



*springer tracts in advanced robotics 88*

Jaydev P. Desai  
Gregory Dudek  
Oussama Khatib  
Vijay Kumar (Eds.)

# Experimental Robotics

The 13th International Symposium  
on Experimental Robotics

 Springer

The Springer logo consists of a white chess knight piece on a pedestal, positioned to the left of the word "Springer" in a white serif font.

## Editors

Prof. Bruno Siciliano  
Dipartimento di Ingegneria Elettrica  
e Tecnologie dell'Informazione  
Università degli Studi di Napoli  
Federico II  
Via Claudio 21, 80125 Napoli  
Italy  
E-mail: siciliano@unina.it

Prof. Oussama Khatib  
Artificial Intelligence Laboratory  
Department of Computer Science  
Stanford University  
Stanford, CA 94305-9010  
USA  
E-mail: khatib@cs.stanford.edu



## Editorial Advisory Board

Oliver Brock, TU Berlin, Germany  
Herman Bruyninckx, KU Leuven, Belgium  
Raja Chatila, ISIR - UPMC & CNRS, France  
Henrik Christensen, Georgia Tech, USA  
Peter Corke, Queensland Univ. Technology, Australia  
Paolo Dario, Scuola S. Anna Pisa, Italy  
Rüdiger Dillmann, Univ. Karlsruhe, Germany  
Ken Goldberg, UC Berkeley, USA  
John Hollerbach, Univ. Utah, USA  
Makoto Kaneko, Osaka Univ., Japan  
Lydia Kavraki, Rice Univ., USA  
Vijay Kumar, Univ. Pennsylvania, USA  
Sukhan Lee, Sungkyunkwan Univ., Korea  
Frank Park, Seoul National Univ., Korea  
Tim Salcudean, Univ. British Columbia, Canada  
Roland Siegwart, ETH Zurich, Switzerland  
Gaurav Sukhatme, Univ. Southern California, USA  
Sebastian Thrun, Stanford Univ., USA  
Yangsheng Xu, Chinese Univ. Hong Kong, PRC  
Shin'ichi Yuta, Tsukuba Univ., Japan

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices of EURON (European Robotics Research Network)



Jaydev P. Desai · Gregory Dudek  
Oussama Khatib · Vijay Kumar  
Editors

# Experimental Robotics

The 13th International Symposium  
on Experimental Robotics

 Springer

*Editors*

Jaydev P. Desai  
Department of Mechanical Engineering  
Robotics, Automation, and  
Medical Systems (RAMS) Laboratory  
Maryland Robotics Center,  
Institute for Systems Research  
University of Maryland  
College Park  
USA

Gregory Dudek  
Centre for Intelligent Machines  
McConnell Engineering Building  
McGill University  
Montreal  
Canada

Oussama Khatib  
Department of Computer Science  
Artificial Intelligence Laboratory  
Stanford University  
Stanford  
USA

Vijay Kumar  
School of Engineering and Applied Science  
University of Pennsylvania  
Philadelphia  
USA

ISSN 1610-7438

ISSN 1610-742X (electronic)

ISBN 978-3-319-00064-0

ISBN 978-3-319-00065-7 (eBook)

DOI 10.1007/978-3-319-00065-7

Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013937066

© Springer International Publishing Switzerland 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Foreword

Robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into human environments and vigorously engaged in its new challenges. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

Beyond its impact on physical robots, the body of knowledge robotics has produced is revealing a much wider range of applications reaching across diverse research areas and scientific disciplines, such as: biomechanics, haptics, neurosciences, virtual simulation, animation, surgery, and sensor networks among others. In return, the challenges of the new emerging areas are proving an abundant source of stimulation and insights for the field of robotics. It is indeed at the intersection of disciplines that the most striking advances happen.

The *Springer Tracts in Advanced Robotics (STAR)* is devoted to bringing to the research community the latest advances in the robotics field on the basis of their significance and quality. Through a wide and timely dissemination of critical research developments in robotics, our objective with this series is to promote more exchanges and collaborations among the researchers in the community and contribute to further advancements in this rapidly growing field.

Since its inception, the *International Symposium on Experimental Robotics (ISER)* was published by Springer. Since the four past editions, ISER has found a more suitable home under STAR, together with other thematic symposia devoted to excellence in robotics research.

The Thirteenth edition of *Experimental Robotics* edited by Jaydev P. Desai, Gregory Dudek, Oussama Khatib and Vijay Kumar offers in its fourteen-chapter volume a collection of a broad range of topics in field and human-centered robotics. The contents of these contributions represent a cross-section of the current state of robotics research from one particular aspect: experimental work, and how it reflects on the theoretical basis of subsequent developments. Experimental validation of algorithms, concepts, or techniques is the common thread running through this large collection of widely diverse contributions, spanning from design to dynamics and control, from manipulation to planning and control, from learning to sensing

and navigation, from multi-robot to human-robot interaction, from aerial to marine robotics, from social robotics to life-science applications.

From its warm social program to its excellent technical program, ISER culminates with this unique reference on the current developments and new directions of experimental robotics – a genuine tribute to its contributors and organizers!

Naples, Italy  
January 2013

Bruno Siciliano  
STAR Editor

# Preface

The International Symposium on Experimental Robotics (ISER), which began in 1989, is a biennial symposium bringing together researchers from all over the world. Sponsored by the International Foundation of Robotics Research (IFRR), this symposium's primary goal is to bring together researchers who are at the forefront of experimental robotics research to present their work in a single-track format. ISER contributions focus on developments of novel experimental platforms, new validation methods, innovative techniques for solving challenging experimental problems, and new robotic systems.

The 13<sup>th</sup> International Symposium on Experimental Robotics was held in Québec City, Canada, at the Fairmont Le Château Frontenac, on June 18–21, 2012. There was an open call for contributions, and after a review process organized by the international steering committee, 65 papers were selected for presentation at the symposium. The technical program included sessions on design, dynamics and control, aerial robotics, multi-robot, learning, social robotics, manipulation, applications to the life sciences, planning and control, field robotics, marine robotics, sensing and navigation, and human-robot interaction. In addition, the program included an interactive session that brought exciting multimedia presentations. Each session began with a dedicated introduction by its chair summarizing and connecting the associated contributions.

The program also featured two invited presentations at the symposium given by Masayuki Inaba from The University of Tokyo and Howie Choset from Carnegie Mellon University. A highlight of the interaction created during the symposium was the panel organized at the scenic Montmorency Falls, which brought much exciting and stimulating discussion among the participants.

This volume includes the collection of contributions presented at the symposium as well as the session introductions, which provide summaries of the accepted papers in each session. We are grateful to Philippe Giguère and his team from Laval University for the outstanding organization of the various activities during the symposium. We would also like to thank Krys Dudek for all the efforts she devoted to the design and management of the symposium website. The greatest words of thanks go

of course to the authors and participants for all their contributions to the excellence in the technical program and social interaction, making this edition of ISER a great success.

Jaydev P. Desai, University of Maryland, College Park  
Gregory Dudek, McGill University  
Oussama Khatib, Stanford University  
Vijay Kumar, University of Pennsylvania

# Contents

<b>Part I: ISER Session Summary on “Design”</b> . . . . .	1
<i>Clément Gosselin</i>	
<b>On the Development of a Programmable Inertia Generator</b> . . . . .	3
<i>Clément Gosselin, Alexandre Lecours, Thierry Laliberté, Frédéric Lessard</i>	
<b>Design of Distributed End-Effectors for Caging-Specialized Manipulator (Design Concept and Development of Finger Component)</b> . . . . .	15
<i>Rui Fukui, Keita Kadowaki, Yamato Niwa, Weiwei Wan, Masamichi Shimosaka, Tomomasa Sato</i>	
<b>Experiments in Underactuated In-Hand Manipulation</b> . . . . .	27
<i>Lael U. Odhner, Raymond R. Ma, Aaron M. Dollar</i>	
<b>Towards a Self-contained Soft Robotic Fish: On-Board Pressure Generation and Embedded Electro-permanent Magnet Valves</b> . . . . .	41
<i>Andrew D. Marchese, Cagdas D. Onal, Daniela Rus</i>	
<b>An Empirical Study of Static Loading on Piezoelectric Stick-Slip Actuators of Micromanipulators</b> . . . . .	55
<i>Aayush Damani, Manikantan Nambi, Jake J. Abbott</i>	
<b>Part II: ISER Session Summary on “Dynamics and Control”</b> . . . . .	71
<i>Patrick van der Smagt</i>	
<b>Rapid Prototyping of Planning, Learning and Control in Physical Human-Robot Interaction</b> . . . . .	73
<i>Martin Lawitzky, José Ramón Medina Hernández, Sandra Hirche</i>	



<b>Identification of Human Limb Stiffness in 5 DoF and Estimation via EMG</b> .....	89
<i>Dominic Lakatos, Daniel Rüschen, Justin Bayer, Jörn Vogel, Patrick van der Smagt</i>	
<b>Motor vs. Brake: Comparative Studies on Performance and Safety in Hybrid Actuations</b> .....	101
<i>Dongjun Shin, Xiyang Yeh, Takashi Narita, Oussama Khatib</i>	
<b>Examining the Effect of Rear Leg Specialization on Dynamic Climbing with SCARAB: A Dynamic Quadrupedal Robot for Locomotion on Vertical and Horizontal Surfaces</b> .....	113
<i>Bruce Miller, Camilo Ordonez, Jonathan E. Clark</i>	
<b>Part III: ISER Session Summary on “Interactive Session”</b> .....	127
<i>Henrik I. Christensen</i>	
<b>Real-Time Clustering for Long-Term Autonomy</b> .....	129
<i>Lionel Ott, Fabio Ramos</i>	
<b>3-Dimensional Tiling for Distributed Assembly by Robot Teams</b> .....	143
<i>James Worcester, Rolf Lakaemper, Mong-ying Ani Hsieh</i>	
<b>JediBot—Experiments in Human-Robot Sword-Fighting</b> .....	155
<i>Torsten Kröger, Ken Oslund, Tim Jenkins, Dan Torczynski, Nicholas Hippenmeyer, Radu Bogdan Rusu, Oussama Khatib</i>	
<b>Development of Aerobots for Satellite Emulation, Architecture and Art</b> .....	167
<i>Inna Sharf, M.S. Persson, David St-Onge, Nicolas Reeves</i>	
<b>Experimental Multi-Vehicle Path Coordination under Communication Connectivity Constraints</b> .....	183
<i>Pramod Abichandani, Kenneth Mallory, Mong-ying Ani Hsieh</i>	
<b>Proactively Approaching Pedestrians with an Autonomous Mobile Robot in Urban Environments</b> .....	199
<i>Daniel Carton, Annemarie Turnwald, Dirk Wollherr, Martin Buss</i>	
<b>Multitask Humanoid Control with a Brain-Computer Interface: User Experiment with HRP-2</b> .....	215
<i>Pierre Gergondet, Abderrahmane Kheddar, Christoph Hintermüller, Christoph Guger, Mel Slater</i>	
<b>Coordination Strategies for Multi-robot Exploration and Mapping</b> .....	231
<i>John G. Rogers III, Carlos Nieto-Granda, Henrik I. Christensen</i>	

**Experiments Comparing Precision of Stereo-Vision Approaches for Control of an Industrial Manipulator** . . . . . 245  
*John-David Yoder, Jeffrey West, Eric Baumgartner, Mathias Perrollaz, Michael Seelinger, Matthew Robinson*

**Part IV: ISER Session Summary on “Aerial Robotics”** . . . . . 257  
*Gaurav S. Sukhatme*

**Environmental Sensing Using Land-Based Spectrally-Selective Cameras and a Quadcopter** . . . . . 259  
*Jnaneshwar Das, William C. Evans, Michael Minnig, Alexander Bahr, Gaurav S. Sukhatme, Alcherio Martinoli*

**State Estimation for Indoor and Outdoor Operation with a Micro-Aerial Vehicle** . . . . . 273  
*Shaojie Shen, Nathan Michael*

**Influence of Aerodynamics and Proximity Effects in Quadrotor Flight** . . . . . 289  
*Caitlin Powers, Daniel Mellinger, Aleksandr Kushleyev, Bruce Kothmann, Vijay Kumar*

**On the Consistency of Vision-Aided Inertial Navigation** . . . . . 303  
*Dimitrios G. Kottas, Joel A. Hesch, Sean L. Bowman, Stergios I. Roumeliotis*

**Part V: ISER Session Summary on “Multi-Robot”** . . . . . 319  
*Nathan Michael*

**Accurate Localization with Ultra-Wideband: Tessellated Spatial Models and Collaboration** . . . . . 321  
*Amanda Prorok, Alcherio Martinoli*

**Cooperative Multi-robot Estimation and Control for Radio Source Localization** . . . . . 337  
*Benjamin Charrow, Nathan Michael, Vijay Kumar*

**Real-Time Optimized Rendezvous on Nonholonomic Resource-Constrained Robots** . . . . . 353  
*Sven Goyal, Alcherio Martinoli*

**Part VI: ISER Session Summary on “Learning”** . . . . . 369  
*Matthew T. Mason*

**Learning Autonomous Driving Styles and Maneuvers from Expert Demonstration** . . . . . 371  
*David Silver, J. Andrew Bagnell, Anthony Stentz*

<b>Unsupervised Feature Learning for RGB-D Based Object Recognition</b> .....	387
<i>Liefeng Bo, Xiaofeng Ren, Dieter Fox</i>	
<b>Learning to Parse Natural Language Commands to a Robot Control System</b> .....	403
<i>Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, Dieter Fox</i>	
<b>A Data-Driven Statistical Framework for Post-Grasp Manipulation</b> ....	417
<i>Robert Paolini, Alberto Rodriguez, Siddhartha S. Srinivasa, Matthew T. Mason</i>	
<b>Part VII: ISER Session Summary on “Social Robotics”</b> .....	433
<i>Maja J. Matarić</i>	
<b>Grasping with Your Face</b> .....	435
<i>Jonathan Weisz, Benjamin Shababo, Lixing Dong, Peter K. Allen</i>	
<b>Human Aware Navigation for Assistive Robotics</b> .....	449
<i>Dízan Vasquez, Procópio Stein, Jorge Rios-Martinez, Arturo Escobedo, Anne Spalanzani, Christian Laugier</i>	
<b>Socially Assistive Robot Exercise Coach: Motivating Older Adults to Engage in Physical Exercise</b> .....	463
<i>Juan Fasola, Maja J. Matarić</i>	
<b>Interpreting and Executing Recipes with a Cooking Robot</b> .....	481
<i>Mario Bollini, Stefanie Tellex, Tyler Thompson, Nicholas Roy, Daniela Rus</i>	
<b>Part VIII: ISER Session Summary on “Manipulation”</b> .....	497
<i>Evan Drumwright</i>	
<b>Load Equalization on a Two-Armed Robot via Proprioceptive Sensing</b> .....	499
<i>Roxana Leontie, Evan Drumwright, Dylan A. Shell, Rahul Simha</i>	
<b>Mapping Grasps from the Human Hand to the DEXMART Hand by Means of Postural Synergies and Vision</b> .....	515
<i>Fanny Ficuciello, Gianluca Palli, Claudio Melchiorri, Bruno Siciliano</i>	
<b>Manipulation with Multiple Action Types</b> .....	531
<i>Jennifer Barry, Kaijen Hsiao, Leslie Pack Kaelbling, Tomás Lozano-Pérez</i>	
<b>A Constraint-Aware Motion Planning Algorithm for Robotic Folding of Clothes</b> .....	547
<i>Karthik Lakshmanan, Apoorva Sachdev, Ziang Xie, Dmitry Berenson, Ken Goldberg, Pieter Abbeel</i>	

**Part IX: ISER Session Summary on “Applications to the Life Sciences”** ..... 563  
*Jaydev P. Desai*

**Towards the Development of a Master-Slave Surgical System for Breast Biopsy under Continuous MRI** ..... 565  
*Bo Yang, U-Xuan Tan, Alan McMillan, Rao Gullapalli, Jaydev P. Desai*

**Motion Compensated Catheter Ablation of the Beating Heart Using Image Guidance and Force Control** ..... 579  
*Samuel B. Kesner, Robert D. Howe*

**Robotic Micropipette Aspiration of Biological Cells** ..... 591  
*Ehsan Shojaei-Baghini, Yu Sun*

**Quantitative Analysis of Locomotive Behavior of Human Sperm Head and Tail** ..... 603  
*Jun Liu, Zhe Lu, Clement Leung, Yu Sun*

**Characterization and Control of Biological Microrobots** ..... 617  
*Islam S.M. Khalil, Marc P. Pichel, Lars Zondervan, Leon Abelmann, Sarthak Misra*

**Part X: ISER Session Summary on “Planning and Control”** ..... 633  
*Nikhil Chopra*

**Computed-Torque Control of a Four-Degree-of-Freedom Admittance Controlled Intelligent Assist Device** ..... 635  
*Alexandre Lecours, Clément Gosselin*

**Sampling-Based Direct Trajectory Generation Using the Minimum Time Cost Function** ..... 651  
*Oscar Chuy, Emmanuel Collins, Damion Dunlap, Aneesh Sharma*

**Antagonistic Control of Multi-DOF Joint** ..... 667  
*Koichi Koganezawa*

**Lyapunov Based Sampling for Adaptive Tracking Control in Robot Manipulators: An Experimental Comparison** ..... 683  
*Pavankumar Tallapragada, Nikhil Chopra*

**Linguistic Composition of Semantic Maps and Hybrid Controllers** ..... 699  
*Neil Dantam, Carlos Nieto-Granda, Henrik I. Christensen, Mike Stilman*

**Part XI: ISER Session Summary on “Field Robotics”** ..... 715  
*Stephen Nuske*

**Energy-Efficient Path Planning for Solar-Powered Mobile Robots** ..... 717  
*Patrick A. Plonski, Pratap Tokekar, Volkan Isler*

<b>Change Detection Using Airborne LiDAR: Applications to Earthquakes</b> . . . . .	733
<i>Aravindhnan K. Krishnan, Edwin Nissen, Srikanth Saripalli, Ramon Arrowsmith, Alejandro Hinojosa-Corona</i>	
<b>Automated Crop Yield Estimation for Apple Orchards</b> . . . . .	745
<i>Qi Wang, Stephen Nuske, Marcel Bergerman, Sanjiv Singh</i>	
<b>Spatial Interpolation for Robotic Sampling: Uncertainty with Two Models of Variance</b> . . . . .	759
<i>Young-Ho Kim, Dylan A. Shell, Colin Ho, Srikanth Saripalli</i>	
<b>Acoustic Masking of a Stealthy Outdoor Robot Tracking a Dynamic Target</b> . . . . .	775
<i>Ashley Tews, Matthew Dunbabin</i>	
<b>Part XII: ISER Session Summary on “Marine Robotics”</b> . . . . .	787
<i>Daniela Rus</i>	
<b>Autonomous Adaptive Underwater Exploration Using Online Topic Modeling</b> . . . . .	789
<i>Yogesh Girdhar, Philippe Giguère, Gregory Dudek</i>	
<b>Active and Adaptive Dive Planning for Dense Bathymetric Mapping</b> . . . .	803
<i>Geoffrey A. Hollinger, Urbashi Mitra, Gaurav S. Sukhatme</i>	
<b>Exploring Space-Time Tradeoffs in Autonomous Sampling for Marine Robotics</b> . . . . .	819
<i>Rishi Graham, Frédéric Py, Jnaneshwar Das, Drew Lucas, Thom Maughan, Kanna Rajan</i>	
<b>Autonomous, Localization-Free Underwater Data Muling Using Acoustic and Optical Communication</b> . . . . .	841
<i>Marek Doniec, Iulian Topor, Mandar Chitre, Daniela Rus</i>	
<b>Local-Search Strategy for Active Localization of Multiple Invasive Fish</b> . . . . .	859
<i>Joshua Vander Hook, Pratap Tokekar, Elliot Branson, Przemyslaw G. Bajer, Peter W. Sorensen, Volkan Isler</i>	
<b>Part XIII: ISER Session Summary on “Sensing and Navigation”</b> . . . . .	875
<i>Liz Murphy</i>	
<b>Experimental Comparison of Odometry Approaches</b> . . . . .	877
<i>Liz Murphy, Timothy Morris, Ugo Fabrizi, Michael Warren, Michael Milford, Ben Upcroft, Michael Bosse, Peter Corke</i>	
<b>Building Large Scale Traversability Maps Using Vehicle Experience</b> . . . .	891
<i>Steven Martin, Liz Murphy, Peter Corke</i>	

Contents	XV
<b>Automatic and Full Calibration of Mobile Laser Scanning Systems</b> . . . . .	907
<i>Jan Elseberg, Dorit Borrmann, Andreas Nüchter</i>	
<b>Part XIV: ISER Session Summary on “Human Robot Interaction”</b> . . . . .	919
<i>Alonzo Kelly</i>	
<b>Hallucinating Humans for Learning Robotic Placement of Objects</b> . . . . .	921
<i>Yun Jiang, Ashutosh Saxena</i>	
<b>Hand Shape Classification with a Wrist Contour Sensor (Comparison of Feature Types and Observation of Resemblance among Subjects)</b> . . . . .	939
<i>Rui Fukui, Masahiko Watanabe, Masamichi Shimosaka, Tomomasa Sato</i>	
<b>Experimental Validation of Operator Aids for High Speed Vehicle Teleoperation</b> . . . . .	951
<i>Alonzo Kelly, Nicholas Chan, Herman Herman, Randy Warner</i>	
<b>Intention-Aware Pedestrian Avoidance</b> . . . . .	963
<i>Tirthankar Bandyopadhyay, Chong Zhuang Jie, David Hsu, Marcelo H. Ang Jr., Daniela Rus, Emilio Frazzoli</i>	
<b>The UBC Visual Robot Survey: A Benchmark for Robot Category Recognition</b> . . . . .	979
<i>David Meger, James J. Little</i>	
<b>Author Index</b> . . . . .	993

# Part I: ISER Session Summary on “Design”

Clément Gosselin

Laboratoire de robotique, Département de génie mécanique  
Université Laval, Québec, Qc, Canada

## Session Summary

The session on design covered a variety of topics ranging from novel haptic devices to underwater robots. Although these topics may seem unrelated, the articles presented in this session conveyed a common message: robot design is a challenging task that is best addressed using a global approach focusing on the tasks to be performed by the robots. Therefore, the presentations included both analytical derivations and clever engineering design. The first paper presented the development of an inertia generator. The goal of this work is to design a hand held device whose inertia can be programmed by moving internal masses in reaction to the acceleration of the frame, measured using accelerometers. A one-dof prototype was shown, which demonstrated the feasibility of the principle. The second paper dealt with the design of caging end-effectors. Caging end-effectors depart from traditional grasping systems in that they are based on purely geometric constraints. In many applications, caging may be used to relieve robots from high precision demands. A prototype based on permanent magnet inductive traction rails was demonstrated and experimental results were discussed. The third paper presented conceptual and experimental results demonstrating the ability of underactuated hands to manipulate small objects between their fingertips. Underactuated hands are considered a promising concept for robotics because they include a limited number of actuators. However, fingertip manipulation is a challenge for such hands. Some solutions were proposed in this paper to alleviate this drawback. The fourth paper introduced a robotic fish and its propulsion and control principle. Miniature CO<sub>2</sub> cartridges are used in conjunction with a custom pressure regulating system and embeddable electro-permanent magnet valves. A prototype was demonstrated that emulates natural caudal fin and peduncle movements. Finally, the last paper addressed the modelling of stick-slip actuators, which are the foundation of modern micromanipulation. The effect of static loads on these actuators is studied, which leads to better models that were confirmed experimentally. Globally, the work presented in the session on design raised issues that are crucial to modern robotics and novel effective solutions were proposed to address them.

# On the Development of a Programmable Inertia Generator

Clément Gosselin, Alexandre Lecours, Thierry Laliberté, and Frédéric Lessard

**Abstract.** This paper presents a preliminary investigation on a one-degree-of-freedom programmable inertia generator. An inertia generator is a hand-held haptic device that has a programmable inertia. By moving internal masses in reaction to accelerations induced by the user, the effective inertia of the device is modified in order to render a prescribed perceived inertia. In this paper, a one-degree-of-freedom device with one internal moving mass is proposed. The dynamic modelling of the system is first presented. Then, a controller is designed to produce the appropriate motion of the internal mass in reaction to the acceleration induced by the user. A prototype is presented and experimental results are discussed.

## 1 Introduction

It is common, for training or entertainment purposes, to actively prescribe the dynamics rendered by a robotic system. For instance, impedance control is used in haptics or physical human-robot interaction (pHRI) in order to simulate virtual environments. This approach can be implemented using fixed-base haptic devices which are controlled to produce a desired behaviour (see for instance [7] and many others).

In interactive systems (e.g. computer games), hand-held devices are also often used. These devices are typically passive [13] and are unable to produce kinesthetic feedback. However, as shown in [1], [11], it is possible to include moving masses in hand-held devices in order to produce an illusion of an external force. This approach can also be used with rotating mechanisms in order to produce the illusion of an external moment [2], [15].

The work reported in this paper addresses a similar but different challenge. The long-term objective of this initiative is to design a hand-held device that has a

---

Clément Gosselin · Alexandre Lecours · Thierry Laliberté · Frédéric Lessard

Département de génie mécanique, Université Laval, Québec, Qc, Canada

e-mail: {gosselin, thierry}@gmc.ulaval.ca,

{alexandre.lecours.1, frederic.lessard.2}@ulaval.ca



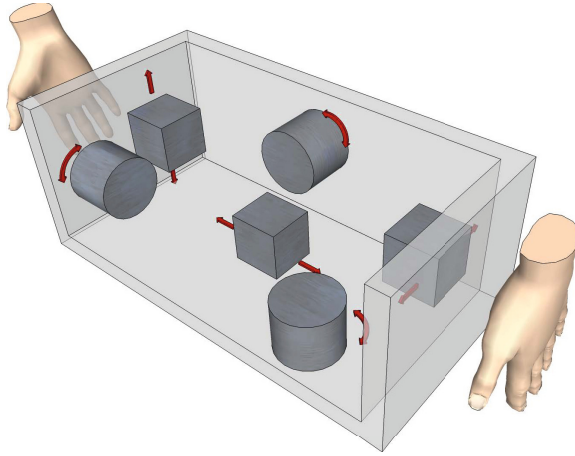
programmable inertia. The effective inertia of such a device is modified by moving internal masses in reaction to the accelerations imparted to the hand-held system by the user. Using this approach, the inertia perceived by the user manipulating the hand-held device can be prescribed arbitrarily, within the physical constraints of the mechanical system.

In this paper, a preliminary investigation on a one-degree-of-freedom (one-dof) device is reported. The main objective of this first phase of the work is to assess the feasibility of the concept. In order to vary the perceived inertia, a moving mass is mounted on a slider within the manipulated one-dof physical interface. By moving the internal mass in reaction to an accelerometer signal, a prescribed effective inertia can be rendered. The paper is structured as follows: after providing a general description of the concept of inertia generator, the dynamic model of the one-dof mechanical system studied here is derived. Then, a controller is designed to produce the appropriate motion of the internal mass in reaction to the accelerations. A description of the physical one-dof prototype is then provided. Finally, experimental results are given and interpreted.

## 2 General Concept of Inertia Generator

The general concept of inertia generator is represented schematically in Fig. 1. Consider a box which is held by a user and inside of which a set of masses are mounted on actuated sliders or revolute joints. For example, three masses could be mounted on orthogonal actuated rails and three inertias could be mounted on orthogonal actuated pivots. Alternatively, one single rigid body could be attached to the end-effector of a 6-dof parallel mechanism that can produce translations and rotations of this mass in arbitrary directions. When the user imparts accelerations to the box, the latter are measured by a set of accelerometers and the masses are displaced in order to render a prescribed inertia. If the ratio of the moving masses to the mass of the frame of the box is large enough, accelerating the internal masses will produce a significant change in the external apparent inertia of the box. For instance, if the user is accelerating the box along the  $x$  axis and it is desired to render an inertia that is smaller than the physical inertia of the system, the mass(es) will be accelerated in the opposite direction in order to reduce the effective inertia. Similarly, if it is desired to render an inertia that is larger than that of the physical system, then the mass(es) will be moved in the direction of the acceleration imparted by the user in order to increase the apparent inertia.

The principle of the inertia generator is akin to that of motion simulators[14] and to that of acceleration compensation for vibration isolation[6, 5]. In such applications and in this work, the concept of *washout filter* is important[8]. The principle of the washout filter is to include a low-frequency command in the control loop that aims at bringing the mechanism to a *neutral* configuration so that it is ready for the next acceleration input. The neutral configuration is defined as one in which all directions of motion are feasible with approximately the same range of motion in all directions. This concept is discussed in the control section of the paper.

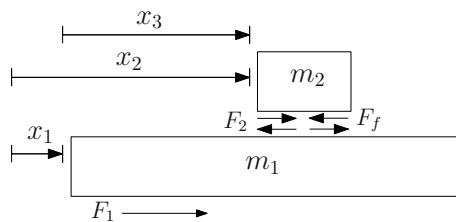


**Fig. 1** Schematic representation of the concept of inertia generator

### 3 Dynamic Modelling of a One-Degree-of-Freedom Inertia Generator

A schematic representation of the one-dof inertia generator is given in Fig. 2. The external frame of the device has a mass  $m_1$  and is subjected to the external force  $F_1$  applied by the user. It is assumed that the direction of motion is horizontal. An actuator is included in the device to move a second mass  $m_2$  mounted on a slider attached to the frame of mass  $m_1$ . The force applied by the actuator on mass  $m_2$  is noted  $F_2$  while the friction force between the two masses is noted  $F_f$ . The position of the frame of the device with respect to a fixed inertial frame is noted  $x_1$ , the position of mass  $m_2$  with respect to the inertial frame is noted  $x_2$  and  $x_3$  denotes the position of mass  $m_2$  with respect to mass  $m_1$ . Therefore, one has

$$x_2 = x_1 + x_3, \quad \ddot{x}_2 = \ddot{x}_1 + \ddot{x}_3. \tag{1}$$



**Fig. 2** Schematic representation of the one-dof inertia generator

It is desired to render a prescribed inertia, noted  $m_a$ , when the user applies forces on the device. Therefore, the desired behaviour can be expressed as

$$F_1 = m_a \ddot{x}_1 \quad (2)$$

where  $\ddot{x}_1$  is the acceleration of mass  $m_1$  with respect to the fixed inertial frame.

Applying Newton's second law to each of the moving masses, one obtains

$$F_1 - (F_2 - F_f) = m_1 \ddot{x}_1 \quad (3)$$

$$F_2 - F_f = m_2 \ddot{x}_2 \quad (4)$$

where  $\ddot{x}_2$  is the acceleration of mass  $m_2$  with respect to the fixed inertial frame. Substituting eq.(1) into eqs.(3) and (4) and rearranging, one has

$$F_1 = (m_1 + m_2) \ddot{x}_1 + m_2 \ddot{x}_3 \quad (5)$$

$$F_2 = m_2 \ddot{x}_1 + m_2 \ddot{x}_3 + F_f. \quad (6)$$

Equations (5) and (6) represent the dynamics of the two-dof system comprising the mobile frame of mass  $m_1$  and the sliding mass  $m_2$ . Referring to eq.(2), it is desired to obtain an expression for the force  $F_2$  to be applied by the actuator in order to render the inertia  $m_a$ . To this end, eq.(2) is first substituted into eq.(5), which leads to

$$m_2 \ddot{x}_3 = (m_a - m_1 - m_2) \ddot{x}_1. \quad (7)$$

Substituting the latter equation into eq.(6) then leads to

$$F_2 = (m_a - m_1) \ddot{x}_1 + F_f. \quad (8)$$

Finally, eq.(7) is rearranged in order to determine the acceleration required at the actuator, namely by rewriting it as

$$\ddot{x}_{d3} = \frac{(m_a - m_1 - m_2) \ddot{x}_1}{m_2} \quad (9)$$

where  $\ddot{x}_{d3}$  is the desired relative acceleration.

## 4 Controller Design

Based on the dynamic model presented in the previous section, a control scheme can be developed in order to render the prescribed inertia  $m_a$ . The system is designed to react to the acceleration of the moving frame of mass  $m_1$ . To this end, an accelerometer is mounted on the frame, which provides a measurement of its acceleration  $\ddot{x}_1$ . The control strategy is based on the combination of three terms, namely: a feedforward term (including friction compensation), a feedback term, and a washout term. Each of these contributions is now detailed.

## 4.1 Feedforward and Friction Compensation

The feedforward term is based on the dynamic model developed in the preceding section. Equation (8) is used to compute an estimation of the force to be applied at the actuator based on the measured acceleration  $\dot{x}_1$  and on an estimation of the friction force  $F_f$ . A simple friction compensation can be written as:

$$F_f = f_c + f_v \quad (10)$$

where  $f_c$  and  $f_v$  are respectively the Coulomb and viscous friction forces with

$$f_c = c(1 - e^{-\alpha|\dot{x}_{d3}|})\text{sign}(\dot{x}_{d3}) \quad (11)$$

$$f_v = v\dot{x}_{d3} \quad (12)$$

where  $c$  is the Coulomb friction coefficient,  $v$  the viscous friction coefficient and  $\alpha$  is a tuning parameter. The exponential term is used to reduce the chattering induced by friction compensation when the velocity is near zero. The desired velocity is used for friction compensation in order to reduce the command noise, although the measured velocity could also be used. Other more complex friction models could also be used (see for instance [16, 4, 3, 9, 10]), including stiction for example, but the simple friction compensation given in eqn. (10) provided good experimental results.

## 4.2 Feedback

The desired acceleration of mass  $m_2$  can be computed using eq.(9). In order to achieve this relative acceleration, it would be possible to use a feedback control (e.g. PID control) with a relative acceleration measurement (e.g. accelerometer or second derivative of the position). However, acceleration control is not very practical mainly because the measured acceleration is known to be very noisy. Instead, velocity or position control can be implemented using an integration technique.

First, the discrete desired velocity required to render the desired acceleration is obtained with a zero-order-hold integration<sup>1</sup>:

$$\dot{x}_{d3}(k) = \dot{x}_{d3}(k-1) + \ddot{x}_{d3}(k)T_s \quad (13)$$

while the position is obtained by integrating a second time, namely:

$$x_{d3}(k) = x_{d3}(k-1) + \dot{x}_{d3}(k-1)T_s + \frac{1}{2}\ddot{x}_{d3}(k)T_s^2 \quad (14)$$

where  $T_s$  is the sampling period,  $k$  is the time step and  $x_{d3}$ ,  $\dot{x}_{d3}$  and  $\ddot{x}_{d3}$  are respectively the desired position, velocity and acceleration.

One should note that this integration method is used to achieve acceleration control in physical human-robot interaction with admittance control schemes [12] and that, although it is preferable to use the desired velocity of the preceding step

<sup>1</sup> Alternatively, a bilinear discretization can be used.

( $\dot{x}_{d3}(k-1)$ ), the measured velocity can alternatively be used in the above equations. The desired acceleration is then rendered using velocity or position control, which is more practical and can be achieved using a simple PID controller or more complex algorithms.

### 4.3 Washout

As explained in a preceding section, the goal of the washout filter is to ensure that the moving mass is kept as close as possible to its neutral position in order to be ready to accommodate arbitrary acceleration inputs. For the one-dof system studied here, this amounts to keeping mass  $m_2$  as close as possible to its mid-range position in order to avoid the mechanical limits (end of stroke). To this end, a virtual spring-damper system modelled as follows is used:

$$F_w = -K_w(x_3 - x_w) - C_w\dot{x}_3 \quad (15)$$

where  $F_w$  stands for the washout force,  $K_w$  is the washout spring stiffness,  $C_w$  is the washout damping factor and  $x_w$  is the neutral position. The adjustment of the washout parameters is based on the analysis of the above second-order system. Knowing the mass to be moved  $m_2$ , one obtains

$$K_w = m_2 \omega_w^2 \quad (16)$$

$$C_w = 2m_2 \zeta_w \omega_w \quad (17)$$

where  $\omega_w$  is the washout natural frequency and  $\zeta_w$  is the damping factor. The washout natural frequency should be chosen low enough so that it does not significantly impact the rendering. On the other hand, it should be high enough to ensure that mass  $m_2$  is kept close enough to its neutral position. In order to cope with this compromise, the washout natural frequency is adjusted according to the acceleration of the frame,  $\ddot{x}_1$ . The following heuristic rule is used:

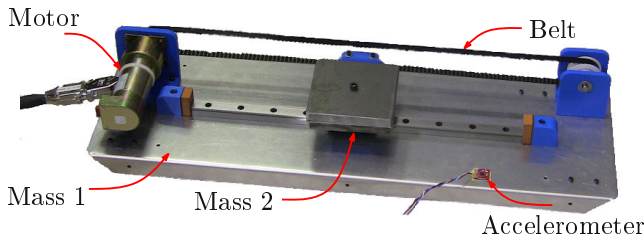
$$\omega_w = \begin{cases} 0 & \text{if } |\ddot{x}_1| > \ddot{x}_{1t} \\ \omega_{w0} \left(1 - \frac{\ddot{x}_1^2}{\ddot{x}_{1t}^2}\right) & \text{otherwise.} \end{cases} \quad (18)$$

where  $\omega_{w0}$  is the default washout natural frequency and  $\ddot{x}_{1t}$  is an acceleration threshold above which the washout is deactivated. In this work,  $\omega_{w0} = 2s^{-1}$  and  $\ddot{x}_{1t} = 0.3ms^{-2}$  are used.

## 5 Experiments

The prototype used in the experiments reported in this paper is shown in Fig. 3. Mass  $m_1$  consists of an aluminium plate equipped with wheels that can roll with

low friction on a table top or on a floor. A DC motor is mounted on the plate together with a rail and pulleys while mass  $m_2$  consists of a small steel plate and a trolley. The motion of mass  $m_2$  is actuated by the DC motor and transmitted with a closed-loop belt. An accelerometer is mounted on the base plate in order to measure the acceleration of the frame,  $\ddot{x}_1$ . Also, an ATI MINI-40 force/torque sensor can be attached to the plate to measure the force  $F_1$  applied by the user on the inertia generator and a second accelerometer can be mounted on mass  $m_2$ . The latter two measurements (force  $F_1$  and acceleration  $\ddot{x}_2$ ) are not used by the controller but only for experimental validation and analysis.



**Fig. 3** Prototype of a one-dof inertia generator used in the experiments

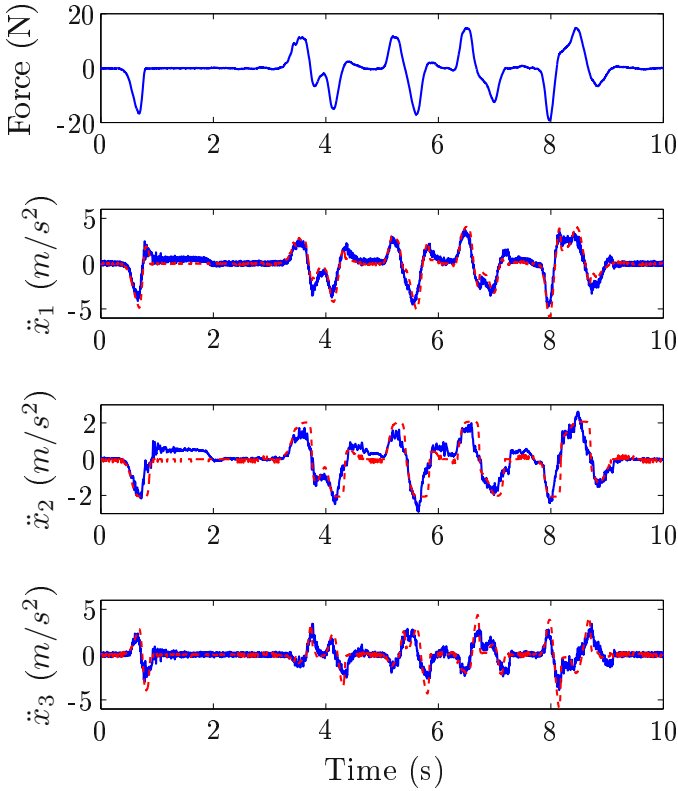
### 5.1 Model Validation

A first experiment was performed in order to validate the dynamic model. In this experiment, the frame (mass  $m_1$ ) is moved manually while the second mass ( $m_2$ ) is free to slide on the rail. The interaction force ( $F_1$ ) and the acceleration of each of the masses ( $\ddot{x}_1$  and  $\ddot{x}_2$ ) are measured. The measured accelerations are then compared with those computed using the measured force and the dynamic equations, in order to validate the model. The results are shown in Fig. 4. It can be observed that, although the results are not perfect, the model is sufficiently realistic to be used for control purposes.

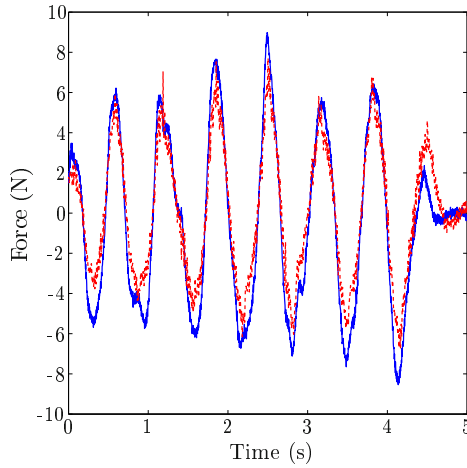
### 5.2 Interaction Experiments

Experiments were performed using the prototype described above, which has the following characteristics:  $m_1 = 2.45\text{kg}$  and  $m_2 = 2.32\text{kg}$ , for a total moving mass of  $4.77\text{kg}$ . The effect of the prescribed inertia  $m_a$  is easily perceived by the user. However, in order to obtain quantitative data, the force and acceleration data were recorded in order to compare the results with the desired dynamics given by eqn. (2).

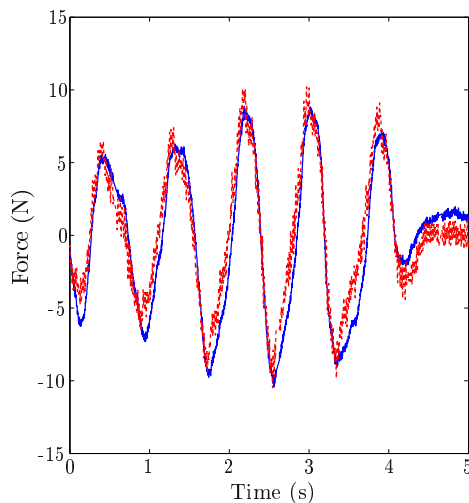
Figures 5 to 8 present the results obtained with a prescribed mass varying from  $m_a = 1.25\text{kg}$  to  $m_a = 6.00\text{kg}$ . On the graphs, the force applied by the user (measured) is compared with the rendered force ( $m_a\ddot{x}_1$ ) where  $\ddot{x}_1$  is the measured acceleration.



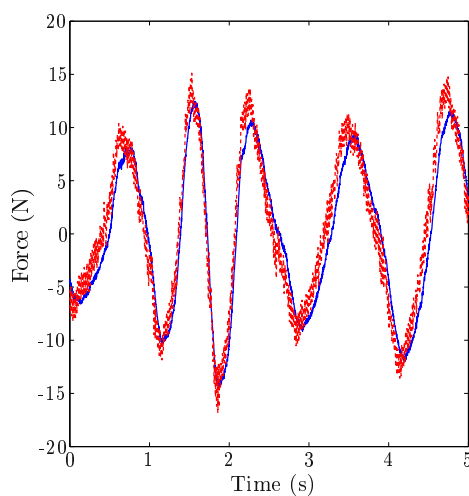
**Fig. 4** Model validation. The solid lines represent measured quantities while the dashed lines represent accelerations computed using the dynamic model and the measured force.



**Fig. 5** Experimental results with  $m_a = 1.25kg$ . The solid line represents the measured force while the dashed line is the model force  $(m_a \ddot{x}_1)$  where  $\ddot{x}_1$  is measured with an accelerometer.

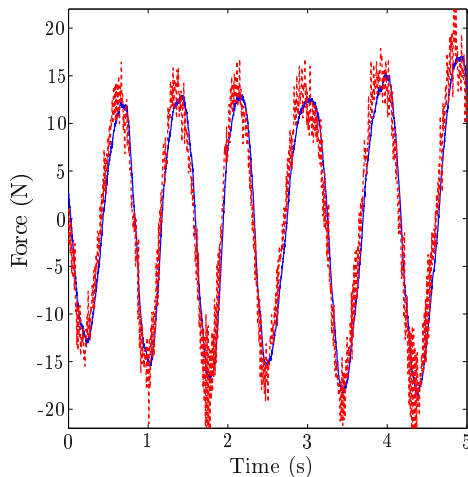


**Fig. 6** Experimental results with  $m_a = 2.45\text{kg}$ . The solid line represents the measured force while the dashed line is the model force ( $m_a \ddot{x}_1$ ) where  $\ddot{x}_1$  is measured with an accelerometer.



**Fig. 7** Experimental results with  $m_a = 4.77\text{kg}$ . The solid line represents the measured force while the dashed line is the model force ( $m_a \ddot{x}_1$ ) where  $\ddot{x}_1$  is measured with an accelerometer.





**Fig. 8** Experimental results with  $m_a = 6.00\text{kg}$ . The solid line represents the measured force while the dashed line is the model force ( $m_a \ddot{x}_1$ ) where  $\ddot{x}_1$  is measured with an accelerometer.

## 6 Discussion

It can be observed from the experimental results that the prescribed inertia is generally well rendered. When the prescribed mass is close to the total mass ( $m_1 + m_2$ ), e.g., with  $m_a = 4.77\text{kg}$ , the results are obviously very good since the demands on the actuator are very low. These results can serve as a basis for the analysis of the results obtained in more demanding situations. Basically, if  $m_a$  is very close to ( $m_1 + m_2$ ), the errors correspond to the estimation errors introduced by the control loop and the measurements themselves.

When the prescribed inertia is more significantly different from the total mass, the actuator is much more solicited and the error tends to increase, as observed in Figs. 5 and 8. Nevertheless, the results are still acceptable, especially considering that the resolution capabilities of the user are limited. For example, with  $m_a = 1.25\text{kg}$  (Fig. 5) the rendered mass is approximately  $1.5\text{kg}$ . It can be observed in the latter figure that the measured forces are slightly larger than the prescribed forces. On the other hand, if the prescribed inertia is larger than the total mass, e.g.,  $m_a = 6.00\text{kg}$  (Fig. 8), the measured forces tend to be slightly smaller than the prescribed forces. These results demonstrate the feasibility of the concept. They also highlight the importance of properly selecting the ratio between  $m_2$  and  $m_1$ , which in turn raises the issue of the power to mass ratio of the actuator. Ideally, the mass ratio ( $m_2/m_1$ ) should be maximized so that the impact of the moving mass is maximized. One possible design avenue is to include the actuator in the moving mass in order to increase the mass ratio. Using this principle, the power to mass ratio of the actuator becomes less important.

## 7 Conclusion

The concept of inertia generator was proposed in this paper. In a hand-held inertia generator, internal masses are moved in reaction to accelerations induced by the user such that the effective inertia of the device is modified in order to render a prescribed perceived inertia. This paper presented preliminary investigations on a one-dof inertia generator that has the capability to render a translational inertia in one direction. The dynamic model of the system was first derived. Based on this model, a controller was proposed that uses the measured acceleration of the frame as an input and determines the motion of the internal mass as an output. The controller is based on an integration of the acceleration signal in order to alleviate the difficulties associated with noisy accelerometer signals. A velocity or position control can therefore be used. Experimental results show that, although the power to mass ratio and the moving mass to frame mass ratio of the prototype are not high, the latter is capable of rendering a significant range of inertias. Future work includes the investigation of multi-dof inertia generators and the experimentation with more advanced prototypes.

**Acknowledgement.** This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), by the Canada Research Chair program and by the Fonds de Recherche du Québec – Nature et Technologie (FRQNT). The authors would also like to thank Michaël Fortin for his help with some of the figures.

## References

1. Amemiya, T., Ando, H., Maeda, T.: Lead-me interface for a pulling sensation from hand-held devices. *ACM Transactions on Applied Perception* (2008)
2. Ando, H., Sugimoto, M., Maeda, T.: Wearable moment display device for nonverbal communication. *IEICE Transactions on Information and Systems E87-D* (2004)
3. Armstrong-Helouvy, B., Dupont, P., Canudas De Wit, C.: A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica* 30(7), 1083–1138 (1994)
4. Cao, L., Schwartz, H.: Stick-slip friction compensation for pid position control. In: *Proceedings of the American Control Conference* (2000)
5. Ebert-Uphoff, I., Dang, A.: Active acceleration compensation for transport vehicles carrying delicate objects. *IEEE Transactions on Robotics* 20(5), 830–839 (2004), doi:10.1109/TRO.2004.832791
6. Graf, R., Dillmann, R.: Active acceleration compensation using a stewart platform on a mobile robot. In: *Proc. 2nd Euromicro Workshop on Advanced Mobile Robots* (1997)
7. Hannaford, B., Okamura, A.: Haptics. In: Siciliano, Khatib (eds.) *The Springer Handbook of Robotics* (2008)
8. Hassouneh, M., Lee, H., Abed, E.: Washout filters in feedback control: benefits, limitations and extensions. In: *American Control Conference, Boston, USA* (2004)
9. Kostic, D., de Jager, B., Steinbuch, M., Hensen, R.: Modeling and identification for high-performance robot control: an rrr-robotic arm case study. *IEEE Transactions on Control Systems Technology* 12(6), 904–919 (2004), doi:10.1109/TCST.2004.833641

10. Kostic, D., Steinbuch, M., de Jager, B.: Modeling and identification for robot motion control. In: *Robotics and Automation Handbook* (2004)
11. Laitinen, P., Mawnpaa, J.: Enabling mobile haptic design: piezoelectric actuator technology properties in hand-held devices. In: *IEEE International Workshop on Haptic Audio Visual Environments and their Applications* (2006)
12. Lecours, A., Mayer-St-Onge, B., Gosselin, C.: Variable admittance control of a four-degree-of-freedom intelligent assist device. In: *IEEE International Conference on Robotics and Automation* (2012)
13. Moen, J.: From hand-held to body-worn: embodied experiences of the design and use of a wearable movement-based interaction concept. In: *First International Conference on Tangible and Embedded Interaction* (2007)
14. Nahon, M., Reid, L.: Simulator motion-drive algorithms, a designer's perspective. *Journal of Guidance, Control and Dynamics* 13(2), 356–362 (1990)
15. Nakamura, N., Fukui, Y.: Development of a force and torque hybrid display 'gyrocubestick'. In: *IEEE World Haptics Conference* (2006)
16. Canudas de Wit, C., Nol, P., Aubin, A., Brogliato, B.: Adaptive Friction Compensation in Robot Manipulators: Low Velocities. *The International Journal of Robotics Research* 10(3), 189–199 (1991)

# Design of Distributed End-Effectors for Caging-Specialized Manipulator

## (Design Concept and Development of Finger Component)

Rui Fukui, Keita Kadowaki, Yamato Niwa, Weiwei Wan,  
Masamichi Shimosaka, and Tomomasa Sato

**Abstract.** In this paper, we propose a novel design of end-effectors that is specialized in caging manipulation. Caging manipulation has several advantages comparing with traditional grasping manipulation. For example, caging can allow small gap/margin between end-effectors and a target object, making the manipulator relieved from constant contact and precise control. Therefore, caging manipulator can avoid many problems from dynamics. Regardless of its advantages, intelligent caging manipulators have not been realized. This is because, for one thing, it may demand many actuators to realize flexible geometrical constraint (caging), for the other thing, kinematic constraints of a general purpose manipulator prevent us from applying direct caging approaches. We address this problem by introducing a novel design/framework of end-effectors that is inspired by ROBOTWORLD. The framework utilizes permanent magnet inductive traction method. The method is suitable for coexistence of multiple robots and for reduction of actuator number by sharing the same actuators. We discuss the concept and the basic framework of the proposed caging manipulator and development of a finger component prototype. After that we conduct basic experiments to evaluate the feasibility of caging manipulation and to reveal the obstacles (challenges) for our manipulator.

## 1 Introduction

Our research group aims to realize a manipulation robot in logistics as shown in Fig. 1. We are especially focusing on stable manipulation of packed objects.

---

Rui Fukui · Keita Kadowaki · Yamato Niwa · Weiwei Wan ·  
Masamichi Shimosaka · Tomomasa Sato

Department of Mechano Informatics,

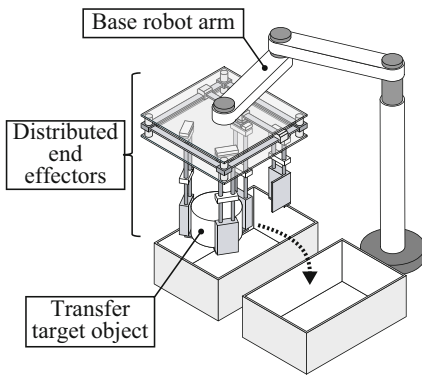
Graduate School of Information Science and Technology,

The University of Tokyo, 113-8656, 7-3-1 Hongo Bunkyo-ku, Tokyo, Japan

e-mail: {fukui, kadowaki, niwa, wan,

simosaka, tsato}@ics.t.u-tokyo.ac.jp

As an example of market products for logistical use, KIVA Systems Corp. developed an automatic object transfer robot system[3]. In that system, robots can manipulate stockers, but the item-level manipulation is executed by human workers. We would like to realize an item-level manipulation robot for logistical use. The right-below text area shows the target specifications of our manipulator.



**Fig. 1** Conceptual sketch of a caging manipulator in logistics applications

- Target objects (Packed daily-use objects are assumed)
  - Shape : hexahedron family and cylinder family
  - Size : maximum is A4 size (210 × 297 [mm]), minimum is cylinder from 30 to 50 [mm] in diameter.<sup>a</sup>
  - Weight : less than 1 [kg]
- Entire weight : less than 5 [kg]
  - ⇒ The end-effectors are supposed to be installed on a general-purpose robot arm.
- Time to transfer an object : within 10 [s]

<sup>a</sup> The maximum size is determined by considering the previous work[4] and the minimum size assumes objects such as a PET bottle.

## 1.1 Related Work

To realize stable object manipulation, there are two major problems. The first is difficulty in acquisition of a precise geometrical model for the target object, while the second is difficulty in precise recognition of the surface properties for the target object.

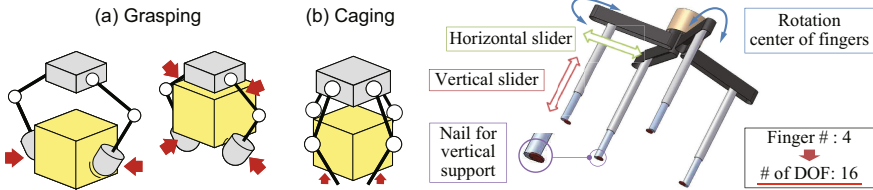
As for the first problem, thanks to the recent technology improvement of stereo cameras or depth imagers, it becomes feasible to acquire geometrical information from sensors and control a robot based on the acquired information[12]. However, they suffer from problems of eye direction and occlusion and cannot get full and precise information. To overcome the model insufficiency, some researchers compare the acquired information with data-base, and construct a manipulation strategy based on the limited information[5, 2, 1].

Regarding the second problem, in traditional robotic manipulation, “Force Closure” is the basis of the manipulation, but the simplified model is too fragile to address practical problems. In contrast, some researches developed tactile contact sensors to realize human-like haptic sense[8, 6]. But the human haptic sense is so complex that it is still far from full imitation of the human capability.

Consequently we focus on more flexible manipulation framework “Caging”[9, 7, 10]. As shown in Fig. 2, caging can constrain an object in its “cage” geometrically. The caging condition can allow small gap between the robot hand and the object. That means the caging manipulation can escape from the force control. In addition, the caging condition is independent from the object’s surface properties.

## 1.2 Problem Statement

If we develop an anthropomorphous caging hand as shown in Fig. 3, the number of actuators is **16**. Comparing with a simple gripper hand that needs only one actuator, 16 actuators are too redundant. Therefore, this paper discusses design of more sophisticated and concise framework for the caging manipulator.



**Fig. 2** Conceptual images of grasping and caging **Fig. 3** Conceptual image of anthropomorphous caging hand

## 2 Technical Approach and Basic Design

First of all, in the caging condition, constant contact between the end-effectors and an object is unnecessary. To utilize the advantage of its loose restriction, our manipulator surrounds the object gradually by moving constraint structures one by one. The basic idea of the framework is inspired by Robotworld[14]; where several robots work in the shared workspace. To realize a caging operation, following functions are required.

**Function 1:** To measure the position and shape of the target object.

**Function 2:** To plan the alignment of the constraint structures that leads to caging condition.

**Function 3:** To locomote the constraint structures one by one.

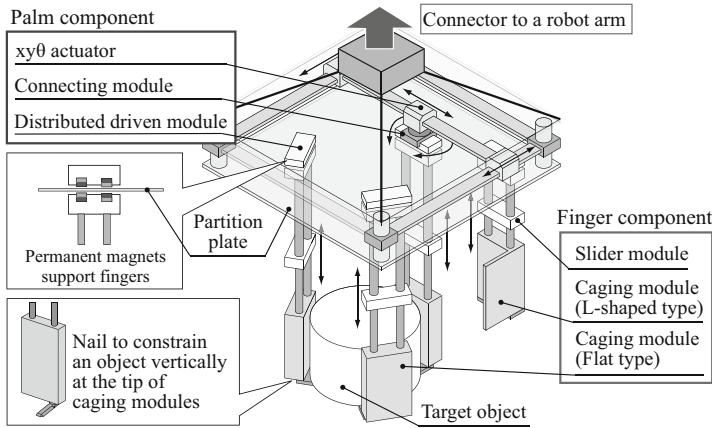
**Function 4:** To constrain the object in required directions.

**Function 5:** To transfer the object while keeping the caging condition.

It is effective to reduce the required number of actuators in Function 3 for a concise framework. To realize the reduction, permanent magnet inductive traction method [13] can be a powerful key technology. Fig. 4 shows the proposed framework. In the framework, each function is assigned to each component. We designate this framework as “distributed end-effectors”. The end-effectors comprise following three components.

### (1) Sensing Component

A depth imager (e.g. Microsoft Kinect, Swiss Ranger) and two dimensional code (2D-Code) reader are the main instruments of this component. They acquire the shape and position of target object.



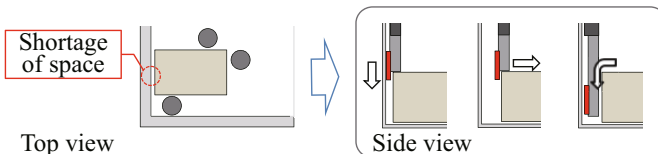
**Fig. 4** Framework of the proposed end-effectors (caging manipulator)

(2) Palm Component

This component consists of a  $xy\theta$  actuator, distributed driven modules and a connecting module. The distributed driven modules support the finger components under the partition plate with permanent magnet pairs. The connecting module is installed at the motion part of the  $xy\theta$  actuator, and it makes connection with the distributed driven module by inserting connection pins. The connecting module also drives the vertical motion of the slider module in the finger component. This palm component is the key part to save the number of required actuators.

(3) Finger Component

This is the main component to constrain the target object. Horizontal constraint is realized by a body plate of the finger front. In contrast, vertical constraint is realized by a nail installed at the bottom of the component. When an object is too close to a neighbor object, another nail is inserted at the edge of the target object and the palm component pushes the object to make enough space for inserting the caging module (i.e. the distributed fingers) as shown in Fig. 5.



**Fig. 5** Conceptual image of drag function

## Summary of the End-Effectors' Framework

We use names “palm” and “finger” to make it easy to understand the framework, however, the structure itself is quite different from human hand. Especially almost all motion axes are orthogonal to each other, therefore, it is very easy (intuitive) to design the finger alignment strategy. Fig. 6 shows the task flow. In this framework, the number of actuators is **nine** (one in each four fingers + five in the palm). This is drastically concise comparing with the anthropomorphous caging hand that has 16 actuators.

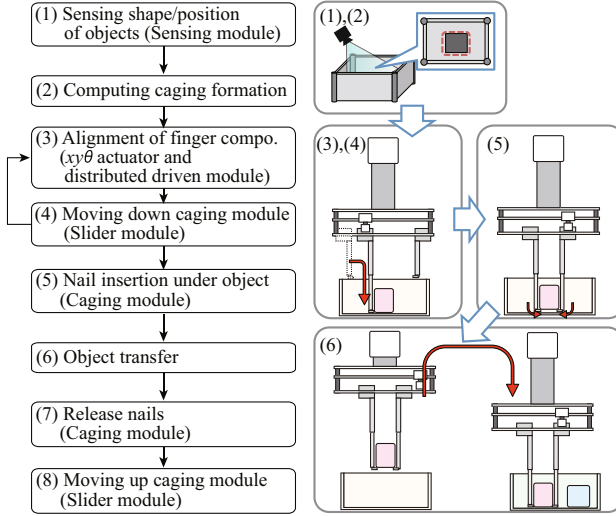


Fig. 6 Task flow of the caging manipulator

## 3 Development of Finger Component Prototype

Fig. 7 shows the developed finger component. Its weight is 300 [g], and the thickness of the caging module is 10 [mm]. The caging module gives geometric constraint to a target object, and the slider module drives the caging module vertically.

### 3.1 Design Details of the Caging Module

The caging module has three functions; (a) To give horizontal geometric constraint, (b) To give vertical geometric constraint and (c) To drag the object horizontally.

The side body plate of the caging module realizes function (a). The plate has two variations as shown in Fig. 8. If the object is hexahedron family, it can be caged with two L type caging modules. Because the L type caging modules is equivalent to two flat type caging modules, but it is faster and easier to operate. A horizontal nail realizes function (b). The nail is installed at the bottom of the caging module and



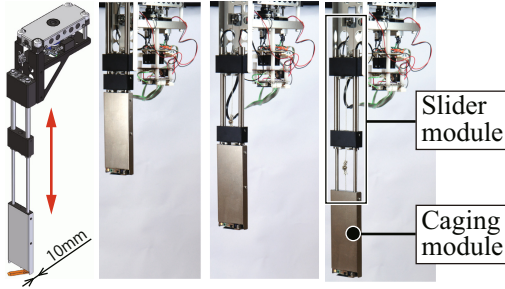


Fig. 7 The developed finger component

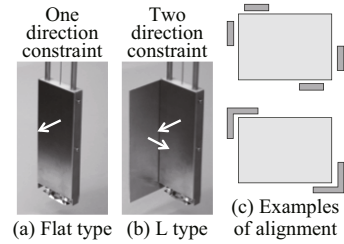


Fig. 8 Variations of caging module

it is rotated and inserted under an object to support weight of the object. Function (c) is realized by inserting a thin nail between the wall of a box and an object (or between objects), and by dragging the object horizontally as shown in Fig. 5.

The most important design key point is how to actuate the two nails by one actuator. As shown in Fig. 9, the both nails are driven by gear transmission mechanisms. The center gear (b) has half non-toothed part, and the part makes it possible to select/switch a driving nail.

### 3.2 Design Details of the Slider Module

The slider module realizes height control of the caging module. Generally speaking, precise height control is essential because the height of the caging module has a large effect to the nail insertion force. However, it is very difficult to detect precisely the bottom boundary of an object. That means precise vertical mechanisms such as lead screw and rack & pinion are useless.

The caging module hangs via a wire, and is driven vertically by winching the wire as shown in Fig. 10. The connecting module rotates a pulley to winch the wire via a magnet coupling. This crane mechanism needs a tensioner to keep the wire tension constant, and the tensioner can detect a contact with the floor or an object.

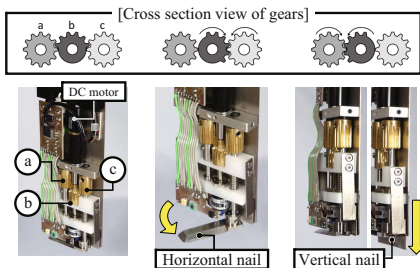


Fig. 9 Motion of horizontal and vertical nails

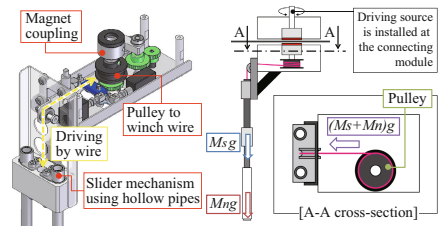


Fig. 10 The crane mechanism using wire and magnet coupling

The crane mechanism is suitable for making a coherent state between the caging module and the floor. Therefore, it does not need precise height control to align the nail at the boundary.

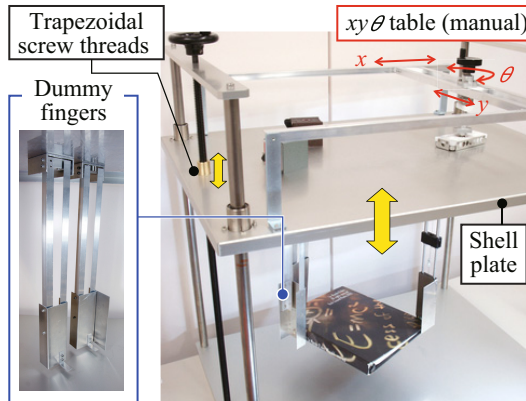
## 4 Experiments

### 4.1 Feasibility Test of Caging Manipulation

Fig. 11 shows the experimental test bench to evaluate the feasibility of caging manipulation. The test bench can imitate a vertical pick-up motion (One DOF) of a robot arm that is equipped with the distributed end-effectors. As a driving source of the distributed driven modules, a manual  $xy\theta$  table was implemented. One developed finger component and two dummy fingers are used for constraining an object. The dummy finger has the same dimension (size) and hangs under the partition plate with permanent magnets. But it is not installed with the actuators and sensors, therefore, the dummy finger needs to be actuated by human hands.

Eight kinds of daily-use objects are selected as manipulation target objects. Fingers' alignment is designed empirically as shown in Fig. 12. As a basic idea, formation (a) is applied for cylinder shape, formation (b) for general hexahedron and formation (c) is applied for hexahedron with high aspect ratio (thin box). In the future, we will use the our developed algorithm [16] to realize automatic planning.

The experimental procedure is as follows; (1) To align the finger component and dummy fingers around the object, (2) To control the height of slider module and insert the horizontal nail under the object, (3) To brake the slider and simulate the arm vertical motion, (4) To check the robustness of the caging by applying external force from the outside of the cage. By executing this experiment, it was confirmed that all eight objects can be caged and resist against external force even if there are 5 [mm] margins between the caging module and the object. Fig. 13 shows the experimental results.



**Fig. 11** Experimental environment

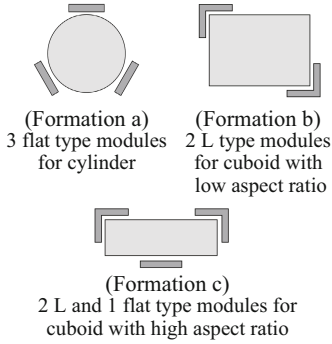


Fig. 12 Caging formations



Fig. 13 Experimental results (Snapshots of constrained target objects)

### 4.2 Performance Experiment of Horizontal Nail Insertion

The horizontal nail insertion is the most uncertain process in the task flow. We evaluated its performance by experiments. Fig. 14(a) shows the experimental setup where  $\theta$  is the angle of chamfer and  $M$  is the object mass. Chamfered distance is fixed at

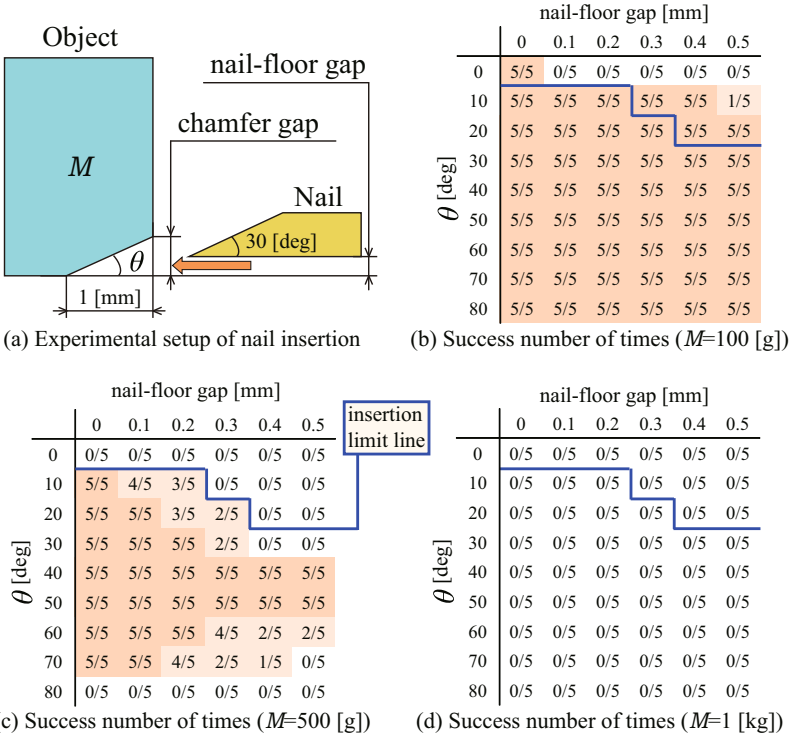


Fig. 14 Results of the horizontal nail insertion experiment

1 [mm], meanwhile the angle of chamfer is varied. The horizontal nail is actuated by a DC motor. Each configuration is examined five trials, and each success rate is counted.

Fig. 14(b, c, d) is the results of the nail insertion when the object mass ( $M$ ) is 100 [g], 500 [g], and 1 [kg] respectively. In the table, each box is painted in deep orange where the nail insertion succeeded in all five trials, in light orange where it succeeded from one to four trials, and white where all trials failed.

A gap made by chamfer (chamfer gap) is the allowable limit of nail-floor gap to insert the nail. Blue lines (insertion limit line) are boundary lines between an area where the nail-floor gap is smaller/larger than the chamfer gap. In Fig. 14(b), the nail was inserted over insertion limit line, and this is because the object was lift up by the insertion force. In Fig. 14(c), the insertion limit line has enormous influence, and the nail insertion becomes impossible outside of it. In addition, the angle of chamfer ( $\theta$ ) also has influence on the nail insertion, and its influence is remarkable when the nail-floor gap is large. Smaller nail-floor gap makes it easier to insert the nail especially when  $\theta$  is between 10 [deg] and 70 [deg]. Unfortunately, the nail insertion under the 1 [kg] object was impossible, although the motor has theoretically sufficient power to insert the nail.

## 5 Conclusion

Thorough this work, we have two main experimental insights as following.

### (1) What are OBSTACLES for our caging manipulator?

Through the development and experiments, we have found four obstacles for our manipulator.

- (1) Size/Dimension of the horizontal nail in the caging module
- (2) Stiffness of the slider module and the partition plate
- (3) Three dimensional rotation of the target object with high aspect ratio
- (4) Too small gap or chamfer between the target object and the floor

**(1) Size/Dimension of the horizontal nail:** The horizontal nails are important parts to support weight of the target object. However the size is restricted by the thickness and width of the caging module. In addition, the size of the horizontal nail has large relation to the allowable margin/gap between the caging module and the target object. Consequently in the experiment A, we needed more number of fingers comparing with initial intuition. That is, we used three finger components to constrain the objects with high aspect ratio.

**(2) Stiffness of the slider module and the partition plate:** When inserting the horizontal nail under the target object or lifting up the object, vertical force is applied to the tip of caging module. To reduce the unintended deformation of the caging module, not only the slider module but also the partition plate should be stiff enough. This is because the bend of the partition plate can induce the misalignment of the

caging module. In current condition, we selected aluminum plate for the partition, but in the next prototype we will try more stiff material such as non-magnetic stainless steel.

**(3) Three dimensional rotation of the target object:** In the feasibility experiment, two L type caging modules and one flat type caging module are necessary to cage an object with high aspect ratio. If we use only two L type caging modules, the end-effectors need to support the vertical force (weight) at the opposing corners. However in this condition, the object rotates in the axis of the diagonal line. Consequently we need to introduce a sensing process and a finger alignment algorithm that is suitable for manipulating an object with high aspect ratio. Or we can introduce the “grasping by caging”[11, 15] technology to build a loose-contact-based grasping by starting from contact free caging.

**(4) Too small gap or chamfer:** The horizontal nail insertion is the only process that needs to consider the effect of friction. In the experimental result (Fig. 14(d)), it is found that the motor requires more power than expected. There is tiny round part at the tip of the horizontal nail, consequently the round part may collide with the chamfer at the bottom of the target object. To overcome the problem, we need to execute more experiments and estimate the uncertain effect of the friction between the nail and target object.

**(2) Advantages and disadvantages of caging manipulation**

Table 1 summarizes the advantages and disadvantages of our proposed caging manipulator compared with the traditional grasping manipulator. The grasping approach has a large advantage in its versatility, hence many researchers adopt this

**Table 1** Summary of qualitative comparison in grasping and caging manipulators

Approach	Grasping (Force Closure)	Caging
Principle	Force constraint by grasping force or frictional force	Geometrical constraint by Caging
Advantages	<ul style="list-style-type: none"> <li>• Can control object posture inside the hand</li> <li>• Can be realized with small number of actuators only for power grasp (e.g. 1 DOF gripper)</li> <li>• Possible to manipulate soft/deformable objects</li> </ul>	<ul style="list-style-type: none"> <li>• Allow substantial error of a geometrical object model</li> <li>• Needless of constant contacts between end-effectors and an object</li> <li>• Needless of force control</li> <li>• Can manipulate a solid object regardless of its surface properties</li> <li>• Wide range of the manipulation target size (Large object)</li> <li>• Concise structures compared with anthropomorphic robot hands</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>• Allow little error of a geometrical object model</li> <li>• Need sophisticated force control to realize constant contact</li> <li>• Difficult to configure optimal internal force</li> <li>• Difficult to evaluate manipulation stability in its operation</li> </ul>	<ul style="list-style-type: none"> <li>• Impossible to control its posture inside the hand</li> <li>• Difficult to manipulate soft/deformable objects</li> <li>• Need to prepare multiple structures to realize a solid cage</li> <li>• Complex structures compared with 1 DOF gripper</li> </ul>

strategy. In contrast, the caging manipulator doesn't need force control and it is robust against the surface properties of the target object. Unfortunately a caging manipulator may not be good at operation of a soft and deformable object. Human-like robot hand has a large potential to realize versatile manipulation, in contrast, the proposed caging manipulator is promising to perform stable object manipulation that cannot be realized by human-like robot hand.

## References

1. Bohg, J., et al.: Mind the gap - robotic grasping under incomplete observation. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 686–693 (2011)
2. Brook, P., et al.: Collaborative grasp planning with multiple object representation. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 2851–2858 (2011)
3. D'Andrea, R., et al.: Future challenges of coordinating hundreds of autonomous vehicles in distribution facilities. In: Proceedings of IEEE International Conference on Technologies for Practical Robot Applications, pp. 80–83 (2008)
4. Fukui, R., Mori, T., Sato, T.: Home-use object transfer/storage robot system with compliant strategy and mechanism (commodities management and its extended application of daily life support for the elderly). *Journal of Robotics and Mechatronics* 23(4), 532–543 (2011)
5. Goldfeder, C., et al.: The Columbia grasp database. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 1710–1716 (2009)
6. Hasegawa, H.: et al.: Robot hand whose fingertip covered with net-shaped proximity sensor / moving object tracking using proximity sensing. *Journal of Robotics and Mechatronics* 23(3), 328–337 (2011)
7. Makita, S., et al.: 3D multi-fingered caging: Basic formulation and planning. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2697–2702 (2008)
8. Noda, K., et al.: A shear stress sensor for tactile sensing with the piezoresistive cantilever standing in elastic material. *Sensors and Actuators A* 127(2), 295–301 (2006)
9. Pereira, G.A.S., Campos, M.F.M., Kumar, V.: Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research* 23, 783–795 (2004)
10. Rodriguez, A., Mason, M., Ferry, S.: From caging to grasping. In: Proceedings of Robotics: Science and Systems (2011)
11. Rodriguez, A., Mason, M.T., Ferry, S.: From caging to grasping. *International Journal of Robotics Research* 31(7), 886–900 (2012)
12. Rusu, R.B., et al.: 3D is here: Point cloud library (PCL). In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 6500–6503 (2011)
13. Sato, T., Fukui, R., Morishita, H., Mori, T.: Construction of ceiling adsorbed mobile robots platform utilizing permanent magnet inductive traction method. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 552–558 (2004)
14. Scheinman, V.: Robotworld: A multiple robot vision guided assembly system. In: Proceedings of the 4th International Symposium on Robotics Research (1987)

15. Wan, W., Fukui, R., Shimosaka, M., Sato, T., Kuniyoshi, Y.: Grasping by caging: A promising tool to deal with uncertainty. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 5142–5149 (2012)
16. Wan, W., Fukui, R., Shimosaka, M., Sato, T., Kuniyoshi, Y.: On the caging region of a third finger with object boundary clouds and two given contact positions. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 4154–4161 (2012)

# Experiments in Underactuated In-Hand Manipulation

Lael U. Odhner, Raymond R. Ma, and Aaron M. Dollar

**Abstract.** This paper shows conceptually and experimentally that underactuated robotic hands can stably grasp and manipulate objects placed between the fingertips. Small objects grasped between a planar pair of two-link underactuated fingers having one tendon each are shown to be manipulable over a range of in-hand configurations that can be predicted analytically. The manifold of predicted stable configurations is found by seeking the minimum energy configuration of the elastic fingers under constraints from the actuator tendons and the contact constraints with the grasped objects. Experimental results are shown from HANDLE, a novel underactuated hand capable of a variety of dexterous in-hand tasks.

## 1 Introduction

Although many years of effort have gone into the development of anthropomorphic, highly-actuated hands, many of the hands in regular use by consumers and researchers (those not directly involved in the design of hands) are parallel jaw grippers, such as those used by the Kuka YouBot [1] or the Willow Garage PR2 [2], or hands with simplified multilink fingers, such as the Barrett Technologies BarrettHand [3] or the Robotiq Adaptive Grippers [4]. This widespread preference for simple hands is due in large part to limitations on the force capabilities of most arms; the durability of simpler hands compared to more complex ones is also a compelling factor. However, the advantages of simple hands come at a cost: such hands are generally designed to acquire and maintain static grasps, sacrificing the ability to manipulate objects within the hand.

As part of a collaboration with iRobot Corporation and Harvard University, the authors have been working on the design of a new robotic hand for the DARPA

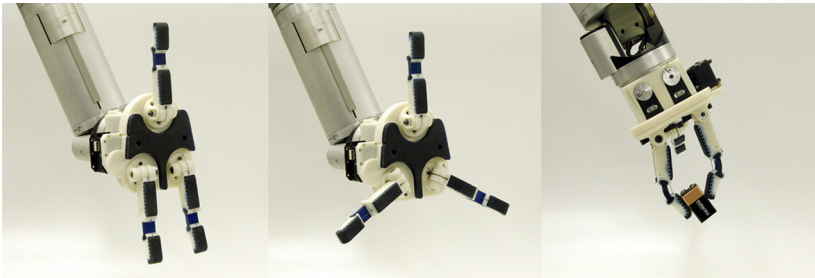
---

Lael U. Odhner · Raymond R. Ma · Aaron M. Dollar  
Yale University, Department of Mechanical Engineering and Material Science,  
New Haven, CT, USA  
e-mail: {lael.odhner, raymond.ma, aaron.dollar}@yale.edu



ARM-H program. ARM-H was conceived to design next-generation robotic hands that are simple enough for widespread use, but also capable of performing more demanding manipulation tasks. The resulting product, the Hardened Adaptive Novel Low-cost End-effector (HANDLE), has five actuators and is packaged so that it fits on the end of a Barrett WAM, and is shown in Fig. 1. Our work on HANDLE has focused on designing underactuated fingers that are capable of manipulating a grasped object within the workspace of the hand. Several underactuated hands in the literature can pinch an object by locking the finger joints against clutches or hard stops [5, 6], or by introducing a concave shape to the fingertip [7]. The results presented in this paper improve on the state of the art demonstrating an underactuated hand capable of moving a pinched object continuously in the plane defined by two opposed fingers. No clutches or hard stops are used; instead, the passive elastic properties of the hand are exploited to alter the minimum energy configuration of a pinched object.

This paper is organized in the following order: In Section 2, the design of the HANDLE fingers will be introduced, and the mechanics of obtaining a two-fingered pinch grasp with the hand will be explained. Section 3 discusses how the underactuated fingers can be used to reposition a pinched object within the workspace of the hand. Because of the limited number of actuators, not all object configurations can be reached. However, the manifold of reachable configurations can be computed using a grid search algorithm. Finally, Section 4 shows experimental results from HANDLE manipulating several small objects of varying width and curvature.

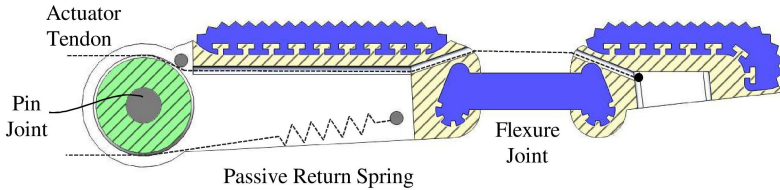


**Fig. 1** The iRobot/Harvard/Yale HANDLE is a novel lightweight robot hand capable of performing a variety of precision grasping and in-hand manipulation tasks

## 2 Finding Stable Pinch Grasps with HANDLE

The design of HANDLE was derived from the SDM Hand, an underactuated hand whose fingers are molded as a single solid piece [8]. Similarly, the fingers of HANDLE, shown in Fig. 2, employ in-molded polyurethane flexures as joints between the proximal and distal finger links. The proximal links are attached to the hand with pin joints via a free-wheeling pulley. The fingers are actuated by a single flexor tendon running the length of the finger. The pulley radius on the proximal joint and

the distance from the tendon to the flexure midplane on the distal joint are both 9 mm, so that the actuated torque on the two joints is approximately equal. The elastic restoring force on the distal joint is provided by the polyurethane flexure; a return spring attached to the proximal joint is tuned to provide 1/5 the angular stiffness (Nm/rad) of the distal joint, in accordance with previous optimizations for reliable underactuated grasping [9].



**Fig. 2** The fingers on HANDLE are underactuated, having a single flexor tendon which runs the length of the finger

## 2.1 Acquiring Stable Pinch Grasps

HANDLE has been experimentally found to be capable of acquiring pinch grasps on a range of objects, such as the 9 Volt battery shown at the right in Fig. 1. This grasp is not a planar form closure in the sense that the actively exerted forces do not fully span the space of wrenches on the grasped object. Nonetheless, the grasp is stable and repeatable because the fingers seek a reliable minimum energy configuration while satisfying contact constraints between the fingertips and the objects. In this section we will explain the modeling assumptions used to predict the stability and equilibrium position of the hand while pinching an object.

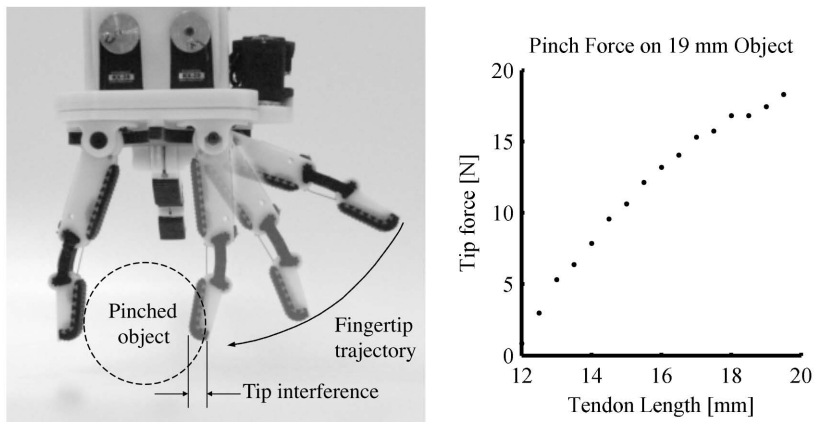
The behavior of the HANDLE fingers is similar to that of other serially underactuated fingers, such as the linkage-based fingers on SARAH [10], or the pulley-based fingers on Hirose’s soft gripper [11]. As a finger is actuated, it closes inward on an arc defined by the minimum energy of the finger for the applied tendon length constraint [12]. Because of the underactuated design, contact on the finger from an object will deform the finger, and produce an elastic reaction force at the point of contact proportional to the magnitude of the disturbance. Due to the design of the fingers on HANDLE, this contact force is large enough that an object in between two fingers will see a significant contact force when two fingers sweep in to a rest position which interferes with the object, as seen at left in Fig. 3. The plot in Fig. 3 shows a plot of the measured internal forces on a grasped object (a 19 mm wide load cell) as a function of tendon excursion, starting at 12 mm of tendon travel, when the fingers first touch the object. The peak fingertip force of the prototype hand shown here is 19 N for an object of this size.

## 2.2 Modeling Stable Pinch Grasps

To evaluate the stability of a grasp, assumptions must first be made about the disturbance forces the object might experience. The rank-based criterion for assessing force closure stability (that the positive span of the wrenches on the grasped object encompasses an arbitrarily large disturbance) is not useful, as the compliance of the underactuated fingers limits the magnitude of the forces that can be applied to the object. Instead, we assume that disturbance forces will be small. This is actually not a bad assumption to work under for the manipulation of small objects; the effect of gravity is often insignificant, especially if the pinch forces on the object are sufficient, and many such tasks occur in environments where unplanned collisions are unlikely. Under such circumstances, we will consider a grasp to be stable if it lies at an energy minimum with respect to the configuration of both the hand and the grasped object [16]. To ascertain whether this is true, a model of the hand was constructed using the Freeform Manipulator Analysis Toolkit (FMAT), an extensible toolkit for quasi-static modeling of rigid-body and flexible mechanisms [13]. FMAT is based on energy-based models of rigid links and deformable elastic bodies, and finds the constrained equilibrium configuration  $\mathbf{q}^*$  of any mechanism having a configuration vector  $\mathbf{q}$  by solving the energy minimization problem:

$$\mathbf{q}^* = \arg \min_{\mathbf{q}, \lambda} (U(\mathbf{q}) - \lambda^T \mathbf{c}(\mathbf{q})) \quad (1)$$

Here  $U(\mathbf{q})$  is the potential energy in the fingers as a function of elastic joint deformation, and  $\mathbf{c}(\mathbf{q})$  is a vector of constraints on the hand and the grasped object having a set of corresponding constraint forces  $\lambda$ . The links of the finger were constrained by the tendons (a constraint on the length of the tendon running along each finger).

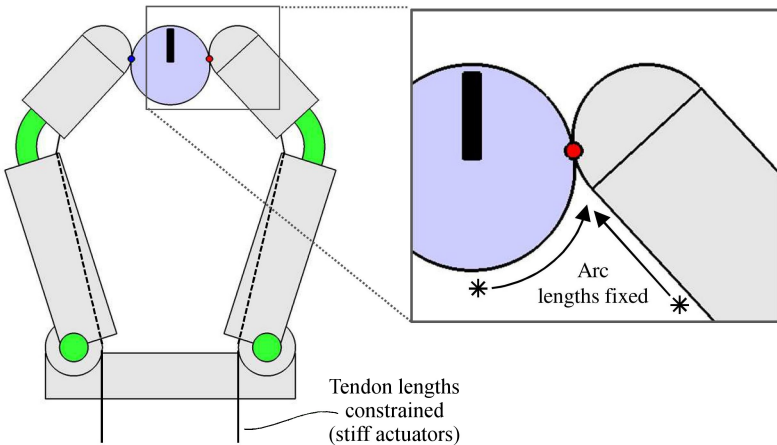


**Fig. 3** Internal force in an underactuated pinch grasp is generated by driving the fingers to an equilibrium configuration which interferes with the pinched object

The fingertip constraints were represented as normal constraints and no-slip rolling conditions between a grasped object and the fingertips. These contact conditions were modeled by constraining the difference in arc length along the surface of the grasped object and the arc length along the fingertip (shown in Fig. 4). To find the initial position of the no-slip contact, the configuration of the object was assumed to be the symmetric configuration at which the fingertips just touch the object; this initial pose was used to provide an initial condition for rolling constraints. The models of the fingers are available for download along with the Freeform Manipulator Analysis Toolkit [13]. Modeling a hand with continuum joints such as those on the SDM Hand and HANDLE is difficult; a bending flexure has in principle an infinite number of degrees of freedom. The FMAT implements a model previously developed by the authors relying on the fact that the Euler-Bernoulli bending of a flexure, while high-dimensional, is smooth, and thus can be projected on a basis of smooth curvature functions [14]. Using this model, the in-plane behavior of the flexure can be adequately represented by only three parameters,  $q_1, q_2$  and  $q_3$ , which do not correspond to serial joint angles in the same way as in a pseudo-rigid body model [15]. Instead, the effect of each parameter is distributed along the length of the flexure. This model is more convenient than a pseudo-rigid body model because the energy is quadratic with respect to the configuration parameters:

$$U_{dist} = \frac{EI_{dist}}{2} \left( q_1^2 + \frac{q_2^2}{3} + \frac{q_3^2}{5} \right) \quad (2)$$

Here  $EI_{dist}$  represents Young's modulus and the bending moment of the distal flexure joint. The kinematics of the flexure joints are also represented in terms of these parameters; details on the computation of the finger kinematics as a function of the smooth curvature basis can be found in [14].



**Fig. 4** Once an internal force is maintained on the grasped object, no-slip rolling contact was assumed between the object and the finger pad. The position-controlled tendons are treated as rigid constraints.

The rotation of the proximal finger joint is described by a fourth parameter  $q_4$ , and the joint stiffness by  $k_{prox}$ :

$$U_{prox} = \frac{k_{prox}}{2} q_4^2 \quad (3)$$

In addition to these parameters needed to describe the deformation of each finger, the in-plane configuration of the pinched object  $(x, y, \theta)$  must be added to the vector of configuration parameters for the hand/object system. These configuration parameters do not carry any associated energy if gravity is negligible, but interact with the joint energy through the fingertip constraints. Because all of the energy functions associated with motion of the hand and grasped object are quadratic functions of the configuration parameters, finding a static equilibrium position using Castigliano's theorem is equivalent to solving a quadratic program with nonlinear constraints corresponding to the fingertip constraints and the constraints imposed by the flexor tendons. The Freeform Manipulator Analysis toolkit uses Matlab's optimization toolkit to find these equilibrium positions.

### 2.3 Assessing Constraint Validity

Given an equilibrium configuration and the constraint forces on tendons and fingertips, the validity of the grasp can be further assessed by checking the conditions on each constraint. For example, normal forces between the fingertips and a grasped object must be positive in order to maintain contact; the tendon forces must be negative. Additionally, limits must be set to ensure that a pinched object cannot roll off the fingertips, either into or out of the hand. A routine was written to automatically check these kinematic and kinetostatic bounds on stability, returning a boolean value determining stability. No-slip contacts were not considered in this stability check, because the coefficient of friction with a surface varies from object to object. Instead, the stability check routine returns a scalar value  $\mu_{crit}$  corresponding to the minimum coefficient of friction needed to keep the no-slip contact:

$$\mu_{crit} = \max \left( \frac{\lambda_{s,i}}{\lambda_{n,i}} \right) \quad (4)$$

Here  $\lambda_{s,i}$  and  $\lambda_{n,i}$  are the shear and normal forces at the  $i$ th no-slip contact between the fingertips and the object.

### 2.4 Summary

Modeling the configuration of the fingers and grasped object in HANDLE's stable pinch grasps can be posed as a problem of energy minimization, in the same fashion one might analyze the deformation of a monolithic rigid body under constraints. The critical difference in the case of pinch grasp modeling is that the necessary conditions for the fingertip constraints must be met if a pinch grasp is to remain

stable. These conditions have been expressed in a form where grasp stability can be quantified, in terms of the boolean conditions (normal constraints and tendon tension constraints), and in terms of scalar conditions (minimum friction coefficient needed for stability).

### 3 Manipulation

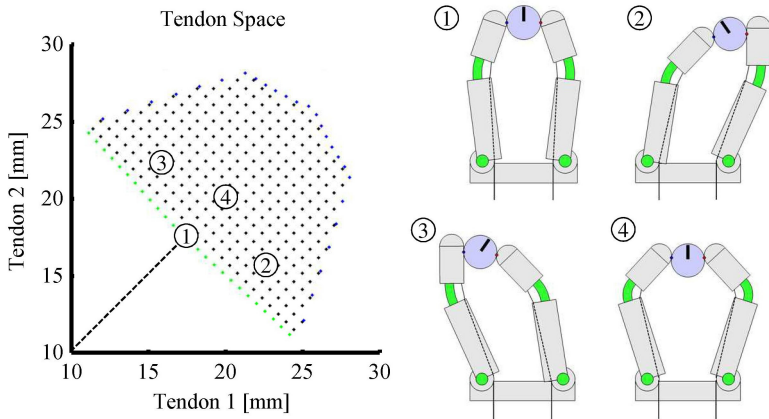
From the model just developed for finding equilibrium configurations of a pinched object, it is straightforward to generalize to a model of manipulation within the workspace of the hand. Modeling manipulation is a matter of determining the change in configuration of an object held between the fingers as the tendon length constraints are varied. Thus, one manipulates with the underactuated hand by using the tendon constraints to shift the equilibrium configuration of the grasped object. Another way of looking at this is to see the manifold of tendon excursions and the manifold of object configurations as related by some mapping, which is a homeomorphism if the constrained energy function is convex and thus returns unique solutions. However, because there are only two actuators in the pinching fingers described here, the entire manifold of object configurations  $(x, y, \theta)$  will not be reachable. Instead, it will be a sub-manifold embedded in this space.

#### 3.1 Manifold Exploration

In order to understand what the manifold of manipulable object configurations looks like, methods of discrete approximation were developed. By slowly contracting the finger tendons to the point where the fingers barely make contact with the grasped object, a single stable point on the manifold can be identified. A grid of points is then expanded out from this initial point (marked as point 1 in Fig. 5). At each point, the boolean stability condition for the grasp is checked and the minimum necessary friction coefficient  $\mu_{crit}$  is identified. The approximated manifold of manipulable configurations resulting from this process is shown in Fig. 6 for a 25 mm diameter cylinder. At left, the projections of the manifold onto the  $xy$  plane and the  $x\theta$  plane show that, the rotation of the object is antisymmetric about the center of the object, and that the approximate magnitude of the largest rotation possible is close to 1 radian. The gradient coloring on the manifold shows the quality of the grasp as determined by the value of  $\mu_{crit}$ .

Several features of interest can be observed on this manifold. First, the coefficient of friction needed to maintain a pinch grasp increases as the tendons are both pulled. This can be explained by examining configuration 4 on Fig. 5. As the fingers pinch inward on a round object, the fingertips roll up onto the outside of the object. Thus, the contact forces tend to eventually eject the object inward toward the palm. This is sometimes a desirable property; often one acquires a small object in a pinch grasp and then wishes to transition into a power grasp further in the palm. Another feature of interest is the semicircular region in the  $xy$  projection, corresponding to the fingertip grasps at the edge of the workspace. Looking at this edge on the  $x\theta$

projection, it is clear that the object can be rolled back and forth in this region to rotate it to nearly the full extent possible. This information could either be used for to tailor the finger design (to ensure that a particular class of object can be rolled between the fingers over a desired range) or to plan manipulation operations such as finger gaing.

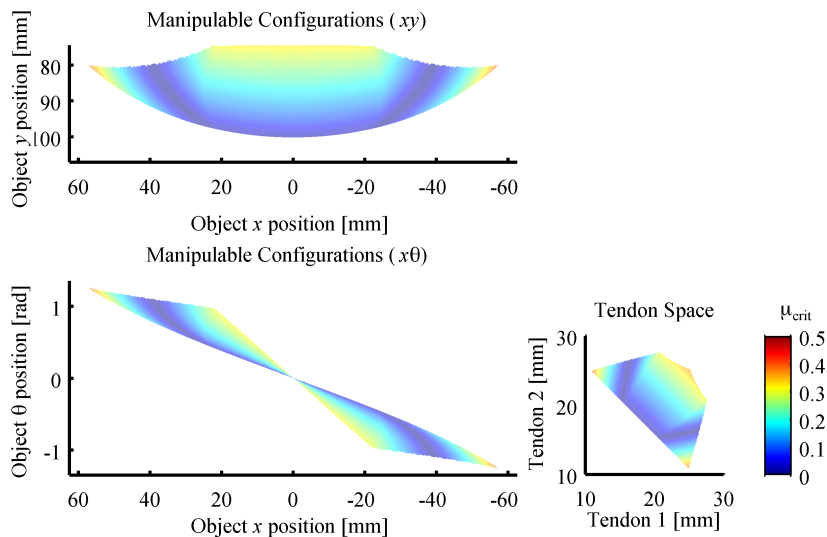


**Fig. 5** The manifold of manipulable configurations for an object can be found by constraining the object at its initial point of contact (1), and varying the tendon constraints to map out a grid of stable configurations. Some examples of the corresponding pinched configurations are shown at right. The origin in tendon excursion space corresponds to the fingers completely extended.

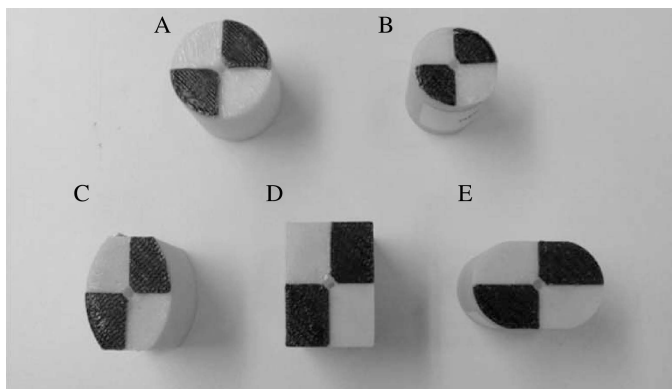
## 4 Experiments

In order to understand how the *a priori* predictions of manipulability made in the previous section relate to the behavior of the real robot hand, several experiments were conducted with a prototype of HANDLE. Several small objects were selected, each having a different width and radius of curvature. The test objects, shown in Fig. 7, were printed on a Stratasys 3D printer. A TrakStar measurement system from Ascension Technology was used to measure the position and orientation of each object.

At the start of each test, the HANDLE manipulator was placed in a planar configuration as shown in Fig. 8 and used to grasp the objects in a centered pinch grasp. The object was placed so that the fingers could make contact without moving the hand. The third finger perpendicular to the plane of the table was held fixed, and not used for the experiment. The path of the tendon trajectory for each object was separately calibrated in order to determine the largest looping trajectory within the hand's workspace that could be realized with without a significant amount of slip. Four waypoints along each object's test trajectory were tuned so that the object



**Fig. 6** The manifold of manipulable configurations for a 25 mm cylinder is shown, along with the corresponding stable region in tendon space. The coloring on the plot indicates  $\mu_{crit}$ , the minimum friction coefficient needed for stability. The origin on the object configuration corresponds to the center of the palm.



**Fig. 7** A series of test objects used to test the predictive value of the pre-computed manipulability mappings

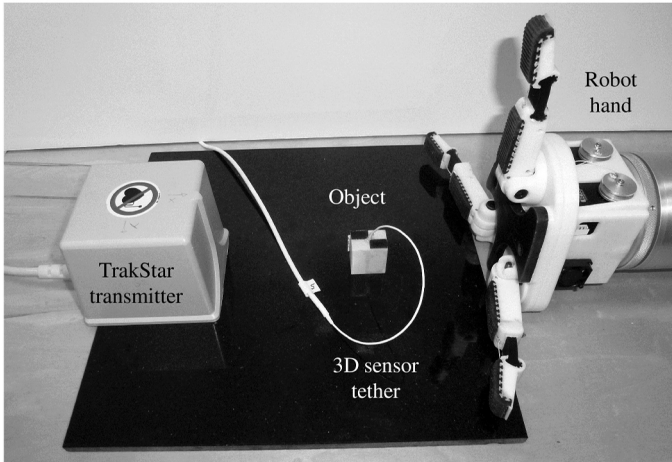
experienced less than 3mm of net displacement after returning to the initial pinch grasp. During this empirical workspace identification, the position and orientation of the object in Cartesian workspace were also recorded for each set of actuation commands.



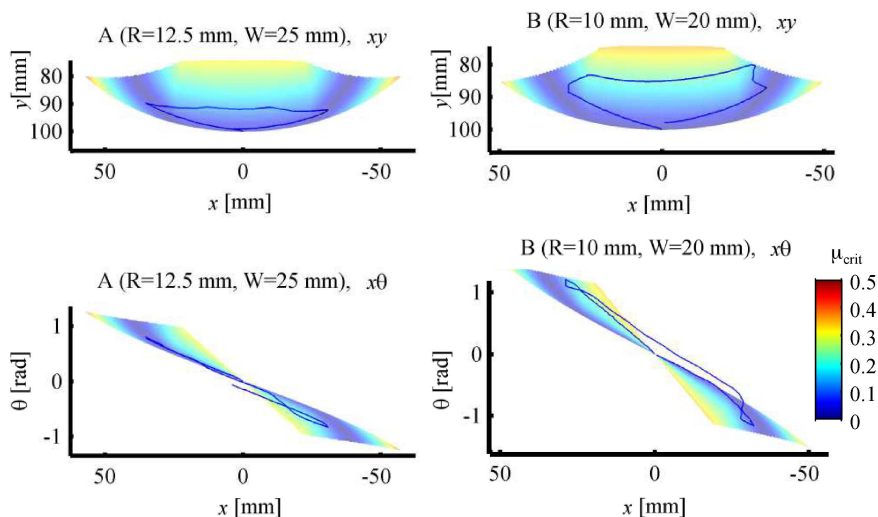
## 4.1 Results

The results of a transit around the workspace with the five objects are shown in Figures 9, 10, and 11. The trajectory of each object is shown superimposed on the computed manifold of manipulable configurations, in the  $xy$  and  $x\theta$  projections. One salient feature from these results was the degree to which the empirically determined “safe” trajectories stayed mostly within regions in which the required coefficient of friction  $\mu_{crit}$  was smaller than about 0.3. This was not as predictive for the objects having larger radii of curvature (such as objects C and D), but supports the idea that hands designed for rolling small objects should be designed with an eye toward this performance metric.

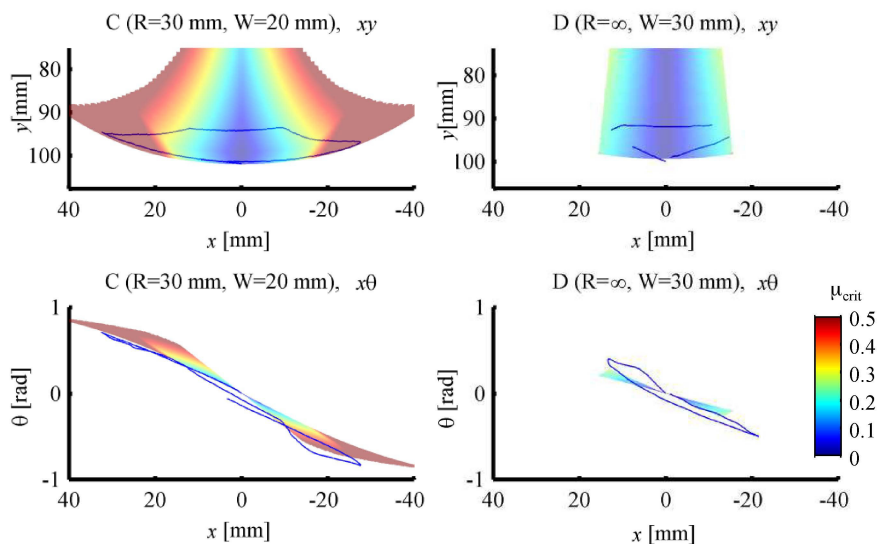
Another significant result was the degree to which the relative angular rotation rate with respect to  $x$  and  $y$  travel tracked the theoretical predictions. The curves trace out the predicted manifolds, deviating from the surface primarily in discrete slip events, as seen with object B (in Fig. 9) and object C (in Fig. 10). The slip events were noticeable by eye when the tests were performed - typically, the object would reach a point in the workspace where the predicted critical coefficient of friction was higher, and the shearing motion of the broken contact would cause a rotation of the object relative to the fingers. The practical implication of this result is that feed-forward models will be of limited usefulness for planning and controlling object position. This is not terribly surprising; however, in these cases the object was never unstable in a larger sense. It remained safely held between the fingers. With a richer set of tactile sensors for determining contact location on the fingers, one could easily imagine a model-based manipulation framework capable of recovering from this kind of error.



**Fig. 8** The apparatus used to measure the position and orientation of a manipulated object

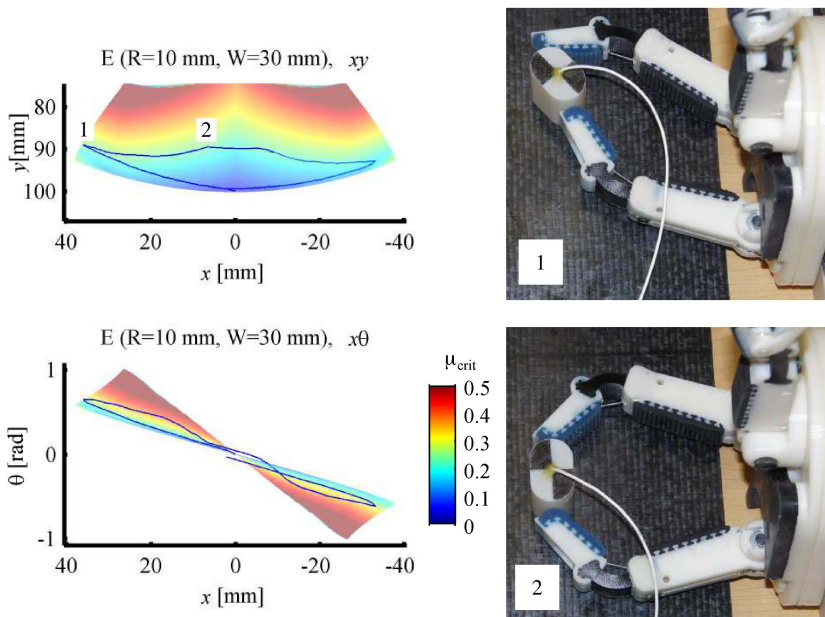


**Fig. 9** Tracking results for two objects, one having a radius of 12.5 mm and a width of 25 mm, (A, left) and one having a radius of 10 mm and a width of 20 mm (B, right). Object B shows a typical angular slip event at the right hand side of the  $x\theta$  projection.



**Fig. 10** Tracking results for objects C (radius 30 mm, width 20 mm) and D (flat edges, width 20 mm). The narrower range of translation and rotation observed in object D were due to properties of the rolling contact for flat objects, compared to the smaller radii in all the other objects.

The final observed source of disagreement between the underactuated manipulability models and the experiments was the the errors caused by breakdown of the ideal contact constraints for configurations near the edge of the workspace. This was particularly true for the flat object (D) and the 30 mm radius object (C). Both of these objects were predicted to have narrow ranges of lateral motion, manifesting as either a massive increase in the required coefficient of friction or a kinematic travel limit, which caused the manifold exploration program to halt. The physical cause of these limits was the limited range of rotation which a flat object can undergo. In practice, the fingertip contact constraints ceased to act as idealized rolling constraints, and instead allowed some slip. For this reason, the angular prediction errors are higher for these objects. Better models for these edge cases could clearly improve the predictive capability of pre-computed manipulability manifolds.



**Fig. 11** Tracking results for object E, and some photographs illustrating the configuration of the hand and object at points along the trajectory

## 4.2 Summary

The experimental results validate the basic premise of the manipulation model put forth here, namely, that altering the equilibrium configuration of a grasped object within the hand can be used to perform basic in-hand manipulation tasks captured by the manifold of manipulable object configurations. The computed manifolds are

representative of the regions in space which can be reached within the workspace of the hand. Errors due to slippage and high internal forces were observed, and tend to be discrete events that do not necessarily affect the stability of the grasped object.

## 5 Conclusions

In this paper, we have shown that fingertip grasping and manipulation of small objects with an underactuated hand is possible, and also practical with no special modifications to a fairly generic fingertip geometry and link configuration for an underactuated hand. The HANDLE fingers can move and reorient small objects within the hand, and the range of motion and rotation of the objects can be approximated with analytical models. The most valuable purpose these results serve is to erode the set of necessary conditions which must be imposed on a robot design in order to consider it fit for manipulation. Underactuated fingers are not capable of obtaining a pinch grasp in which arbitrary disturbance forces can be resisted; however, it is often safe to assume that the acquisition and manipulation of a small object does not involve such disturbances. In these cases, the analysis presented here offers some idea of how the design, planning and execution of in-hand manipulation tasks can be carried out.

**Acknowledgements.** The HANDLE robotic hand was developed in collaboration with Robert Kohout, Mark Claffee and Nick Corson at iRobot Corporation, and Robert Howe, Leif Jentoft, and Yaroslav Tenzer at Harvard University. The authors would like to thank them for their hard work and insights into the many difficult problems encompassed in the area of hand design.

## References

1. Bischoff, R.: KUKA youBot - a mobile manipulator for research and education. In: Proc. Inter. Conf. on Robotics and Automation pp. 1–4 (2011)
2. Willow Garage: PR2 Robot, <http://www.willowgarage.com/> (Cited June 3, 2012)
3. BarrettHand, <http://www.barrett.com/robot/products-hand.htm/> (Cited June 3, 2012)
4. Robotiq Adaptive Robot Gripper, <http://www.robotiq.com> (Cited June 3, 2012)
5. Aukes, D., Kim, S., Garcia, P., Edsinger, A., Cutkosky, M.: Selectively Compliant Underactuated Hand for Mobile Manipulation. In: Proc. Inter. Conf. on Robotics and Automation pp. 2824–2929 (2012)
6. Birglen, L., Gosselin, C.: Kinetostatic analysis of underactuated fingers. IEEE Trans. on Robotics and Automation 20(2), 211–221 (2004)
7. Kragten, G.A., Baril, M., Gosselin, C., Herder, J.L.: Stable Precision Grasps by Underactuated Grippers. IEEE Trans. on Robotics 27(6), 1056–1066 (2011)
8. Dollar, A.M., Howe, R.D.: Joint Coupling Design of Underactuated Hands for Unstructured Environments. Inter. Journal of Robotics Research 30, 1157–1169 (2011)
9. Dollar, A.M., Howe, R.D.: The Highly Adaptive SDM Hand: Design and Performance Evaluation. Inter. Journal of Robotics Research 29(5), 585–597 (2010)

10. Rubinger, B., Fulford, P., Gregoris, L., Gosselin, C., Laliberte, T.: Self-Adapting Robotic Auxiliary Hand (SARAH) for SPDM Operations on the International Space Station. In: Proc. Inter. Symposium on A.I. and Robotics & Automation in Space (2001)
11. Hirose, S., Umetani, Y.: The development of soft gripper for the versatile robot hand. *Mechanism and Machine Theory* 13(3), 251–259 (1978)
12. Birglen, L.: Kinetostatic analysis of underactuated fingers. *IEEE Trans. on Robotics and Automation* 20(2), 211–221 (2004)
13. Odhner, L., Dollar, A.M.: The Freeform Manipulator Analysis Tool, <http://www.eng.yale.edu/grablab/fmat/> (Cited June 3, 2012)
14. Odhner, L., Dollar, A.M.: The Smooth Curvature Model: An Efficient Representation of Euler-Bernoulli Flexures as Robot Joints. *IEEE Trans. on Robotics* (2012) (in press)
15. Su, H.: A Pseudo-rigid-body 3R model for determining large deflection of cantilever beams subject to tip loads. *ASME Journ. of Mechanisms and Robotics* 1(2) (2009)
16. Hanafusa, H., Asada, H.: Stable Prehension by a Robot Hand with Elastic Fingers. In: Proceedings of the 7th International Symposium on Industrial Robots, pp. 361–368 (1977)

# Towards a Self-contained Soft Robotic Fish: On-Board Pressure Generation and Embedded Electro-permanent Magnet Valves

Andrew D. Marchese, Cagdas D. Onal, and Daniela Rus

**Abstract.** This paper details the design, fabrication and experimental verification of a complete, tetherless, pressure-operated soft robotic platform. Miniature CO<sub>2</sub> cartridges in conjunction with a custom pressure regulating system are used as an on-board pressure source and embeddable electro-permanent magnet (EPM) [9] valves [13] are used to address supporting hardware requirements. It is shown that this system can repeatedly generate and regulate supply pressure while driving a fluidic elastomer actuator (FEA) [7, 14, 13]. To demonstrate our approach in creating tetherless soft mobile robots, this paper focuses on an example case-study: a soft robotic fish. An underactuated propulsion system emulating natural caudal fin and peduncle movement is designed, fabricated, and subsequently experimentally characterized.

## 1 Introduction

One key requirement in creating robots that are integral parts of our daily lives is body elasticity. Elasticity is a form of intelligence embedded within the mechanics of a robot body. A soft robot is inherently safe and adaptive. It can deform and absorb energy in case of a collision [1]. Environmental uncertainty is less of a problem, reducing complexity in modeling, planning, and control. Soft robots have many potential applications [20], including bio-inspired robotics [6].

Among alternatives, fluid pressure is a suitable actuation method for soft robots. This form of mechanical energy induces stresses directly inside an elastomer [21], to enable a large actuation range, limited only by the mechanical strength of the

---

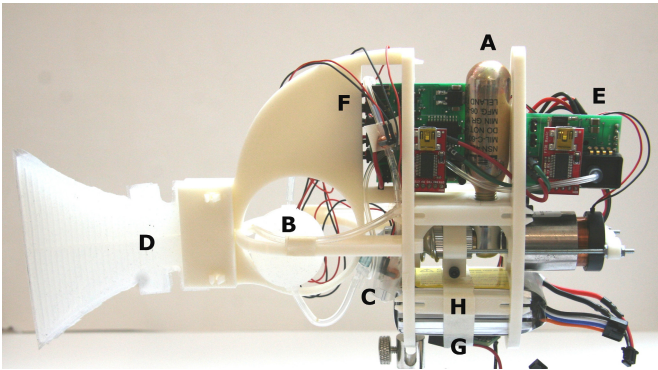
Andrew D. Marchese · Cagdas D. Onal · Daniela Rus  
Computer Science and Artificial Intelligence Laboratory,  
Massachusetts Institute of Technology, 32 Vassar St. Cambridge, MA 02139, USA  
e-mail: {andy, cagdas, rus}@csail.mit.edu

Cagdas D. Onal  
Department of Mechanical Engineering, Worcester Polytechnic Institute,  
100 Institute Road, Worcester, Massachusetts 01609  
e-mail: cdonal@wpi.edu

material. We build our robots relying on a novel actuator technology, tagged as fluidic elastomer actuators (FEAs) [7, 14, 13]. FEAs use synthetic elastomer films as pneumatic or hydraulic bending elements. They are operated by the expansion of embedded fluidic channels under pressure input.

This family of soft robots has a unique set of challenges. A fluidic actuation principle requires a pressure source [8] for operation, which limits mobility and mainstream usage. For mobile applications, these robots need to generate the actuator supply pressure on-board. Furthermore, to address, pressurize, and depressurize actuators, this class of robots requires a valve array, which can occupy considerable real-estate and be cumbersome to install on-board. In order to create completely tetherless fluid-powered soft robots, actuator supply pressure and control valves must be moved on-board. To date, the authors are aware of only a single system [14] that accomplishes this task.

Building on our previous results, this paper presents a complete, tetherless, pressure-operated soft robotic platform by addressing both challenges: (1) We use miniature CO<sub>2</sub> cartridges with custom pressure regulating systems as on-board pressure sources; and (2) we use a new, embeddable, and energy-efficient electro-permanent magnet (EPM) [9] valve [13] to address supporting valve requirements.



**Fig. 1** Fish prototype realized with fluidic actuation system. The system is composed of (A) CO<sub>2</sub> storage and release mechanism, (B) elastic reservoir vessel, (C) embeddable electro-permanent magnet (EPM) valves [13], (D) fluidic elastomer actuator [7, 14, 13], (E) custom CO<sub>2</sub> regulator board, (F) custom EPM driver board, (G) custom boost converter, and (H) lithium polymer battery.

To demonstrate our approach in creating tetherless fluid-powered soft robots, this paper focuses on an example case-study: a soft robotic fish (see Fig. 1). One common way fish achieve forward swimming is by composite caudal fin, peduncle, and body movement [2]. Motion in the fish’s tail elegantly sheds vortices forming a jet with high propulsive efficiency [17] [18]. To emulate such motion, many electromechanical robotic fish utilize several actuators to drive joints within a multi degree-of-freedom tail. MIT’s Robotuna utilizes six, 3 horsepower servomotors in its six DOF tail [4] [3]. G9, a robotic fish developed at the University of Essex, employs a



4 DOF tail [12] and NAF-I utilizes one passive and two active joints [22]. However, because of supporting hardware requirements, we do not yet have the capabilities to drive more than a single fluidic elastomer actuator onboard our robotic platform.

To meet the constraints of having a limited number of available actuators, but needing a wave-like motion, this paper develops an underactuated caudal fin and peduncle requiring only a single, bi-directional FEA, a locomotion scheme which can be realistically integrated within our soft robotic fish. We base our initial fin design on findings from [15], where a single electromechanical actuator was used to drive a compliant caudal fin. Considering the well documented hydrodynamic complexity of fish locomotion [19] [5] [10], we attempt to experimentally characterize various underactuated fish tail designs and control policies.

In the following sections, we present the design, implementation, and experimental verification of this novel, self-contained, fluidic robotic system and its propulsion system. Section 2 individually details the five major subsystems of the platform and their function within the aggregate system. Section 3 details experimental methodology and results used to verify critical aspects of system functionality. Lastly, section 4 explores major experimental insights.

## 2 Technical Approach

There are several enabling subcomponents of the self-contained fluidic actuation system. These subcomponents allow the system to simultaneously generate and regulate actuator supply pressure and control the deformation of FEAs.

In short, CO<sub>2</sub> is released from a high pressure (800 psi) canister to generate low driving pressure (3.5 psi) in an elastic reservoir. Miniature control valves allow gas to deplete from the reservoir into actuators and govern actuator pressurization.

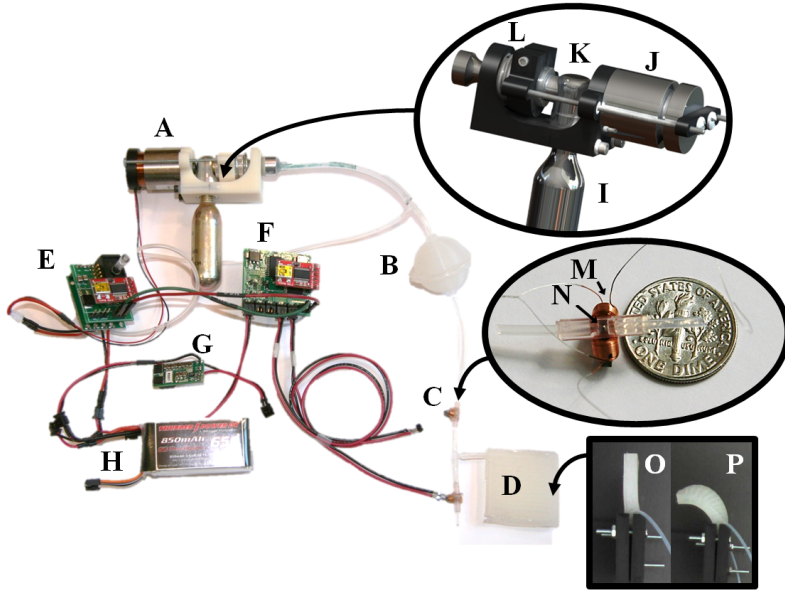
### 2.1 CO<sub>2</sub> Storage and Release Mechanism

Detailed in Fig. 2 (A), this component houses an 8 gram CO<sub>2</sub> canister (I) punctured by a commercially available trigger valve (K). Valve displacement, and correspondingly CO<sub>2</sub> release, is controlled by a linear voice coil (J). To facilitate trigger valve actuation under onboard power constraints, a preload spring (L) is employed.

### 2.2 Elastic Reservoir Vessel

Detailed in Fig. 2 (B), this spherical, thick-walled silicone elastic vessel is filled with gas by A and depleted of gas by running the actuators, (D). Past a certain internal volume, pressure within the vessel remains relatively constant independent of volume changes. Such vessel design is highly advantageous in driving a FEA: air may leave the vessel and enter the actuators while the vessel passively maintains constant driving pressure and alleviates the system from continually replenishing the reservoir.





**Fig. 2** Self-contained fluidic actuation system. The system is composed of (A) CO<sub>2</sub> storage and release mechanism, (B) elastic reservoir vessel, (C) embeddable electro-permanent magnet (EPM) valves [13], (D) fluidic elastomer actuator [7, 14, 13], (E) custom CO<sub>2</sub> regulator board, (F) custom EPM driver board, (G) custom boost converter, and (H) lithium polymer battery.

### 2.3 *Electropermanent Magnet (EPM) Valves*

Detailed in Fig. 2 (C), these embeddable control valves are a vast improvement both in size and energy consumption on the authors' previous work [13]. These valves turn ON and OFF gas flow with only a momentary pulse of energy (8 ms, 5 A) through coils (M) surrounding hard magnetic material and require no input energy to indefinitely maintain a state. A pulse either establishes or cancels a magnetic field in a fluidic channel, moving a small steel ball (N) away from or towards a sealing orifice.

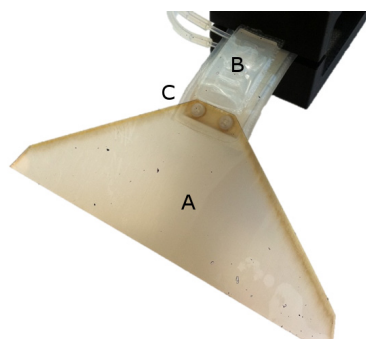
### 2.4 *Control Circuitry*

Detailed in Fig. 2 (E and F). E measures elastic vessel (B) internal pressure as feedback for a bang-bang control routine used to drive the linear voice coil (J). F drives EPM valves (C) by controlling input current pulse duration and direction. E suppresses F when pressure drops below threshold and the reservoir is being filled.

## 2.5 Underactuated Caudal Fin and Peduncle

The caudal fin and peduncle were designed to meet the functional objective of producing forward thrust while meeting the constraint of being driven by a single bi-directional FEA, detailed in Fig. 2 (D). The actuator is shown in a restored, or depressurized state (O) and in a fully displaced, or pressurized state (P). Fig. 3 details the fabricated underactuated caudal fin (A) and peduncle (B) system. The compliant caudal fin is cut from 10 mil PEEK (Polyether ether ketone) film and is secured to the elastomer actuator using 2-56 nylon machine screws (C). The constraining (center) layer of the actuator is also composed of PEEK material and acts to quickly return the elastomer actuator to its restored state during depressurization. The actuator is cast from ECOFLEX 0030 silicone rubber.

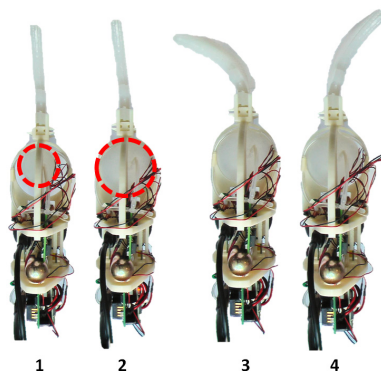
**Fig. 3** Underactuated caudal fin (A) and peduncle (B). The caudal fin is composed of a thin PEEK material and joined to the peduncle, a fluidic elastomer actuator (FEA), through small screws at (C).



## 3 Results

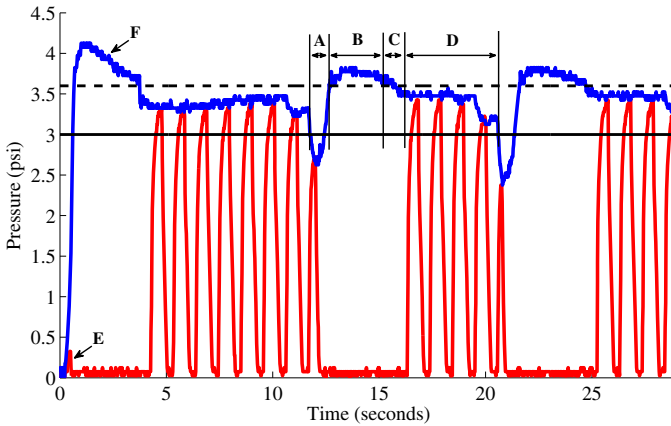
Figure 4 displays top-view snapshots of our prototype system during operation. In this experiment, all sub-components are assembled and programmed to work together to verify system functionality. An elastic reservoir vessel acts as a passive pressure regulator for the high-pressure CO<sub>2</sub> cartridge and EPM valves drive a fin-shaped bidirectional fluidic elastomer actuator at 1 Hz.

**Fig. 4** The fluidic actuation system is used to realize natural, fish-like motion. In (1) the elastic reserve vessel (red-dashed circle) is uninflated. In (2) the reserve vessel is inflated with CO<sub>2</sub> gas. The right and left sides of the fin shaped bidirectional fluidic elastomer actuator are pressurized, curving the fin to the left (3) and right (4) sides, respectively.



### 3.1 Fluidic Actuation System Performance

Fig. 5 depicts the system regulating supply pressure,  $P_{sup}$ , (blue) between a reference (dashed line) and threshold (solid line) in the presence of actuator pressurization and depressurization (red) at 1 Hz. Time period (A) represents the  $CO_2$  valve initially opening, (B) gas filling the elastic vessel (3 seconds), (C)  $CO_2$  valve closing (0.5 seconds), and (D) the elastic pressure vessel maintaining a relatively constant supply pressure while EPM control valves drive rapid actuation. Event (F) represents initial pressurization of the system and event (E) the initial passive closure of the EPM control valves. Pressure is sampled at 50Hz.



**Fig. 5** Fluidic actuation system generating and regulating supply pressure (blue curve) between a reference (dashed line) and threshold (solid line) in the presence of actuator pressurization and depressurization (red curve). As the actuator depletes reservoir volume, properties of the elastic vessel passively regulate supply pressure and enable infrequent  $CO_2$  valve actuation.

The fluidic actuation system was run according to the following algorithm:

---

#### Algorithm 1. Control Algorithm for Fluidic Actuation System

---

```

Error = Reference -  $P_{sup}$ 
if Error < Threshold and valve is closed then
     $CO_2$  valve idle
end if
if Error  $\geq$  Threshold and valve is closed then
    (1) Suppress actuation, (2) Open  $CO_2$  valve
else
    (1) Close the valve, (2) Unsuppress actuation
end if

```

---

Table 1 summarizes the system running for an extended duration of time. Here, a single fluidic actuator was driven at 1 Hz under the pressurization characterized in Fig. 5. Elastic vessel fill volume was varied from test 1 to test 2 and adjusted prior to running the system.

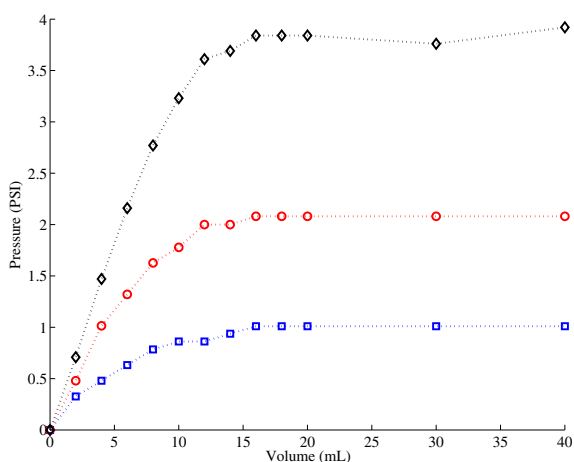
**Table 1** Measured system parameters during extended operation

Test	Duration	Vessel Fills	Actuations	Charge Consumed	Gas Consumed
1	184 <sup>a</sup> (sec)	20	85	96 (mAh)	1080 <sup>b</sup> (mL)
2	180 <sup>a</sup> (sec)	20	65	127 (mAh)	880 <sup>b</sup> (mL)

<sup>a</sup> Test terminated due to excessive heating of boost converter.

<sup>b</sup> Estimate from average vessel expansion (1.9 and 1.8 in. respectively).

The pressure-volume relationship within the reservoir is highly non-linear. Initially, as volume within the vessel increases, pressure increases. However, past a certain internal volume ( 15 mL), pressure within the vessel remains relatively constant independent of volume changes. This “plateau pressure” is proportional to vessel wall thickness and is experimentally characterized in Fig. 6. Here, volume was manually injected into vessels of varying wall thickness (0.125, 0.25, and 0.375 in.) using a 60 mL capacity syringe in 2 mL increments up to 20 mL and 10 mL increments up to 40 mL. At each volume corresponding gauge pressure measurements were collected.



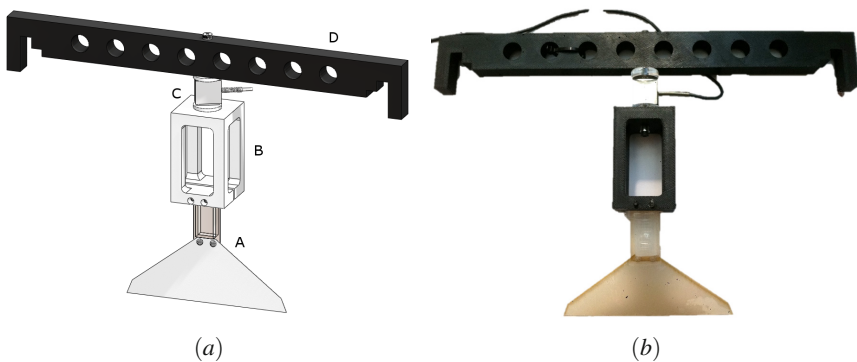
**Fig. 6** The reserve vessel’s internal pressure remains relatively constant independent of volume changes past a specific volume increase ( 15 mL). This “plateau pressure” is proportional to the vessel’s wall thickness. Black represents a wall thickness of 0.375 in., red 0.25 in., and blue 0.125 in. In all vessels, O.D. was initially 1.25 in.

## 3.2 Underactuated Caudal Fin and Peduncle Performance

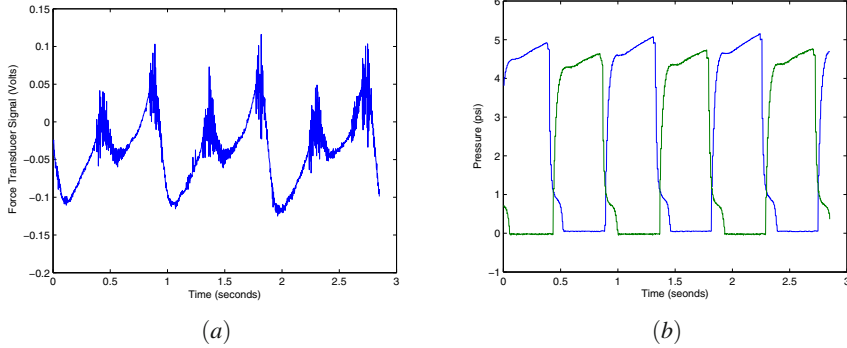
### 3.2.1 Experimental Setup

In order to evaluate the static thrust produced by a variety of caudal fin and peduncle combinations under various control policies in a controlled setting, the experimental assembly detailed in Fig. 7 was fabricated. The purpose of this assembly was to suspend a robotic fish tail and fin (A) vertically in a tank of water (50.8 cm long, 25.4 cm wide, 30.5 cm high) and measure the resulting stationary thrust force of a given control policy through a force transducer (C), (LSB200 2 lb Cap, FUTEK Advanced Sensor Technology, Inc.). Supports were fabricated to both secure the tail to the force transducer (B) and secure the entire described apparatus to the walls of the water tank (D). A similar vertical experimental setup was used in a paper detailing the effects of caudal fin compliance on propulsive force [15].

The amplified force transducer signal was acquired at 500Hz using a USB-6211 DAQ (National Instruments). In software, the force signal was zeroed prior to running a policy and, after a stabilization period, averaged over three policy-dependent cycle periods. Fig. 8 (a) details the force signal over three periods for a control policy resulting in net thrust. An average negative force is produced corresponding to transducer compression or vertical thrust force. Additionally, pressure inside both sides of the fluidic elastomer actuator was measured using two ASDX series 15 psi differential pressure transducers (Honeywell Sensing and Control). Fig. 8 (b) details the corresponding pressure profile for the above referenced control policy. Furthermore, two control valves (100 psi, 2 watt, 411 series, ASCO) were used to pressurize and depressurize each side of the bi-directional actuator (four control valves total). The valve array was driven by a custom embedded controller responsible for realizing policies, detailed in Sec. 2.4. Policies were communicated from a host PC to the low level embedded controller through a MATLAB serial communication interface.



**Fig. 7** Experimental setup used to measure the static thrust produced by various control policies for various caudal fin and peduncle combinations. The apparatus consists of a force transducer (C) secured to the vertically suspended fish tail (A) through a support at (B). A second support (D) suspends the entire apparatus vertically in a tank of water.

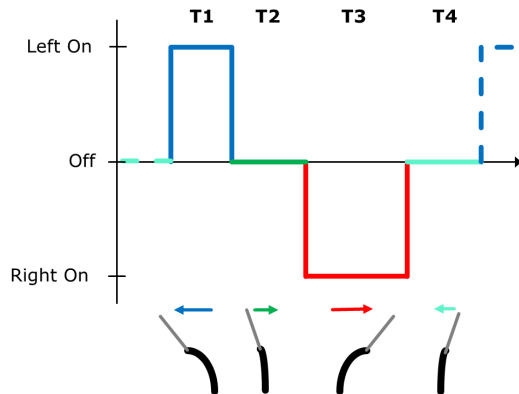


**Fig. 8** Transducer compressive or vertical thrust force signal profile over three strokes of a control policy resulting in a net thrust (a) and internal elastomer actuator pressure profile over the same time period (b)

### 3.2.2 Policy Parametrization

Control policies were parameterized using as few variables as possible. Fig. 9 details policy parametrization. A policy,  $\alpha$ , consists of four time periods, T1 - T4, which when summed define one cycle of a periodic stroke. T1 is the pressurization time of the right elastomer actuator segment, and is realized by closing the right actuator's outlet valve and opening the inlet valve. When the right actuator pressurizes, the peduncle curves to the left. T2 is the depressurization time of the right elastomer actuator, and is realized by closing the actuator's inlet valve and opening the outlet. At the onset of depressurization, the peduncle begins to return to its restored state. T3 and T4 represent identical time periods for the left actuator. Policies were realized with 1 ms precision. A minimum of 25 ms was imposed on T1-T4, as this corresponds to the fastest realizable switching frequency of the solenoid valves. A maximum was empirically determined for each actuator to avoid damaging elastomer channels.

**Fig. 9** Control policies,  $\alpha$ , were parameterized using pressurization and depressurization times of the bidirectional actuator. T1 and T2 correspond to the pressurization and depressurization times of the left-side actuator and T3 and T4 to the right-side actuator, respectively. These four times constitute one cycle of a periodic stroke.



### 3.2.3 Test Scenarios

A total of 64 tests were conducted. Two FEAs, or peduncles, were evaluated: small measuring 1.27 by 2.54 by 0.95 centimeters and large measuring 2.54 by 2.54 by 0.95 centimeters. Two fin areas were evaluated: small measuring 15.5 and large measuring 32.1 square centimeters. Two control strategies were evaluated, active depressurization and passive depressurization. During active depressurization, T2 and T4 were held at control limit minimums (25 ms) allowing the pressurization of one side of the bidirectional FEA to forcefully depressurize the other side. During passive depressurization, T2 and T4 were equal to T1 and T3 allowing each side of the FEA sufficient time to depressurize before the other is pressurized. Lastly, pressurization times, T1 and T3, were set equal and varied incrementally from either 83 or 142 ms (depending on depressurization strategy) to 450 or 750 ms (depending on actuator size limitations).

During each scenario **static force (F)**: the average thrust force over three oscillations after a stabilization period, **wake (W)**: double fin oscillation amplitude, and **Pressure (P)**: mean peak internal actuator pressure after stabilization were calculated. In addition, **efficiency** was estimated for each scenario as:  $\frac{F}{6T_1PV}$  in milliNewtons per unit of input energy, where  $\dot{V}$  is an estimated volumetric flow rate and is constant for all scenarios.

Fig. 10 details the frequency wake product as a function of the system's control input, pressurization time, for the small actuator. In general, independent of fin area or depressurization strategy, the frequency-wake product increased with pressurization time. However, as expected the product begins to plateau as both oscillation amplitude and frequency plateau with linear increases in pressurization time.

**Fig. 10** Frequency-wake product as a function of the system's control input, pressurization time, for the small actuator. Here, red and blue represent large and small fin areas respectively and squares and circles represent passive and active depressurization (vent) strategies respectively.

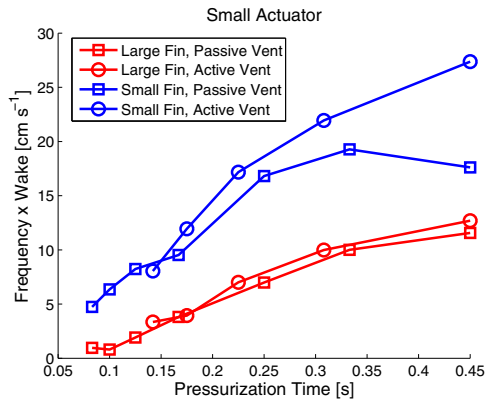
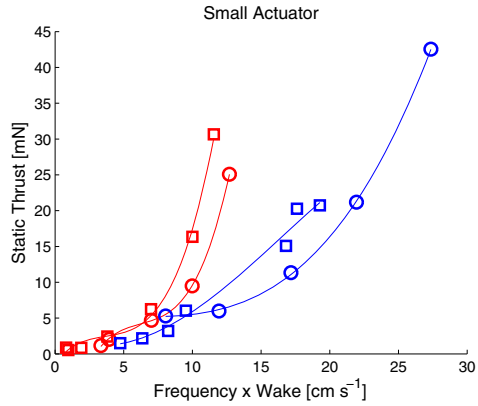


Fig. 11 details the stationary thrust produced as a function of frequency-wake product for the small actuator. In general, independent of fin area or depressurization strategy, the thrust force increased near-exponentially with frequency-wake product.

**Fig. 11** Stationary thrust produced as a function of frequency wake product for the small actuator



Furthermore, Fig. 12 details the estimated efficiency, or output force per unit input energy, as a function of the frequency-wake product for the small actuator. In general, a higher product resulted in greater efficiency.

**Fig. 12** Estimated efficiency, or output force per unit input energy, as a function of the frequency-wake product for the small actuator

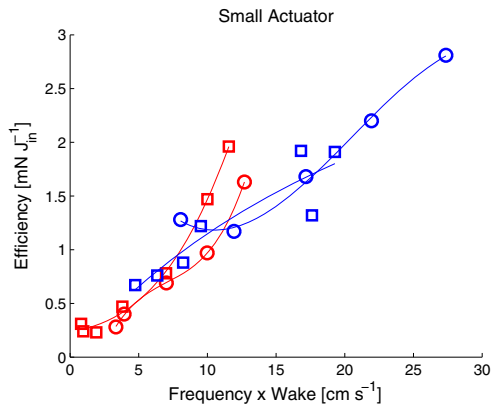
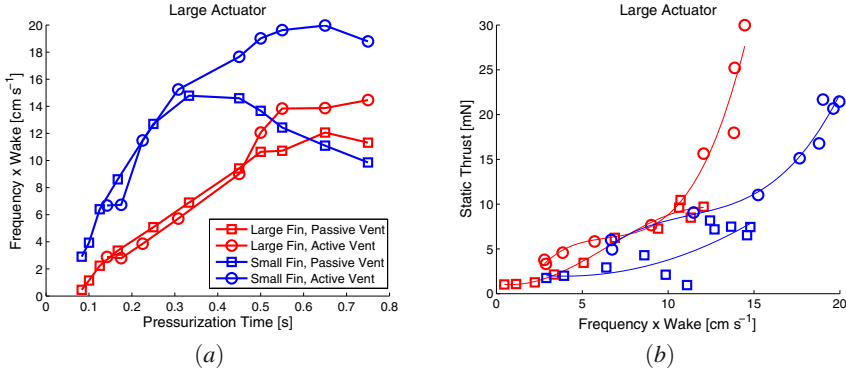


Fig. 13 details both the frequency-wake product as a function of pressurization time and stationary thrust force as a function of frequency-wake product for the larger actuator. In general, the results are similar to the the smaller actuator; however, a notable difference is that during a passive depressurization strategy (squares) the larger actuator’s ability to generate thrust decreases past a certain frequency-wake product. We believe this is due to longer than necessary depressurization times creating a sudden decrease in fin velocity during the stroke. Essentially, when one side of the actuator depressurizes, the fin returns to its neutral, restored state, and if the fin reaches this restored state before the alternate actuator side begins to pressurize the fin can be slowed to a stop in the middle of a stroke.





**Fig. 13** Frequency-wake product as a function of pressurization time and stationary thrust force as a function of frequency-wake product for the large actuator

## 4 Main Experimental Insights

**EPM Valve Efficiency.** During experiments in Table 1, an 8 ms pulse of 5 A was used to drive the EPM control valves. Two valves were used at the inlet and outlet of the single actuator, and 4 pulses were required per actuation cycle. In test 1, the valves consumed an estimated 3.8 mAh during the 85 actuations, a mere 4% for the total energy consumed. In test 2, estimated EPM energy consumption amounted to 2.3%

**CO<sub>2</sub> Capacity Limits.** Using Van der Waals non ideal gas equation, the maximum theoretical volume available in the system is 3.5 Liters.

$$\left(p + \frac{n^2 a}{V^2}\right)(V - nb) = nRT \quad (1)$$

Where,  $p$  is the supply pressure (0.125 MPa),  $T$  gas temperature (296 K),  $R$  gas constant,  $n$  number of moles (0.182),  $a$  measure of attraction between particles (0.364  $\frac{Jm^3}{mol^2}$ ), and  $b$  volume excluded by a mole of particles (0.00004267  $\frac{m^3}{mol}$ ).

**Reservoir Fill Volume.** As the results in Table 1 indicate, for the same duration of time a larger periodic fill volume allows the system to spend more time operating the fluidic elastomer actuators and expend less energy opening and closing the CO<sub>2</sub> valve.

**Underactuated Caudal Fin and Peduncle.** As is shown in the experimental results, pressurization time provides control authority, though limited, over frequency-wake product and correspondingly stationary force generation. Physical limitations on pressurization time and frequency-wake coupling inhibit arbitrarily large frequency wake products.

Considering only the active depressurization strategy scenarios, with the larger fin area both the small and large actuators had the same average efficiency 0.79  $\frac{mN}{J_{in}}$ .

However, with the smaller fin area the small actuator was almost twice as efficient as the larger actuator with average efficiencies of 1.83 and 1.13  $\frac{mN}{J_m}$  respectively.

Many researchers have thoroughly investigated the propulsive efficiency of flapping foils in a *dynamic* setting [11] [16] [17]. For these studies, efficiency and thrust coefficients of oscillating foils were determined as a function of Strouhal number, a critical parameter describing vortex pattern formation behind moving foils, and a parameter that is dependent on, among other quantities, stream/body velocity. This current analysis is limited to zero body velocity, so we cannot assume any of the same underlying hydrodynamic phenomena (i.e. reverse Karman street) are at work.

**Acknowledgements.** This work was done in the Distributed Robotics Laboratory at MIT with partial support from the DARPA DSO “Chembots” project (W911NF-08-C-0060), the Boeing Company, and the National Science Foundation Graduate Research Fellowship Program, Primary Award #1122374. We are grateful for this support.

## References

1. Albu-Schaffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimbock, T., Wolf, S., Hirzinger, G.: Soft robotics. *IEEE Robotics & Automation Magazine* 15, 20–30 (2008)
2. Bainbridge, R.: Caudal fin and body movement in the propulsion of some fish. *Journal of Experimental Biology* 40, 23–56 (1963)
3. Barrett, D.S.: Propulsive efficiency of a flexible hull underwater vehicle. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (1996)
4. Barrett, D.S., Triantafyllou, M.S., Yue, D.K.P., Grosenbaugh, M.A., Wolfgang, M.J.: Drag reduction in fish-like locomotion. *Journal of Fluid Mechanics* 392, 183–212 (1999)
5. Borazjani, I., Sotiropoulos, F., Tytell, E.D., Lauder, G.V.: Kinematics and muscle dynamics of C- and S-starts of carp (*Cyprinus carpio* L.). *Journal of Experimental Biology* 215, 671–684 (2012)
6. Calisti, M., Giorelli, M., Levy, G., Mazzolai, B., Hochner, B., Laschi, C., Dario, P.: An octopus-bioinspired solution to movement and manipulation for soft robots. *Bioinspiration & Biomimetics* 6(3), 036,002 (2011)
7. Correll, N., Önal, Ç.D., Liang, H., Schoenfeld, E., Rus, D.: Soft autonomous materials—using active elasticity and embedded distributed computation. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) *Experimental Robotics. STAR*, vol. 79, pp. 227–240. Springer, Heidelberg (2012)
8. Kazerooni, H.: Design and analysis of pneumatic force generators for mobile robotic systems. *IEEE/ASME Transactions on Mechatronics* 10(4), 411–418 (2005)
9. Knaian, A.N.: Electropermanent magnetic connectors and actuators: devices and their application in programmable matter. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2010)
10. Lauder, G., Lim, J., Shelton, R., Witt, C., Anderson, E., Tangorra, J.: Robotic models for studying undulatory locomotion in fishes. *Marine Technology Society Journal* 45(4), 41–55 (2011)
11. Licht, S., Polidoro, V., Flores, M., Hover, F., Triantafyllou, M.: Design and projected performance of a flapping foil auv. *IEEE Journal of Oceanic Engineering* 29(3), 786–794 (2004)

12. Liu, J., Hu, H.: Biological inspiration: From carangiform fish to multi-joint robotic fish. *Journal of Bionic Engineering* 7, 35–48 (2010)
13. Marchese, A.D., Onal, C.D., Rus, D.: Soft robot actuators using energy-efficient valves controlled by electropermanent magnets. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 756–761 (2011)
14. Onal, C.D., Chen, X., Whitesides, G.M., Rus, D.: Soft mobile robots with on-board chemical pressure generation. In: International Symposium on Robotics Research, ISRR (2011)
15. Park, Y.J., Jeong, U., Lee, J., Kim, H.Y., Cho, K.J.: The effect of compliant joint and caudal fin in thrust generation for robotic fish. In: 2010 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob), pp. 528–533 (2010)
16. Schouveiler, L., Hover, F., Triantafyllou, M.: Performance of flapping foil propulsion. *Journal of Fluids and Structures* 20(7), 949–959 (2005)
17. Triantafyllou, G., Triantafyllou, M., Grosenbaugh, M.: Optimal thrust development in oscillating foils with application to fish propulsion. *Journal of Fluids and Structures* 7(2), 205–224 (1993)
18. Triantafyllou, M., Triantafyllou, G.: An efficient swimming machine. *Scientific America* 272(3), 64–70 (1995)
19. Triantafyllou, M., Triantafyllou, G., Yue, D.K.: Hydrodynamics of fishlike swimming. *Annual Review of Fluid Mechanics* 32, 33–53 (2000)
20. Trivedi, D., Rahn, C., Kier, W., Walker, I.: Soft robotics: Biological inspiration, state of the art, and future research. *Advanced Bionics and Biomechanics* 5(2), 99–117 (2008)
21. Wait, K., Jackson, P., Smoot, L.: Self locomotion of a spherical rolling robot using a novel deformable pneumatic method. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 3757–3762 (2010)
22. Zhong, Y., Chong, C., Zhou, C., Seet, G., Low, K.: Performance predict model for a body and caudal fin (bcf) biomimetics fish robot. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2009, pp. 1230–1235 (2009)

# An Empirical Study of Static Loading on Piezoelectric Stick-Slip Actuators of Micromanipulators

Aayush Damani, Manikantan Nambi, and Jake J. Abbott

**Abstract.** Piezoelectric stick-slip actuators have become the foundation of modern micromanipulation. Due to difficulty in closed-loop control with manipulators that use piezoelectric stick-slip actuators, methods for open-loop control with a human in the loop have been developed. The utility of such methods depends directly on the accuracy of the open-loop models of the manipulator. Prior research has shown that modeling of piezoelectric actuators is not a trivial task as they are known to suffer from nonlinearities that degrade their performance. In this paper, we study the effect of static (non-inertial) loads on a prismatic and a rotary piezoelectric stick-slip actuator, and obtain a model relating the step size of the actuator to the load. The actuator-specific parameters of the model are calibrated by taking measurements in specific configurations of the manipulator. Results comparing the obtained model to experimental data are presented.

## 1 Introduction

Micromanipulation deals with small motions on the order of  $10^{-3}$  to  $10^{-6}$  m. Under the guidance of electron and optical microscopes, micromanipulation is now commonly used in the areas of MEMS construction and characterization, isolation and characterization of individual materials, and manipulation of single cells. The development and use of commercial manipulators like the Kleindiek MM3A [3], the Zyvex Nanomanipulator [5], Imina Technologies miBot [2], SmarAct Actuators [4], and the Attocube Nanopositioners [1] has increased with the demand for precise standardized tools for micromanipulation.

Piezoelectric stick-slip actuators have become the foundation of modern micromanipulation due to their simple structure, high positional accuracy, unlimited

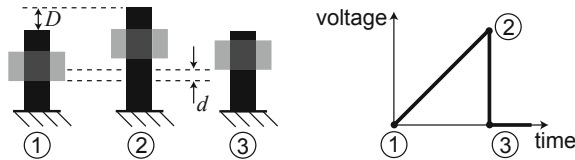
---

Aayush Damani · Manikantan Nambi · Jake J. Abbott  
Department of Mechanical Engineering,  
University of Utah, Salt Lake City, Utah, USA  
e-mail: {aayush.damani, m.nambi, jake.abbott}@utah.edu

movable distance, and high stability due to support by guiding surfaces [10]. Due to their useful characteristics, these actuators have been extensively used in manipulation of micro/nano-sized objects, medical devices, camera lens actuation systems, and in bio-sciences [7, 19]. These actuators consist of a piezoelectric element and a sliding mass that moves relative to the piezoelectric element (Fig. 1). Typically, these actuators have no sensor feedback (with the exception of SmarAct Actuators [4]), and hence, the individual joints of the manipulators are controlled open-loop, using one knob per joint. Due to difficulty in implementing real-time closed-loop controllers (which are generally based on vision feedback [9, 11]) for micromanipulators, methods to control them open-loop that capitalize on the intelligence of the human user are being developed [17, 20]. The utility of such methods depends directly on the accuracy of the open-loop models of the manipulator used.

Modeling of piezoelectric actuators is not a trivial task as they are known to suffer from nonlinearities such as hysteresis, creep, and drift, which degrade their performance [14, 12, 16]. A number of researchers have mathematically modeled the dynamics of piezoelectric stick-slip actuators [6, 8]. Peng *et al.* [18] used a pre-sliding friction model to explain the dynamics of stick-slip actuators, and obtained an empirical model for the effect of end-effector mass on the step size of the actuator. Lockwood *et al.* [15] found that when gravitational force was acting parallel to the axis of their stick-slip actuator, the step size and corresponding displacement rate in the downward direction was observed to be 14.7% greater than in the upward direction. Thus, it is known that static (i.e., noninertial) loads in the direction of motion of the actuator increases the step size and vice-versa. However, this effect has not been well characterized in the past.

In this paper, we study the effect of static loads on a prismatic and a rotary piezoelectric stick-slip actuator, obtain an empirical model relating the step size to the load, and develop a method to calibrate the parameters of the empirical model using measurements from the actuators. The modeling experiments presented herein were performed for the coarse (stepping) mode of operation of the actuator



**Fig. 1** Functional description of a piezoelectric stick-slip actuator. A saw-tooth voltage is applied to the piezoelectric element. As the voltage slowly increases from 1 to 2, the piezoelectric element stretches by a distance  $D$ , and due to friction between the piezoelectric element and the sliding mass, the sliding mass also advances (stick phase). When the voltage is quickly reduced from 2 to 3, the piezoelectric element quickly shrinks, but the inertia of the sliding mass prohibits it from moving backward as quickly, resulting in a net forward displacement of the sliding mass of  $d < D$  (slip phase). This is also known as the coarse mode of operation of the actuator. In the fine (traditional) mode, the voltage signal between 1 and 2 is controlled to achieve fine positioning.

(Fig. 1). The empirical models derived can be used with algorithms developed in [17] to perform intuitive teleoperation of the micromanipulator's end-effector, rather than controlling individual joints. With piezoelectric stick-slip actuators, the step size is stochastic, with a hard-to-model variance about a load-dependent mean. The method presented in this paper deals with modeling this mean. The method is primarily designed to provide an accurate estimate of the size of the next commanded step, such that a user's desired motion command can be accurately mapped to a required number of joint steps. Having a more accurate model of joint stepping could also lead to a method to estimate the joint configuration in manipulators without joint sensing, but such estimation methods would be subject to drift, and as such would need to incorporate additional sensing methods to be useful in practice.

## 2 Technical Approach

The commonly used Kleindiek MM3A manipulator is used in this study (Figs. 2 and 3). It has three degrees of freedom (DOF) with two rotary joints and one prismatic joint, which use piezoelectric stick-slip actuators. Due to the discrete step nature of these actuators, as well as the MM3A's controller, commands are given in the form of number of steps to be taken along a given joint. The joints of the MM3A lack sensor feedback, hence, it is difficult to obtain accurate measurements of the step size. To study the effect of static loads on the step size of a joint  $j$ , we use the average step size given by:

$$\bar{\gamma}_{ji} = \frac{R_j}{N_{ji}} \quad (1)$$

where  $R_j$  is the total range of joint  $j$  ( $4\pi/3$  rad for the rotary joints and 12 mm for the prismatic joint),  $N_{ji}$  is the total number of steps required by joint  $j$  to travel through  $R_j$ , and  $i \in \{+, -\}$  indicates the direction of joint motion.

As the step size for each joint is small (on the order of  $1 \mu\text{m}$ ), it is difficult to visually detect when a joint reaches its end of travel. However, the actuators make a distinct noise when they hit a mechanical stop. This knowledge is used to develop an audio limit switch that detects the end of travel for a joint. Custom software monitors the sound from a microphone at each instant and computes the Fast Fourier Transform (FFT) of the audio signal. The change in sound when a joint hits a mechanical stop is detected as a peak in the power of the FFT. The frequency at which this peak occurs, and the intensity of the peak, is different for each joint and has to be tuned before each experiment.

By measuring  $\bar{\gamma}_{ji}$  at different configurations of the manipulator, we study the effect of gravitational loads on the rotary and the prismatic joint (no other external forces are acting on the manipulator). Because an individual joint cannot distinguish a gravitational load due to the distal links from an equivalent load due to a force applied at the end-effector (passing through the manipulator's Jacobian), our results generalize to all static (i.e., noninertial) loads. Nonlinear regression is used to fit a function, based on our knowledge of the load acting on the actuator, to the empirical data, to obtain a relation for the step size of the form  $\gamma_{ji} = \Gamma_{ji}(g, \alpha_{ji})$ , where  $\alpha_{ji}$  is a

set of actuator specific parameters, and  $g$  is the gravity vector. The actuator specific parameters  $\alpha_{ji}$  of the model are then calibrated for by using  $\bar{\gamma}_{ji}$  measurements at selected configurations for each joint. Significance of unmodeled factors such as change in environmental conditions from day to day are analyzed by performing ANOVA on the data obtained for  $\bar{\gamma}_{ji}$ .

### 3 Results

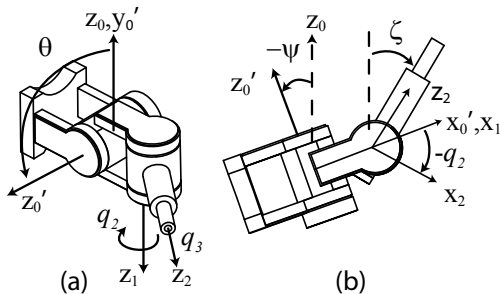
This section contains the main empirical modeling results of this paper. The experiments that were conducted to obtain these results are detailed in Section 4.

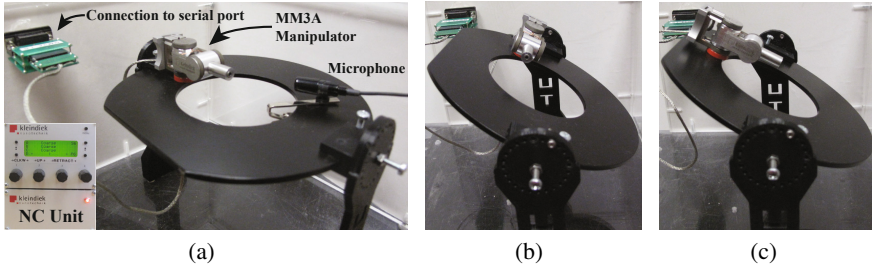
#### 3.1 Effect of Unmodeled Factors

Environmental conditions (e.g., temperature, humidity) are uncontrolled in our experiments, so we will not incorporate these factors into our model (although it is possible that they could be incorporated in the future [13]). To minimize these unmodeled effects on the open-loop control of the Kleindiek MM3A, we propose to calibrate the joints before each session of use. This assumes that there is a significant change from day to day that warrants such recalibration. To substantiate this claim, the average step size for prismatic joint 3 and the rotary joint 2 in the positive ( $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{2+}$ ) and negative ( $\bar{\gamma}_{3-}$  and  $\bar{\gamma}_{2-}$ ) directions were taken on two different days, which would incorporate a change in environmental conditions. The positive direction for the prismatic joint means moving out from 0 mm to 12 mm as defined by the  $z_2$  direction. For the rotary joint, the positive direction is defined by the right-hand rule about the  $z_1$  axis. For the prismatic joint, the configuration of the manipulator was kept constant at  $q_2 = -\pi/2$ ,  $\theta = 0$ , and  $\psi = 0$  on both days, and three readings each of the step size values  $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{3-}$  were taken on each day. For the rotary joint,  $\bar{\gamma}_{2+}$  and  $\bar{\gamma}_{2-}$  was recorded at  $q_3=0$  mm,  $\theta = -\pi/2$ , and  $\psi = 0$ . In these configurations, there is no effect of gravity on the joint being investigated, isolating the unmodeled factors of interest.

An ANOVA test on the data shows that the difference in step size on different days is statistically significant ( $p < 0.05$ ) for both positive and negative directions

**Fig. 2** Kleindiek MM3A. With the  $z_0$  axis initially vertical, the base frame is rotated by  $\theta$  about  $x_0$  and then rotated by  $\psi$  about the new  $y_0$ . (a) Isometric view at  $\theta = 90^\circ$ . No gravitational loads acting on joints 2 or 3. (b) Side view at  $\theta = 0^\circ$ , with gravitational loads acting on both joints 2 and 3.  $\zeta = \psi - q_2$ .





**Fig. 3** The Kleindiek MM3A manipulator is shown at different orientations. (a)  $q_2 = -\pi/2$ ,  $\theta = 0$ , and  $\psi = 0$  (b)  $q_2 = -\pi/2$  and  $\psi = 0$  at a particular  $\theta$  (c)  $q_2 = -\pi/2$  and  $\theta = 0$  at a particular  $\psi$ .

for both the prismatic and the rotary joints. The ANOVA test also shows a significant difference in the step size between the positive and negative directions within a given day for both joints. Thus, calibration is recommended each time the manipulator is to be used, and different calibration parameters should be found for each direction of motion.

### 3.2 Modeling of a Prismatic Joint

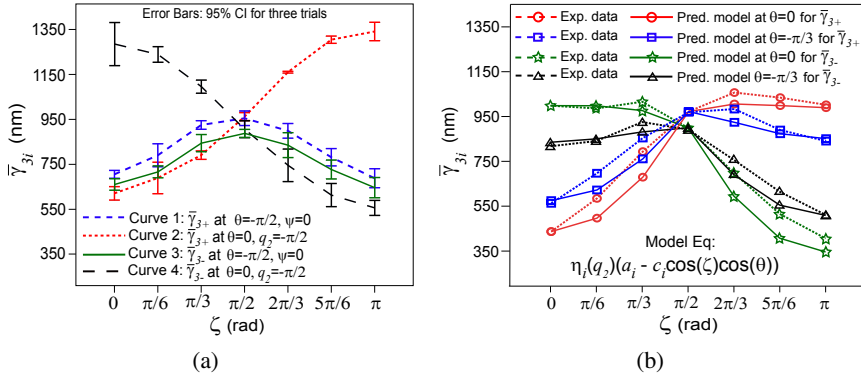
Fig. 4 shows the results for the modeling experiments on the prismatic joint. The gravitational load on the prismatic joint is varied by changing the angles  $q_2$ ,  $\theta$ , and  $\psi$  (see Fig. 2). Curve 1 in Fig. 4a shows  $\bar{\gamma}_{3+}$  recorded at  $\theta = -\pi/2$  such that there is no load due to gravity along the joint regardless of  $q_2$ . At  $q_2 = -\pi/2$  on curve 1, the entire structure of the manipulator is aligned with the axis of the prismatic joint, absorbing the recoil caused due to the quick stepping nature of the actuator, resulting in a maximum value for  $\bar{\gamma}_{3+}$ . The result from curve 1 is converted into an efficiency factor as:

$$\eta_i(q_2) = 1 - b_i |\cos q_2| \quad (2)$$

Joint 3 has a maximum stepping efficiency of 1 at  $q_2 = -\pi/2$ . The reduced step size (i.e., the reduction in stepping efficiency) at values of  $q_2$  other than  $-\pi/2$  is likely due to the component of the recoil force of the actuator acting perpendicular to the link connecting joint 1 to joint 2 causing a small deflection in the link (which is not infinitely rigid). This effect is captured by the  $|\cos(q_2)|$  term in  $\eta_i(q_2)$ . The free parameter  $b_i$  captures the loss of stepping efficiency when the prismatic joint is fully perpendicular to the maximum-efficiency configuration.

To isolate the effect of gravity without any loss of stepping efficiency due to recoil,  $q_2$  is fixed at  $-\pi/2$  such the manipulator arm is always outstretched, and the gravitational load is changed by varying  $\psi$ ; results of this experiment are shown by curve 2, which is the pure effect of gravity on  $\bar{\gamma}_{3+}$ . Results for  $\bar{\gamma}_{3-}$  are similar to





**Fig. 4** (a) Experimental data for the step size of the prismatic joint ( $\bar{y}_{3i}$ ) as a function of  $\zeta$ , plotted at  $\theta = 0$  and  $-\pi/2$ , with data recorded on three different days. (b) Model equation fitted to experimental data for a single day (Day 1) at  $\theta = 0$  and  $-\pi/3$ , with  $\psi = 0$ . Calibrated parameter values of  $a_+ = 972$ ,  $b_+ = 0.27$ ,  $c_+ = 372$ ,  $a_- = 899$ ,  $b_- = 0.25$ , and  $c_- = -436$  were found using the three calibration configurations described in the text.

$\bar{y}_{3+}$ , but mirrored about  $\zeta = \pi/2$  as can be seen from curves 3 and 4 in Fig. 4a, indicating that moving joint 3 outward with  $\zeta = 0$  is equivalent to moving joint 3 inward at  $\zeta = \pi$ .

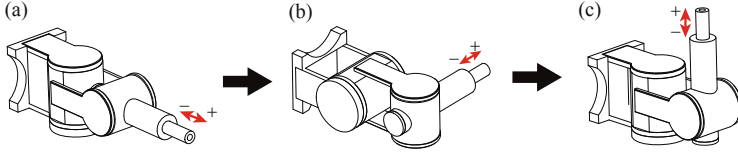
We hypothesized a model that combines the information in curves 1 and 2 as:

$$\bar{y}_{3i} = \eta_i(q_2)(a_i - c_i \cos(\zeta) \cos(\theta)) \quad (3)$$

The model has six actuator-dependent parameters ( $\alpha_3 = \{a_+, a_-, b_+, b_-, c_+, c_-\}$ ) that can be identified by measuring  $\bar{y}_{3+}$  and  $\bar{y}_{3-}$  at the three different configurations:  $(q_2, \theta, \psi) = (-\pi/2, 0, 0)$ ,  $(0, -\pi/2, 0)$ , and  $(0, 0, 0)$ . This process of finding the free parameters for the prismatic joint is explained in Section 3.3. The parameter  $a_i$  represents the basic step size of the joint when no gravitational load or recoil inefficiency is acting on the joint, measured at  $(-\pi/2, 0, 0)$ . It can be seen that curve 1 and curve 2 intersect at the value of  $a_i$ . The term  $c_i \cos(\zeta) \cos(\theta)$  is a function of the component of the gravitational load due to the weight of the distal link acting along the axis of the joint. The parameter  $b_i$  was defined above.

### 3.3 Calibration Procedure for a Prismatic Joint

The step size model for the prismatic joint as described in Eq. 3 has six unknown parameters that can be calibrated for by taking six measurements of  $\bar{y}_{3i}$  as shown in Fig. 5. For simplicity, the average step size at a known configuration of  $q_2$ ,  $\theta$ , and  $\psi$  is denoted by  $\bar{y}_{3i}(q_2, \theta, \psi)$ . The following procedure is used to identify the six free parameters  $\alpha_3 = \{a_+, a_-, b_+, b_-, c_+, c_-\}$  of the prismatic joint:



**Fig. 5** Calibrating configurations (in sequence) for identifying the six unknown parameters of the model of the prismatic joint (joint 3). (a)  $a_i$  is calculated by measuring  $\bar{\gamma}_{3i}$  at  $(q_2, \theta, \psi) = (-\pi/2, 0, 0)$ , (b)  $b_i$  is calculated using  $a_i$  calculated in the previous step and  $\bar{\gamma}_{3i}$  at  $(q_2, \theta, \psi) = (0, -\pi/2, 0)$ , and (c)  $c_i$  is calculated using the values of  $a_i$  and  $b_i$  above, and  $\bar{\gamma}_{3i}$  at  $(q_2, \theta, \psi) = (0, 0, 0)$ .

1. First,  $\bar{\gamma}_{3+(-\pi/2,0,0)}$  and  $\bar{\gamma}_{3-(-\pi/2,0,0)}$  are measured at  $(q_2, \theta, \psi) = (-\pi/2, 0, 0)$  and by substituting in Eq. 3, we find parameter  $a_i$  of the model by the following relation:

$$a_i = \bar{\gamma}_{3i(-\pi/2,0,0)} \quad (4)$$

2. Next,  $\bar{\gamma}_{3+(0,-\pi/2,0)}$  and  $\bar{\gamma}_{3-(0,-\pi/2,0)}$  are measured at  $(q_2, \theta, \psi) = (0, -\pi/2, 0)$  and using Eq. 3 and the calculated value of  $a_i$ , we find parameter  $b_i$  using the following relation:

$$b_i = 1 - \frac{\bar{\gamma}_{3i(0,-\pi/2,0)}}{a_i} \quad (5)$$

3. Finally,  $\bar{\gamma}_{3+(0,0,0)}$  and  $\bar{\gamma}_{3-(0,0,0)}$  are measured at  $(q_2, \theta, \psi) = (0, 0, 0)$ , and by substituting these values in Eq. 3 along with  $a_i$  and  $b_i$ , we find parameter  $c_i$  using the following relation:

$$c_i = a_i - \frac{\bar{\gamma}_{3i(0,0,0)}}{1 - b_i} \quad (6)$$

The order of the three steps above, which correspond to steps (a), (b), and (c), respectively, does not have to be carried out in any specific order. In practice, it may be more efficient to conduct the calibration in a different order that requires less joint movements (e.g., (b), (c), (a)).

Fig. 4b shows the model plotted against experimental data for a single day at  $\theta = 0$  and  $-\pi/3$  with  $\psi$  fixed at 0. The value of  $\theta = -\pi/3$  is not included as one of the calibration configurations mentioned above, yet the model captures the step size of the joint as a function of the configuration. We observe similar results in other configurations. Thus, the calibrated parameters can completely characterize the effect of the load due to gravity on the prismatic joint in any arbitrary configuration of the manipulator.

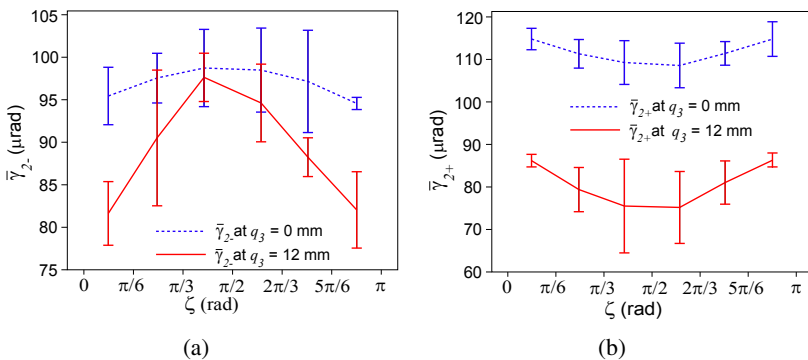
### 3.4 Modeling of a Rotary Joint

Two experiments were performed to study the effect of gravitational loads on the rotary joint 2 (in an effort to study static loading in general). In experiment 1,  $\theta$  is kept at  $-\pi/2$  such that there is no load due to gravity on the joint, in an attempt to verify that the joint has a consistent behavior throughout its range of motion if other factors are controlled. Variation in  $\bar{\gamma}_{2i}$  is studied in different sub-ranges of  $q_2$ , for  $q_3 = 0$  mm and 12 mm. From the results of experiment 1, it is safe to conclude that the step size of the rotary joint is relatively constant throughout its range of motion when no load due to gravity is acting on the joint, since the variation in step size for different values of  $q_2$  in this configuration is found to be less than  $\pm 2\%$ , with no discernible trend in the data. Fig. 6 shows the experimental results for experiment 2 in which  $\theta$  is kept at zero such that there is load due to gravity on the joint; here the gravitational load on joint 2 is a function of its own position  $q_2$ .

The model for static loading on the rotary joint is derived based on the physics that, if  $\theta = 0$ , the torque on joint 2 is related to gravitational loads as  $\tau_2 \propto g \sin(q_2)$ , where  $g$  is the acceleration due to gravity; the constant of proportionality is related to the mass and lengths of the distal links, which are unknown to us. The empirical model to predict the step size for the rotary joint is formulated as:

$$\bar{\gamma}_{2i} = \bar{\gamma}_{2i, \theta = \pm\pi/2} + d_i \sin(\zeta) \quad (7)$$

where  $\bar{\gamma}_{2i, \theta = \pm\pi/2}$  denotes the direction-dependent step size of the rotary joint when there is no effect of gravity on the link (i.e., at  $\theta = \pm\pi/2$ ),  $d_i$  is a free parameter that denotes the maximum increase in step size over the baseline step size  $\bar{\gamma}_{2i, \theta = \pm\pi/2}$ , and  $\zeta = \psi - q_2$  as described in Fig. 2. We assume that the step size at  $\theta = -\pi/2$  and  $\theta = \pi/2$  would be equal to the step size at  $q_2 = 0$  and  $q_2 = -\pi$  when  $\psi = \theta = 0$ , since there is no torque due to gravity on the joint in any of these cases.



**Fig. 6** Step size (a)  $\bar{\gamma}_{2-}$  and (b)  $\bar{\gamma}_{2+}$  as a function of  $\zeta$  at  $q_3 = 0$  and 12 mm,  $\psi = 0$ , and  $\theta = 0$ .  $\bar{\gamma}_{2i}$  is recorded for intervals of  $\pi/6$  from  $-\pi/6$  to  $7\pi/6$  and is plotted at the midpoint of each interval as explained in Section 4.

It can be seen that the nature of step size in the positive direction is an inverted form of its nature in the negative direction. This is attributed to the fact that the load due to gravity acts against the direction of motion of the joint in the positive direction, and with it in the negative direction. Hence, the step size obtained in the positive direction,  $\bar{\gamma}_{2+}$ , will be less than that obtained at  $\theta = -\pi/2$  where no gravitation load is acting on the joint. The opposite holds true for the step size in negative direction,  $\bar{\gamma}_{2-}$ . In other words, downward steps are bigger than horizontal steps, which in turn are bigger than upward steps, as we would expect.

If the manipulator were to be tilted by an angle  $\theta \neq 0$ , then the torque due to gravity on joint 2 would become proportional to the cosine of the gravitational component, such that the model of Eq. 7 should be modified as:

$$\bar{\gamma}_{2i} = \bar{\gamma}_{2i,\theta=\pm\pi/2} + d_i \sin(\zeta) \cos(\theta) \quad (8)$$

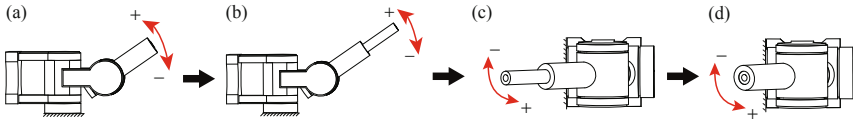
### 3.5 Calibration Procedure for a Rotary Joint

Fig. 7 shows the calibration sequence for rotary joint 2, which gives us values for the joint-specific parameters,  $d_i$  and  $\bar{\gamma}_{2i,\theta=\pm\pi/2}$ , for the rotary joint in the positive and negative directions. For simplicity, the average step size at a known configuration of  $q_3$ ,  $\theta$ , and  $\psi$  is denoted by  $\bar{\gamma}_{2(q_3,\theta,\psi)}$  unless otherwise mentioned. The following procedure is followed to obtain the free parameter:

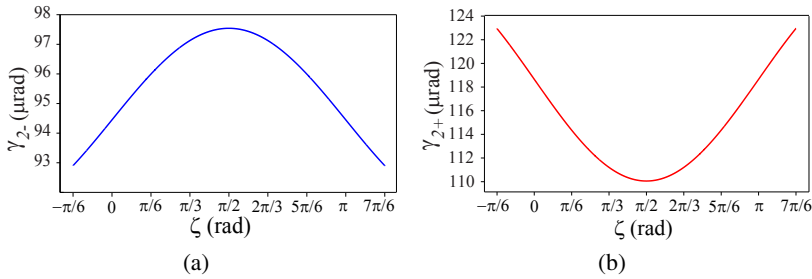
1.  $\bar{\gamma}_{2-(0,0,0)}$  and  $\bar{\gamma}_{2+(0,0,0)}$  are measured by driving joint 2 across its range from  $q_2 = \pi/6$  to  $-7\pi/6$  in the negative direction and then in the positive direction at  $(q_3, \theta, \psi) = (0, 0, 0)$ .
2. The prismatic joint is then fully extended.  $\bar{\gamma}_{2-(12,0,0)}$  and  $\bar{\gamma}_{2+(12,0,0)}$  are measured by driving joint 2 across its range from  $q_2 = \pi/6$  to  $-7\pi/6$  in the negative direction and then in the positive direction at  $(q_3, \theta, \psi) = (12 \text{ mm}, 0, 0)$ .
3. The manipulator is then tilted by setting  $\theta = -\pi/2$  such that there is no gravitational torque on joint 2.  $\bar{\gamma}_{2-(12,-\pi/2,0)}$  and  $\bar{\gamma}_{2+(12,-\pi/2,0)}$  are measured by driving joint 2 across its range from  $q_2 = \pi/6$  to  $-7\pi/6$  in the negative direction and then in the positive direction at  $(q_3, \theta, \psi) = (12 \text{ mm}, -\pi/2, 0)$ .
4. The prismatic joint is then fully retracted.  $\bar{\gamma}_{2-(12,-\pi/2,0)}$  and  $\bar{\gamma}_{2+(12,-\pi/2,0)}$  are measured by driving joint 2 across its range from  $q_2 = \pi/6$  to  $-7\pi/6$  in the negative direction and then in the positive direction at  $(q_3, \theta, \psi) = (0 \text{ mm}, -\pi/2, 0)$ .

Since  $\gamma_2$  is a function of  $q_2$  at each instant, it not a trivial task to calculate the parameter  $d_i$  from Eq. 8 by using the average step size values ( $\bar{\gamma}_{2i}$ ) that are available to us based on the entire range of motion. A simulation of the model shown in Eq. 8 was implemented wherein a number of different values of the free parameters  $\bar{\gamma}_{2i,\theta=-\pi/2}$  and  $d_i$  were given to the simulation as inputs, and the simulation returns the step size at each instant and the total number of steps required to move through the joint's entire range. The total number of steps obtained is then used to calculate the simulated average step size  $\bar{\gamma}_{2i,\theta=0}$ .

Fig. 8 shows the simulation results obtained for the rotary joint after stepping  $q_2$  through its full range of motion from  $\pi/6$  to  $-7\pi/6$  for fixed arbitrary (typical)



**Fig. 7** Calibrating configurations (in sequence) for identifying the six unknown parameters of the model of the rotary joint 2.  $q_2$  is driven across its full range from  $\pi/6$  to  $-\pi/6$  in the negative and positive directions at (a)  $q_3 = 0$  mm,  $\theta = 0$ , and  $\psi = 0$ ; (b)  $q_3 = 12$  mm,  $\theta = 0$ , and  $\psi = 0$ ; (c)  $q_3 = 12$  mm,  $\theta = -\pi/2$ , and  $\psi = 0$ ; and (d)  $q_3 = 0$  mm,  $\theta = -\pi/2$ , and  $\psi = 0$ .

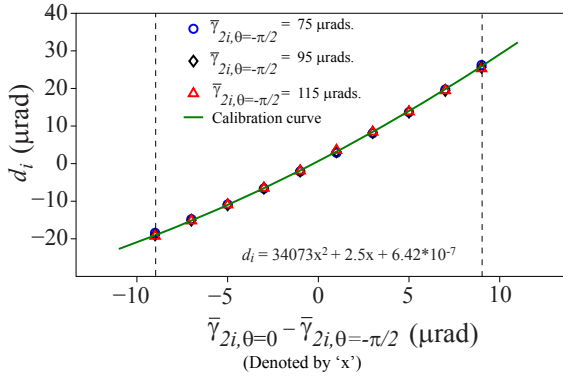


**Fig. 8** Simulated model of the step size of joint 2 in (a) the negative direction, and in (b) the positive direction. The values of  $\bar{\gamma}_{2i,\theta=0}$  and  $\bar{\gamma}_{2i,\theta=-\pi/2}$  are fixed, and  $\theta$  was kept at zero in simulation. The nature of data obtained in simulation agrees with experimental results shown in Fig. 6.

values of  $\bar{\gamma}_{2i,\theta=-\pi/2}$  and  $d_i$ . The figure shows the dependence of step size on the current configuration. This validates our model of the rotary joint 2 with data observed in experiments (Fig. 6). It was found that the difference between the two average step size values  $\bar{\gamma}_{2i,\theta=0}$  and  $\bar{\gamma}_{2i,\theta=-\pi/2}$  have a quadratic relation with the free parameter  $d_i$  as shown in Fig. 9. Irrespective of the *individual* values of  $\bar{\gamma}_{2i,\theta=0}$  and  $\bar{\gamma}_{2i,\theta=-\pi/2}$ , the free parameter value  $d_i$  remains the same (difference of less than  $1 \mu\text{rad}$ ) for the same *difference* between the two step size values. The simulation was performed such that the range of values for  $\bar{\gamma}_{2i,\theta=0} - \bar{\gamma}_{2i,\theta=-\pi/2}$  obtained in simulation was from  $-9.2 \mu\text{rad}$  to  $9.2 \mu\text{rad}$ , because this was the range of  $\bar{\gamma}_{2i,\theta=0} - \bar{\gamma}_{2i,\theta=-\pi/2}$  observed in experiments. A relation for computing  $d_i$  was formulated by fitting the simulation results obtained to a quadratic function as shown in Fig. 9. The equation formulated using nonlinear least-squares regression is:

$$d_i = 3.41(\bar{\gamma}_{2i,\theta=0} - \bar{\gamma}_{2i,\theta=-\pi/2})^2 + 0.025(\bar{\gamma}_{2i,\theta=0} - \bar{\gamma}_{2i,\theta=-\pi/2}) + 6.42 \times 10^{-7} \quad (9)$$

From Fig. 6, we see that  $\bar{\gamma}_{2i,\theta=-\pi/2}$  and  $d_i$  are a function of  $q_3$ , as  $q_3$  changes the inertial load on joint 2. Different values of  $d_i$  can be calculated when  $q_3 = 0$  and  $12$  mm using Eq. 9, and the effect of changing  $q_3$  is reflected in the values of  $\bar{\gamma}_{2i,\theta=0}$



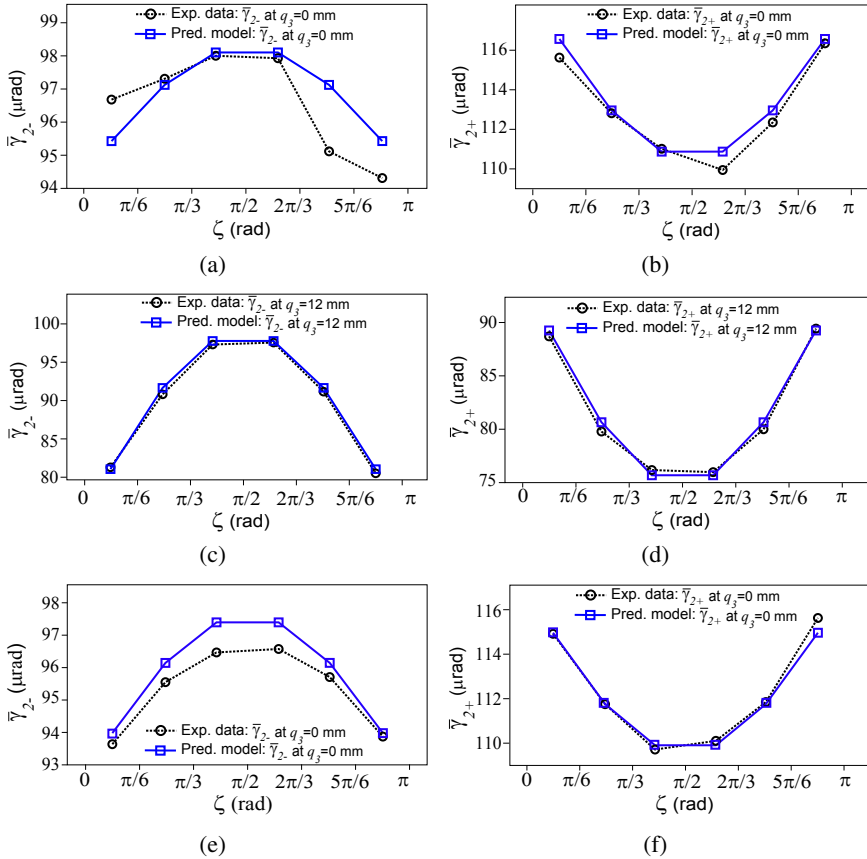
**Fig. 9** Variation of the free parameter  $d_i$  with respect to change in difference between  $\bar{\gamma}_{2i,\theta=0}$  and  $\bar{\gamma}_{2i,\theta=-\pi/2}$  at different values of  $\bar{\gamma}_{2i,\theta=-\pi/2}$ . The vertical dash lines represent the range of this difference as observed in experiments.

and  $\bar{\gamma}_{2i,\theta=-\pi/2}$ . The relation between  $d_i$  and  $q_3$  cannot be derived with just two data points, and this change in step size due to inertial loading will be studied in the future, but we find a simple linear interpolation provides accurate results.

Fig. 10a-d shows the predicted model for  $\gamma_{2i}$  after computing  $d_i$  via calibration against experimental data collected on a single day with  $q_3 = 0$  and 12 mm. Fig. 10e-f shows the data collected at  $\theta = -\pi/4$  which is used to test the validity of the model for  $\theta \neq 0$ . The predicted models obtained after calibration are found to be accurate to within  $1 \mu\text{rad}$  ( $\pm 2\%$ ). Thus, the free parameters for the rotary joint can be calibrated for by using eight  $\bar{\gamma}_{2i}$  measurements.

## 4 Experiments

The experiments in this paper were designed to isolate and study the effect of static loads on the rotary and prismatic joints of a Kleindiek MM3A. The gravitational load acting on the prismatic joint (joint 3) along the direction of its motion can be described by the angles  $q_2$ ,  $\theta$ , and  $\psi$  (Fig. 2). To study the effect of gravity on the prismatic joint, data was recorded in two different experiments. For each value of  $q_2$ ,  $\theta$ , and  $\psi$ ,  $\bar{\gamma}_{3+}$  was first recorded followed by  $\bar{\gamma}_{3-}$ . In experiment 1,  $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{3-}$  were recorded at different values of  $q_2$  and  $\theta$  in the range of 0 to  $\pi$  and 0 to  $-\pi/2$ , respectively, in increments of  $\pi/6$  with  $\psi$  fixed at 0. For each value of  $q_2$ , data was recorded for different values of  $\theta$  before moving on to the next value of  $q_2$ . In experiment 2,  $q_2$  was fixed at  $-\pi/2$  (outstretched) and the gravitational load was varied by changing  $\psi$ , with  $\theta = 0$ .  $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{3-}$  were recorded for one condition in experiment 1 followed by the corresponding condition in experiment 2, before recording data for the next condition in both experiments. This distributes any drift in  $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{3-}$  due to time equally in both experiments. One trial for each condition in both experiments was taken per day for three consecutive days



**Fig. 10** (a) Model equation fitted to experimental data taken on a single day for  $\bar{\gamma}_{2i}$  at (a)  $q_3 = 0$  mm,  $\theta = 0$ ,  $\psi = 0$  in the negative direction, with  $d_- = 3.83 \mu\text{rad}$  (b)  $q_3 = 0$  mm,  $\theta = 0$ ,  $\psi = 0$  in the positive direction, with  $d_+ = -8.15 \mu\text{rad}$  (c)  $q_3 = 12$  mm,  $\theta = 0$ ,  $\psi = 0$  in the negative direction, with  $d_- = 23.94 \mu\text{rad}$  (d)  $q_3 = 12$  mm,  $\theta = 0$ ,  $\psi = 0$  in the positive direction, with  $d_+ = -19.40 \mu\text{rad}$  (e)  $q_3 = 0$  mm,  $\theta = -\pi/4$ ,  $\psi = 0$  in the negative direction, with  $d_- = 6.91 \mu\text{rad}$  (f)  $q_3 = 0$  mm,  $\theta = -\pi/4$ ,  $\psi = 0$  in the positive direction, with  $d_+ = -10.24 \mu\text{rad}$

to take into account the effect of unmodeled changes in environmental conditions. Curves 1 and 3 in Fig. 4a are obtained from  $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{3-}$ , respectively, recorded in experiment 1 using the values when  $\theta = -\pi/2$ , for all three days. Curves 2 and 4 are obtained from  $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{3-}$ , respectively, recorded in experiment 2. Fig. 4b shows  $\bar{\gamma}_{3+}$  and  $\bar{\gamma}_{3-}$  for experiment 1 recorded on Day 1 when  $\theta = 0$  and  $-\pi/3$ . Data from experiments 1 and 2 performed on the same day were used to derive the model parameters shown in Fig. 4b.

For modeling a rotary joint, two experiments were performed on joint 2. Initially,  $\bar{\gamma}_{2+}$  and  $\bar{\gamma}_{2-}$  was measured for the entire range of motion for the joint with

$q_3 = 0$  mm and 12 mm. Using a custom-made fixture, joint 2 was then moved in intervals of  $\pi/6$  for  $q_2$  from 0 to  $\pi$  and  $\bar{\gamma}_{2i}$  was calculated for each interval. The average step size of each interval is assumed to be the step size at the midpoint of the interval as shown in Fig. 6. This allows us to study the variation in  $\bar{\gamma}_{2i}$  as a function of  $q_2$ .  $\bar{\gamma}_{2i}$  was recorded in both negative ( $\bar{\gamma}_{2-}$ ) and positive ( $\bar{\gamma}_{2+}$ ) directions. In experiment 1, the mentioned sequence of collecting data was performed at  $\theta = -\pi/2$ . When  $\theta = -\pi/2$ , there is no torque due to gravity on the rotary joint, and the step size observed is purely due to the inertial load on the joint and the inherent properties of the actuator. In experiment 2,  $\theta$  is kept at zero. A gravitational torque is present on the rotary joint, and the step size obtained is influenced by gravitational loading on the joint. One set of data for both experiments was recorded on three different days. Fig. 6 shows the results for  $\bar{\gamma}_{2i}$  in experiment 2 for all three days, with  $\theta = 0$ . Fig. 10a-d shows the data for  $\bar{\gamma}_{2i}$  from experiment 2 for a single day with the predicted model fitted to the experimental data. An additional set of data was recorded at  $\theta = -\pi/4$  to check the validity of the model described in Eq. 8, the results of which are shown in Fig. 10e-f.

## 5 Main Experimental Insights

From the experiments performed in this paper, it was concluded that the step size of a piezoelectric stick-slip actuator can be modeled as having two summed components—a baseline step size that occurs when there is no static load acting on the joint, and a positive/negative contribution due to any static load acting on the joint—and that this two-component step size must be modified to account for the manipulator being in a configuration in which its compliance decreases the efficiency of the stick-slip movement.

Models relating the step size to the static loads were developed for a prismatic (joint 3) and a rotary joint (joint 2) of the Kleindiek MM3A. The actuator-specific parameters of the model can be calibrated for by taking 14 measurements of the average step size (6 for the prismatic joint and 8 for the rotary joint) in specific configurations of the manipulator. The models can accurately predict the step size of the joints at a given manipulator configuration. Kleindiek does not provide specifications for step size of the joints of the MM3A, so we compare the accuracy of our model to a simpler constant-step-size model when there is no static load acting on the joints, i.e.,  $\bar{\gamma}_{3i}$  at  $(q_2, \theta, \psi) = (-\pi/2, 0, 0)$  for the prismatic joint, and  $\bar{\gamma}_{2i}$  at  $(q_3, \theta, \psi) = (0, 0, 0)$  for the rotary joint. The maximum error in the developed model is approximately 15% for the prismatic joint, and 2% for the rotary joint, as compared to 40% and 7% for the prismatic and rotary joints, respectively, when using the constant-step-size model. Changes in environmental conditions have an effect on the parameters of the model; consequently, the model for the joint parameters should be recalibrated each day.

Fig. 6 shows that there is a significant effect of the joint 3 variable  $q_3$  on the step size of the rotary joint 2. Also,  $\bar{\gamma}_{2i, \theta = -\pi/2}$  and  $d_i$  are functions of  $q_3$ . This is expected, as a change in  $q_3$  will lead to a change in inertial load on joint 2 and a



change in the step size of joint 2. The effect of inertial loads on the step size are not addressed herein and will be studied in the future.

Joint 1 is another rotary joint with the same range as joint 2 and having the same properties except for the change in static load value. Hence, the model and calibration routine for joint 2 can be extended to joint 1. The only difference in the calibration routine would be that at  $\theta = 0$  there is no effect of gravity on joint 1, while at  $\theta = -\pi/2$  the gravity is perpendicular to the joint axis. So, in short, the definition of the terms,  $\bar{\gamma}_{2i,\theta=0}$  and  $\bar{\gamma}_{2i,\theta=-\pi/2}$  would be interchanged.

Models developed in this paper for the step size of piezoelectric stick-slip actuators are not perfect. Hence, when these models are used in teleoperation algorithms like the one proposed in [17], there will be drift in the position of the end-effector due to the accumulation of error in the model. However, this problem can be overcome as recently developed piezoelectric actuators have sensors with micro- and nanometer resolution [4]. This sensor feedback could be used to remove drift in the position, but the models of step size will still be necessary to command multiple steps in a single command to the joint before sensor feedback is obtained.

The experiments in this paper were performed in a room without tight climate control. When using the manipulator inside an SEM, frequent recalibration might not be necessary, since the manipulator will be in a vacuum. However, the audio limit switch used to detect end of travel will not work in a vacuum, and will need to be replaced by an accelerometer-based sensor mounted on the manipulator (when sensor feedback is not available) to detect the end of travel during calibration. The experiments in this paper were performed using a Kliendiek MM3A, but we expect the results to generalize to other similar devices that utilize piezoelectric stick-slip actuators.

## References

1. Attocube Nanopositioners, <http://www.attocube.com/>
2. Imina Technologies miBot, <http://www.imina.ch/>
3. Kleindiek Nanotechnik MM3A, <http://www.nanotechnik.com/>
4. SmarAct GmbH, <http://www.smaract.de>
5. Zyvex Nanomanipulator, <http://www.zyvex.com/>
6. Breguet, J.M., Clavel, R.: Stick and slip actuators: design, control, performances and applications. In: Proc. Micromechanics and Human Science, pp. 89–95 (1998)
7. Breguet, J.M., Driesen, W., Kaegi, F., Cimprich, T.: Applications of piezo-actuated micro-robots in micro-biology and material science, pp. 57–62 (2007)
8. Chang, S.H., Li, S.S.: A high resolution long travel friction-drive micropositioner with programmable step size. Review of Scientific Instruments 70(6), 2776–2782 (1999)
9. Fatikow, S., Wich, T., Hülsen, H., Sievers, T., Jähnisch, M.: Microbot system for automatic nanohandling inside a scanning electron microscope. IEEE/ASME Trans. Mechatronics 12(3), 244–252 (2007)
10. Higuchi, T., Yamagata, Y.: Micro robot arm utilizing rapid deformations of piezoelectric elements. Advanced Robotics 6(3), 353–360 (1992)
11. Hötendorfer, H., Giouroudi, I., Bou, S., Ferros, M.: Evaluation of different control algorithms for a micromanipulation system. In: Int. Conf. Engineering and Mathematics (2006)

12. Jung, H., Shim, J.Y., Gweon, D.: New open-loop actuating method of piezoelectric actuators for removing hysteresis and creep. *Review of Scientific Instruments* 71(9), 3436–3440 (2000)
13. Li, J.W., Yang, G.S., Zhang, W.J., Tu, S.D., Chen, X.B.: Thermal effect on piezoelectric stick-slip actuator systems. *Review of Scientific Instruments* 79(4), 046108 (2008)
14. Liaw, H.C., Shirinzadeh, B., Smith, J.: Enhanced sliding mode motion tracking control of piezoelectric actuators. *Sensors and Actuators A: Physical* 138(1), 194–202 (2007)
15. Lockwood, A.J., Wang, J.J., Gay, R., Inkson, B.J.: Characterising performance of TEM compatible nanomanipulation slip-stick inertial sliders against gravity. *Journal of Physics: Conference Series* 126(1) (2008)
16. Lv, Y., Wei, Y.: Study on open-loop precision positioning control of a micropositioning platform using a piezoelectric actuator. In: *Proc. World Congress on Intelligent Control and Automation*, pp. 1255–1259 (2004)
17. Nambi, M., Damani, A., Abbott, J.J.: Toward intuitive teleoperation of micro/nano-manipulators with piezoelectric stick-slip actuators. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 445–450 (2011)
18. Peng, J.Y., Chen, X.B.: Modeling of piezoelectric-driven stick-slip actuators. *IEEE/ASME Trans. Mechatronics* 16(2), 394–399 (2011)
19. Schurzig, D., Labadie, R.F., Hussong, A., Rau, T.S., Webster III, R.J.: Design of a tool integrating force sensing with automated insertion in cochlear implantation. *IEEE/ASEM Trans. Mechatronics* 17(2), 381–389 (2012)
20. Tonet, O., Marinelli, M., Megali, G., Sieber, A., Valdastrì, P., Menciassi, A., Dario, P.: Control of a teleoperated nanomanipulator with time delay under direct vision feedback. In: *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3514–3519 (2007)

## Part II: ISER Session Summary on “Dynamics and Control”

Patrick van der Smagt

Institute for Informatics,  
Technische Universität München

The ISER is unique in that it focuses on experimental results, without leaving theory out of sight. This session is an excellent example thereof: the four papers demonstrate how their theoretical approaches can be used in practical settings. Interestingly, three out of four people focus on human-robot interaction.

In that, the paper “Identification of human limb stiffness in 5 DoF and estimation via EMG” by Lakatos *et al.* proposes a method to identify impedance parameters of the human arm in realistic settings, and verify their method through actual measurements. These data are very applicable to the paper “Motor vs. Brake: Performance and Safety Analysis in Hybrid Actuation for Human-Friendly Robots” by Shin *et al.*, where a new, variable-impedance actuator is proposed which combines pneumatic with electric actuation, combining the high power density of the former with the accuracy of the latter. Methods can be evaluated with the approach described in “Rapid Prototyping of Planning, Learning and Control in Physical Human-Robot Interaction” by Lawitzky *et al.* In this paper, a hierarchical system of abstraction layers allows for faster evaluation of parts of the system. Finally, biological models and robotic implementation join in “Effective Use of Rear Legs in Quadrupedal Dynamic Climbing” by Miller *et al.*, where two biology-based walking models are extended to four-legged walking—where the models originally stemmed from.

# Rapid Prototyping of Planning, Learning and Control in Physical Human-Robot Interaction

Martin Lawitzky, José Ramón Medina Hernández, and Sandra Hirche

**Abstract.** Physical human-robot interaction (pHRI) is a highly challenging research topic: it requires real-time decision making capabilities by the robot; it involves the human as a source of uncertainty in the coupled dynamical system; and the quality of interaction cannot be evaluated by classical objective measures only but requires psychological experiments. Here we propose a rapid prototyping system in order to develop and evaluate methods for planning, learning, and control enabling *pro-active and goal-directed physical robotic assistance* to the human. With this rapid prototyping system we are able to quantify the benefits of two novel methods that combine feedback planning and learning from demonstration in a cooperative load-transport task.

## 1 Introduction

Recent advances in robotics research render the prospect of robotic assistants entering weakly structured, daily-life domestic, and industrial scenarios within reach of the next decade. Some of the most challenging problems are in the area of *pro-active and goal-directed physical robotic assistance* to the human. Research in this area imposes a number of particular challenges beyond those of mobile manipulation: The direct physical coupling with the human requires real-time planning and instant decision making capabilities of the robot. In addition, the uncertain and temporally varying behavior of the human is difficult to model. As result predictions or simulations of human behavior suffer from high uncertainty. Anticipation of the partner's action, however, is key for successful joint action [27]. Furthermore, methods for

---

Martin Lawitzky · José Ramón Medina Hernández · Sandra Hirche  
Institute for Information-Oriented Control,  
Technische Universität München, Munich, Germany  
e-mail: ml@tum.de  
<http://www.itr.ei.tum.de>

continual negotiation and role allocation using the haptic channel are required as the motion plans of the human and the robot will be different in the most general case. The quality of interaction cannot be evaluated based on classical purely objective measures only, but requires psychological experiments. This, in turn, requires robust, safe, and flexibly changeable experimental systems suitable for studies with *naïve* users.

The contribution of this work is twofold: i) We present a rapid prototyping experimental environment for the development of integrated planning, learning and control schemes for pHRI, their objective evaluation, and psychological evaluation in naïve-user studies. It includes a high-fidelity virtual reality (VR) system with a two degrees-of-freedom (DoF) haptic interface and the seamless transfer of the implemented schemes to a highly integrated mobile robot using a real-time capable modular software system. ii) We present novel results on the *combination* of planning and learning mechanisms for goal-directed assistance in physical human-robot interaction using this rapid-prototyping setup. In particular, the complementary properties of a learning algorithm [3] and a feedback-planning algorithm [31] are examined, to derive strategies for a synergetic combination of these two approaches. As prototypical task we investigate the cooperative transport of an object from an initial to a final configuration through a cluttered environment. Similar conditions are found in mobility assistance to humans, physical rehabilitation, and computer-aided assembly. The synergy strategies are superior over the individual schemes as validated in a naïve-user study and a proof-of-concept full-scale experiment.

## 1.1 Related Work

Most of the existing approaches towards motion generation for goal-oriented robotic physical assistant behavior can be classified as either a) planning-based approaches or b) learning-based approaches. In particular, feedback-planning approaches suit the need for instant decision making while explicitly incorporating environment constraints. Learning approaches, in turn, enable the (probabilistic) modeling and prediction, i.e. intention recognition, of the human partner after a training phase. Incremental learning approaches, which are considered here, continuously adapt the prediction model, i.e. are also capable to capture temporal variations in human behavior. While feedback-planning approaches find kinematic solutions neglecting task dynamics, no prior training data set is required to apply them in new scenes.

Motion planning in the context of physical robotic assistants finds only little reference in the literature. Taïx et al. present a method improve the sampling strategy of a path planner through haptic input while supporting the human guide to find the goal in a two-dimensional virtual reality scenario [29]. Feedback motion planning for haptic guidance using a cell-decomposition method is proposed in the context of computer-aided

design [24]. While planning-based approaches estimate the desired direction of motion indirectly by assuming a shortest-path cost function, it is well known that the estimation of the human partner’s intention is crucial for intuitive physical human-robot interaction [7]. Methods to estimate human intention based on the minimum-jerk assumption [8] are often deployed [20, 4]. Within the programming-by-demonstration framework a significant body of methods to learn motion patterns from human demonstration, to recognize such, and to estimate human intention are developed. A few works successfully apply probabilistic approaches like Gaussian mixture models [11, 12] and time-based Hidden Markov Models [19, 21] to the intention recognition problem in pHRI, see also [2] for an excellent survey. So far, most of the existing works investigate individual algorithms for planning [29, 24], learning/prediction [20, 4, 12, 19], and control [26, 16, 15, 15, 32, 6] for physical human-robot interaction in a separate way. Only very few works consider the combination of learning and control in pHRI [12, 21]. To the best of the authors knowledge there exists no work — except our own [17] — on the combination of planning and learning for pHRI.

The remainder of this article is organized as follows: A structural rapid-prototyping framework for pHRI is proposed in Section 2, the experimental setup is described in Section 3. Section 4 describes the evaluation methods leading to the results presented in Section 5.

Notation: Bold characters are used for vectors and matrices. The configuration space of the manipulated rigid object is denoted  $\mathcal{C}$  which is a manifold  $\mathcal{C} = \mathbb{R}^2$  in two-dimensional case without rotation and  $\mathcal{C} = \mathbb{R}^3 \times \mathbb{RP}^3$  in the three-dimensional case with rotation. The *obstacle region* is denoted  $\mathcal{C}_{\text{obs}} \subseteq \mathcal{C}$ . The remaining configurations are called *free space* denoted  $\mathcal{C}_{\text{free}} = \mathcal{C} \setminus \mathcal{C}_{\text{obs}}$ .

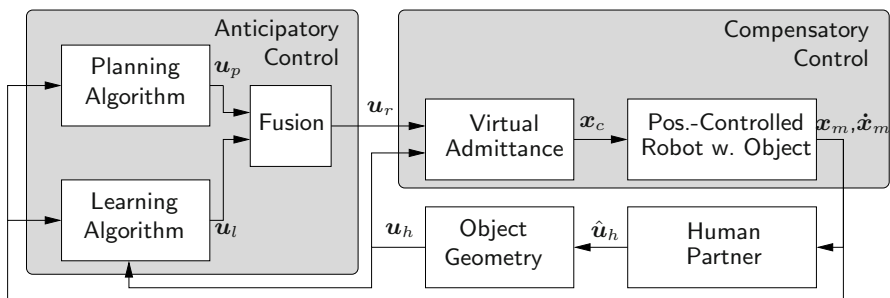
## 2 Technical Approach

The quality of assistance benefits from goal-oriented robot behavior – in contrast to purely reactive behavior – as shown for example in cooperative load transport tasks [18, 23]; the behavior can be generated from plan-based strategies [14] as well as motion reproduction strategies [21, 22]. For simplicity and clarity we focus here on the physical assistance in the redundant degrees of freedom of the task, i.e. a more pro-active robot reduces the required human effort to solve the task.

### 2.1 Architecture for Integrated Planning, Learning and Control

The complexity of the problem of goal-oriented behavior design in pHRI requires a systematic decomposition into simpler modules Mixed real-time

criticality and the continual negotiation process with the human partner suggests a decomposition inspired by the cognitive architecture known as the Extended Control Model (ECOM) [9]. We propose a dynamic feedback control structure that embeds anticipatory control mechanisms as well as compensatory schemes to cope with unexpected behavior of the human partner or the environment. The proposed architecture for integrated planning, learning, and control in pHRI is depicted in Fig. 1. The anticipatory behavior is composed by a) a feedback planning scheme that generates a goal-directed force  $\mathbf{u}_p$  from the current object configuration  $\mathbf{x}_m$ , b) a learning-based method that generates a force  $\mathbf{u}_l$  depending on the observations of configuration  $\mathbf{x}_m$ , velocity  $\dot{\mathbf{x}}_m$ , and the human effective wrench  $\mathbf{u}_h$  (calculated from the force applied at the grasp point  $\hat{\mathbf{u}}_h$ ), and c) a situation-dependent fusion technique that decides on the robot's virtual exerted force. For compensation of deviations from the anticipated trajectories, a compliance control scheme is used.



**Fig. 1** Architecture for integrated planning, learning and control in pHRI

Following this structure, different alternative implementations of anticipatory and compensatory control layers are experimentally evaluated on our rapid prototyping experimental system:

**Anticipatory Control.** In this work, two concurrent implementations of the anticipatory layer, the sampling-based neighborhood graph (SNG) for feedback planning [31] and the time-based HMM (tHMM) learning, recognition and a learning-based framework [19, 21] are briefly reviewed. Alternatively, joint planning through automatic segmentation and verbal communication in combination with intention recognition implements perception and modeling of the task and the partner behavior [21]. Motion generated from the learned human motion model and the estimated human intention or plan-based alternatives [14] implement alternative anticipatory building blocks. Alternatives for the implementation of compensatory control include static role allocation (leader/follower) [18] or dynamic effort sharing strategies [23].

Alternatively, a risk-sensitive optimal feedback control scheme generates an intuitive robot force contribution depending on observed human execution variability [22].

**Compensatory Control.** We assume the robotic assistant is feedback-controlled such that the commonly manipulated object follows a virtual object impedance that is rendered with inertia matrix  $\mathbf{M}$ , and virtual viscous friction  $\mathbf{D}$ .

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} = \mathbf{u}_h + \mathbf{u}_r, \quad (1)$$

where  $\mathbf{x}$  denotes the object pose and  $\mathbf{u}_h$  and  $\mathbf{u}_r$  the effective wrenches by human and robot, respectively. Motion generation algorithms continuously produce wrenches  $\mathbf{u}_p$ ,  $\mathbf{u}_l$  that act on a virtual admittance in superposition with the human partner’s force  $\mathbf{u}_h$  exerted through the object on the end effector of the robot.

### 3 Experimental Setups of the Rapid Prototyping System

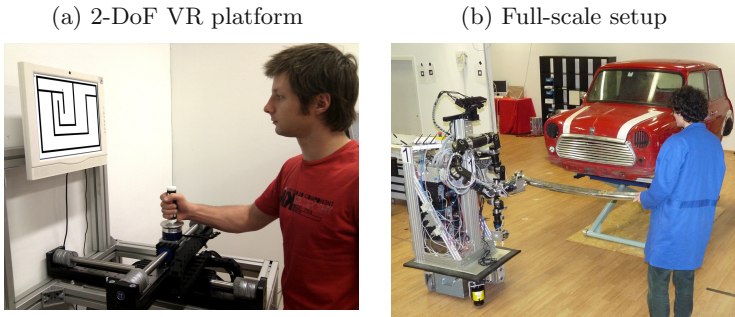
A rapid prototyping experimental system is developed in which complex algorithms for pHRI can be evaluated in plug-and-play fashion — also in user studies — using modular building blocks. In a first stage the developed algorithms are tested on a 2-DoF haptic device in a virtual haptic environment. Afterwards, they are directly transferred to a large-scale anthropomorphic mobile robotic system with integrated visual tracking and laser-range based localization schemes.

#### 3.1 High-Fidelity Virtual Reality System in 2-DoF

Psychological and other quantitative measurements require a safe experimental environment with controlled conditions. Therefore, a 2-DoF VR system (see Fig. 2(a)) is deployed for quantification of the performance of novel methods. The virtual-reality environment consists of a 2-DoF linear-actuated device (*ThrustTube*) with a free-spinning handle at the grasp point. A force/torque sensor (*JR3*) attached to the handle measures the human interaction force. The virtual scene is visually represented on a display placed on top of the interface. The scene information can be imported from virtually any standard monochrome pixel or vector graphics file and is automatically rendered as stiff environment. A virtual rigid polygon or ellipsoid object with distributed grasp points can be specified and emulated.

For haptic rendering of arbitrary scenes the scene information is imported from any standard monochrome pixel or vector graphics file and is





**Fig. 2** Experimental Systems

automatically rendered as stiff environment. The control algorithm is implemented in *Matlab/Simulink's Real-Time Workshop* and executed on *Linux Preempt/RT* at a frequency of 1 kHz.

### 3.2 6-DoF Experimental System with Mobile Manipulator

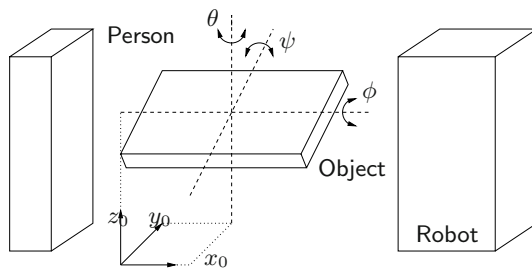
After successful evaluation in the 2-DoF VR system the algorithms are transferred to the a full-scale cooperative transport scenario with a human-sized mobile manipulator in a highly integrated experimental environment. In particular, properties such as scalability to higher degrees of freedom, robustness to noise in autonomous mobile manipulation and safe behavior outside the expected motion corridor are evaluated. The mobile robot used in this experiment, see Fig. 2(b), locomotes with a four-wheeled omni-directional mobile platform, which offers roughly human-like maneuverability and smooth motion [13]. Two identical anthropomorphic 7-DoF manipulators provide a human-like work space [28]. The manipulators are equipped with a *Schunk PG70* two-finger parallel grippers which allow a tight grasp of objects [21]. They are mounted onto *JR3* force/torque sensors. A point cloud of the environment is acquired using a tilted *Hokuyo UBG* laser range finder. Two *Sick S300* laser range finders scan for obstacles on floor level. A ceiling camera system consisting of 40 cameras covers a space of 10x10m and supports the localization of humans and robots. The software framework is based on the modular real-time architecture *ARCADE* [1]. It interfaces to *ROS* in a seamless manner; for example *ROS* algorithms are utilized for self-localization. The admittance-type control algorithm is implemented analogously to the VR setup and communicates with other software modules through the high-bandwidth real-time data base *RTDB* [10] at a frequency of 1 kHz.

Similar to the VR scenario, a virtual admittance is rendered in world coordinates at the robot's end effector. The mobile platform is controlled such that the manipulability is maximized and joint limits are avoided. Following

this strategy, transferability of methods and results from the VR setup is simplified as only the end effector configuration remains to be controlled by the anticipatory scheme.

## 4 Evaluation in the Rapid Prototyping System

To examine goal-oriented behavior strategies in our proposed rapid prototyping system, we consider the problem of a human-robot dyad cooperatively moving an object from a start configuration  $\mathbf{x}(0) \in \mathcal{C}_{\text{free}}$  to a different final configuration  $\mathbf{x}(t_{\text{goal}}) \in \mathcal{C}_{\text{free}}$  through a cluttered environment. The human partner and the robot are assumed to control their posture such that they keep a constant configuration relative to the object on the two-dimensional ground plane  $(x_0, y_0)$ , see Fig. 3. Aspects of approximation of person-object-robot representations for motion planning are discussed in [5].



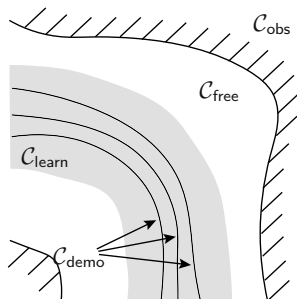
**Fig. 3** 6-DoF reduced geometric model of the cooperation partners and the object used for calculation of free configuration space  $\mathcal{C}_{\text{free}}$

We deploy one planning-based and one learning-based approach for goal-directed robotic assistance and proposed two different schemes to combine the strengths of both approaches.

### 4.1 Feedback Motion Planning

A feedback motion planning algorithm generates a feedback function  $\mathbf{K}(\mathbf{x})$  for all positions  $\mathbf{x} \in \mathcal{C}_{\text{free}}$  in the accessible configuration space  $\mathcal{C}_{\text{free}}$ . The SNG is a very comprehensible method, sufficiently efficient to cover higher-dimensional configuration spaces applicable for settings with 6-dimensional object poses at a viable resolution as required for our large-scale scenario. Therefore, the entire configuration space is randomly clustered into overlapping hyperballs, and Dijkstra's algorithm is applied to plan on the connected graph from the ball containing the initial configuration to the ball containing the final configuration. Finding the shortest path within hyper balls is straightforward.

In this planning-based approach the robot computes its motion plan based on environmental constraints and possibly under consideration of manipulability constraints by the human partner. In consequence the robot plan will generally differ from the human plan and an online negotiation is required.



**Fig. 4** Subsets of the configuration space  $\mathcal{C} = \mathbb{R}^2$

Given a desired magnitude of velocity and a desired maximum robot force level, the force<sup>1</sup> vector  $\mathbf{u}_p$  is calculated straightforwardly with the virtual object's inverse dynamical model.

## 4.2 Learning from Demonstration

Instead of directly calculating a goal-directed plan, learning-based approaches to physical robotic helpers rely on human-behavior anticipation. This principle requires a model of the task and the human partner, which is acquired from previous demonstrations.

Here we deploy time-based Hidden Markov Models (tHMM) to represent human demonstrations in a compact manner. Advantageously, tHMMs constrain the human motion prediction to the zone of influence of the configuration space  $\mathcal{C}_{\text{learn}} \subseteq \mathcal{C}_{\text{free}}$  around the demonstrated configurations  $\mathcal{C}_{\text{demo}} \subseteq \mathcal{C}_{\text{learn}}$ . This feature is crucial in order to increase safety and ensure intuitive motion generation in direct interaction with a human partner. The added explicit time information and Gaussian Mixture Regression provides a generalization of smooth trajectories of the encoded demonstrated human behavior [19]. The human-motion predictions serve as a goal for the robot's anticipatory motion generation, as explained in [21] and are used to generate a learning-based virtual force  $\mathbf{u}_l$ .

In this learning-based approach the robot initially behaves *passive*, i.e.  $\mathbf{u}_r = 0$  during the task execution and observes the executed trajectory which can be assumed to be close to the human partner's intended path. In subsequent

<sup>1</sup> Force in the 2D case, wrench in the 6D case.

trials the robot adopts this as its own motion plan. Note that the space of learned trajectories is generally only a subspace of the free space, but generally the subspace of learned trajectories is larger than the space of demonstrated trajectories  $\mathcal{C}_{\text{demo}} \subseteq \mathcal{C}_{\text{learn}} \subseteq \mathcal{C}_{\text{free}}$ , see Fig. 4. Algorithms for motion learning and reproduction have limited generalization capabilities and can thus cover regions neighboring the demonstrated trajectories.

### 4.3 Synergy Strategies Based on Planning and Learning

Given the advantages of each individual scheme for motion generation, we derive two strategies to exploit synergies in the following.

**Prediction-Quality-Based Homotopy Blending.** This method exploits the probabilistic characteristics of the motion recognition and prediction algorithm. As recognition uncertainty grows, the goal-oriented fall-back solution generated by the feedback planning algorithm dominates. The proposed homotopy blends between the output functions  $\mathbf{u}_p$  and  $\mathbf{u}_l$  of the planning and learning schemes

$$\mathbf{u}_r = \gamma \mathbf{u}_l + (1 - \gamma) \mathbf{u}_p$$

with prediction certainty  $\gamma \in [0, 1]$ . The outputs of the planning algorithm and the learning algorithm are denoted  $\mathbf{u}_p$  and  $\mathbf{u}_l$  respectively. We regard the unnormalized likelihood of the state estimate as a measure for certainty, see also [30]. This strategy overcomes the limitation of the learning-based algorithm to produce active task contributions only within the zone of influence  $\mathcal{C}_{\text{learn}}$  and blends continuously into the feedback motion plan valid for the entire accessible configuration space  $\mathcal{C}_{\text{free}}$ .

**Cost-Based Fusion of Strategies.** This method is based on the parallel evaluation of the strategies and a cost-based fusion using a sophisticated hierarchical multi-criteria decision making (MCDM) algorithm on the efficient (Pareto) frontier. For each of the planning and the learning-based approaches an individual utility is evaluated. The utility function  $\mathcal{U}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u})$  of the planning scheme is calculated through simulation of the effects of different forces  $\mathbf{u}_p$  and  $\mathbf{u}_l$  on the *cost to go* by evaluation feedback plan. The utility is maximal for force vectors along the planning algorithm output  $\mathbf{u}_p$  and minimal for the opposite direction. The utility value of 0 is cost-neutral and is reached for directions orthogonal to the planned direction of motion.

The utility function of the learning-based algorithm is described by the prediction certainty  $\gamma \in [0, 1]$ . The certainty value of 1 is reached at the mean direction of the motion prediction according to the unnormalized likelihood of the prediction.

A hierarchical MCDM-scheme inspired by [25] is deployed: The output of the learning-based approach is evaluated for its utility  $\mathcal{U}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}_l)$ . In case the utility is positive, the output of the learning-based approach is accepted,  $\mathbf{u}_r = \mathbf{u}_l$ . Otherwise, the closest direction to  $\mathbf{u}_l$  on the efficient frontier with non-negative utility  $\mathcal{U}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}_r) \geq 0$  is selected

$$\mathbf{u}_r = \underset{\mathbf{u} \in \{\mathbf{u}_i | \mathcal{U}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}_i) \geq 0 \cap |\mathbf{u}_i| \leq |\mathbf{u}_l|\}}{\arg \max} \mathbf{u}^T \mathbf{u}_l. \quad (2)$$

This results in a rotation of any  $\mathbf{u}_l$  with negative utility  $\mathcal{U}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}_l)$  onto the hyperplane orthogonal to  $\mathbf{u}_p$ .

## 5 Quantitative Evaluation Results

In order to quantify the performance of the proposed algorithms, a quantitative user study was performed utilizing the proposed rapid prototyping setup. Four conditions are tested: a) assisted by feedback planning, b) assisted by motion reproduction, c) homotopy switching based on prediction certainty, and d) a multi-criterion decision making algorithm.

The proposed approaches are evaluated in experiments with cooperating humans. This section describes the evaluation criteria and the quantitative results. For illustration, the feedback plan as well as the learned task models in pHRI in the small scale experiment in 2 DoF are shown in Figs. 6(a)-6(c), and for the large scale setup in 6 DoF in Figs. 6(d) -6(f).

A small pilot study in a virtual reality scenario was conducted yielding the results depicted in Fig. 5. Seven non-paid participants (age mean: 27.1, std: 1.5) were asked to move a virtual point mass object of 100 kg, through a simple maze from a starting configuration to a final configuration through the scene without colliding with the virtual obstacles visually and haptically displayed. The virtual object was exposed to a virtual viscous friction of 400 Ns/m.

Each participant repeated the experiment five times per each of the four conditions after five training trials without assistance.

### 5.1 Quantitative Measures

We evaluate the following criteria in order to rate the performance of the proposed approaches:

- Mean root-mean-square (RMS) deviation from the participant's path  $\tau_{\mathbf{x}, H5}$  after five trials

$$x_{\text{RMS}} = \sqrt{\frac{1}{T} \int_0^T d(\mathbf{x}(t), \tau_{\mathbf{x}, H5})^2 dt}$$

with distance for point  $\mathbf{x}(t)$  to path  $\tau_{\mathbf{x}, H5}$

$$d(\mathbf{x}(t), \tau_{\mathbf{x}, H5}) = \min_{\mathbf{x}_\tau \in \tau_{\mathbf{x}, H5}} \|\mathbf{x}_\tau - \mathbf{x}(t)\|.$$

This measure indicates the rate of convergence towards routine.

- Mean disagreement  $\mathbf{u}_D$  which can be defined orientation invariant:

$$\mathbf{u}_D = \begin{cases} \frac{-\mathbf{u}_h}{\|\mathbf{u}_h\|} \cdot \mathbf{u}_r, & \text{if } \mathbf{u}_h \cdot \mathbf{u}_r < 0 \wedge \mathbf{u}_h \neq \mathbf{0} \\ 0, & \text{otherwise.} \end{cases}$$

Larger values of disagreement  $\mathbf{u}_D$  indicate that the human and the robotic partner produce a higher amount of counteracting, and therefore inefficient forces.

- Mean completion time  $T_{\text{mean}}$  as indicator of efficiency of the cooperation.

Regarding the RMS deviation over trials, the conditions c) Homotopy blending and d) MCDM show fastest convergence and are in a similar range as the b), the pure learning-based approach. The pure planning-based approach a) leads to slower convergence. The measurements of mean disagreement show that the conditions a), c) and d) show equally low disagreement. The completion times of the fusion strategies c) and d) are similar to those of the planning-based approach and lower than in the learning-based strategy.

## 5.2 Experiments in 6 DoF

Our large-scale experiment is used to validate the approaches presented in Section 5 in a lifelike scenario. The cooperative transport of a car part is considered through the cluttered lab environment, see Fig. 5 for a map. The cooperatively manipulated object, a 1.2 m car part (b) weighing 1.9 kg. For fast computation, parallelized implementations on an *Intel Core i7 920 at 2.67GHz* were utilized. The computation times for our prototypical 6D-problem are given in Table 1.

**Table 1** Computation times of SNG and tHMM for robotic partner behavior generation in 6D

Computation step	SNG	tHMM
Generate neighborhood graph from point cloud ( $\alpha = P_c = 0.985$ )	639 s	
Planning per new goal configuration	< 1 s	
Expectation Maximization after each observation		~ 5 s
Direction lookup per control iteration	< 50 $\mu$ s	
Viterbi and regression during execution		< 500 $\mu$ s

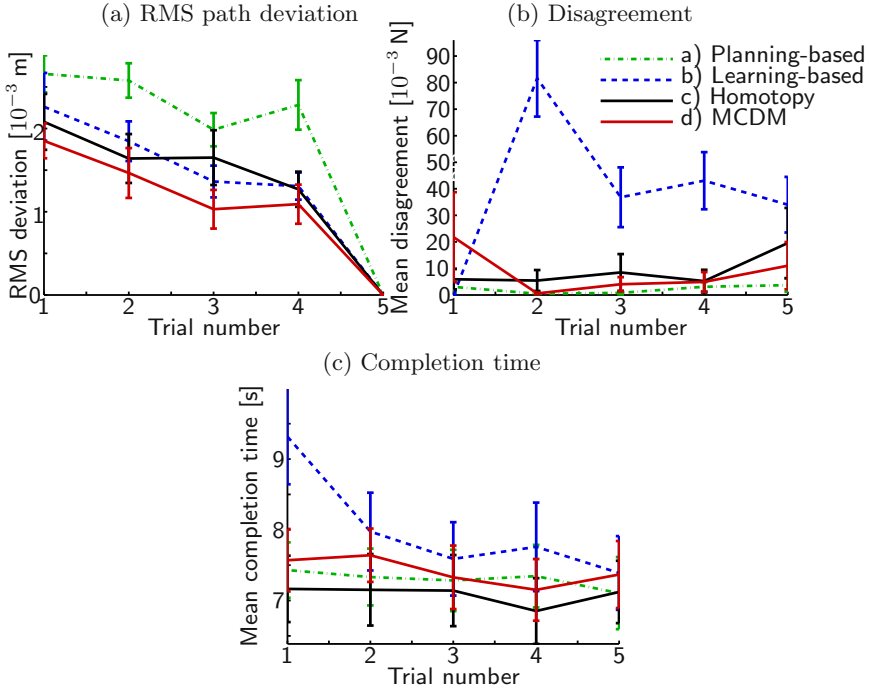
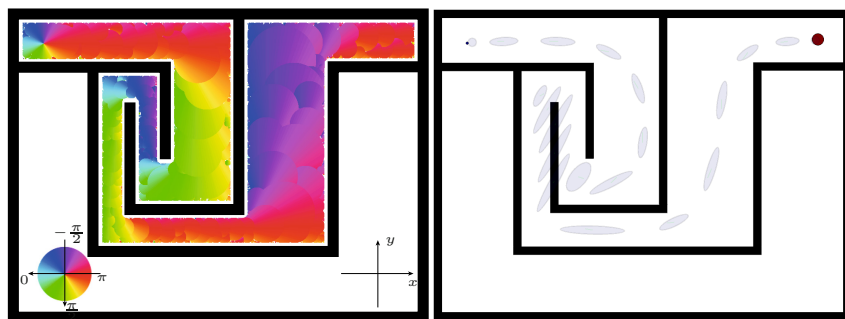


Fig. 5 Evolution of quantitative measures over trials

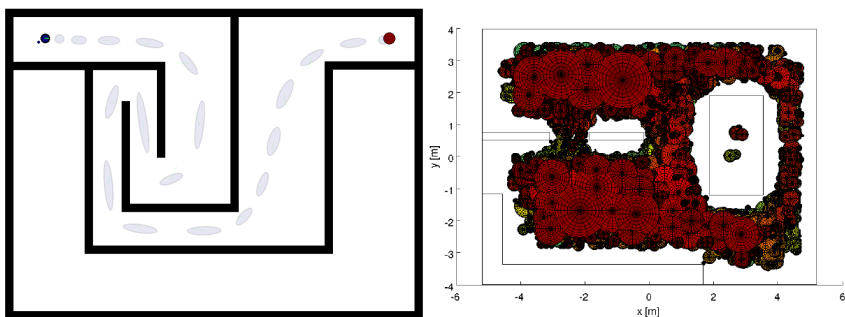
### 5.3 Main Experimental Insights

Our rapid prototyping environment is instrumental to investigate properties relevant to the application of cooperative load transport. A few insights are given here: The behavior of the planning-based approach does not evolve with the number of trials whereas the learning-based approach renders a purely reactive behavior in the first trial but converges towards the human partner's desired trajectory. Outside the area of observed demonstrations, however, the learning scheme is bound to observe the human but the planning approach is active for the entire free configuration space. This is due to the fact that the environment representation used in the feedback planning scheme is explicit and not implicitly given through rough training data. The planning scheme requires a desired velocity magnitude and a planning precision to operate. In contrast, the learning approach is sensitive to the number of states, the number of Gaussians per state, and the weighting of input importance. An advantage of the learning scheme is the linear complexity with respect to the input dimension whereas the feedback-planning approach scales exponentially with the output dimension. Depending on the application, both individual algorithms show different strengths and can render the preferable solution.



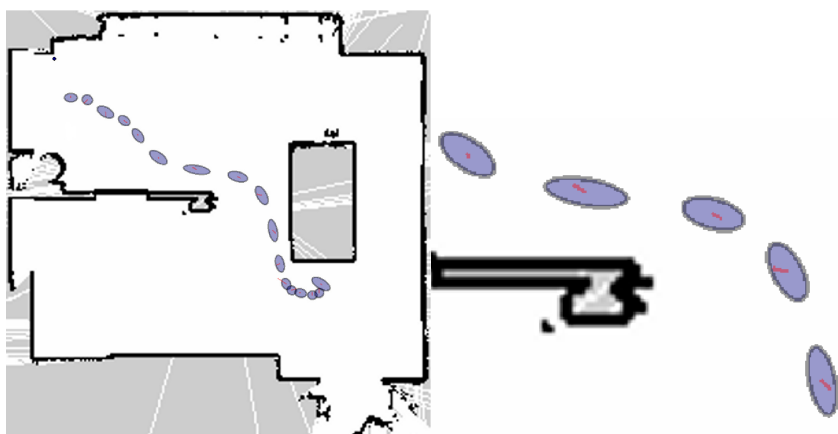
(a) Feedback motion plan

(b) Ergodic tHMM



(c) Left-to-right tHMM

(d) Planar components of feedback plan



(e)  $x/y$  components of Gaussians

(f) Magnified central region of Fig. 6(e)

**Fig. 6** Motion generation algorithms in a 2 DoF virtual haptic maze (a)-(c), and in 6 DoF in the 10x10m experimental area mapped by a laser-range finder (d)-(f)



The presented framework also provides a structured way to identify strategies to exploit synergies between both approaches, showing the additional complementary benefits of fusion-based strategies. From the quantitative results it is visible that the fusion strategies c) and d) combine the strengths of the planning-based and the learning-based approaches a) and b). These fusion methods outperform the planning-based approach regarding the required adaptation of the human and leads to small disagreement compared to the learning-based approach. The completion time under these conditions is on the level of the planning approach throughout all trials and is significantly lower than the unassisted condition in the first trial of the learning-based approach.

## 6 Conclusion

In this work, we demonstrate that our novel proposed modular architecture for prototyping of goal-directed physical robotic assistants embeds well various different approaches for anticipatory control. Transfer of methods and results from user studies on a 2-DoF high-fidelity VR setup to a 6-DoF leaves computational complexity as greatest remaining challenge. As an exemplary study, novel fusion-based strategies between planning and learning-based approaches show additional benefits towards interaction performance and validate the applicability of the presented framework.

**Acknowledgement.** The authors would like to thank Dongheui Lee for supporting this work with valuable advice. This research is partly supported by the DFG excellence initiative research cluster “Cognition for Technical Systems – CoTeSys”.

## References

1. Althoff, D., Kourakos, O., Lawitzky, M., Mörtl, A., Rambow, M., Rohrmüller, F., Bršćić, D., Wollherr, D., Hirche, S., Buss, M.: An Architecture for Real-time Control in Multi-robot Systems. In: Cognitive Systems Monog.: Human Centered Robot Systems, pp. 43–52 (2009)
2. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot Programming by Demonstration. In: Siciliano, B., Khatib, O. (eds.) Handbook of Robotics, pp. 1371–1394. Springer (2008)
3. Calinon, S., D’Halluin, F., Sauter, E.L., Caldwell, D.G., Billard, A.G.: Learning and reproduction of gestures by imitation. *IEEE Robot. Autom. Mag.* 17(2), 44–54 (2010)
4. Corteveille, B., Aertbelien, E., Bruyninckx, H., De Schutter, J., Van Brussel, H.: Human-inspired Robot Assistant for Fast Point-to-point Movements. In: Proc. IEEE ICRA, pp. 3639–3644 (2007)
5. Esteves, C., Archevalata, G., Laumond, J.: Motion Planning for Human-Robot Interaction in Manipulation Tasks. In: Proc. IEEE ICMA, pp. 1766–1771 (2005)

6. Evrard, P., Kheddar, A.: Homotopy Switching Model for Dyad Haptic Interaction in Physical Collaborative Tasks. In: Proc. EHS EuroHaptics, pp. 45–50 (2009)
7. Fernandez, V., Balaguer, C., Blanco, D., Salichs, M.A.: Active Human-Mobile Manipulator Cooperation Through Intention Recognition. In: Proc. IEEE ICRA, pp. 2668–2673 (2001)
8. Flash, T., Hogan, N.: The coordination of arm movements: An experimentally confirmed mathematical model. *J. Neurosci.* 5, 1688–1703 (1985)
9. Gauthereau, V., Hollnagel, E.: Planning, Control, and Adaptation: A Case Study. *European Management Journal* 23(1), 118–131 (2005)
10. Goebel, M., Färber, G.: A Real-Time-capable Hard- and Software Architecture for Joint Image and Knowledge Processing in Cognitive Automobiles. In: Proc. IEEE IV, pp. 734–740 (2007)
11. Gribovskaya, E., Khansari-Zadeh, S.M., Billard, A.: Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators. *Int. J. Rob. Res.* 30(1), 80–117 (2010)
12. Gribovskaya, E., Kheddar, A., Billard, A.: Motion learning and adaptive impedance for robot control during physical interaction with humans. In: Proc. IEEE ICRA (2011)
13. Hanebeck, U., Saldic, N., Schmidt, G.: A Modular Wheel System for Mobile Robot Applications. In: Proc. IEEE/RSJ IROS, pp. 17–22 (1999)
14. Kirsch, A., Kruse, T., Sisbot, E., Alami, R., Lawitzky, M., Brščić, D., Hirche, S., Basili, P., Glasauer, S.: Plan-Based Control of Joint Human-Robot Activities. *Künstl. Intell.* 24(3), 223–231 (2010)
15. Kosuge, K., Hirata, Y.: Human-Robot Interaction. In: Proc. IEEE ROBIO, pp. 8–11 (2004)
16. Kosuge, K., Yoshida, H., Fukuda, T.: Dynamic control for robot-human collaboration. In: Proc. IEEE Workshop on Robot and Human Communication, pp. 398–401 (1993)
17. Lawitzky, M., Medina, J.R., Lee, D., Hirche, S.: Feedback Motion Planning and Learning from Demonstration in Physical Robotic Assistance: Differences and Synergies. In: Proc. IEEE/RSJ IROS (2012)
18. Lawitzky, M., Mörtl, A., Hirche, S.: Load Sharing in Human-Robot Cooperative Manipulation. In: Proc. IEEE Ro-Man, pp. 185–191 (2010)
19. Lee, D., Ott, C.: Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Auton. Robot.*, 1–17 (2011)
20. Maeda, Y., Hara, T., Arai, T.: Human-robot cooperative manipulation with motion estimation. In: Proc. IEEE/RSJ IROS, vol. 4, pp. 2240–2245 (2001)
21. Medina, J.R., Lawitzky, M., Mörtl, A., Lee, D., Hirche, S.: An Experience-Driven Robotic Assistant Acquiring Human Knowledge to Improve Haptic Cooperation. In: Proc. IEEE/RSJ IROS, pp. 2416–2422 (2011)
22. Medina, J.R., Lee, D., Hirche, S.: Risk Sensitive Optimal Feedback Control for Haptic Assistance. In: Proc. IEEE ICRA (2012)
23. Mörtl, A., Lawitzky, M., Küçükylmaz, A., Sezgin, M., Basdogan, C., Hirche, S.: The Role of Roles: Physical Cooperation between Humans and Robots. *Int. J. Robot. Res.* (2012) (to appear)
24. Rosell, J., Vázquez, C., Pérez, A., Iñiguez, P.: Motion Planning for Haptic Guidance. *J. Intell. Robotic. Syst.* 53(3), 223–245 (2008)
25. Saaty, T.L.: *The Analytic Hierarchy Process*. McGraw-Hill International, New York (1980)

26. Schneider, S., Cannon, R.: Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results. *IEEE Trans. Robot. Automat.* 8(3), 383–394 (1992)
27. Sebanz, N., Knoblich, G.: Prediction in Joint Action: What, When, and Where. *Topics in Cognitive Science* (2), 353–367
28. Stanczyk, B., Buss, M.: Development of a Telerobotic System for Exploration of Hazardous Environments. In: *Proc. IEEE/RSJ IROS*, pp. 2532–2537 (2004)
29. Taïx, M., Flavigné, D.: Motion Planning with interactive devices. In: *Proc. IEEE ECMS*, pp. 1–6 (2011)
30. Thobbi, A., Gu, Y., Sheng, W.: Using Human Motion Estimation for Human-Robot Cooperative Manipulation. In: *Proc. IEEE/RSJ IROS*, pp. 2873–2878 (2011)
31. Yang, L., LaValle, S.: The Sampling-Based Neighborhood Graph: An Approach to Computing and Executing Feedback Motion Strategies. *IEEE Trans. Robot. Automat.* 20(3), 419–432 (2004)
32. Yoshikawa, T., Khatib, O.: Compliant motion control for a humanoid robot in contact with the environment and humans. In: *Proc. IEEE/RSJ IROS*, pp. 211–218 (2008)

# Identification of Human Limb Stiffness in 5 DoF and Estimation via EMG

Dominic Lakatos, Daniel Rüschen, Justin Bayer,  
Jörn Vogel, and Patrick van der Smagt

**Abstract.** To approach robustness and optimal performance, biological musculo-skeletal systems can adapt their impedance while interacting with their environment. This property has motivated modern robotic designs including variable-impedance actuators and control methods, based on the capability to vary visco-elastic properties actively or passively. Even though variable-impedance actuation and impedance control in robotics is resolved to a great part, a general set of rules by which impedance is adjusted related to the task at hand is still lacking. This paper aims to fill this gap by providing a method to estimate the stiffness of the human arm in more than two degrees of freedom by perturbation. To overcome ill-conditionedness of the impedance and inertial matrices, we propose and validate methods to separately identify inertial and stiffness parameters. Finally, a model is proposed to estimate the joint stiffness from EMG-measurements of muscle activities.

## 1 Motivation, Problem Statement, Related Work

Dynamic interaction with the environment means handling impacts and unknown contact forces. Therefore compliant systems are active topics of research in the

---

Dominic Lakatos · Daniel Rüschen · Jörn Vogel  
Institute of Robotics and Mechatronics,  
German Aerospace Center (DLR),  
D-82234 Oberpfaffenhofen, Germany  
e-mail: {dominic.lakatos,daniel.rueschen,joern.vogel}@dlr.de

Justin Bayer  
Chair for Robotics and Embedded Systems of the Department of Informatics  
of the Technische Universität München, D-80333 München  
e-mail: bayer.justin@googlemail.com

Patrick van der Smagt  
Institute for Informatics, Technische Universität München

field of robotics. Surpassing traditional rigid robots, the control loops of modern robotic systems are extended with additional impedance parameters, viz. stiffness and damping.

Even though the implementation of impedance control in robotics is resolved to a large part, one important issue still needs to be addressed: how are the impedance parameters set to optimally perform a task? Traditionally, robotic tasks are only defined in target end-effector positions or, in some cases, end-effector trajectories; but the impedance around these positions or trajectories remains a matter of common sense, at best. For instance, when performing a peg-in-hole task, high stiffness in the perpendicular and low stiffness in the lateral directions, so as to allow for imprecise positioning while solving the task, appears to be useful. But how do we find general rules-of-thumb for setting these extra parameters?

Beside heuristic methods tuning the impedance parameters, mimicking the behavior of the human arm is an auspicious field of research, and leads to what we call biologically-inspired robotics. By measuring and subsequently analyzing human arm impedance parameters, we can attempt to extract general rules and project these to the robotic domain.

The human arm's capability to alter its impedance has motivated multiple developments of robotic manipulators and control methods. It provides advantageous during manipulation such as robustness against external disturbances and task adaptability. However, how the impedance of the arm is set depends on the manipulation situation; a general procedure is lacking.

The only direct method to measure stiffness in a functioning feedback system is to apply external force perturbations to the limb and to measure the resulting displacements; such measurements have only been satisfactorily realized in planar (2D) movements [8, 12, 4, 5, 10, 2, 11]. To date, no fully satisfactory methods exist to investigate the time-varying impedance during movements. Early efforts were subject to error because they assume that subjects perform the same movement on repeated trials and they ignore the non-linear inertial properties of the musculo-skeletal system.

We provide a method to identify human arm impedance in more than 2 degrees of freedom. We do this by initially identifying the kinematic and inertial parameters of the arm through movement. Subsequently we identify stiffness parameters of the human arm in 5 degrees of freedom (shoulder, elbow, and lower arm rotation), while taking the numerical stability of the data into account. The data are related to a representation of the stiffness by electromyography (EMG) signals which, in combination with the kinematics, gives us a 3D Cartesian identification of the impedance parameters of the human arm.

## 2 Technical Approach

An adequate model describing the human limb dynamics can be separated in two power interconnected subsystems: the mass inverse dynamics of the skeleton (including the mass distribution of the muscles)

$$\Gamma(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\xi}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}}, \quad (1)$$

where  $\mathbf{q} \in \mathbb{R}^n$  are joint positions,  $\boldsymbol{\xi} \in \mathbb{R}^{\leq 10n}$  are base inertial parameters and  $\boldsymbol{\tau}_{\text{ext}}$  are external torques and general impedance functions of the muscular system, acting as force elements on the joints:

$$\boldsymbol{\tau} = -\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{a}). \quad (2)$$

We assume  $\mathbf{h} : \mathbf{q}, \dot{\mathbf{q}}, \mathbf{a} \rightarrow \boldsymbol{\tau}$  to be continuous, while the muscle activities  $\mathbf{a}$  are motor commands, which are able to shift the equilibrium point of the impedance. Thus, linearization in the working point  $\mathbf{x}_d := (\mathbf{q}(t=0), \dot{\mathbf{q}}(t=0), \mathbf{a}(t=0))$  yields:

$$\mathbf{h}^* = \underbrace{\mathbf{h}|_{\mathbf{x}_d}}_{\boldsymbol{\tau}_d} + \underbrace{\frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{a})}{\partial \mathbf{q}} \Big|_{\mathbf{x}_d}}_{\mathbf{K}_q} \tilde{\mathbf{q}} + \underbrace{\frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{a})}{\partial \dot{\mathbf{q}}} \Big|_{\mathbf{x}_d}}_{\mathbf{D}_q} \dot{\tilde{\mathbf{q}}} + \frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{a})}{\partial \mathbf{a}} \Big|_{\mathbf{x}_d} \tilde{\mathbf{a}} + \dots \quad (3)$$

Additionally, we assume that activations  $\mathbf{a} = \text{const.}$  (this can be fulfilled by certain experimental conditions); consequently the joint torques acting due to muscles can be approximated by:

$$\boldsymbol{\tau} = -\boldsymbol{\tau}_d - \mathbf{K}_q \tilde{\mathbf{q}} - \mathbf{D}_q \dot{\tilde{\mathbf{q}}}, \quad (4)$$

where  $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$  is the tracking error,  $\boldsymbol{\tau}_d$  are equilibrium torques and  $\mathbf{K}_q, \mathbf{D}_q$  are joint stiffness and damping matrices, respectively.

Identifying the complete parameter set (i.e.,  $\boldsymbol{\xi}, \mathbf{K}_q$  and  $\mathbf{D}_q$ ) from measurements of  $\boldsymbol{\tau}_{\text{ext}}$  would lead to an ill-conditioned least-squares problem [7], thus we estimate  $\boldsymbol{\xi}$  separately by projecting the inertial forces to the subject's mounting base where they can be measured with a force / torque sensor, i.e.,

$$\chi_0(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\xi}) = \chi_{\text{sensor}}. \quad (5)$$

Once the inertial parameters  $\boldsymbol{\xi}$  are known, the not directly measurable joint torques  $\boldsymbol{\tau}$  can be estimated via the inverse dynamic model and the identification model reduces to

$$\mathbf{K}_q \tilde{\mathbf{q}} + \mathbf{D}_q \dot{\tilde{\mathbf{q}}} = \boldsymbol{\tau}_{\text{ext}} - \Gamma(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\xi}) - \boldsymbol{\tau}_d, \quad (6)$$

where only the left hand side is unknown. This separation allows severed identification of parameters for each subsystem—to overcome the problem of badly scaled least-squares estimations [7].

The complete 5-DoF identification procedure requires the following steps:

1. Identification of center of rotation for the 3-DoF shoulder joint and 2-DoF elbow joint, respectively;
2. Solving inverse kinematics, which gives an approximation of the Jacobian matrix;
3. Estimating the inertial parameters via kinematics data and base force / torque sensing;
4. Separated identification of the impedance parameters while EMG signals of the active muscles are recorded;
5. Train a model in order to predict stiffness from EMG data which finally gives the possibility to estimate stiffness without mechanical measurements.

## 2.1 Kinematic Identification

To identify the positions of the human limb joints, a method proposed by [1] is conducted. We assume that at least two axes of rotation intersect. The absolute position  $\mathbf{r}$  of the point of intersection can be represented via markers placed at the joint adjacent body segments, i.e.

$$\mathbf{r}_1 = \mathbf{p}_1 + \mathbf{R}_1^T \mathbf{d}_1, \quad (7)$$

$$\mathbf{r}_2 = \mathbf{p}_2 + \mathbf{R}_2^T \mathbf{d}_2, \quad (8)$$

where  $\mathbf{p}_i$  and  $\mathbf{R}_i$  denotes the absolute position and orientation of a marker and  $\mathbf{d}_i$  is the distance of the intersection point w.r.t. the  $i$ th marker. Thus minimizing the integral error

$$S = \frac{1}{T} \int_0^T (\mathbf{r}_1 - \mathbf{r}_2)^T (\mathbf{r}_1 - \mathbf{r}_2) dt, \quad (9)$$

determines the unknown  $\mathbf{d}_1$  and  $\mathbf{d}_2$ .

## 2.2 Inverse Kinematics

The kinematics of the human limb consists of uncertainties, e.g., non-ideal joints and varying segment lengths. In order to minimize these errors we propose a numerical solution of the inverse kinematics, i.e.

$$\arg \min \| \mathbf{T}(\mathbf{q}) \mathbf{T}_d^{-1} - \mathbf{I} \|_F, \quad (10)$$

where  $\mathbf{T}(\mathbf{q})$  and  $\mathbf{T}_d$  are the parameterized and desired homogeneous transformation matrix to the wrist, respectively and  $\|\cdot\|_F$  denotes the Frobenius matrix norm. This optimization problem is continuous and unconstrained and can be solved with, e.g., a quasi-Newton method.

## 2.3 Inertial Parameter Model

For the identification of the inertial parameters, a model can be considered where the dynamical forces / torques are projected to a coordinate system at the subject's mounting base, i.e., under the seat (a similar approach was proposed by [13]). The equations of this model can be obtained analytically by means of the iterative Newton-Euler formalism:

$$\chi_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{n}_i \end{bmatrix} = \begin{bmatrix} \mathbf{F}_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \xi) + \mathbf{R}_{i,i+1}(\mathbf{q}) \mathbf{f}_{i+1} \\ \mathbf{N}_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \xi) + \mathbf{R}_{i,i+1}(\mathbf{q}) \mathbf{n}_{i+1} + \tilde{\mathbf{p}}_{i,i+1}(\mathbf{q}) (\mathbf{R}_{i,i+1}(\mathbf{q}) \mathbf{f}_{i+1}) \end{bmatrix}, \quad (11)$$

for  $i = n_{\text{body}}, n_{\text{body}} - 1, \dots, 0$ ,

where  $\chi_i$  is the wrench acting on the  $i$ th body.  $\mathbf{R}_{i,i+1}$  and  $\mathbf{p}_{i,i+1}$  are the relative rotation and distance between body  $i$  and  $i + 1$ , respectively. Furthermore,  $\mathbf{f}_{i+1}$  and  $\mathbf{n}_{i+1}$  are

Cartesian forces and torques propagated from the  $(i + 1)$ st body. Finally,  $\mathbf{F}_i$  and  $\mathbf{N}_i$  are the forces and torques due to the inertial dynamics. They can be computed by:

$$\mathbf{F}_i = m_i \dot{\mathbf{v}}_i + \tilde{\omega}_i \mathbf{S}_i + \tilde{\omega}_i (\tilde{\omega}_i \mathbf{S}_i) , \quad (12)$$

$$\mathbf{N}_i = \boldsymbol{\Theta}_i \dot{\omega}_i + \tilde{\omega}_i (\boldsymbol{\Theta}_i \omega_i) + \tilde{\mathbf{S}}_i \dot{\mathbf{v}}_i . \quad (13)$$

Herein  $\tilde{\omega} \in \mathbb{R}^{3 \times 3}$  denotes the skew symmetric tensor composed of components  $\omega \in \mathbb{R}^3$ .  $\mathbf{v}_i$ ,  $\omega_i$  and  $\dot{\mathbf{v}}_i$ ,  $\dot{\omega}_i$  are absolute translational and angular velocities and accelerations of the  $i$ th body. The inertial parameters  $m_i \in \mathbb{R}$ ,  $\mathbf{S}_i \in \mathbb{R}^3$ , and  $\boldsymbol{\Theta}_i \in \mathbb{R}^{3 \times 3}$  (i.e., mass moments zeroth, first, and second order) are linear in the base base projected model  $\chi_0(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \boldsymbol{\xi}) = \chi_{\text{sensor}}$  and can be identified by common least-squares estimations.

## 2.4 Impedance Identification

Due to the constraint of energy conservation the force field generated by mechanical stiffness must be integrable [6], i.e., stiffness matrices are symmetric and positive definite (SPD). In order to enhance the robustness of the identification procedure we also determine the stiffness separated from damping. Therefore we take the reduced model

$$\mathbf{K}_q \tilde{\mathbf{q}} = \boldsymbol{\tau}_{\text{ext}} - \boldsymbol{\Gamma}(\mathbf{q}, \boldsymbol{\xi}_{\text{red}}) - \boldsymbol{\tau}_d , \quad (14)$$

into account. This model is valid for the stationary case  $\tilde{\mathbf{q}} = \dot{\mathbf{q}} \approx 0$ . In that case the stiffness balance the (non-linear) gravity and external torques.

The identification model is linear in  $\mathbf{K}_q$  and consists of the form

$$\mathbf{A} \mathbf{X} = \mathbf{B} , \quad (15)$$

where  $\mathbf{A} = \tilde{\mathbf{q}}^T$ ,  $\mathbf{X} = \mathbf{K}_q^T$  and  $\mathbf{B} = (\boldsymbol{\tau}_{\text{ext}} - \boldsymbol{\Gamma}(\mathbf{q}, \boldsymbol{\xi}_{\text{red}}) - \boldsymbol{\tau}_d)^T$ . To ensure the SPD constraint the area criterion proposed by [3]:

$$f(\mathbf{Y}) = \|\mathbf{A} \mathbf{Y} - \mathbf{B} \mathbf{Y}^{-T}\| , \quad (16)$$

where  $\mathbf{X} = \mathbf{Y} \mathbf{Y}^T$ , will be minimized. When  $\mathbf{P} = \mathbf{A}^T \mathbf{A}$  and  $\mathbf{Q} = \mathbf{B}^T \mathbf{B}$  the unique solution is given by

$$\hat{\mathbf{K}}_q = \hat{\mathbf{X}} = \mathbf{U}_P \boldsymbol{\Sigma}_P^{-1} \mathbf{U}_{\tilde{\mathbf{Q}}} \boldsymbol{\Sigma}_{\tilde{\mathbf{Q}}} \mathbf{U}_{\tilde{\mathbf{Q}}}^T \mathbf{U}_P^{-1} \mathbf{U}_P , \quad (17)$$

where

$$\mathbf{P} = \mathbf{U}_P \boldsymbol{\Sigma}_P^2 \mathbf{U}_P^T , \quad (18)$$

$$\tilde{\mathbf{Q}} = \boldsymbol{\Sigma}_P \mathbf{U}_P^T \mathbf{Q} \mathbf{U}_P \boldsymbol{\Sigma}_P = \mathbf{U}_{\tilde{\mathbf{Q}}} \boldsymbol{\Sigma}_{\tilde{\mathbf{Q}}}^2 \mathbf{U}_{\tilde{\mathbf{Q}}}^T , \quad (19)$$

are the Schur decompositions of  $\mathbf{P}$  and  $\tilde{\mathbf{Q}}$ , respectively.



## 2.5 Stiffness Determination from EMG

To predict a stiffness matrix  $\mathbf{Z}_i$  from EMG data  $\mathbf{x}_i$ , a nonlinear two-layer model was used. As  $\mathbf{Z}_i$  is symmetric and positive definite, we note that it can be decomposed into  $\mathbf{Z}_i = \mathbf{L}_i \mathbf{L}_i^T$  via the Cholesky decomposition.

Constraining the output of our model to be positive definite and symmetric can thus be done by not modeling  $\mathbf{Z}_i$ , but  $\mathbf{L}_i$  instead.

In more detail, given  $N$  time windows  $\{\mathbf{x}_i\} \subset \mathbb{R}^{n \times m}$  where  $n$  is the length of the time windows and  $m$  is the number of EMG electrodes, we predict the components of the Cholesky decomposition via

$$l_i = \frac{1}{N} \mathbf{W}_1 \sigma(\mathbf{W}_2 \phi(\mathbf{x}_i) + \mathbf{b}_2) + \mathbf{b}_1, \quad (20)$$

where  $\phi$  is a function that extracts features from each time window and  $\sigma$  is a nonlinear function applied component-wise. We then turn  $l_i$  into a lower-triangular matrix  $\mathbf{L}_i$  by rearranging the components from vector into matrix form. The final prediction is subsequently formed by  $\mathbf{Y}_i = \mathbf{L}_i \mathbf{L}_i^T$ .

The parameters of the model  $\theta = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_2, \mathbf{b}_1\}$  are either matrices  $\mathbf{W}_1$  and  $\mathbf{W}_2$  or vectors  $\mathbf{b}_1$  and  $\mathbf{b}_2$ . To learn such model, we assume that the measurements of the stiffness matrices  $\{\mathbf{Z}_i\}$  are subject to Gaussian noise and minimize the negative log likelihood:

$$\log \lambda \cong \sum_i \|\mathbf{Y}_i - \mathbf{Z}_i\|^2. \quad (21)$$

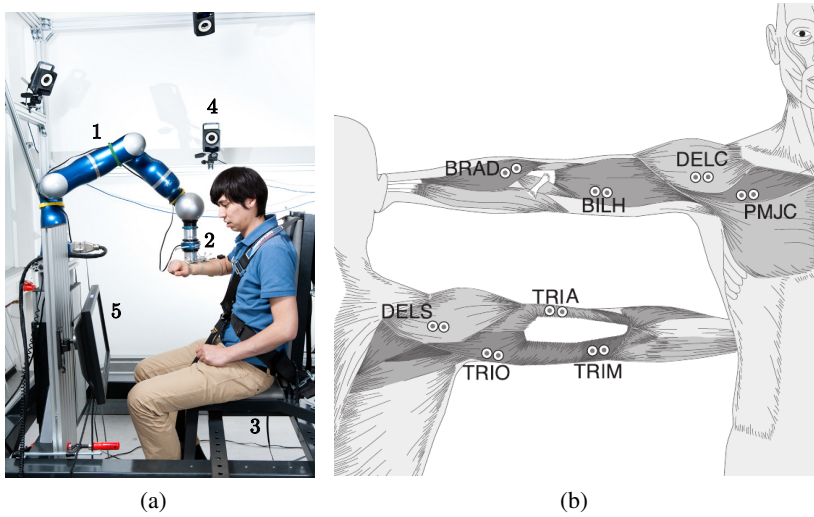
The resulting optimization problem is unconstrained and continuous. The gradients are efficiently computed via dynamic programming and the chain rule. Thus, standard off-the-shelf optimizers are used to find good solutions for  $\theta$ .

## 3 Experiments

During the whole experiment subjects were seated on a special chair depicted in Figure 1(a) while the upper body was restrained by a seat belt. At the wrist a plastic cuff supported the connection to the robot's end-effector. JR3 force / torque sensors were placed at the interconnection (between robot and limb) and at the subject's mounting base (under the seat). The data of both force / torque sensors were sampled at 2 kHz. To estimate the kinematic configuration optical tracking markers were placed at the upper body, upper arm and forearm, respectively. We used data from Vicon T10 cameras to track the markers position and orientation, sampled at 500 Hz. To map EMG to stiffness, we recorded EMG signals from eight sources on the arm (see Fig. 1(b)). We used Delsys Trigno wireless electrodes, sampled at 2 kHz. Experimental instructions and visual feedback were given to the subject via a display.

At the beginning of an experimental session, data from the optical tracking system was recorded to identify the subject's individual joint positions and compute the arm kinematics. Here the subject was instructed to move all joint axes of the limb randomly. After this, data (40 trials) for the inertial parameters were gathered,

while different predefined kinematic limb configurations had to be reached in free movement. These initial recordings were followed by the final stiffness identification procedure in which subjects had to fulfill a force task. The subject's wrist was coupled to the light-weight robot's end-effector and desired and actual interaction forces / torques were displayed. After holding a certain force / torque level (4 levels in Cartesian X and Z direction each) for a random duration between 1.5 and 2.5 seconds, the robot perturbs the limb in one direction randomly chosen from the 10 possibilities (two for each joint DoF). All smooth (polynomial fifth order) displacements were planned in joint coordinates of the human limb with an amplitude of  $\approx 0.08$  rad via the Jacobian matrix, i.e.,  $\Delta \mathbf{x}_{\text{robot}} = \mathbf{J}_{\text{limb}}(\mathbf{q}) \Delta \mathbf{q}$ . Typical disturbances are shown in Figure 2.



**Fig. 1** **a** Experimental setup: (1) DLR light-weight robot applies disturbances to the human arm, (2) JR3 force / torque sensor measures interaction forces, (3) JR3 force / torque sensor measures subject's mounting base forces, (4) Vicon T10 optical tracking system, (5) Subject's visual feedback. **b** EMG electrode placement for estimating stiffness from EMG. A total number of 8 electrodes are placed. EMG signals of dominant muscles involved in shoulder and elbow joint movements are gathered: brachioradialis (*BRAD*), biceps long (*BILH*), deltoid clavicular (*DELC*), pectoralis major clavicular (*PMJC*), deltoid scapular (*DELS*), triceps long (*TRIO*), triceps lateral (*TRIA*), and triceps medial (*TRIM*).

## 4 Results

### 4.1 Estimated Stiffness and Prediction via EMG

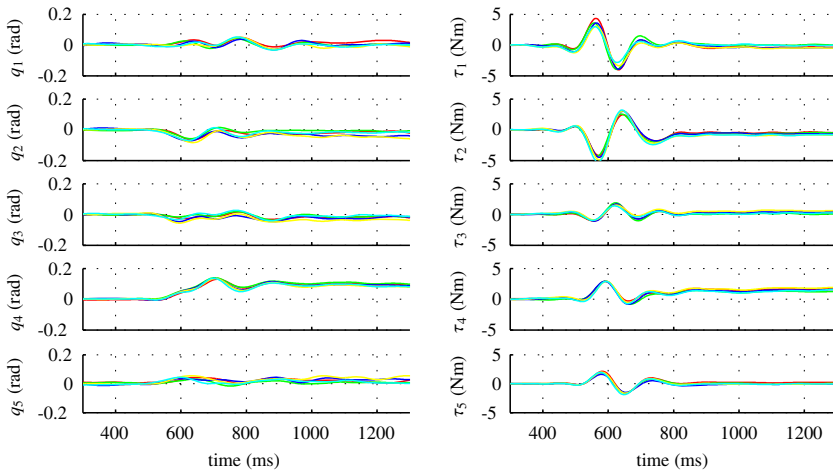
After estimating the joint positions, the inverse kinematics of all trials (i.e., inertial and stiffness measurements) were computed. First identification results are obtained

from stationary parts of the recorded data, i.e., where the velocity is lower than a certain threshold. Therefore the inertial identification model reduces to

$$\mathcal{X}_0(\mathbf{q}, \boldsymbol{\xi}_{\text{red}}) = \mathcal{X}_{\text{sensor}} \quad (22)$$

where  $\boldsymbol{\xi}_{\text{red}}$  contains the mass moments zeroth and first order. To estimate  $\boldsymbol{\xi}_{\text{red}}$ , mean values of the joint angles  $\mathbf{q}$  (where  $\dot{\mathbf{q}} \approx 0$ ) and mean values of associated base wrench components  $\mathcal{X}_{\text{sensor},i}$  where  $i = 3, 4, 5$  were used for least-squares regression.

The stiffness identification procedure was also based on the reduced model where  $\mathbf{q}_d$  and  $\boldsymbol{\tau}_d$  was obtained by taking the mean values in the time interval before the onset of the disturbance. Analogously, a second interval for  $\mathbf{q}$  and  $\boldsymbol{\tau}$  was chosen after the displacement. For descriptive reasons, typical estimated joint stiffness matrices are transformed to Cartesian coordinates and visualised as stiffness ellipsoids in Figure 3. Each stiffness matrix  $K_q$  was determined from 50 disturbance measurements; consequently each map was constructed from 50 sets of  $8 \times 400$  data points. We preprocessed the EMG data with a full wave rectification and split the data into time windows of length 70 afterwards. For  $\phi$  we picked the maximum along each of the signals followed by two layers of unsupervised feature extraction using the approach of [9]. We chose 100 soft rectified linear units as the nonlinearity in our model:  $\sigma(x) = \ln(1 + \exp x)$ . All hyper parameters of the learning process were selected by random search and picking those which performed best on a held out validation set. The average normalized root mean squared error for the multi-layered model was 0.3378. In contrast, a linear model never achieved 0.39 or better. For an example of the predicted stiffness matrices, see Figure 4.

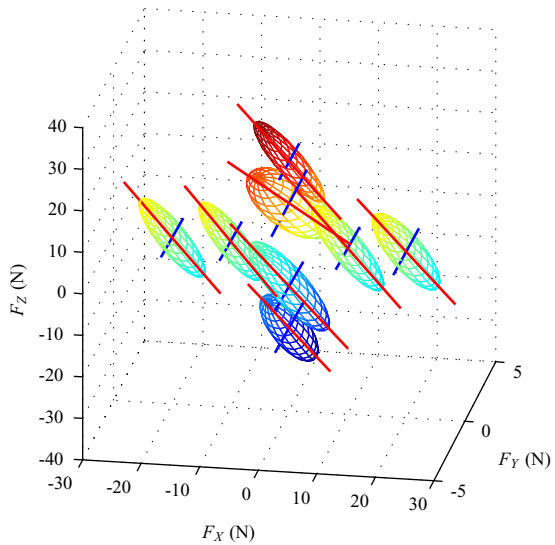


**Fig. 2** Typical disturbances exerted by the light-weight robot, mapped to human arm's joint space. To demonstrate the repeatability the data is aligned along the time axis.

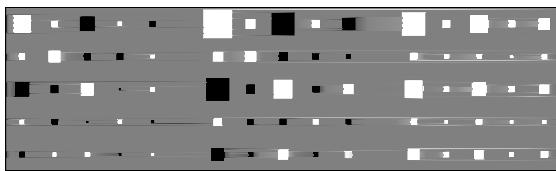
### 4.2 Comments

Estimating joint stiffness in more than two degrees entails a chain of model assumptions, while model uncertainties accumulate from kinematics over inertial to stiffness identification. The identification of inertial parameters depend on measured joint angles  $\mathbf{q}$  (and their derivatives) and are based on force / torque data measured on the mounting base, i.e., the measurement range must include human’s whole body mass. These are affected by the following assumptions:

- For human arm’s kinematics it is assumed that the shoulder joint is an ideal spherical joint and the elbow joint consist of two orthogonal, intersecting axes. Furthermore, it is assumed that optical markers placed on the upper and lower arm do not move relative to the skeleton. Both issues, extensively studied in [1], lead to biased estimates of joint angles  $\mathbf{q}$  and their time derivatives.



**Fig. 3** Cartesian stiffness ellipsoids of estimated joint stiffness matrices. Each ellipsoid represents the force-field generated due to spherical displacements (here, the radius  $r = 5$  mm). The ellipsoid’s origins are shifted to the point of pretension forces  $\mathbf{F} = (F_x, F_y, F_z)^T$ . Additionally, the principle axes (eigenvalues) of stiffness ellipsoids are displayed as straight lines.



**Fig. 4** Hinton diagrams of the stiffness matrices based on EMG prediction (left), estimation from force perturbations (middle) and the absolute value of their difference (right). Black boxes correspond to negative, white to positive values while the size represents the magnitude. Data was taken from the testing set.

- In particular for some small mass moments of inertia, the reaction force at the mounting base undercuts the sensitivity of the force / torque sensor.
- For the stiffness identification we assume that muscle activation  $\mathbf{a}$  are constant. This implies fast perturbations, where displacements are stabilized in a short time (cf. [7]). Otherwise a change in activations would have changed the impedance.

This chain of assumptions forces us to analyze the results depicted in Fig. 3 intensively. For instance, from planar measurements it is known [5] that stiffness ellipses align their major axis in the direction of the pretension force applied. For the present estimations this effect can be observed only in the direction of  $F_Z$ .

## 5 Main Experimental Insights

In this work we have introduced a new and unique method to measure the stiffness of the human arm in 5-DoF joint space, viz. 3 shoulder DoF, the elbow flexion, and the lower arm rotation. Identification of arm kinematics and deriving the Jacobian matrix allows for transferring the measured joints stiffnesses to the Cartesian domain. We thus pioneered the measurement of human arm impedance in more than 2 Cartesian coordinates.

Furthermore, we have proposed and incorporated a multi-layered regression model which maps surface EMG signals to joint stiffness. With this method, combined with a detailed kinematic model, we can accurately estimate arm impedance without the need of mechanical perturbations. This is essential in order to determine human arm impedance not only in static positions but also along a trajectory during task execution, without the need of perturbation measurements.

Given this framework we are now able to investigate how humans modulates arm impedance in any task. The resulting measurements can be used to derive methods of impedance modulation for robotic arms.

## References

1. Biryukova, E., Roby-Brami, A., Frolov, A., Mokhtari, M.: Kinematics of human arm reconstructed from spatial tracking system recordings. *Journal of Biomechanics* 33, 985–995 (2000)
2. Burdet, E., Osu, R., Franklin, D.W., Yoshioka, T., Milner, T.E., Kawato, M.: A method for measuring endpoint stiffness during multi-joint arm movements. *Journal of Biomechanics* 33, 1705–1709 (2000)
3. Chen, Y., McInroy, J.E.: Estimation of symmetric positive-definite matrices from imperfect measurements. *IEEE Transaction on Automatic Control* 47, 1721–1725 (2002)
4. Gomi, H., Kawato, M.: Human arm stiffness and equilibrium-point trajectory during multi-joint movement. *Biological Cybernetics* 76, 163–171 (1997)
5. Gomi, H., Osu, R.: Task-dependent viscoelasticity of human multijoint arm and its spatial characteristics for interaction with environments. *The Journal of Neuroscience* 18, 8965–8978 (1998)
6. Hogan, N.: The mechanics of multi-joint posture and movement control. *Biological Cybernetics* 52, 315–331 (1985)

7. Lakatos, D., Petit, F., van der Smagt, P.: Conditioning vs. excitation time for estimating impedance parameters of the human arm. *IEEE Humanoids* (2011)
8. Mussa-Ivaldi, F.A., Hogan, N., Bizzi, E.: Neural, mechanical, and geometric factors subserving arm posture in humans. *The Journal of Neuroscience* 5, 2732–2743 (1985)
9. Ngiam, J., Koh, P.W., Chen, Z., Bhaskar, S., Ng, A.: Sparse filtering. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems* 24, pp. 1125–1133 (2011)
10. Perreault, E.J., Kirsch, R.F., Acosta, A.M.: Multiple-input, multiple-output system identification for characterization of limb stiffness dynamics. *Biological Cybernetics* 80, 327–337 (1999)
11. Perreault, E.J., Kirsch, R.F., Crago, P.E.: Effects of voluntary force generation on the elastic components of endpoint stiffness. *Experimental Brain Research* 141, 312–323 (2001)
12. Tsuji, T., Morasso, P.G., Goto, K., Ito, K.: Human hand impedance characteristics during maintained posture. *Biological Cybernetics* 72, 475–485 (1995)
13. Venture, G., Yamane, K., Nakamura, Y., Yamamoto, T.: Identification of Human Limb Viscoelasticity using Robotics Methods to Support the Diagnosis of Neuromuscular Diseases. *The International Journal of Robotics Research* 28(10), 1322–1333 (2009)

# Motor vs. Brake: Comparative Studies on Performance and Safety in Hybrid Actuations

Dongjun Shin, Xiyang Yeh, Takashi Narita, and Oussama Khatib

**Abstract.** Human-centered robotics draws growing interest in utilizing pneumatic artificial muscles (PAMs) for robots to cooperate with humans. In order to address the limited control performance which prevents PAMs from being more widely used, a hybrid actuation scheme has been proposed to combine PAMs and a low inertia DC motor, and presented significantly improved control performance without loss of robot safety. While the DC motor provides high precision and reliability, the small motor has, however, difficulties in dealing with the large stored energies of the PAMs, especially in the events of PAMs failure and large initial load changes. In order to further ensure robot safety, we developed a new hybrid actuation scheme with PAMs (macro) and a particle brake (mini), which provides high torque-to-weight ratio and inherent stability. We then conducted comparative studies between hybrid actuations with (1) a DC motor and (2) a brake in terms of robot safety and performance. Experimental comparisons show that the hybrid actuation with PAMs and a brake provides higher energy efficiency for control bandwidths under 2 Hz, and is capable of effectively reducing large impacts due to the brake's high torque capacity and passive energy dissipation. These comparative studies provide insight that the hybrid actuation with PAMs and a brake can be a competitive solution for the applications that require high efficiency, but accept a relatively low control performance, for example, a waist joint.

## 1 Introduction

Human-centered robotics draws growing interest in inherently safe actuations for robots to cooperate with humans. Notable achievements are series elastic actuator [1], variable impedance actuator [2], and distributed macro-mini actuation [3].

---

Dongjun Shin · Xiyang Yeh · Takashi Narita · Oussama Khatib  
Artificial Intelligence Laboratory,  
Stanford University, Stanford, CA 94305, USA  
e-mail: djshin@robotics.stanford.edu

In addition, considerable research has employed pneumatic artificial muscles (PAMs) for their high force-to-weight ratios and inherent compliances [4–6]. Low output impedance of PAMs over a wide frequency range enables PAMs to reduce large impact forces during unforeseen collisions.

However, PAM's limited control performance prevents it from being more widely used. Due to their air compressibility and viscous/coulomb friction in a braided shell, force and position control bandwidths are limited.

**Hybrid Actuation:** In order to address the performance limitations, Shin et al. proposed a hybrid actuation approach, which consists of a pair of PAMs coupled in parallel to a low-inertia DC motor [7]. The low-pass behavior of PAMs naturally partitions the reference input torques into low frequency (macro) and high frequency (mini) actuation components. The macro torque component is primarily sustained by the muscles while the resultant torque error is compensated by the DC motor. This configuration improves the overall bandwidth as the fast DC motor compensates for slow dynamics of the muscles.

Other works have employed a magnetic brake along with PAMs. Ahn et al. used a magneto rheological brake (MR-brake) with phase switching control in order to improve the control performance of PAMs [8]. Senkal and Gurocak developed a haptic joystick using PAMs and a spherical MR-brake [9]. Nagai and Nakamura implemented the position and vibration control of PAMs by actively adjusting viscosity coefficients using a MR-brake [10].

**Problem Statement:** The hybrid actuation schemes significantly reduces possible dangers by employing the novel design for low effective inertia and utilizing inherent damping and limited flow rate of PAMs.

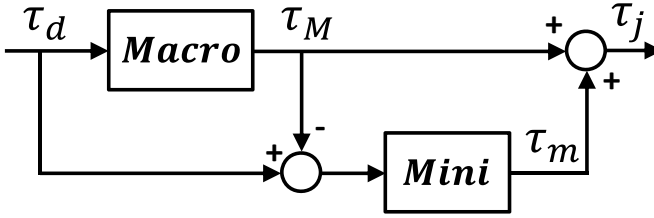
Brakes provide higher torque-to-weight ratios than DC motors and stable dissipative operations, which are robust to non-ideal velocity feedback due to noise and phase lag issues [11]. These characteristics positively contribute to robot safety. Their slightly non-linear characteristics, however, result in some difficulties in control, and their passive natures allow for torque compensation only in the opposite direction of motion. Furthermore, all of previous works have been focused on performance improvements, but effects on robot safety without compromising control performance remains to be addressed.

On the other hand, while the hybrid actuation with PAMs and a DC motor remarkably improves control performance while minimizing loss of robot safety, the limited torque capacity of the mini DC motors may result in difficulties in responding to changes in large inertial load at high speeds (e.g., accidental muscle puncture).

In order to understand characteristics of each hybrid actuation and optimally select a mini actuator for a desired application, we investigate and compare robot safety and performance of hybrid actuations with (1) a DC motor or with (2) a brake.

Section 2 provides a technical approach which includes control strategies and hardware descriptions of both hybrid actuations. Section 3 and 4 describe the experimental setup and show results, respectively. Finally, Section V concludes the paper with discussion.





**Fig. 1** Hybrid actuation scheme [7]. PAM is employed as the macro actuation while mini actuation can be a DC motor or a brake. Note that the brake compensates for the torque error in the opposite direction of motion.

## 2 Technical Approach

In order to achieve low effective inertia but high torque capacity, we employed PAMs as a low frequency actuation (macro actuation). Since torque and position control bandwidth is significantly limited by their slow dynamics, a small high frequency actuation (mini actuation) is essential to increase a dynamic range with maintaining/improving robot safety. In this paper, we employed a brake or a motor as a mini actuation to investigate advantages and disadvantages of each actuation.

### 2.1 Control Strategy

The basic control strategy in this paper is the hybrid actuation concept proposed in [12]. The motor or brake is coupled in parallel to the robot joint, which is driven by PAMs. An adaptive force feedback controller was implemented for PAMs as follows:

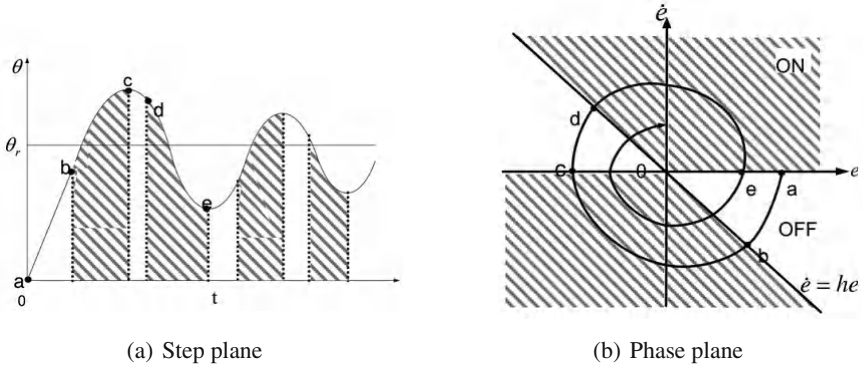
$$C(s) = \frac{22.5}{K} \frac{s+6}{s+300} \frac{s+25}{s+0.01} \tag{1}$$

where,  $K$  is an adaptive controller gain, which is a function of muscle length.

Using the hybrid actuation concept, the motor or brake torque command is obtained from joint torque error by PAMs as shown in Fig. 1. In case of a motor, mini actuation is a simple open-loop control since it is a bidirectional actuator and output torque is quite linear to input current. Unlike a motor, the brake can only produce unilateral opposing torques; only the overshoot torques can be reduced. In our control framework (Fig. 1), when the brake is used in place of the DC motor, we control the brake using a phase switching controller [8] as follows:

$$\tau_m = K_{ed} |\dot{\theta}| \varepsilon \tag{2}$$

where,  $K_{ed}$  is the damping gain for the joint velocity  $\dot{\theta}$  and  $\varepsilon = 0$  or  $1$  for the off and on state for the brake, respectively. The phase plane switching surfaces were



**Fig. 2** Phase plane switching control strategy [8]. Brake activation can be fine-tuned using the parameter  $h$ . During the active state ( $b \sim c$ ,  $d \sim e$ ), the brake produces opposing torque as outlined in Equation (2) to compensate overshoots and undershoots. Proper selection of  $h$  improves control performance by selectively deactivating the brake when the joint accelerates to the set point, for example in  $a \sim b$ .

constructed similarly as [8]. The control strategy allows the user to fine-tune when the brake needs to activate such that fast dynamic response can be achieved despite the brake's dissipative nature. Fig. 2 illustrates how each switching surface parameter affects the dynamics of the system. The parameters were empirically found as  $h = -50$  and  $K_{ed} = 0.5$  with the error defined as  $e = q_d - q$ .

Note that a linear model relating torque and current is used for the control of brakes. Interestingly, the brake does not transfer torque viscously in shear mode [13]. This means the brake torque is practically determined by a magnetic field, i.e., an input current, rather than an input velocity unless the velocity is significantly low (0.04 rad/s). Since the brake torque is proportional to the input current alone, the brake is easily controllable within its torque range.

The high torque density of a brake enables it to hold a joint with high stiffness, reduce overshoots, and maintain stability under large inertial changes while keeping low effective inertia. Due to its intrinsic passivity, a brake does not incorrectly inject energy into a system even in case of non-ideal velocity feedback because of noise and phase lag. These characteristics further ensure robot safety under various circumstances.

## 2.2 Hardware Design

In order to effectively investigate and compare a motor and brake, we incorporated a brake in the existing hybrid actuation testbed, which combines PAMs and a motor. All hardware descriptions of the existing testbed are included in [12].

**Table 1** Actuator Comparison

Actuator	Advantage	Disadvantage
PAM	High force density Compliance	Difficult to control
DC motor	Accurate Easy to control	Low torque density
Particle brake	High torque density Inherently stable	Passive actuator Static friction

A magnetic particle brake is employed in the hybrid actuation in the form of a physically dissipative damper to stably reduce large impact forces. When current is applied to the coil of the brake, magnetic flux binds magnetic particles in the powder cavity. As current increases, the bond between the particles becomes stronger, which results in higher resistance/braking torque to the rotor.

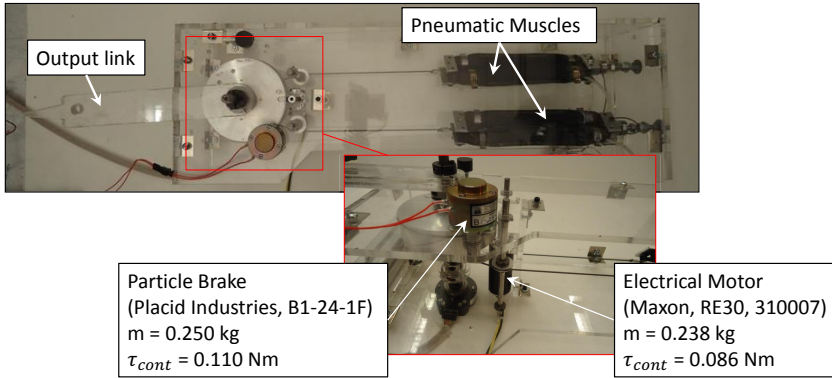
Although MR-brakes have higher torque-to-weight ratio than particle brakes [9], the available MR-brakes in the market are too large and heavy for human-friendly robot applications. Electro rheological (ER) brakes, on the other hand, need extremely high control voltage which is potentially unsafe [8], and hysteresis brakes have very low torque-to-weight ratio [14]. Table 1 shows a comparison of the characteristics of an electric motor, a PAM, and a magnetic particle brake.

One drawback of the particle brake is its static torque, which is typically determined by static friction and rotor inertia. Although this torque is compensated by a controller, the smallest brake that meets torque requirement should be chosen.

### 3 Experimental Setup

In order to evaluate and compare the performance of various hybrid actuation schemes, we set up an experimental testbed as shown in Fig. 3. The testbed employs three actuators: PAMs for macro actuation, and a DC motor or a particle brake for mini actuation. A pair of antagonistically actuated PAMs (Shadow Robot, 20 mm diameter) is connected to the joint with 15 mm-radius circular pulleys. Additionally, we had a particle brake (Placid Industries, B1-24-1F) and a DC motor (Maxon Motor, RE30, 310007) connected in parallel with PAMs. For fair comparison of performance and safety, we closely matched the weight of the brake with that of the DC motor, although a larger brake has substantially higher torque density. For the best mini actuation performance, we also employed a highly stiff cable-driven pulley transmission with the reduction ratio of 10:1. All three actuators are connected to the joint in parallel. Table 2 shows the technical specifications of mini actuations in the testbed.

For the experiments of impact reduction, we used another actuator to exert impulsive forces on the robotic link of the testbed.



**Fig. 3** Testbed of hybrid actuation combining PAMs and a DC motor or a particle brake, all of which are connected to the joint in parallel. For fair comparison of performance and safety, we closely matched the weight of the brake with that of the DC motor.

**Table 2** Technical specifications of the motor and brake

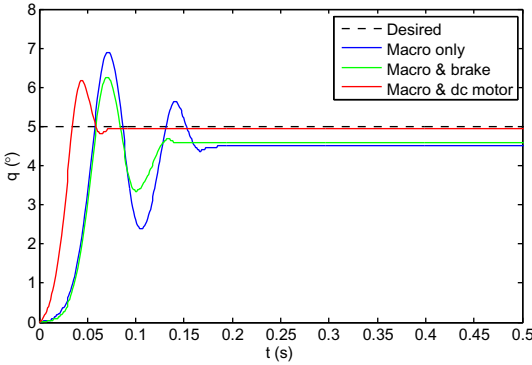
Actuator	Motor	Brake
Manufacturer	Maxon motors	Placid Industries
Product Model	RE30 - 310007	B1-24-1F
Max Continuous Torque	86.2 mNm	110.0 mNm
Torque Constant	25.9 mNmA <sup>-1</sup>	1466.7 mNmA <sup>-1</sup>
Mechanical Time constant	3 ms	4-8 ms
Weight	0.238 kg	0.250 kg

## 4 Experimental Results

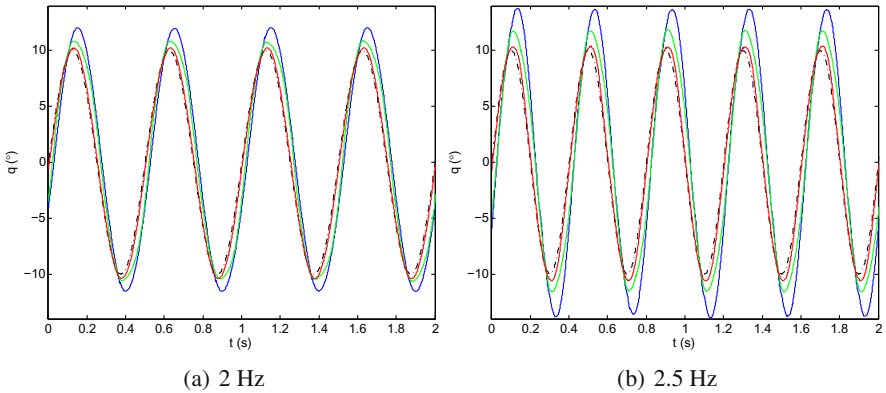
To investigate tracking performance and safety of two hybrid actuation schemes, a series of experiments was conducted.

### 4.1 Performance Analysis

Fig. 4 and 5 show the 5° step response and 10° sinusoidal position tracking of each actuation scheme, respectively. Hybrid actuation with PAMs and a DC motor achieves the best performance. Since the DC motor is able to produce bidirectional active torques at high frequencies, it improves performance in the transient and steady states. Meanwhile, the addition of the particle brake manages to dampen overshoots and to improve settling time.



**Fig. 4** Step response comparison. Macro only actuation results in the largest overshoots. The addition of the particle brake dampens the overshoots and reduces settling time. Macro and DC motor actuation achieves the best performance due to the high frequency torque contribution from the DC motor.



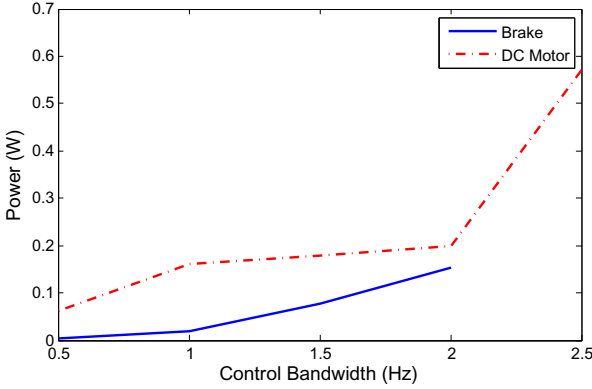
**Fig. 5** Sinusoidal position tracking for  $10^\circ$ . The hybrid actuation with PAMs and a brake achieves a position control bandwidth of 2 Hz, while the DC motor mini actuation increases the bandwidth to 6 Hz [12].

Fig. 6 shows the energy consumption comparison with respect to position control bandwidth. At low control bandwidths, the hybrid actuation with PAMs and a brake provides significantly more efficient operation, although the maximum achievable control bandwidth is limited to 2 Hz. A larger brake has higher energy efficiency, and thus can further improve the efficiency of the system.

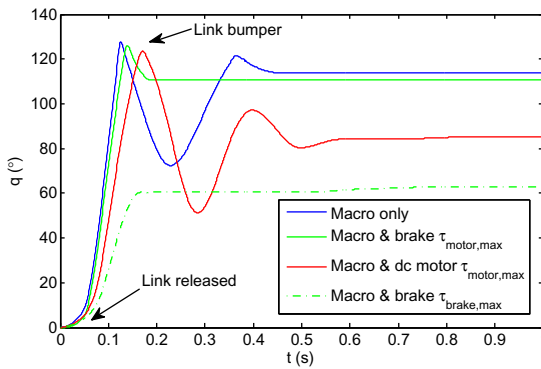
### 4.2 Safety Analysis

To investigate the safety of the joint under various schemes, we simulated a mechanical fault by disconnecting one of the muscles and releasing the joint while keeping the other muscle preloaded. The preload was achieved by controlling the pressure of the muscle at 207 kPa.

To reduce the velocity and impact force of the swinging link, we commanded the brake or the DC motor to apply an opposing torque. Fig. 7 shows the capability of



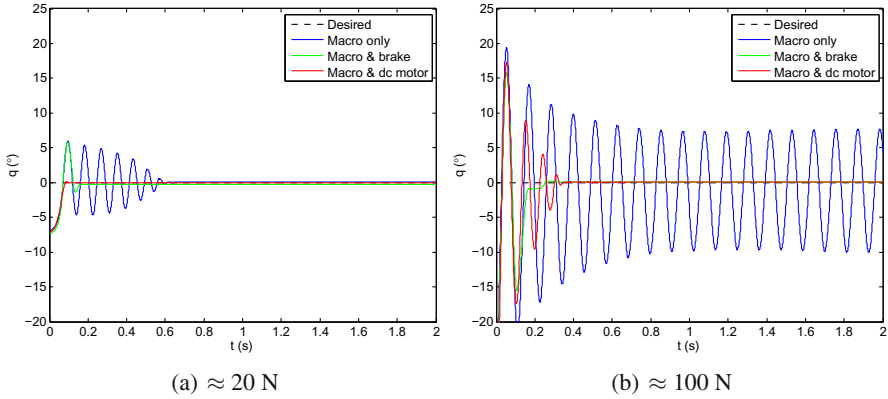
**Fig. 6** Energy consumption comparison with respect to position control bandwidth. At low control bandwidths, the hybrid actuation with PAMs and a brake provides significantly more efficient operation, although the maximum achievable control bandwidth is limited to 2 Hz.



**Fig. 7** Response comparison to instantaneous load changes. At 0.5 Nm opposing torque which is the output torque due to the maximum DC motor torque after the transmission, neither the particle brake nor the DC motor is able to significantly slow down the moving link. With full brake torque is exerted, the brake is able to bring the motion of the link to a halt at 0.05 sec. Note that the change in direction of motion is a result of collision with the link bumper.

the DC motor and the particle brake to dampen the motion of the swinging link. With the maximum compensation torque of 0.5 Nm, which is the output torque capacity corresponding to the maximum DC motor torque after the transmission, neither the brake nor the DC motor is able to reduce the link velocity significantly. Compared to the brake, the DC motor, with its faster dynamics, is able to more effectively dampen the link’s motion. However, when the brake exerted its full torque, the link is brought to a halt at approximately 0.05 sec later.

In addition, we conducted the experiments of impact reduction. Fig. 8 shows that the hybrid actuation with a DC motor achieves better rejection performance for a small amplitude impact, while the brake mini actuation more effectively reduces a large amplitude impact. Since brakes are able to passively dissipate energies, the new hybrid actuation is able to achieve high damping capability. Meanwhile, the damping property of the hybrid actuation with a DC motor is limited by noise, phase



**Fig. 8** Comparison of impact reduction. Both hybrid actuation schemes achieves almost the same rejection performance for a small amplitude impact, while the brake mini actuation more effectively reduces a large amplitude impact. The new hybrid actuation is able to achieve high damping due to the brake’s capability to passively dissipate energy.

lag, and interference between muscle and motor. This limitation may result in dangerous motions in the case of large impacts, which the motor cannot effectively cope with.

In summary, experimental results show that the hybrid actuation with a DC motor generally provides better tracking performance than with a brake. In addition, a DC motor provides a faster dynamic response to instantaneous (small) changes in load or unexpected impacts. On the other hand, while maximum achievable control bandwidth is limited, the new hybrid actuation with a brake consumes less energy given the same control performance. Furthermore, a brake can effectively respond to a large change in an inertial load and even stably reduce a large amplitude impact due to its high torque capacity and passive dissipative nature. This prevents the robot from generating unfavorable and unsafe motion in the case of unexpected large impacts.

## 5 Conclusion

In order to further ensure robot safety, we have developed and implemented a new hybrid actuation with PAMs and a magnetic particle brake within a macro-mini actuation concept. The high torque density of a brake maintains low effective inertia, while significantly damping large accidental torque due to muscle failure. Furthermore, the brake’s physically dissipative damping enables the new actuation to stably reduce large amplitude impacts, which may result in unfavorable motions and dangers. These characteristics enable the brake mini actuation not only to effectively reduce large impacts, but also to improve dynamic control performance over the PAMs-alone actuation.

In order to understand characteristics of each hybrid actuation and optimally select a mini actuator for a desired application, we have investigated and compared robot safety and performance of hybrid actuations with (1) a DC motor or with (2) a brake. Experimental results show that the hybrid actuation with PAMs and a DC motor achieves better position tracking performance and faster dynamic response to instantaneous load changes or unexpected small impacts ( $\approx 20$  N) than PAMs and a brake. Meanwhile, the hybrid actuation with PAMs and a brake consumes less energy for the tracking bandwidths less than 2 Hz, which is the maximum achievable control bandwidth of the current system with a brake. Furthermore, the new actuation can both effectively respond to changes in a large inertial load due to brake's high torque capacity and robustly reject large impacts ( $\approx 100$  N) due to brake's passive energy dissipation. These properties help to prevent the robot from generating unfavorable or unsafe motions. For the applications that requires high efficiency, but accepts a relatively low control performance, for example, a waist joint, the hybrid actuation with PAMs and a brake can be a competitive solution.

Similarly to the hybrid actuation discussed in [15, 16], brake sizing is essential to determine robot performance and safety. Smaller brake size will improve control performance due to its lower inertia and faster response time, but its smaller braking torque significantly limits the capability for large disturbances and accident impacts. In addition, mini actuator sizing is substantially dependent on joint stiffness and muscle pressure [16]. Therefore, PAMs with an optimal brake should be considered. The analysis of experimental results suggests that the combination of a DC motor and a brake as mini actuations may provide meaningful results. However, the development of this actuation should be accompanied by not only optimal sizing of a DC motor and a brake, but also appropriate actuator torque participation, minimizing the interference between two actuators.

**Acknowledgements.** We gratefully acknowledge the generous advice and comments of members of the Stanford Artificial Intelligence Laboratory and the Biomimetic Dexterous Manipulation Laboratory.

## References

1. Pratt, G., Williamson, M.: Series elastic actuators. In: Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 399–406 (1995)
2. Bicchi, A., Tonietti, G.: Fast and soft arm tactics: Dealing with the safety-performance trade-off in robot arms design and control. *IEEE Robotics and Automation Magazine* 11, 22–33 (2004)
3. Zinn, M., Roth, B., Khatib, O., Salisbury, J.K.: New actuation approach for human-friendly robot design. *International Journal of Robotics Research* 23(1), 379–398 (2004)
4. Colbrunn, R.W., Nelson, G.M., Quinn, R.D.: Design and control of a robotic leg with braided pneumatic actuators. In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 992–998 (2001)
5. Vanderborght, B., Verrelst, B., Van Ham, R., Lefeber, D.: Controlling a bipedal walking robot actuated by pleated pneumatic artificial muscles. *Robotica* 24(4), 401–410 (2006)



6. Yeh, T.J., Wu, M.-J., Lu, T.-J., Wu, F.-K., Huang, C.-R.: Control of McKibben pneumatic muscles for a power-assist, lower-limb orthosis. *Mechatronics* 20(6), 686–697 (2010)
7. Shin, D., Sardellitti, I., Khatib, O.: A hybrid actuation approach for human-friendly robot design. In: *Proc. of the 2008 IEEE International Conference on Robotics and Automation*, pp. 1747–1752 (2008)
8. Tu, D.C.T., Ahn, K.K.: Intelligent phase plane switching control of pneumatic artificial muscle manipulators with magneto-rheological brake. *Mechatronics* 16, 85–95 (2006)
9. Senkal, D., Gurocak, H.: Haptic joystick with hybrid actuator using air muscles and spherical MR-brake. *Mechatronics* 21, 951–960 (2011)
10. Nagai, S., Tomori, H., Midorikawa, Y., Nakamura, T.: The position and vibration control of the artificial muscle manipulator by variable viscosity coefficient using MR brake. In: *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 307–312 (2011)
11. Radulescu, A., Howard, M., Braun, D.J., Vijayakumar, S.: Exploiting variable physical damping in rapid movement tasks. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 141–148 (2012)
12. Shin, D., Sardellitti, I., Park, Y.-L., Khatib, O., Cutkosky, M.: Design and control of a bio-inspired human-friendly robot. *The International Journal of Robotics Research* 29(5), 571–584 (2010)
13. Walker, D.S., Thoma, D.J., Niemeyer, G.: Variable impedance magnetorheological clutch actuator and tele robotic implementation. In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 2885–2891 (2009)
14. Genduso, F., Miceli, R., Rando, C., Galluzzo, G.R.: Back emf sensorless-control algorithm for high-dynamic performance pmsm. *IEEE Transaction on Industrial Electronics* 57(6), 2092–2100 (2010)
15. Shin, D., Khatib, O., Cutkosky, M.: Design methodologies of a hybrid actuation approach for a human-friendly robot. In: *Proc. of the 2009 IEEE International Conference on Robotics and Automation*, pp. 4369–4374 (2009)
16. Shin, D., Seitz, F., Khatib, O., Cutkosky, M.: Analysis of torque capacities in hybrid actuation for human-friendly robot design. In: *Proc. of the 2010 IEEE International Conference on Robotics and Automation*, pp. 799–804 (2010)

# Examining the Effect of Rear Leg Specialization on Dynamic Climbing with SCARAB: A Dynamic Quadrupedal Robot for Locomotion on Vertical and Horizontal Surfaces

Bruce Miller, Camilo Ordonez, and Jonathan E. Clark

**Abstract.** Recent investigations into biological locomotion have resulted in the development of reduced order templates that emphasize the role of lateral dynamics in achieving rapid and robust fore-aft movement, such as the Full-Goldman model for dynamic climbing and the Lateral Leg Spring model for horizontal plane running. The observation of individual animals demonstrating locomotion via both of these models motivates the development of a single platform that can do so as well. However, a drawback in developing a robot directly from these models stems from both having a bipedal configuration. While a bipedal robot could be designed, the restriction of control approaches, reduction in stability, and preclusion of leg differentiation motivates the development of a platform with additional limbs. In this study, we describe the development of the first quadrupedal platform capable of instantiating the Full-Goldman model, as well as the Lateral Leg Spring model. In particular, the climbing behavior is characterized and the effect of rear leg posture is examined for locomotion on a vertical surface. We demonstrate that climbing behavior can be impacted by the configuration of the rear legs and that minimizing the magnitude of rear leg sprawl may improve efficiency, while rear sprawl postures with a larger magnitude may improve robustness.

## 1 Introduction

Animals have shown the capacity to rapidly and nimbly navigate unstructured environments and utilize diverse forms of locomotion to traverse different regimes [8, 13, 17, 21]. To better understand the characteristics that lead to the fast, robust movement exhibited by biological creatures, ‘templates’ have been developed to examine the dynamics that enable high-performance locomotion [12]. Templates are reduced-order dynamical models that simplify complex biological locomotory

---

Bruce Miller · Camilo Ordonez · Jonathan E. Clark  
FAMU & FSU College of Engineering, Tallahassee, FL, USA  
e-mail: [jeclark@fsu.edu](mailto:jeclark@fsu.edu)

dynamics into a more tractable problem, facilitating the understanding of how animals move with the speed and dexterity observed in nature. Several robots have used templates as a locomotive basis to produce high speed, robust locomotion, demonstrating how the effective application of principles gleaned from biology can lead to improved performance in mechanical systems [9, 15, 19, 24].

Recent studies of biological locomotion have highlighted the important role of lateral dynamics in achieving robust, high-speed locomotion [4, 13]. This has led to the development of reduced-order models for scansorial and terrestrial locomotion that incorporate lateral force generation via a sprawled leg posture. Two such models are the Full-Goldman (FG) model for dynamic climbing [13] and the Lateral Leg Spring (LLS) model for horizontal plane running [21]. Though these models operate in different regimes, there are several parallels in the system dynamics and configurations, suggesting a natural compatibility for their incorporation in a single platform capable of utilizing both locomotion modalities. Furthermore, studies have shown that as individual animals move from horizontal to vertical surfaces, the forces generated switch from pushing, characteristic of LLS running, to pulling, emblematic of FG climbing [20]. Although robotic platforms have been developed for the individual models [9, 23], no robot has yet instantiated two dynamic locomotion models on the same platform.

While these models and platforms provide insight into the dynamics of biological locomotion, the simplification to bipedal configurations reduces their scope. Having only two limbs restricts control approaches that can be considered, limits navigable environments, complicates the maintenance of balance and stability, and precludes the utilization of leg differentiation in steady state locomotion. This last point is accentuated by studies of animals and other multi-legged robots that have demonstrated that improved locomotion performance can be realized via the utilization of fore-aft leg specialization [10, 11].

In this study, we extend the bipedal FG climbing model to a quadrupedal configuration and describe the design and fabrication of SCARAB (Scansorial and Cur-sorial Ambulation with a Robust, Adaptive roBot), the first quadrupedal platform capable of dynamic climbing on a planar surface. The robot is also designed to run on level and sloped surfaces in a manner described by the LLS running model and to transition between scansorial and terrestrial regimes as well. This platform enables and motivates the investigation as to how the configuration of the rear legs affects dynamic climbing and how it can be optimized to improve climbing performance.

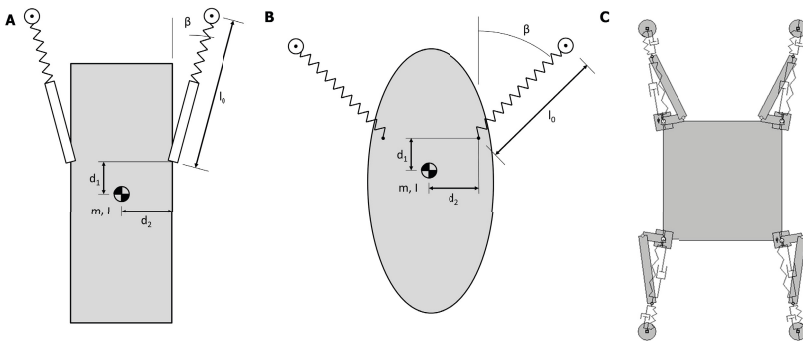
The remaining sections of this paper are organized as follows. Section 2 presents the FG climbing model and LLS running model used as the basis for the development of the physical platform, as well as scaling arguments and a dynamic simulation of the platform. Section 3 describes the design of SCARAB and its components. Section 4 details the experimental procedures utilized in both the simulation and experimental studies. The results of the simulation and physical experiments and a discussion of their implications are given in Section 5. Section 6 summarizes the contributions of this study and suggests avenues for future developments.

## 2 Modeling and Simulation

SCARAB was designed to be able to rapidly traverse both level and vertical surfaces. To guide the development of this platform, two biologically inspired models for dynamic locomotion were utilized as templates. These models, the Full-Goldman model for dynamic climbing and the Lateral Leg Spring model for horizontal plane running, are described in Sections 2.1 and 2.2, respectively. Both of these models were developed at a  $2g$  scale, necessitating the use of dynamic scaling to scale the models' elements to a size that is more suited to platform development. The details of dynamic scaling are described in Section 2.3. Finally, a simulation was developed to more accurately model the robot, which is presented in Section 2.4.

### 2.1 Full-Goldman Climbing Model

Animals of varying size, leg number, attachment mechanism, and morphology have demonstrated the capacity for rapid vertical locomotion. While it would seem that differing climbing strategies would be adopted for these animals, the center of mass trajectories and ground reaction force profiles show similar characteristics [13]. Two key points can be extracted from these observations. First, the center of mass trajectories appear to be pendular even though climbing animals typically have multiple legs in contact with the climbing substrate at the same time, restricting free pendular dynamics. Second, significant lateral forces are generated, almost half those generated in the fore-aft direction, indicating that lateral dynamics play a crucial role in rapid vertical locomotion [16]. The dynamics generating this climbing behavior are captured in the reduced-order Full-Goldman (FG) model for dynamic climbing. A schematic of the model is shown in Fig. 1A.



**Fig. 1** Models for dynamic locomotion. (A) Schematic of the FG dynamic climbing model. (B) Schematic of the LLS horizontal plane running model. (C) Model utilized for the dynamic simulation of climbing and running with a quadrupedal platform.

The model is composed of a rigid, distributed mass body of mass  $m$  and moment of inertia  $I$  attached to two massless legs with a nominal length of  $l_0$  consisting of a spring of stiffness  $k$  in series with a linear actuation element. The base of the legs are fixed a distance  $d_1$  above the center of mass and  $d_2$  to the left and right. These attachment points, the hips, are rigid joints, locking the legs at a prescribed sprawl angle  $\beta$ , defined as the angle between the longitudinal axis of the body and the axis along the length of each leg. At the end of each leg is a foot that can attach to the climbing substrate as a freely rotating pin-joint.

During climbing, each step begins with a touch-down event, which occurs when one foot establishes contact with the climbing surface, beginning the stance phase for that leg. At this point the leg spring is at its rest length and the linear actuation element is maximally extended. The actuation element then begins to contract, pulling the body towards the stance foot pivot as the body swings as a pendulum about this point. While the stance leg contracts, the opposing leg extends the actuation element to the maximal length to prepare for the next step. The stance foot maintains contact with the climbing substrate until the actuation element has fully contracted, at which point the contact is broken for the stance foot while the opposing foot establishes contact and begins the next step. This leg follows the same behavior until the first foot again touches down. The period between two touch-down events of the same leg is defined as a stride. The model continues to take alternating steps to generate the climbing behavior.

## 2.2 *Lateral Leg Spring Running Model*

As with rapid climbing, similar locomotion characteristics have been exhibited by animals of various sizes and morphologies for locomotion on level surfaces. The sagittal plane dynamics have been well described by the Spring Loaded Inverted Pendulum (SLIP) model [5, 6]. However, this model neglects the lateral plane dynamics, which play an integral role in locomotion, particularly in the case of sprawled posture animals. The Lateral Leg Spring (LLS) model was developed to account for these effects and to describe the dynamics of horizontal plane of running animals [22]. A schematic of this model is shown in Fig. 1B.

The configuration of the LLS running model is similar to that of the FG climbing model. Two axially elastic legs are attached to a rigid, distributed mass body. However, two important differences exist between the models. First, the attachment of the legs to the body at the hip serves as a freely rotating pin joint rather than locking the legs at a fixed sprawl angle. Second, there is no axial actuation element in the legs, as the model is energetically conservative. This could not be the case for dynamic climbing since kinetic energy is constantly being converted to gravitational potential and must be replenished through actuation.

A running step begins with a touch-down event, at which point the foot beginning stance establishes contact with the running surface with the leg fully extended and at the desired sprawl angle. The body moves towards the stance foot under its own momentum, compressing the spring and storing energy as elastic potential while

rotating around the stance foot and hip joint. The spring then extends, accelerating the body away from the stance foot and rotating the body. When all elastic potential has been returned to the system, the stance foot detaches and the flight foot establishes contact, beginning the next step. Since the hip acts as a freely rotating pin joint about which the body can rotate during stance, in the flight phase, the leg must be reset to the desired initial sprawl angle. The process is symmetrical between the left and right legs and the passage of two steps (one with the left leg and one with the right leg) constitutes a stride.

### 2.3 Dynamic Scaling

The similarities in configuration and locomotion of the FG climbing model and LLS running model simplifies the development of a platform capable of exemplifying both locomotion modalities. However, a significant obstacle to designing a platform based on these models is scale. Since the inspiration for these models was cockroaches and small geckos, the model parameters are specified for an overall mass of approximately 2g. The desired mass for the platform was set at 2kg, following power density arguments suggesting the upper limit for a dynamic climbing platform using conventional DC motors to be at approximately this size [16]. To preserve the dynamic characteristics that are the reason for embedding these models in the platform, dynamic scaling is utilized [21]. The derivation of the dynamic scaling laws have been previously described [21, 10, 1]; therefore, in this paper we only present the scaling relations and the resulting parameter values, which can be found in Table 1.

While the parameter values for the scaled models were similar in most aspects, the desired leg stiffnesses show significant deviation. Since the nominal stiffness for climbing was twice that for running, it was desirable to develop a mechanism by which the effective leg stiffness could change when operating in the different modalities. This was done by developing a leg that had a different stiffness when loaded while extending (utilized for LLS running) and contracting (utilized for FG climbing), which is described in detail in Section 3.

**Table 1** Dynamic scaling relations and parameter values for scaling the reduced-order models to 2kg. For model and scaled values, the left value corresponds to the FG (climbing) model value while the right value corresponds to the LLS (running) model value. Note that there is no actuation length for the LLS model because it is energetically conservative.

Parameter	Model Values	Scale Factor	Scaled Value	Robot Values
Mass (kg)	0.002 / 0.0025	$\alpha_M$	2 / 2	1.88
Leg Stiffness ( $Nm^{-1}$ )	6 / 3.5	$\alpha_M^{2/3}$	600 / 300	640 / 320
Leg Length (m)	0.0154 / 0.017	$\alpha_M^{1/3}$	0.154 / 0.158	0.200
Actuation Length (m)	0.0092 / -	$\alpha_M^{1/3}$	0.092 / -	0.088
Stride Frequency (Hz)	9 / 10	$\alpha_M^{-1/6}$	2.85 / 3.28	Varies
Expected Velocity ( $ms^{-1}$ )	0.20 / 0.25	$\alpha_M^{1/6}$	0.63 / 0.76	Varies

## 2.4 *Dynamic Simulation*

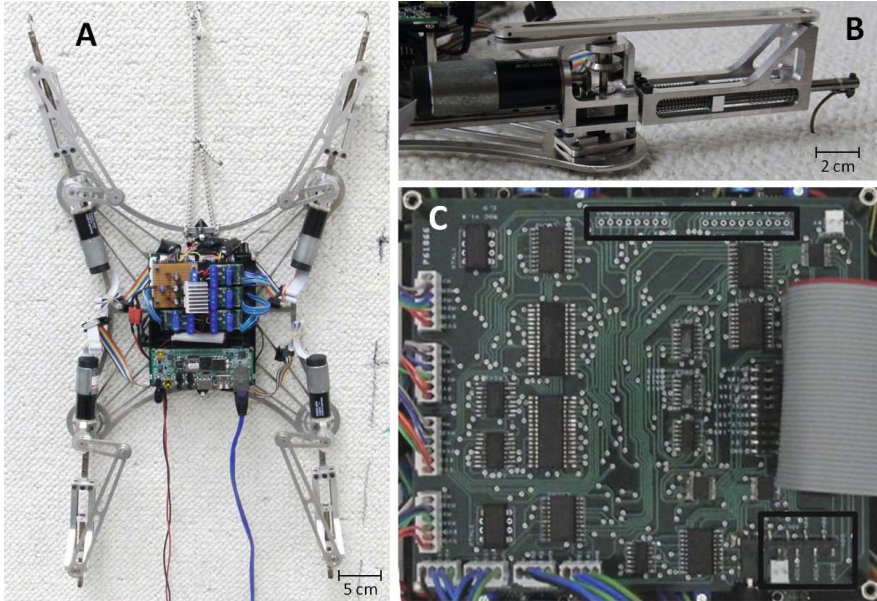
After determining the dynamically scaled values for a 2kg platform, a dynamic simulation was developed in Working Model 2D<sup>®</sup>. The simulation has a quadrupedal configuration and was designed with mass distributed in the legs and actuation mechanisms to better model the physical system, as shown in Fig. 1C. This simulation was used to verify dynamic similarity to the template, as well as for motor selection, controller design, and investigation of the effect of varying rear leg posture. It will additionally be used in performing parameter variation studies and examining transitions between climbing and running. While the model was developed for and is capable of demonstrating both climbing and running behavior, only the climbing simulation is detailed in this work. A detailed examination of the running behavior is omitted in order to allow a more in depth analysis of the climbing behavior, in particular, the role of fore-aft leg specialization.

## 3 **Physical Platform**

After demonstrating that the dynamic simulation preserved the climbing behavior of the FG model, the first prototype of SCARAB was designed, consisting of four legs for actuation and attachment, an electronics package for control of the robot, and a central body to connect the individual components. The fully assembled platform, shown in Fig. 2A, is approximately 50cm long and 30cm wide, depending on the configuration of the legs, and has a mass of 1.88kg.

Each leg, shown in Fig. 2B, is actuated via a Maxon RE-max 24 motor (Maxon 222049) with a 24:1 planetary gearhead (Maxon 14397). The motor is in series with a 3:2 bevel gear set, which drives a crank-slider mechanism to vary the rest length of the leg. This mechanism is used to add (or remove) energy during locomotion. Additionally, each leg has a linear spring in series with the crank-slider, which reduces the peak ground reaction forces, lowering the stress on the leg and reducing peak loads on the motor. It also assists with attachment, allowing the loading on the foot to increase gradually. As mentioned in Section 2.3, the shin, which houses the leg spring, was designed so it exhibited a different effective stiffness during running and climbing in a manner similar to that used on RiSE v1 [3]. The mechanism utilized for this was a slider braced between two sets of compression springs, which can be seen in Fig. 2B. Since the springs were not attached to the slider, moving the slider one way or the other would only engage one set of springs. By appropriate selection of the spring stiffnesses on both sides, the leg could behave with two separate effective stiffnesses depending on the direction of compression. The design of the leg is intentionally similar to a previous sagittal plane runner [2], facilitating the future implementation of SLIP-like running on level and sloped surfaces.

A foot is attached to the end of each leg for attachment to the climbing substrate. At this stage of development, the goal of the attachment scheme was to utilize a simple, passive mechanism that would enable reliable attachment without impacting the dynamic performance of the robot. To this end, the foot is designed to utilize



**Fig. 2** Assembled robot and individual components. (A) Photograph of SCARAB on the climbing surface. (B) Close up of the rear left leg, which is identical to the rear right leg. The front legs are the same except for the toe, which is reversed for attachment when the leg is extended. (C) Picture of the custom expansion board for the electronics system. The fully assembled electronics package has this board as the bottom layer, while the top layer can be seen in (A).

a hook-and-loop attachment mechanism. Each foot uses a single toe to allow the attachment point to function as a pin joint. The toe is a debarbed fish hook that has been bent to allow the point to catch the climbing surface. While this mechanism is fairly simple, it demonstrates directional adhesion in a manner comparable to that utilized by animals and previous climbing robots [14, 24, 18]. Additionally, the length of the toes on the front leg and back leg are different to position the robot at approximately a  $10^\circ$  angle relative to the climbing substrate. This allows the robot to pull itself towards the wall while in stance to improve the success of attachment at touch-down. The climbing substrate is a  $2.5m$  by  $1.25m$  vertical wooden wall with Berber carpet affixed to the climbing surface.

For platform operation, independent control of four actuators at a  $1kHz$  update rate was desired. In addition, the control system needed to be compact and light-weight. This motivated the development of a custom electronics package around the Gumstix Overo<sup>®</sup> Fire. This controller was chosen due to its small footprint, high clock frequency, expandable memory capacity, and the availability six pulse width modulation (PWM) and analog-to-digital lines. The primary drawback was the lack of sufficient general purpose input/output (GPIO) lines. To overcome this limitation, a custom expansion board, shown in Fig. 2C, was designed based on the architecture



of several Intel microprocessors [7], employing a multiplexed address/data bus and a separate control bus. This configuration is able to address up to 16 8-bit devices (or up to 128 individual GPIOs) to communicate with the central processor, though the current design only utilizes 4 of these device addresses. The electronics package enabled the utilization of two dual quadrature decoders and four single channel motor drivers to track and control the motors, while weighing only 298g and fitting into a 11cm by 13cm footprint.

The body to which the legs and electronics are attached is a 30cm by 30cm aluminum frame. The legs are each attached at the hips, located 12.5cm in both fore-aft and lateral directions from the center of the frame, while the electronics are mounted directly over the center of the body. This configuration places the geometric center of mass of the platform approximately in the center of the robot.

For both climbing and running, SCARAB utilizes a trotting gait, in which the front right and rear left legs are in stance while the front left and rear right legs are in flight and vice versa. To control positioning of the legs and to maintain the phase offset of the trotting pairs, position control of the individual leg motors is utilized. Each motor is given a desired trajectory that is prescribed by the stride frequency without any information about the position of the other motors. The motor tracks this desired trajectory using a proportional controller.

## 4 Experimental Procedure

The aim of the experimental tests described in this work is two-fold. The first goal is to verify the similarity of the physical platform, simulation, and the FG climbing model. The second is to examine the effect of rear leg configuration on climbing behavior. The development of the quadrupedal SCARAB enables the examination of fore-aft leg specialization, which has been shown to be beneficial for running robots but had yet to be tested in the scansorial regime.

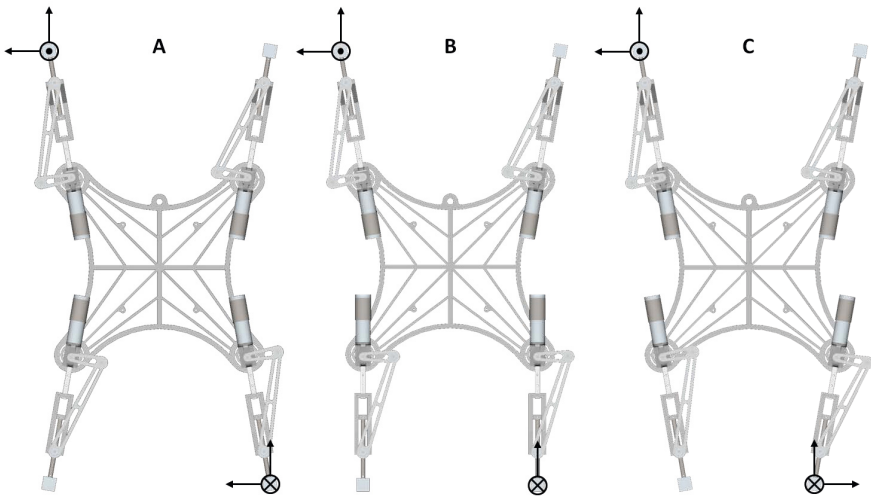
To quantify the behavior of the platform, experimental data was gathered using motion capture and current and voltage sensing. Whole body position and velocity data was obtained via a motion capture system using a high-speed digital camera (Casio Exilim EX-F1) that tracked 2 LED markers located above and below the center of mass of the robot. Motion tracking data was captured at 300fps and analyzed in MATLAB<sup>®</sup> using a custom point-tracking script. Power consumption was calculated from the current draw of the robot and the input voltage. A Vektrex<sup>™</sup> VCS10 current sensor and a Sparkfun<sup>®</sup> Logomatic V2 was used to measure and log the current draw of the robot, which was synchronized with the motion capture data and input voltage to determine the total system power consumption and efficiency during climbing.

The first experimental goal of verifying the similarity between SCARAB, the quadrupedal simulation, and the FG model was performed by comparing center of mass trajectories as well as fore-aft and lateral velocity profiles over the course of a stride. Since the FG model utilizes a sprawl angle of 10°, this angle was selected for the sprawl angle of the front legs on the quadrupedal platforms. A rear sprawl angle

of  $10^\circ$  was also selected for this comparison. For the physical platform, the motion capture system was used to gather data from 15 trial runs, which were compiled to generate an average center of mass trajectory and velocity profiles over the course of a stride. In each trial, the robot was placed at the bottom of the wall and climbed to the top. To help minimize transients, only the data from the final stride was used for the verification. The simulated data was obtained through forward simulation of the quadrupedal model. The simulation was allowed 15 strides to reach steady state before data was gathered from a single stride.

In the second set of experiments, the effects of rear leg posture were investigated. Several rear leg configurations were examined, as shown in Fig. 3. The first case was an outward-sprawled configuration, which showed similarity to the observed leg orientation of sprawled posture animals during climbing (though not necessarily their force generation). The second case was a zero-sprawl configuration, which was expected to demonstrate similar rear leg function to cockroaches [13]. The third case was an inward-sprawled configuration and was expected to reproduce ground reaction forces observed in the rear legs of geckos when climbing rapidly [4]. Overall, five rear sprawl angles were examined, ranging from  $-20^\circ$  to  $20^\circ$  in  $10^\circ$  increments, with negative angles corresponding to inward-sprawl and positive angles corresponding to outward-sprawl.

Three behavior characteristics were examined for each configuration: fore-aft velocity, lateral velocity, and efficiency. The fore-aft velocity corresponds to the mean fore-aft velocity during the stride. The lateral velocity was calculated as the



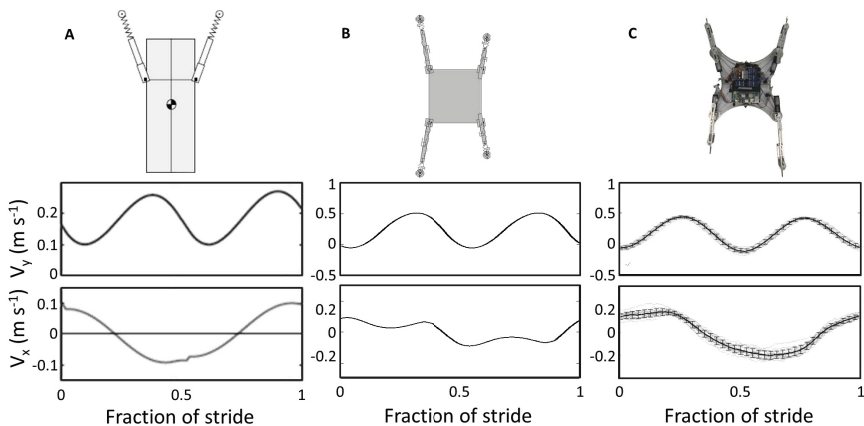
**Fig. 3** Examined configurations for the rear legs of the platform and expected fore-aft and lateral ground reaction forces which would be generated when the front-left and rear-right legs are in stance. (A) Configuration in which rear legs are sprawled outward. (B) Configuration in which the rear legs have no sprawl. (C) Configuration in which the rear legs are sprawled inward.

root-mean-square lateral velocity during the stride. Finally, the efficiency was determined via specific resistance (SR), calculated by  $SR = P/mgv$ , where  $P$  is the average power consumption during the stride,  $m$  is the mass of the robot,  $g$  is the acceleration due to gravity, and  $v$  is the average fore-aft velocity. Note that lower values for SR are more efficient.

## 5 Results

In preliminary experiments, the robot was run on vertical and horizontal surfaces and demonstrated mean speeds of up to  $0.17 \text{ ms}^{-1}$  when climbing a vertical surface and up to  $0.43 \text{ ms}^{-1}$  when running across level ground. The climbing results are further discussed below while the running will be more fully examined in future work to allow for an improved analysis of the two locomotion modalities.

The velocity profiles over the course of a single stride for the FG model, the dynamic simulation, and SCARAB are shown in Fig. 4. Both fore-aft and lateral velocity exhibit similar profiles for all three. However, the fore-aft speed is lower than predicted by dynamic scaling (see Table 1). This is the result of running the quadruped at a  $1.5\text{Hz}$  rather than the dynamically scaled frequency of  $2.85\text{Hz}$ , which was due to failure to establish attachment with the front feet at high stride frequencies. While several factors likely contributed, the most significant is probably that the stiffness of the rear legs being less than the dynamically scaled values. Since the rear toes were made long enough to produce a  $10^\circ$  angle between the body and the wall, they also acted as a cantilever beam in series with the leg, which was not accounted for in the development of the platform. This resulted in a lower natural frequency of the leg, such that when attempting to climb at the dynamically

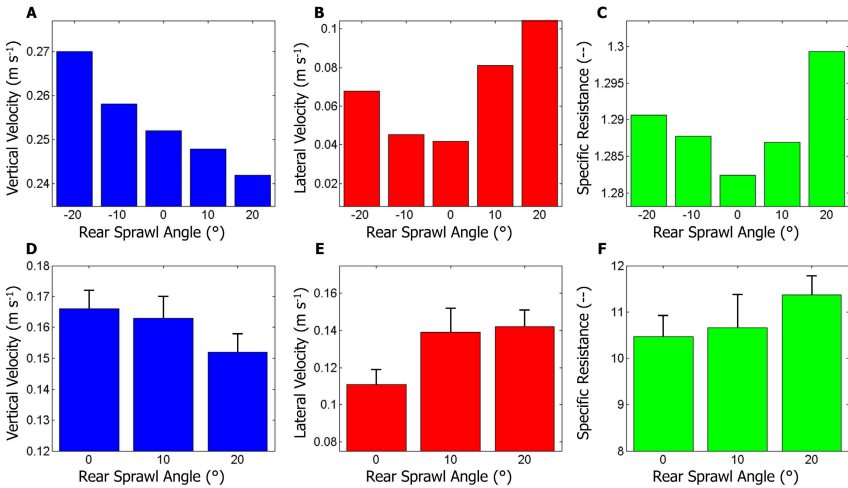


**Fig. 4** Fore-aft ( $V_y$ ) and lateral ( $V_x$ ) velocity profiles during a stride. (A), (B), and (C) show velocity profiles for the FG model [13], the quadrupedal simulation, and SCARAB, respectively. The error bars in (c) show the standard deviation for the velocity at increments of the stride.

scaled frequency, the rear springs would not be able to return the energy stored during leg extension. It also pivoted the body about the rear toe attachments, pitching the front feet away from the wall. However, the similarity of the qualitative shape of the profiles demonstrates that SCARAB still exhibits the lateral force generation and oscillating velocity profile characteristic of the climbing template. Furthermore, increasing the speed of the dynamic simulation results in the expected mean climbing speed being realized.

Rear leg sprawl variations were also examined, as shown in Fig. 5. From these results, several trends can be noted. First, moving from more outward-sprawled postures to more inward-sprawled postures increases vertical climbing speed by 10%. Second, zero-sprawled postures exhibit the lowest peak lateral velocities while a sprawled posture, whether inward or outward, increases the lateral velocity, up to twice the magnitude of the zero-sprawled configuration. While low lateral velocity may at first seem desirable, lateral velocities of approximately half the climbing speed have been shown to correspond to more stable climbing [9]. Third, although the magnitude of the difference is small (only about 1%), a similar trend to lateral velocity is observed for specific resistance. This indicates that climbing with a zero-sprawled posture may slightly improve efficiency while more sprawled postures may be better suited when stability is most essential.

The effects of utilizing an inward-sprawled posture were unable to be tested on the physical robot due to attachment failure. This is a result of out of plane roll that was not captured in the 2D dynamic simulation. The roll results in the front foot



**Fig. 5** Behavior characteristics as a function of rear sprawl angle. (A), (B), and (C) show the effect of varying rear sprawl angle on the mean fore-aft velocity, the peak lateral velocity, and the specific resistance of the quadrupedal simulation, respectively. (D), (E), and (F) show effect of varying rear sprawl angle for the same behavior characteristics on the experimental platform. The error bars show the standard deviation in the experimental results.

missing the climbing substrate when the leg begins contracting, causing the robot to fall off the wall. This effect has been seen in previous dynamic climbing robots and has been dealt with in the past through stabilization via a roll bar extended off the rear of the platform [9]. While the lack of a roll bar keeps the experimental platform from utilizing inward-sprawled configurations, the robot is able to climb successfully with no rear sprawl or outward-sprawled legs, suggesting that the rear legs, when not sprawled inward, reduce out of plane roll.

Results for the outward and zero-sprawl configurations show similar trends to the simulation data. To assess the significance of the trends, two-sample t-tests were performed between the behavior characteristics of the zero-sprawl and the 20° outward-sprawled configurations. A significant decrease in SCARAB's mean climbing speed ( $p < 0.001$ ) was observed as the sprawl angle was increased from 0° to 20°, as well as increases in both peak lateral velocity ( $p < 0.0001$ ) and specific resistance ( $p < 0.001$ ). It is worth noting that the improvement in efficiency on the physical platform is greater than observed in simulation, and is likely the result of measuring efficiency via electrical power on SCARAB rather than mechanical power, as was done in simulation. These results corroborate the simulation findings and suggest a trade off between efficiency and stability.

## 6 Conclusions

In this work, we present the design and experimental validation of SCARAB, the first quadrupedal robotic platform capable of dynamic climbing on a planar surface and horizontal plane running on level ground. Validation studies showed that the platform exhibited dynamically similar behavior profiles to the biologically-inspired Full-Goldman climbing template. In addition, SCARAB was used to investigate the effects of functional leg specialization on climbing performance. Simulation results showed consistent trends among the behavior characteristics. First, increased speed can be achieved through the utilization of inward-sprawled gaits. Second, increased magnitude of rear leg sprawl can increase the maximum lateral velocity during climbing. This has been previously shown to correlate to, and potentially contribute to, improved stability to perturbations. Third, efficiency is best for zero-sprawl configurations, though the improvement in specific resistance is slight. The results from the experimental platform agree with the simulation findings. However, the utilization of inward-sprawled configurations failed due to unmodeled out of plane roll that caused attachment failure on the robot.

This study has provided a preliminary examination of quadrupedal dynamic climbing, but there are still several avenues that deserve further attention. In particular, a thorough exploration of the stability and robustness of the SCARAB platform will be undertaken, as well as an investigation of attachment mechanisms, such as those used by RiSE [24] and StickyBot [18], to explore options for climbing on various natural and man-made surfaces and to assess the challenges of using these mechanisms during dynamic locomotion.

While only the dynamic climbing behavior has received in depth analysis thus far, preliminary investigations into running on level surfaces have been conducted and a detailed analysis is forthcoming, as well as examinations of both the running and climbing performance of SCARAB on sloped surfaces. The capacity for locomotion via both modes enables the investigation of why animals change the lateral force generation from pushing to pulling as the slope of a surface increases [20], a distinguishing characteristic between the FG and LLS models, and which may have a significant effect on the stability and efficiency of running on inclined surfaces.

**Acknowledgements.** This work was supported by the collaborative participation in the Robotics Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD 19-01-2-0012. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation thereon.

## References

1. Alexander, R.M.: Principles of Animal Locomotion. Princeton University Press (2003)
2. Andrews, B., Miller, B., Schmitt, J., Clark, J.E.: Running over unknown rough terrain with a one-legged planar robot. *Bioinspiration & Biomimetics* 6(2), 1–15 (2011)
3. Autumn, K., Buehler, M., Cutkosky, M., Fearing, R., Full, R., Goldman, D., Groff, R., Provancher, W., Rizzi, A., Saranli, U., Saunders, A., Koditschek, D.: Robotics in scensorial environments. *Proc. SPIE* 5804, 291–302 (2005)
4. Autumn, K., Hsieh, S.T., Dudek, D.M., Chen, J., Chitaphan, C., Full, R.J.: Dynamics of geckos running vertically. *Journal of Experimental Biology* 209(2), 260–272 (2006)
5. Blickhan, R.: The spring-mass model for running and hopping. *Journal of Biomechanics* 22(11-12), 1217–1227 (1989)
6. Blickhan, R., Full, R.J.: Similarity in multilegged locomotion: Bouncing like a monopode. *Journal of Comparative Physiology A-Sensory Neural and Behavioral Physiology* 173(5), 509–517 (1993)
7. Brey, B.B.: The Intel Microprocessors, 8th edn. Prentice-Hall (2008)
8. Cavagna, G.A., Heglund, N.C., Taylor, C.R.: Mechanical work in terrestrial locomotion: two basic mechanisms for minimizing energy expenditure. *American Journal of Physiology - Regulatory, Integrative and Comparative Physiology* 233(5), R243–R261 (1977)
9. Clark, J., Goldman, D., Lin, P., Lynch, G., Chen, T., Komsuoglu, H., Full, R., Koditschek, D.: Design of a bio-inspired dynamical vertical climbing robot. In: *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA (June 2007)
10. Clark, J.E., Cutkosky, M.R.: The effect of leg specialization in a biomimetic hexapedal running robot. *Journal of Dynamic Systems Measurement and Control* 128(1), 26–35 (2006)
11. Full, R.J., Blickhan, R., Ting, L.H.: Leg design in hexapedal runners. *Journal of Experimental Biology* 158(1), 369–390 (1991)
12. Full, R.J., Koditschek, D.E.: Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology* 202(23), 3325–3332 (1999)
13. Goldman, D.I., Chen, T.S., Dudek, D.M., Full, R.J.: Dynamics of rapid vertical climbing in cockroaches reveals a template. *Journal of Experimental Biology* 209(15), 2990–3000 (2006)

14. Gorb, S.N.: Biological attachment devices: exploring nature's diversity for biomimetics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 366(1870), 1557–1574 (2011)
15. Kim, S., Clark, J.E., Cutkosky, M.R.: iSprawl: Design and tuning for high-speed autonomous open-loop running. *The International Journal of Robotics Research* 25(9), 903–912 (2006)
16. Lynch, G.A., Clark, J.E., Lin, P., Koditschek, D.E.: A bioinspired dynamical climbing climbing robot. *The International Journal of Robotics Research* 31(8) (2012)
17. McMahon, T.A.: Mechanics of locomotion. *The International Journal of Robotics Research* 3(2), 4–28 (1984)
18. Santos, D., Kim, S., Spenko, M., Parness, A., Cutkosky, M.: Directional adhesive structures for controlled climbing on smooth vertical surfaces. In: 2007 IEEE International Conference on Robotics and Automation, ICRA (April 2007)
19. Saranlı, U., Buehler, M., Koditschek, D.E.: RHex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research* 20(7), 616–631 (2001)
20. Schmitt, J., Bonnono, S.: Dynamics and stability of lateral plane locomotion on inclines. *Journal of Theoretical Biology* 261(4), 598–609 (2009)
21. Schmitt, J., Garcia, M., Razo, R.C., Holmes, P., Full, R.J.: Dynamics and stability of legged locomotion in the horizontal plane: a test case using insects. *Biological Cybernetics* 86(5), 343–353 (2002)
22. Schmitt, J., Holmes, P.: Mechanical models for insect locomotion: dynamics and stability in the horizontal plane i. theory. *Biological Cybernetics* 83(6), 501–515 (2000)
23. Shill, J., Miller, B., Schmitt, J., Clark, J.E.: Design of a dynamically stable horizontal plane runner. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 4749–4754 (May 2010)
24. Spenko, M.J., Haynes, G.C., Saunders, J.A., Cutkosky, M.R., Rizzi, A.A., Full, R.J., Koditschek, D.E.: Biologically inspired climbing with a hexapedal robot. *Journal of Field Robotics* 25(4-5), 223–242 (2008)

# **Part III: ISER Session Summary on “Interactive Session”**

Henrik I. Christensen

RIM@GT, Georgia Tech, Atlanta, GA 30332

## **Session Summary**

Following the success at the previous ISER and IROS-2011 this years ISER also had an interactive session, where authors have an opportunity to provide a more 1-2 interaction as they discuss their research. This is a great model for having interactive presentations rather than the usual one-way presentation of research. The model is particularly well suited for experimental research, which typically has simulation models, video or detailed graphics, all of which is suited for interactive discussions.

The interactive session included 9 papers that cover the areas of Human-Robot Interaction, Multi-Robot Coordination, Mapping and Long-term Autonomy, and Perception and Control. The paper by Carton et al discusses design of a service robot that can safely approach people in dynamic environments. The method utilizes human behavior data to plan safe paths and to design associated control strategies. The method in particular considers speed and distance parameters to evaluate optimal control strategies. The second paper on human-robot interaction describes the Jedi-bot system that was developed at Stanford. The robot is design for execution of the adversarial game of sword fighting. To achieve this a kinect sensor is used to detect the sword and its trajectory and compute an intersecting trajectory. The system uses a highly reactive control strategy and was implemented on the KUKA Light Weight Robot 4 using the fast research interface. The paper by Gergondet et al describe a system for tele-operation of small humanoid robotics using a brain-computer interface to allow users that may be paralyzed or motor impaired to interact with intelligent devices such as a robot. Abichandani et al report on design of a multi-robot system for path coordination. The method utilizes a receding horizon approach applied to mixed integer non-linear programming. They approach takes communication constraints into account and has been evaluated on teams of robots that include up to 5 members and performing in complex work environments. Worchester et al present a multi-robot task decomposition method. The objective is to perform multi-robot assembly of large-scale structures. The system uses a sensor-based approach to detect the current state and pieces available to complete the assembly process. The system generates a plan for sub-assemblies and uses that for control of mobile platforms for cooperative assembly. Rogers et al present a system for multi-robot exploration and mapping of indoor environments. The exploration process is driven by a set of evolving frontiers that control the robot coordination process.



The system has been tested in office like environments and includes teams of up to 9 robots. The system has been used to explore environments that cover more than 10,000 sq ft. Ott & Ramos present an approach for unsupervised clustering of sensory data to allow for on-line generation of semantics and doing so while honoring real-time processing constraints. To achieve this the method utilizes affinity propagation and introduction of meta-points. Yodar presents a method for using stereovision to estimate the pose of work pieces to be picked up. In small series manufacturing CAD models may not be readily available and for such situations it is desirable to directly use stereo based pose estimation. Three different strategies for estimation of pose are evaluated and grasping use a marker based end-effector strategy is presented. Finally, Sharf presents a strategy to use Blimps as part of art-displays. Two different blimp designs are considered together with two different control strategies, one using a fast motion with less accuracy and another that has higher accuracy but is significantly slower.

# Real-Time Clustering for Long-Term Autonomy

Lionel Ott and Fabio Ramos

**Abstract.** In the future robots will have to operate autonomously for long periods of time. To achieve this, they need to be able to learn directly from their environment without human supervision. The use of clustering methods is one possibility to tackle this challenge. Here we present extensions to affinity propagation, a clustering algorithm proposed by Frey and Dueck [5], which makes it suitable for real-time and long-term use in robotics applications. The proposed extension, called meta-point affinity propagation, introduces so called meta-points which increase the performance of the clustering and allows for incremental usage. Additionally we propose a method that enables us to obtain probabilistic cluster assignments from any affinity propagation based clustering method. We show experimental results on the quality and speed of meta-point affinity propagation as well as the probabilistic cluster assignments. Furthermore, we demonstrate how meta-point affinity propagation allows us to process data sets much larger than what affinity propagation is able to handle.

## 1 Introduction and Related Work

Our long-term vision is to enable robotic systems to explore and build models of unknown environments over extended periods of time without human supervision or prior knowledge. This requires methods that allow the robot to build a model from observations in an unsupervised manner. Such methods need to be efficient as robots typically have limited resources and the operations need to be performed in a timely manner. In addition to creating a model, another important part is the exploitation of the model for higher-level tasks crucial for autonomy such as obstacle avoidance, exploration or scene understanding.

---

Lionel Ott · Fabio Ramos  
Australian Centre for Field Robotics,  
School of IT, University of Sydney  
e-mail: {l.ott, f.ramos}@acfr.usyd.edu.au

We approach building such environment models from a clustering perspective. Clustering allows us to group similar data together into clusters in an unsupervised way. In our case, the data are the sensory readings of the robot. However, with no prior information about the data, only methods that can infer the number of clusters automatically are suitable.

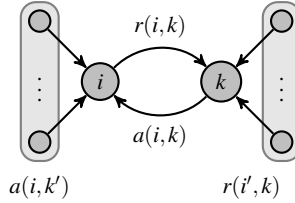
The contributions presented in this work are two fold. First, we introduce meta-point affinity propagation, a novel method to efficiently cluster large amounts of noisy data. Secondly, we present a method that allows any algorithm based on affinity propagation to obtain probabilistic clustering assignments. We show how these contributions allow us to cluster large numbers of points in a fraction of the time it takes standard affinity propagation. Additionally, we demonstrate how the probabilistic interpretation of cluster assignments can be used to evaluate the quality of clustering results.

## ***1.1 Related Work***

The most prominent clustering methods that are capable of inferring the number of clusters from data are latent Dirichlet allocation [2], spectral clustering [10], DB-SCAN [4] and more recently affinity propagation [5]. All of these methods differ in the types of assumptions they make, their complexity and flexibility. Unsupervised learning has been used by Happold et al. [7] in order to learn colour based models which enables them to predict terrain traversability from image data. A different approach, with the same goal, was proposed by Kim et al. [8] in which they use the experience of a robot as it drives through the environment to learn a model that maps visual appearance to terrain traversability. Visual appearance can also be used directly, for example, Giguere et al. [6] use k-means clustering to learn the model of a coral reef for the purpose of steering a robot such that it remains above the coral reef. The work by Steinberg et al. [13] uses Dirichlet process mixture models to learn models of the benthic habitats present in image data gathered by an AUV. Such methods are not limited to image data as shown by Modayil and Kuipers [9], who learn object models from laser scan data using a simple clustering method. A different method that learns models from 3D point clouds is presented by Ruhnke et al. [12], who employ spectral clustering to cluster models based on their consistency.

## **2 Approach**

We cast the task of building a model of the environment as a clustering problem. We use affinity propagation [5], a state-of-the-art clustering method, to build the model of the environment. Affinity propagation has many advantages for our purpose. First, there is no need to define the number of clusters a priori as they are determined from the data itself. Second, the only inputs required are the similarity values between data points which can be any sensible value in the context of the application, and is not required to be a metric. These similarity values are then used



**Fig. 1** Messages exchanged by affinity propagation between nodes in each iteration. Both messages take into account the accumulated values of the other message at the given node.

by affinity propagation to compute the clustering solution by iteratively computing two messages, availability and responsibility. Availability  $a(i, k)$  is the message sent from point  $k$  to point  $i$  and encodes how good of an exemplar  $k$  would be for  $i$  based on evidence available to point  $k$ . While responsibility  $r(i, k)$ , sent from point  $i$  to  $k$ , encodes how suited point  $k$  is as an exemplar for point  $i$  given the information available in  $i$ . After initialising all messages to 0 they are computed iteratively until convergence is achieved. The actual equations used to compute the messages are shown below:

$$r(i, k) = s(i, k) - \max_{k' \text{ s.t. } k' \neq k} (a(i, k') + s(i, k')) \quad (1)$$

$$a(i, k) = \min \left( 0, r(k, k) + \sum_{i' \text{ s.t. } i' \notin \{i, k\}} \max(0, r(i', k)) \right) \quad (2)$$

$$a(k, k) = \sum_{i' \text{ s.t. } i' \neq k} \max(0, r(i', k)), \quad (3)$$

where  $s(i, k)$  is the similarity between points  $i$  and  $k$ . Figure 1 shows how these two messages interact with each other. One can see that both messages are computed using the values of the other messages accumulated at the node.

In this work, we use cameras to perceive the robot's environment and thus need a compact way to represent the visual appearance of observations. To this end we split the images observed by the robot into small rectangular patches. For a  $640 \times 480$  image the patches are typically  $80 \times 60$  in size. From these, we extract HSV colour space histograms and histograms of local binary patterns [11], thus capturing both colour and texture information. The similarity between features obtained from observations in this way is computed as the sum of the histogram similarities, i.e.:

$$\text{sim}(H_1, H_2) = -\text{dist}(H_1^{\text{colour}}, H_2^{\text{colour}}) - \text{dist}(H_1^{\text{texture}}, H_2^{\text{texture}}), \quad (4)$$

where  $H_1$  and  $H_2$  are the histograms of the colour and texture information for each of the image patches. The distance  $\text{dist}$  between two histograms is computed using the Bhattacharyya distance of two histograms.

## 2.1 *Meta-point Affinity Propagation*

While standard affinity propagation produces good results, it is too slow to process thousands of data points in a few seconds. We therefore propose a method called meta-point affinity propagation which is inspired by ideas presented in Cao et al. [3]. The main idea is that data points which are close in feature space can be grouped together and replaced by a single meta-point. In robotics, similar observations occur frequently for example multiple observations made from a similar pose. By replacing such redundant observations with a single aggregated one, we effectively reduce the number of points involved in the computation of affinity propagation.

A meta-point  $\mathbf{P}_i$  stores the following information:

$$\mathbf{P}_i = \{\text{count, mean, exemplar, last-update}\}, \quad (5)$$

with the fields having the following meaning:

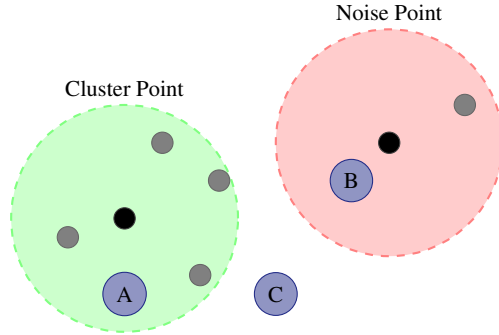
$\mathbf{P}_i.\text{count}$	number of points represented by the meta-point
$\mathbf{P}_i.\text{mean}$	the mean value of all represented data points
$\mathbf{P}_i.\text{exemplar}$	representative raw data point for this meta-point
$\mathbf{P}_i.\text{last-update}$	time the meta-point has been updated last

Besides the immediate effect of reducing the computational burden, the concept of meta-points has two additional benefits:

- the number of meta-points is dependant on the size of the feature space;
- random observations can be dealt with in a straight forward way.

The first point is a direct consequence of the usage of meta-points instead of raw data points. If a robot moves in a static environment all observations will be mapped to one of the meta-points after a while and thus no new meta-points will be created. The second point requires us to distinguish between two types of meta-points: Cluster-points that represent the points used for clustering, and noise-points which are ignored during the clustering. A meta-point is considered a cluster-point once it represents enough raw data points, otherwise it is considered a noise-point. This allows us to discard points generated from random observations such as spurious readings from a laser scanner. Put differently we can detect and ignore outliers in our observations.

The most important part of meta-point affinity propagation is the handling of new observations. The pseudo code in Algorithm 1 shows the steps performed in order to add a point  $p$  into either the set of cluster-points  $\mathbf{P}$  or the set of noise-points  $\mathbf{N}$ . Figure 2 shows the possible cases described above and in Algorithm 1. We keep these two sets separate for performance reasons. A new data point is added to an existing data point, either cluster-point or noise-point, if the meta-point is similar enough to the data point. Otherwise a new meta-point is created from the new raw data point. In case that the data point was added to a noise-point and this one now represents enough points to be considered a cluster-point is moved to the set of cluster points  $\mathbf{P}$ .



**Fig. 2** Visualisation of meta-points and the different cases that can occur when adding a new data point. *A* is merged into the cluster-point while *B* is merged into the noise-point. Finally, *C* is used to create an entirely new meta-point.

Computing the actual clustering result is then performed using standard affinity propagation using the cluster-point data. The two parameters required by meta-point affinity propagation are,  $\{\theta_{\text{min-points}}, \theta_{\text{similarity}}\}$ . They define the minimal number of points required for a meta-point to be considered during the clustering and the maximal difference in similarity between a meta-point and a new point for it to be considered part of that meta-point respectively. The similarity threshold is tied to the range of values the chosen similarity measure can take on. The minimum number of points is related to the rate at which observations are made. If the value is too low many points that can be considered noise will be added and if it is too high actual clusters that appear only rarely may not be added. In order to prevent noise-points to turn into meta-points by accumulating over long periods of time one can also prune noise-points that have not been observed for a set period of time. In the extreme case where both parameters are set to zero we recover the original affinity propagation algorithm.

This form of merging data points obviously assumes that small changes in the feature space distance result in no noticeable change of the object class to be clustered. Additionally the handling of noise only addresses noise which results from random measurements or one off sensing failures. It does not detect or handle complete failure of a sensor or systematic noise, as these produce consistent and continuous observations.

## 2.2 Probabilistic Cluster Assignments

Affinity propagation performs hard cluster assignments, i.e. each data point is assigned to exactly one cluster. Often times, however, assignments are not this clear-cut. Furthermore, probabilistic methods are widely used in robotics since they enable us to deal with uncertainties of representations in a principled way. Thus a probabilistic interpretation of the clustering would be highly beneficial. As it turns

```

ADD-DATA-POINT( $p$ )
1   $nn = \text{NEAREST-NEIGHBOUR}(\mathbf{P}, p)$ 
2  if  $\text{DIST}(nn, p) < \theta_{\text{similarity}}$ 
3     $\text{UPDATE-META-POINT}(nn, p)$ 
4  else
5     $nn = \text{NEAREST-NEIGHBOUR}(\mathbf{N}, p)$ 
6    if  $\text{DIST}(nn, p) < \theta_{\text{similarity}}$ 
7       $\text{UPDATE-METAPOINT}(nn, p)$ 
8      if  $nn.\text{count} \geq \theta_{\text{min-points}}$ 
9         $P = P \cup nn$ 
10        $N = N \setminus nn$ 
11    else
12       $\text{noise} = \text{CREATE-META-POINT}(p)$ 
13       $N = N \cup \text{noise}$ 

```

**Algorithm 1.** Pseudo code detailing the steps performed by meta-point affinity propagation when a new data point is added.  $\mathbf{P}$  is the set of cluster-points,  $\mathbf{N}$  the set of noise meta-points and  $\theta$  the parameters.

out we can derive such an interpretation by analysing the internal representation of affinity propagation in the following way.

We start by finding the set of exemplars  $E$  in the typical way of affinity propagation, that is by selecting set of indices  $k$  for which:

$$e_k \in E : a(k, k) + r(k, k) > 0, \quad (6)$$

holds, i.e. all points for which the value of the self-availability and self-responsibility is greater then zero. The probability of an observation  $i$  belonging to cluster  $k$  is then defined as

$$p(i = k) = \frac{1}{Z} f(a(i, k) + r(i, k)), \quad (7)$$

where

$$Z = \sum_{e \in E} f(a(i, e) + r(i, e)) \quad (8)$$

is the normalization factor and  $f$  is a function that maps its inputs to the range  $[0, 1]$ . In our case we use the logistic function:

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (9)$$

as it covers the values typically taken on by affinity propagation well. From this probability distribution we can obviously recover the hard assignment affinity propagation makes as:

$$\underset{k}{\operatorname{argmax}} p(i = k) \quad (10)$$

However, we also gain the ability to reason about an observation in terms of its fit with respect to the current clustering. From the entropy of the assignment, obtained as:

$$H(i) = - \sum_{k=1}^n p(i=k) \log p(i=k), \quad (11)$$

we can tell how well the clustering can explain the observation. A small entropy value indicates a well explained data point, i.e. a peaked distribution, whereas a large value indicates an observation that can be explained similarly well by multiple clusters.

In addition to the evaluation of a single observation we can extend this to the entire clustering solution in order to obtain an overall quality measurement. A measure of the quality of the clustering solution is interesting as for humans it is easy to tell if the result obtained by clustering is meaningful. However, a robot lacks this intuition and thus a way to quantify the quality of clustering results is of great importance. The evaluation of clustering results has been the focus of intense research and there are different methods to obtain such a measure, see [1] for an overview. However, most of the proposed methods are intended to assign a score to a clustering solution in order to compare different clustering methods against each other. As such they require a reference clustering solution as ground-truth. In robotics, however, obtaining ground-truth is often hard, if not impossible, and in our case no reference clustering exists. For this reason methods that rely on a reference clustering are not suitable for our application. A possible way to evaluate the clustering solution with the information we have access to is to compute an average entropy. This indicates how well all individual points clustered can be represented by the end result. Here again a low value indicates a good solution.

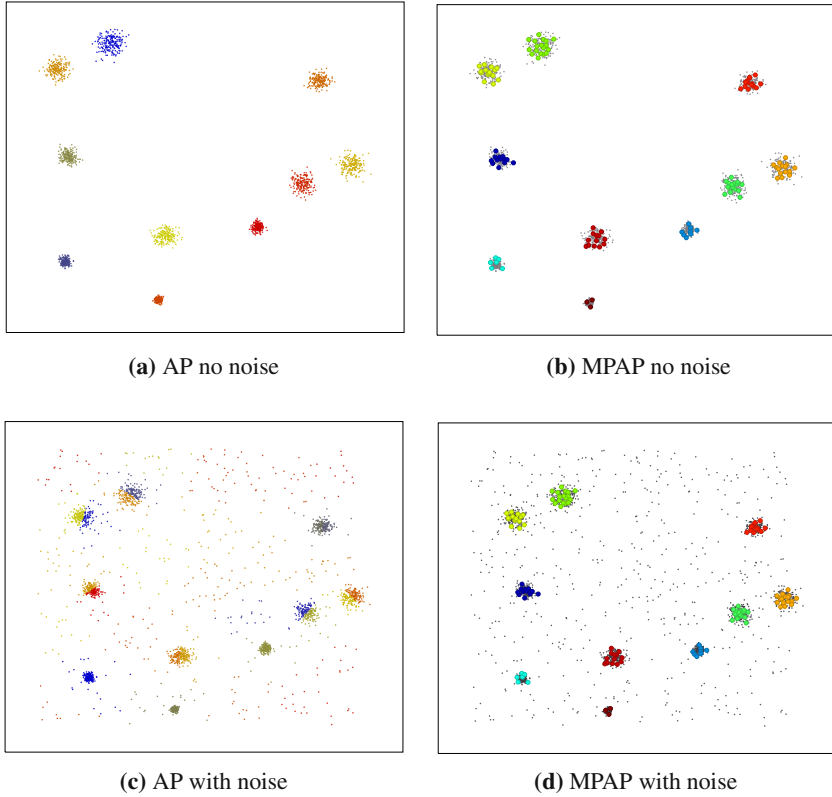
## 3 Experiments

### 3.1 Clustering Results and Speed

In a first set of experiments we compare the clustering results obtained by affinity propagation and meta-point affinity propagation on synthetic data, as shown in Figure 3. We use this to easily show the differences between the two algorithms. In the case without noise we can see that both methods find a good clustering solution, as indicated by the coloured points. The difference between the two is that affinity propagation has to cluster all the data points, whereas meta-point affinity propagation only has to cluster the meta-points, indicated by the bigger circles. This makes a large difference in speed, which can be seen in Table 1.

As mentioned in Section 2.1, an additional advantage of meta-point affinity propagation over affinity propagation is the ability to deal with random observations. For the synthetic data we simulate such random observations by adding data points from a uniform distribution. Figure 3c and Figure 3d show typical results for both methods under these conditions. One can see that affinity propagation creates a larger number of clusters compared to the result obtained on the same data without noise.





**Fig. 3** (a), (b) Exemplary results of affinity propagation and meta point affinity propagation on data without noise. (c), (d) Exemplary results of affinity propagation and meta-point affinity propagation with uniform noise. The small circles indicate the meta-points found by meta-point affinity propagation. The colouring indicates points that have been assigned to the same cluster.

This is due to the fact that affinity propagation has to assign each data point to a cluster and cannot consider some observations as noise and ignore them. This may in some cases, depending on the chosen parameters, result in clusters to be split as can be seen in Figure 3c. Meta-point affinity propagation, on the other hand, first builds meta-points which allows the method to reject points it considers to be noise and then the clustering has only to compute the solution for data without or reduced amount of noise. Comparing the meta-points for the data with and without noise we can see that in both cases they cover the actual clusters. Consequently, as far as meta-point affinity propagation is concerned, there is no noise in the data to be clustered.

From the plots of the synthetic data it is easily visible how meta point affinity propagation represents the original data with fewer data points. This allows meta



**Fig. 4** Exemplars found by both methods for a set of 1200 patches from an outdoor data set. From left to right we can see: brick wall, asphalt, tree, grass, wood chips and red concrete.



**Fig. 5** The exemplars obtained from batch-processing around 900 images collected in an outdoor environment using meta-point affinity propagation. The exemplars are not as unique as in Figure 4, however, the data was not pre-processed either.

point affinity propagation to be significantly faster than affinity propagation, as the numbers in Table 1 show. As affinity propagation has quadratic runtime the gaps between the two methods will keep increasing if more points from the same underlying model are added as meta-point affinity propagation will only update the meta-point statistics while affinity propagation has to handle entirely new points.

Figure 4 shows the exemplars obtained when clustering 1200 image patches that represent tree, grass, brick wall, asphalt, red concrete and wood chips. One can see that both methods find the same types of clusters with the big difference being the number of points involved in the clustering and the resulting speed which is shown in Table 1. The timing values include the entire processing of the data, i.e. feature extraction and management of meta-points. The reduction in data points by meta-point affinity propagation is quite drastic as only around 10% of the original data is retained while still producing the same clustering result.

Finally, we used meta-point affinity propagation to batch-process images captured while the robot was moving through the environment for 15 to 30 minutes. The clustering of these images requires several thousands of data points to be handled. The “Large-Scale Data” section in Table 1 shows the exact numbers. Standard affinity propagation can’t handle this amount of data in a reasonable amount of time. However, using meta-point affinity propagation we are able to reduce these numbers

**Table 1** Results for different clustering tasks. Synthetic data shows the results of the 2D example, outdoor data shows the results for the 1200 outdoor image patches data set and large-scale data shows results for entire sequences captured by a robot moving through the environment. For each of the data sets we show the number of clusters obtained, the number of actual data points clustered and the total run-time.

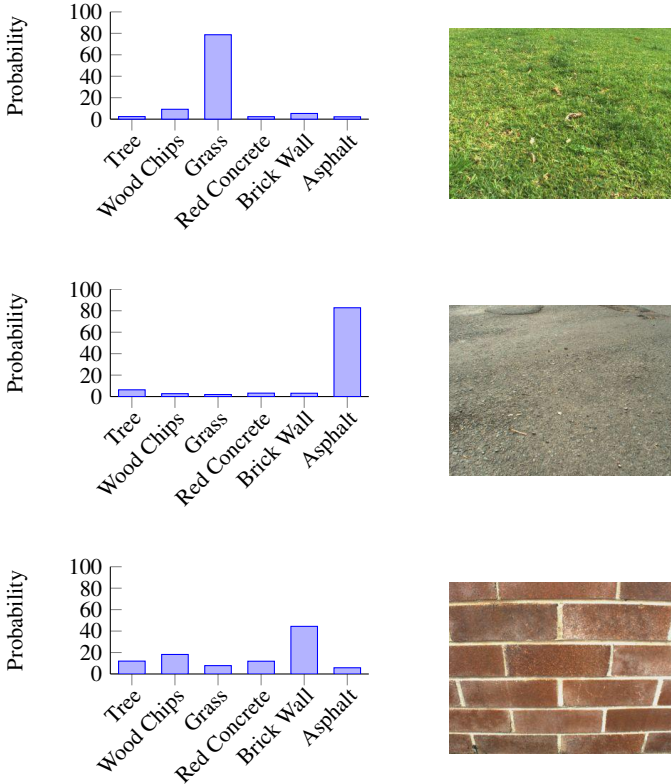
Synthetic Data				
Method	Clusters	No. raw points	No. clustered points	Duration (s)
AP no noise	10	2000	2000	60.61
AP with noise	35	2500	2500	63.43
MPAP no noise	10	2000	121	0.08
MPAP with noise	10	2500	122	0.08
Outdoor Data				
Method	Clusters	No. raw points	No. clustered points	Duration (s)
AP	6	1200	1200	8.79
MPAP	6	1200	79	1.84
Large-Scale Data – MPAP only				
Dataset	Clusters	No. raw points	No. clustered points	Duration (s)
Outdoor	10	31464	594	185
Indoor	10	48600	626	307

to manageable sizes, requiring just over one percent of the original amount of data to be retained. The durations shown are obtained by processing large batches of images to meta-point affinity propagation at once and then running the clustering. This is repeated until all images have been added. The final exemplars obtained in this way for the outdoor dataset is shown in Figure 5.

The results shown here from both synthetic and real images shows that meta-point affinity propagation obtains results that are comparable with affinity propagation but at a fraction of the computational cost. The added robustness of meta-point affinity propagation to noise makes this new method very appealing for use in robotics.

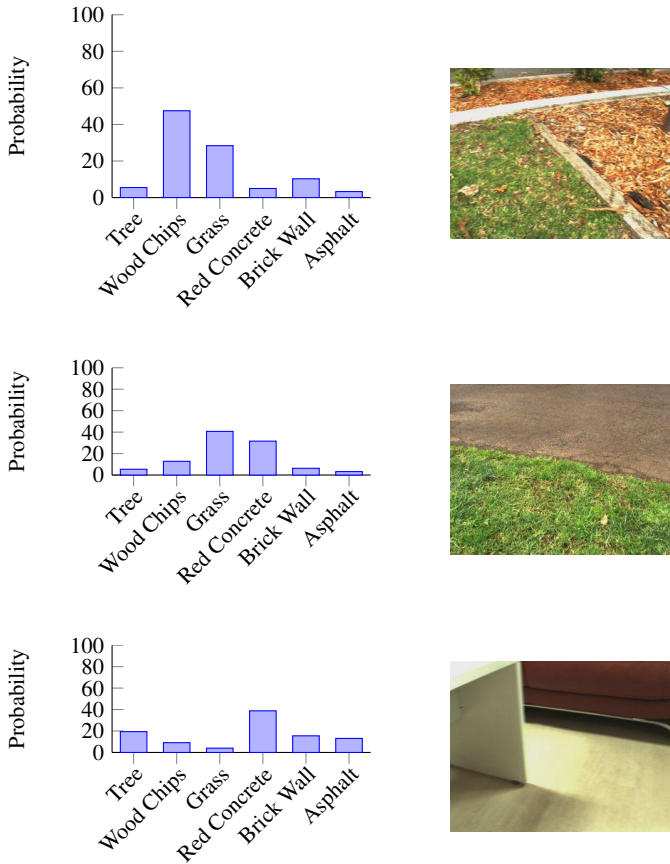
### 3.2 Probabilistic Cluster Assignments

To show the performance of our probabilistic cluster assignment we build a model out of the previously used 1200 image patches representing grass, trees, wood chips, asphalt, brick wall and red concrete. The exemplars obtained from the clustering are shown in Figure 4. We can see that for images containing the appearance of only a single cluster, as shown in Figure 6 the probability distribution is peaked around the corresponding cluster. When we look at the more interesting cases in Figure 7 where



**Fig. 6** Examples of probability distributions for single observations of uniform appearance. From top to bottom we have grass, asphalt and brick wall.

there is more than one cluster represented, as is the case in the top two images, we can see that they contribute the two largest peaks in the distribution. Finally, if we look at the last image in Figure 7 which obviously does not belong to any of the clusters we can see that the overall distribution is rather flat compared to the other cases. Compared to the hard assignment of affinity propagation which would have assigned this image to the “red concrete” class the probability distribution informs us that we can not trust the cluster assignment much.



**Fig. 7** Examples of probability distributions of single observations with mixed or unknown appearance. From top to bottom we have a scene with grass and wood chips, grass and red concrete and finally an indoor seating area which is not covered by the clustering.

## 4 Conclusion

In this paper we presented an extension to affinity propagation called meta-point affinity propagation, which allows us to cluster data in real-time and incrementally. Furthermore, we proposed a generic way to extract probabilistic cluster assignments for affinity propagation based methods. In experiments we show how meta-point affinity propagation obtains results similar to affinity propagation, however, at a much lower computational cost. We also show how the probabilistic cluster assignments can help to evaluate the clustering of a single observation or the entire clustering result. This combination of incremental real-time clustering with probabilistic

assignments will allow us to build meaningful models of the environment a robot operates in. Such models will adapt to changes in the environment and can also be used for example to guide the robot in the exploration of its surroundings.

## References

- [1] Amigó, E., Gonzalo, J., Artilles, J., Verdejo, F.: A comparison of Extrinsic Clustering Evaluation Metrics based on Formal Constraints. *Information Retrieval* (2008)
- [2] Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. *Journal of Machine Learning Research* (2003)
- [3] Cao, F., Ester, M., Qian, W., Zhou, A.: Density-Based Clustering over an Evolving Data Stream with Noise. In: *SIAM International Conference on Data Mining* (2006)
- [4] Ester, M., Kriegel, H., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: *Proc. of the International Conference on Knowledge Discovery and Data Mining* (1996)
- [5] Frey, B.J., Dueck, D.: Clustering by Passing Messages Between Data Points. *Science* (2007)
- [6] Giguere, P., Dudek, G., Prahacs, C., Plamondon, N., Turgeon, K.: Unsupervised Learning of Terrain Appearance for Automated Coral Reef Exploration. In: *Proc. of the Canadian Conference on Computer and Robot Vision* (2009)
- [7] Happold, M., Ollis, M., Johnson, N.: Enhancing Supervised Terrain Classification with Predictive Unsupervised Learning. In: *Proc. of Robotics: Science and Systems, RSS* (2006)
- [8] Kim, D., Sun, J., Rehg, J.M., Bobick, A.F.: Traversability Classification using Unsupervised On-line Visual Learning for Outdoor Robot Navigation. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation, ICRA* (2006)
- [9] Modayil, J., Kuipers, B.: Bootstrap Learning for Object Discovery. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS* (2004)
- [10] Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an algorithm. In: *Proc. of Advances in Neural Information Processing Systems* (2001)
- [11] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002)
- [12] Ruhnke, M., Steder, B., Grisetti, G., Burgard, W.: Unsupervised Learning of 3D Object Models from Partial Views. In: *ICRA* (2009)
- [13] Steinberg, D., Pizarro, O., Williams, S., Jakuba, M.: Dirichlet process mixture models for autonomous habitat classification. In: *OCEANS 2010 IEEE* (2010)

# 3-Dimensional Tiling for Distributed Assembly by Robot Teams

James Worcester, Rolf Lakaemper, and Mong-ying Ani Hsieh

**Abstract.** We consider the assembly of a three dimensional (3D) structure by a team of heterogeneous robots capable of online sensing and error correction during the assembly process. The automated assembly problem is posed as a general 3D tiling problem where the assembly components/tiles consist of various shapes and sizes. For a desired 3D structure, we first compute the partition of the assembly strategy into  $N_c$  sub-components that can be executed in parallel by a team of  $N_c$  assembly robots. To enable online error detection and correction during the assembly process, mobile robots equipped with visual depth sensors are tasked to scan, identify, and track the state of the structure. The objective is to enable online detection of missing assembly components and reassignment of these components to the team of assembly robots. We present the development of the planning, sensing, and control strategies employed and report on the experimental validation of these strategies using our multi-robot testbed.

## 1 Introduction

Distributed autonomous assembly of general two (2D) and three dimensional (3D) structures is a complex task requiring robots to have the ability to: 1) sense and manipulate assembly components; 2) interact with the desired structure at all stages

---

James Worcester · Mong-ying Ani Hsieh  
Drexel University,  
Mechanical Engineering & Mechanics,  
Philadelphia, PA, 19104  
e-mail: {jbw68, mhsieh1}@drexel.edu

Rolf Lakaemper  
Temple University,  
Computer & Information Sciences,  
Philadelphia, PA 19122  
e-mail: lakamper@temple.edu

of the assembly process; 3) satisfy a variety of precedence constraints to ensure assembly correctness; and 4) ensure the stability and structural integrity of the desired structure throughout the assembly process. While the distributed assembly problem represents a class of tightly-coupled tasks that is of much interest in multi-robot systems [1], it is also highly relevant to the development of next generation intelligent, flexible, and adaptive manufacturing and automation.

The execution of tightly-coupled tasks by multi-robot teams has mostly focused on cooperative grasping and manipulation [2, 3]. These works, however, do not address the challenges imposed by the need to satisfy specific precedence constraints during assembly to ensure correctness and stability of the desired structure. Existing approaches to distributed assembly can be broadly classified as micro/nano-scale or self-assembly and macro-scale assembly. In self-assembly, the objective is to devise local rules with global guarantees on assembly of stochastically interacting components [4, 5, 6]. Macro-scale assembly approaches include [7, 8]. In [7], assembly is achieved through a combination of robots with limited sensing and actuation capabilities and assembly components capable of storing and communicating location information with the robots. The focus of this work is on designing a set of consistent local attachment rules that ensure completeness and correctness of the assembly. In [9, 10], a workload partitioning strategy is presented to enable a team of robots to achieve parallel construction at the macro scale. The approach maintains a Voronoi decomposition of the structure based on the assembly robots' locations by minimizing the total difference in the masses of the assembly components in each cell.

In this work, we pose the 3D assembly problem as a three dimensional tiling problem where the team of robots is given a description of the desired structure. The structure is obtained by tiling, or connecting, a collection of assembly components of varying shapes and sizes. The assembly components attached to each based on predefined attachment sites and may differ depending on the geometry and size of the components. Given a desired 3D structure, we build on our previous work [11] to determine an allocation of the assembly task into subcomponents to enable parallel assembly by a team of autonomous robots. The objective is to determine the appropriate partition of the assembly task such that local attachment constraints, specified by the geometry of adjacent assembly tiles/components, and global precedence constraints, specified by structural stability requirements can be satisfied while minimizing workload imbalance among the team. While we have shown that the proposed partitioning strategy ensures the correctness of the distributed assembly strategy, the allocation is performed *a priori* and thus is unable to cope with execution time assembly errors, *e.g.*, incorrect and/or missed placements. To enable online error detection and correction of the assembly process, we consider the addition of a small number of mobile scanning robots capable of providing real-time visual feedback of the state of the structure during the assembly process. The objective is to enable the mobile scanning robots the ability to inform the assembly robots, in real-time, when an assembly tile/component has been incorrectly or not placed. Our main contribution is to experimentally show that a heterogeneous team



of assembly and scanning robots can improve the robustness and enable the online adaptation of any given assembly strategy.

The paper is organized as follows: We describe our methodology in Section 2. The experimental setup and results are presented in Section 3. The experimental insights and lessons learned are reported in Section 4. We conclude with directions for future work in Section 5.

## 2 Methodology

Let  $S_d$  denote the desired 3-D structure and  $M$  denote the number of distinct assembly components/tiles/blocks where  $t_i$  denote a component/tile/block of type  $i$ . We will assume that each tile of type  $i$  can be described as a general polytope and that the robots know the geometries of the different tile types a priori. Furthermore, every tile of type  $i$  will have a fixed number of attachment sites. These attachment sites are locations where tiles can mate and lock onto other tiles.

To assemble the structure  $S_d$ , we assume a team consisting of  $N_a$  assembly robots, and  $N_s$  scanning robots, equipped with visual depth sensors. The scanning robots will be tasked to sense the state of structure during the assembly process. The assembly robots will use the information provided by the scanning robots to ensure correct placement of their respective tiles  $t_i$ .

### 2.1 Task Partitioning

Given  $S_d$  and  $N_a$  assembly robots, we employ the approach described in [11] to determine an appropriate partitioning of the assembly of  $S_d$  into  $N_a$  tasks that can be executed in parallel. The objective is to arrive at a partition that maximizes parallel execution of the assembly while minimizing workload imbalance between the robots without violating any of the placement precedence constraints between the assembly components. The approach uses Dijkstra's algorithm with multiple starting nodes to generate a set of assembly tasks for each robot. This results in a partitioning of components of  $S_d$  such that each robot's task is composed of tiles that are closest to the starting node. The starting nodes are chosen to be equally spaced along the exterior. This initial allocation strategies is then improved with a second phase of node trading to yield a more balanced workload among the robots. The last step of this approach is the generation of an assembly sequence for each robot that minimizes the time a robot must wait for the placement of supporting tile by another robot. This is achieved by maximizing the time between a placement and the placements of any supporting tiles.

It is important to note that the approach described in [11] is a partitioning strategy that is executed *a priori* and generates a distributed assembly strategy for a team  $N_c$  robots given  $S_d$ , and  $\{t_1, \dots, t_M\}$ .

## 2.2 Visual Feedback

To provide information to the robots about the current state of the physical structure  $S_p$  as it is being assembled, we implement a feedback system using visual depth sensors. The objective is to use online sensing to compare  $S_p$  with the robot's internal model of the currently assembled structure  $S_a$ , and to provide control data to the building process, based on differences between  $S_p$  and  $S_a$ . To keep an updated representation of the state of  $S_p$ , we add a sensing robot to the system, which is equipped with a depth sensor  $K_p$ , in our case the Microsoft Kinect sensor. The robot constitutes the system for visual inspection (VI). The VI-robot is independent of construction robots. It runs a prioritized exploration algorithm, which aims to map and update the dynamically changing physical structure with priority on currently targeted building regions. The input to this system is the internal structure  $S_a$  (Figure 1(b)), which models what the system is expected to see from the physical structure  $S_p$  (Figure 1(a)), and the raw visual sensor data (3D point cloud, Figure 1(c)), the output is a state for every block  $t_i$  of the internal structure  $S_a$ , denoting if the tile is present, missing, or occluded (currently no visual information about the tile is available).

Before an assembly robot adds a part  $t_i$  to the physical structure  $S_p$ , it queries the VI-system, if the targeted region data is updated and  $S_p$  matches the expected state of  $S_a$ . For this comparison, we simulate a robot internal system containing  $S_a$  and a virtual Kinect sensor  $K_v$ . Using ray-tracing, we simulate a Kinect scan of  $S_a$ . The outcome of the simulated ray tracing is compared with the real scan of the physical Kinect  $K_p$  to compute the state for each tile  $t_i \in S_a$ . The following sections will explain the VI system in more detail.

### 2.2.1 Coordinate System Matching

To compare the outcome of the physical scan and the virtual scan, we must find  $P_v$ , the pose of  $K_v$  in the virtual system. If we let  $P_p$  denote the pose of  $K_p$  in the physical system, then  $P_v$  has to equal  $P_p$ . The positioning is performed in multiple steps consisting of an overhead localization system, a floor based correction, and an Iterative Closest Point (ICP) alignment. In the following we use a right handed coordinate system. The horizontal plane is described by  $(x, z)$ , height is described by the  $y$ -axis.

First, an overhead localization system gives an estimate of the horizontal  $(x, z)$  position of  $K_p$ . This includes the  $(x, z)$  coordinates as well as the yaw  $\alpha$ , i.e. the rotation angle around the  $y$ -axis. The overhead localization system is provided by a network of cameras with errors in  $(x, y)$  below 5 cm and angular errors in  $\alpha$  of  $\pm 10$  degrees.

To complete  $P_p$ , the missing pose-parameters  $y$  (the Kinect's height) and  $\beta, \gamma$  (pitch and roll, i.e. rotation around  $x$  and  $z$  axis respectively) are determined by a floor-based correction. We perform floor detection in the point cloud  $C_p$  resulting from the physical scan. Since the floor in the physical system defines the  $x-z$  plane, a transformation  $T_f$ , which aligns the floor's normal with the  $y$  axis of the virtual

system completes the estimate of  $P_p$ . We compute  $T_f$  by regression of the floor points to their projections in the  $x - z$  plane (point to plane correspondence). As such, we note that  $T_f$  has no  $y$ -rotation component, i.e. the Kinect's yaw, as previously determined by the overhead system, is not altered by an otherwise ambiguous rotation. In addition, we re-compute the translational part of  $T_f$  in the  $x - z$  plane, such that only the vertical position of  $K_v$  is affected.

While there are errors in the localization and ground plan position estimates and noise in  $C_p$ , they provide a sufficiently good starting point for an Iterative Closest Point (ICP) alignment [12]. We use  $P_p$  as a starting estimate for  $P_v$ , therewith we also transform  $C_p$ : we set  $C_p \leftarrow T_f C_p$ . We perform a 6D (3 location parameters, 3 directional parameters) point to plane ICP, with the goal to align  $C_p$  to  $S_a$ . ICP is a well known technique in robotics and computer vision, successfully applied to align (3D) point clouds, especially for robot mapping [13]. Given two point clouds  $C_1$  and  $C_2$ , it finds, in an iterative way, a (locally) optimal transformation  $T_c$  that minimizes the squared sum of distances between points in  $C_1$  and their iteratively re-determined closest neighbors in  $T_c C_2$ . ICP is known robust and fast as long as a good starting estimate of the point-poses is provided. In practice, the previously described steps to compute  $P_p$  proved to be sufficient as a starting point.

We perform a fast point-to-plane ICP version:  $C_1 \subset C_p$  originates from the physical scan  $C_p$ , and consists of a subset of points, being candidates for points belonging to  $S_p$ .  $C_2$  is iteratively generated as the projection points of  $C_1$  onto the virtual structure  $S_a$ . Internally,  $S_a$  is represented as a set of planar patches, describing the geometry of the tiles  $t_i$ . Storing the tiles of  $S_a$  together with a hierarchy of axis aligned bounding boxes (AABB) allows for fast computation of the projections of  $C_1$  onto  $S_a$ . The hierarchy is given naturally: we store an AABB for the structure  $S_a$ , for each tile  $t_i \in S_a$  and each planar patch  $p \in t_i$ . Using this hierarchy of bounding boxes,  $C_1$  results in a relatively small subset of  $C_p$ . In addition, we omit points that belong to the floor, as determined by the floor detection step. A single Kinect scan in hi-res (640 x 480) contains about 300000 points, the typical point cloud of candidates describing reflections from the structure  $S_p$ , after filtering, typically reduces the number of points to less than 10000. We limit our ICP to a maximum of 10 iterations. ICP results in  $T_{ICP}$ , an accumulated rotation and translation to align  $C_1$  to  $S_a$ . When we apply  $T_{ICP}$  to  $P_p$ , this reduces pose errors from the former computation. We set  $P_v = T_{ICP} P_p$ . See Figure 1(d) for the result of this step.

ICP not only provides the pose  $P_v$  of  $K_v$  in the virtual system, but also the projection points  $\tilde{C}_1$  of  $C_1$  onto the structure  $S_a$ . We therefore compute a connection between the physical point cloud and the virtual structure. In fact, for each tile  $t_i$  in  $S_a$ , we can determine how many projected points, called physical support points  $s_i \subset \tilde{C}_1$  of  $t_i$  are projected on  $t_i$ . The set of support points tells us, if a tile  $t_i$  of the virtual structure  $S_a$  is seen in the physical world. A tile  $t_i$  with a sufficient number of support points is present. However, the converse argument is not valid, since a tile without support could be physically present, but occluded. The next step, ray tracing, solves this problem.

### 2.2.2 Ray Tracing

This step determines the set of reflection points of a scan of the virtual Kinect  $K_v$  with pose  $P_v$  of the virtual building  $S_a$ . We position the virtual Kinect at pose  $P_v$  and simulate a ray-tracing using the Kinect's optical properties (resolution, view angles). Again, since the virtual building  $S_a$  is stored using planar polygons and a hierarchy of axis aligned bounding boxes, the ray intersection can be performed very efficiently. For each ray, we compute the closest intersection with a tile  $t_i$  from the Kinect, resulting in a virtual point cloud  $C_v$ . For each point in  $C_v$ , we know the supported tile  $t_i$  (i.e. the tile the generating ray intersected with). Ray tracing determines the support sets in the virtual system, that is, the support that we *should* see under the condition  $S_a = S_p$ . In contrast,  $\bar{C}_1$  determines the real support, i.e. the support we *do* see. See Figure 1(e) for the result of the ray tracing step.

### 2.2.3 Tile Classification

Differences in support from  $C_v$  and  $\bar{C}_1$  respectively determine if a tile is classified as present, missing, or occluded.

For every tile  $t_i$ , denote the number of physical and virtual support points by  $p_i$  and  $v_i$  respectively. Define  $r$  as the minimum ratio between  $p_i$  and  $v_i$ ,  $r = \min(\frac{v_i}{p_i}, \frac{p_i}{v_i})$ ,  $t_r$  is a threshold value for this ratio, set to 0.7. For our purpose, it proved to be sufficient to only compare the number of support points of each tile, i.e., we are not explicitly using any geometric differences. Support below a threshold of 100 points is set to 0.

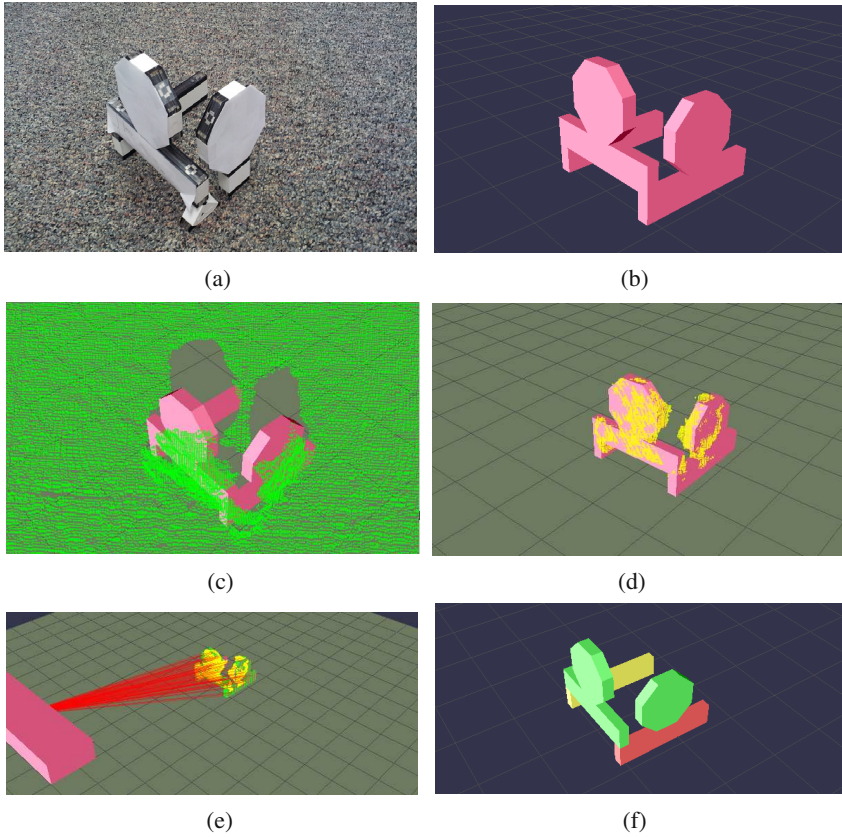
The state of a tile  $t_i$  of  $S_a$  reflects its presence in the physical structure  $S_p$ . We determine this state as follows:

- $v_i = 0 \Rightarrow$  the tile is occluded.
- $v_i \neq 0$  and  $p_i = 0 \Rightarrow$  the tile is missing.
- $v_i \neq 0$  and  $p_i \neq 0$  and  $r \leq t_r \Rightarrow$  the tile is missing. This case implicitly tests geometric differences.
- $v_i \neq 0$  and  $p_i \neq 0$  and  $r > t_r \Rightarrow$  the tile is present.

If the state of a tile  $t_i$  is “missing”, the building robots have to adjust. “Present” signals, that  $t_i \in S_a$  and  $t_i \in S_p$  at the expected position, the building process can continue. If a tile is in state “occluded”, the VI-robot has to re-scan the building from a different position before the building process can proceed. See Figure 1(f) for an example.

## 2.3 Online Error Correction

The VI-robot(s) is responsible for assigning the replacement of any missing tiles it discovers. It does this by managing an auction for each block that should have been placed but is absent. Each assembly robot sends a message to the VI-robot(s) after



**Fig. 1** Vision Feedback System. (a) Physical structure  $S_p$  (b) Virtual structure  $S_a$ . Note that  $S_p$  and  $S_a$  differ in this example: the long rectangle (front right) in the virtual structure is not present in  $S_p$ , it is replaced by a small cube. In the building process, this is an example of a missing/incorrect tile. (c) Green points show raw input data  $C_p$  from the physical Kinect sensor  $K_p$ . The Kinect's pose  $P_v$  in the virtual system was determined by the overhead positioning system. This figure shows the coordinate matching before floor based correction and ICP (d) After floor-based correction, ICP and candidate filtering: the yellow dots show the pose-corrected raw kinect data  $C_1$ , aligned to the virtual building. Points of the original raw data which were unlikely to support the structure were removed (floor- and bounding box based filtering). (e) Ray tracing: the red lines show some rays of the simulated Kinect  $K_v$  scan to determine the visibility of tiles  $t_i \in S_a$ . Yellow dots show the aligned real data  $C_1$ , green dots the result of the virtual scan  $C_v$ . The difference in support for each tile from yellow and green dots (real/virtual support points) is used to determine the state of each tile. (f) Result: Green tiles: present in  $S_a$  and  $S_p$ . Yellow tile: occluded (please note that this tile is occluded from view point  $P_v$ , as seen in (e). Here we rotated the view to make it visible). Red tile: Missing in  $S_p$ . The vision system correctly identified the front right rectangle as missing.

placing a tile. The VI-robot monitors these messages to maintain a state vector  $q$ , where  $q_i$  is 1 if the block has been placed and 0 otherwise. After each placement, the VI-robot reports a sensing vector  $q_j^s$ , where  $q_j^s$  is 1 if the block is definitely present,  $-1$  if it is missing, and 0 if the presence or absence of the block cannot be determined. Then, if  $q_i * q_j^s = -1$ , a block that a robot claims to have placed is determined to be missing. Once the error has been detected, the scanning robot sends a message to inform the assembly robots that the block is missing and asks for bids to determine which robot will replace the missing block. Each robot then constructs a bid based on the following criteria:

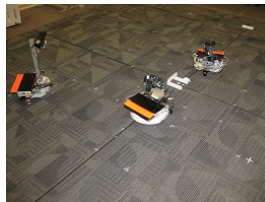
$$b_i = w_i - A * c_i + B * d_{ij}, \quad (1)$$

where  $b_i$  is the bid of the  $i^{\text{th}}$  robot,  $w_i$  is the remaining workload of the  $i^{\text{th}}$  robot,  $c_i$  is the number of blocks still to be placed that are directly supported by the missing block, and  $d_{ij}$  is the distance between the missing block and the  $i^{\text{th}}$  robot's cache. The constants  $A$  and  $B$  are weights that can be optimized experimentally.

### 3 Experimental Validation

#### 3.1 Setup

To evaluate the performance of the proposed online error detection and correction strategy, we implemented the proposed distributed assembly strategy on our multi-robot assembly testbed. The testbed consists of two mini-mobile manipulators (M3 robots), or  $N_c = 2$ , shown in Figure 2, each equipped with an iRobot Create base, a Crustcrawler 5 DOF arm, 802.11b wireless communication, and a Hokuyo URG laser range finder (LRF). The LRF was used by the assembly robots to detect, pick, and place the tiles during the assembly process. In addition to the two M3 robots, the testbed included one scanning robot equipped with a iRobot Create base and a Microsoft Kinect visual depth sensor. Overhead localization for the robots was provided using two visual cameras.

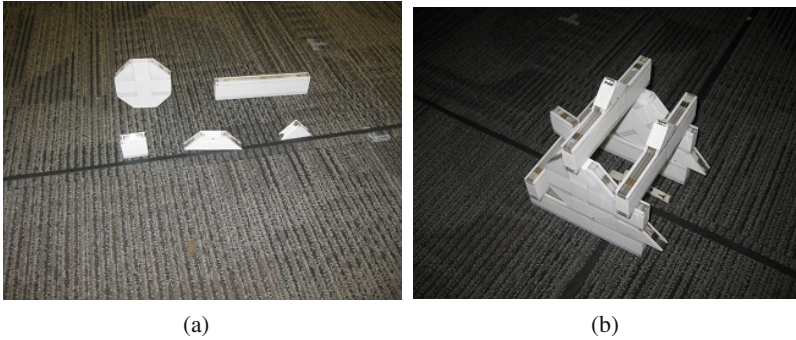


**Fig. 2** Team of two assembly robots and one VI-robot with a partially completed structure



Each robot was given the global position of the structure's center and the positions of their respective parts cache. The assembly parts were plastic tiles of various shapes and sizes (side lengths from 4 – 17 cm), each with a given set of magnetic attachment sites (see Figure 3(a)). Each robot was assigned their respective assembly plans determined by [11]. The assembly plans consisted of a list of tile identifiers in the computed assembly order. Distributed implementation of the plan was achieved by encoding the immediate supports for each component in the plan to ensure robots wait for the placement of a missing support tile by another robot before placing their parts.

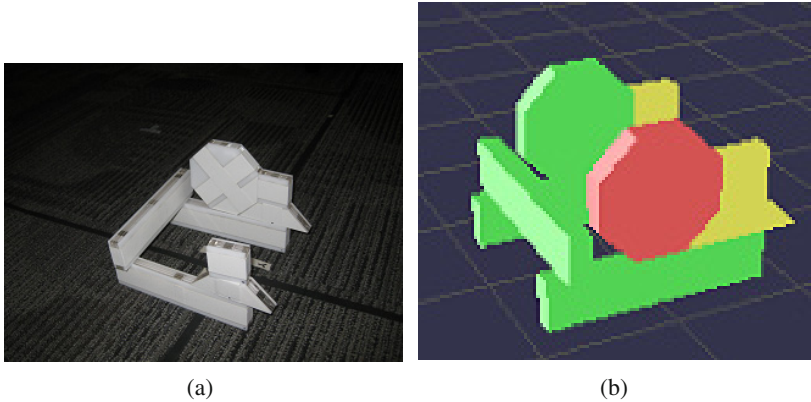
The assembly tiles were grouped by type and placed in predefined locations in the workspace. The idea is to have a separate parts cache for each tile type. In our experiments, we considered the distributed assembly of 3D structures composed of 14 tiles with 5 distinct tile types. Figure 3(b) shows the desired structure for the experiment. To simulate missed placements, random assembly tiles were removed at various times during the assembly process.



**Fig. 3** (a) Sample assembly tiles. (b) Desired structure to be assembled.

## 3.2 Results

Fourteen experimental trials were run on the scanning robot for the desired structure shown in Figure 3(b). During each trial, one or more random assembly tiles were removed at different parts of the assembly process. Figure 4 shows the results of one of the experimental trials where the missing tile was successfully detected by the scanning robot. Out of twenty-two removed blocks, the scanning robot was able to successfully detect twelve of the missing tiles and reported undetermined for the other ten. There were no false positives during these trials, and only one false negative where a tile was reported as missing when it was actually present. The smaller tiles (square and triangle) were always reported as undetermined, while the larger tiles were always detected as missing after they had been removed in these trials.



**Fig. 4** (a) Tile removed. (b) Missing tile reported by the scanning tile.

Table 1 summarizes the assembly partition obtained at the start of an experimental trial for each robot. The tiles allocated to each robot are shown in the order in which they are supposed to be placed. Table 2 shows the updated assembly allocation as tiles are removed during the experiment, including the workload reallocation after the detection of errors.

**Table 1** Initial Allocation for the 3D Structure in Fig. 3(b)

<b>Robot 1</b>	<b>Tile ID</b>	<b>Robot 2</b>	<b>Tile ID</b>
Long Rectangle	3	Trapezoid	7
Trapezoid	6	Octagon	5
Octagon	4	Square	8
Square	9	Square	10
Long Rectangle	11	Long Rectangle	12
Triangle	13	Triangle	14

## 4 Experimental Insights and Lessons Learned

The execution of complex tasks by a team of heterogeneous robots in a complex and dynamic environment with limited resources poses significant challenges. Most existing assembly strategies do not explicitly address the impact of sensing and actuation noise on the performance of a team of autonomous robots tasked to assemble complex three dimensional structures in an actual physical space. In our work, we consider the real-time on-board sensing requirements necessary for online adaptation of any distributed assembly strategy.



**Table 2** Allocation After Detection of a Missing Tile

Robot 1	Tile ID	Robot 2	Tile ID
Long Rectangle	3	Trapezoid	7
		Removed tile	7
Trapezoid	6		
Octagon	4	Trapezoid	7
Removed tile	4		
Square	9	Octagon	5
		Removed tile	5
Octagon	4	Square	8
		Removed tile	8
		Octagon	5
Long Rectangle	11	Square	10
Triangle	13	Long Rectangle	12
Removed tile	13		
Square	8	Triangle	14
		Triangle	13

In our experimental setup, we considered two types of real-time on-board sensing: 1) the ability to localize the individual assembly tiles for pick-up and placement by the assembly robots, and 2) the ability to determine the state of the assembly structure during the entire assembly process. In both cases, the relative small size of the assembly tiles in relation to the sensing and actuation precision of the actuators and sensors used in the system posed significant engineering challenges. However, the ability to overcome these limitations at the small scale suggests that one can be more confident in the performance of the algorithms when employed on larger full scale systems.

## 5 Future Work

In this work, we presented a distributed 3D assembly strategy with online visual feedback to enable realtime error detection and correction. Our approach enables the online verification and adaptation of general 3D assembly strategies. An immediate direction for future work is to improve the visual feedback system to provide more detailed assessment of the state of the assembly structure. In particular, the reduction of false negatives by visually inspecting the structure via different viewpoints. A second direction for future work is to extend the visual feedback system to enable identification of incorrect assembly placements as well as missing tiles. Finally, we would like to enable online adaptation of the assembly strategy in the presence of incorrect tile placements. This, in conjunction with the visual feedback system, can significantly increase the robustness and adaptability of the system.

**Acknowledgements.** James Worcester and M. Ani Hsieh gratefully acknowledge the support of NSF grant CNS-1143941. Dr. Lakaemper is sponsored by the grant ARRA-NIST-10D012 of the National Institute of Standards and Technology (NIST).

## References

1. Chaimowicz, L., Sugar, T., Kumar, V., Campos, M.F.M.: An Architecture for Tightly Coupled Multi-Robot Cooperation. In: Proc. IEEE Int. Conf. on Rob. & Autom., Seoul, Korea, pp. 2292–2297 (May 2001)
2. Mataric, M.J., Nilsson, M., Simsarian, K.: Cooperative Multi-Robot Box-Pushing. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1995), Pittsburgh, Pennsylvania, pp. 556–561 (August 1995)
3. Fink, J., Hsieh, M.A., Kumar, V.: Multi-robot manipulation via caging in environments with obstacles. In: Proc. IEEE International Conference on Robotics and Automation (ICRA 2008), Pasadena, CA, pp. 1471–1476 (May 2008)
4. Klavins, E.: Programmable Self-Assembly. *Control Systems Magazine* 24(4), 43–56 (2007)
5. Evans, W.C., Mermoud, G., Martinoli, A.: Comparing and modeling distributed control strategies for miniature self-assembling robots. In: Proc. of the 2010 Int. Conf. on Robotics and Automation (ICRA 2010), Anchorage, AK, pp. 1438–1445 (May 2010)
6. Matthey, L., Berman, S., Kumar, V.: Stochastic Strategies for a Swarm Robotic Assembly System. In: Proc. 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan, pp. 1953–1958 (May 2009)
7. Werfel, J., Nagpal, R.: Three-dimensional construction with mobile robots and modular blocks. *International Journal of Robotics Research* 27(3-4), 463–479 (2008)
8. Heger, F., Singh, S.: Robust robotic assembly through contingencies, plan repair and re-planning. In: Proceedings of ICRA 2010 (May 2010)
9. Yun, S.K., Schwager, M., Rus, D.: Coordinating construction of truss structures using distributed equal-mass partitioning. In: Proc. of the 14th International Symposium on Robotics Research, Lucerne, Switzerland (August–September 2009)
10. Yun, S.K., Rus, D.: Adaptation to robot failures and shape change in decentralized construction. In: Proc. of the Int. Conf. on Robotics & Automation (ICRA 2010), Anchorage, AK USA, pp. 2451–2458 (May 2010)
11. Worcester, J., Rogoff, J., Hsieh, M.A.: Constrained task partitioning for distributed assembly. In: Proc. 2011 Int. Conf. on Intelligent Robots and Systems, IROS 2011 (September 2011)
12. Besl, P., McKay, N.: A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 239–256 (1992)
13. Nüchter, A., Lingemann, K., Hertzberg, J., Surmann, H.: Heuristic-Based Laser Scan Matching for Outdoor 6D SLAM. In: Furbach, U. (ed.) KI 2005. LNCS (LNAI), vol. 3698, pp. 304–319. Springer, Heidelberg (2005)

# JediBot – Experiments in Human-Robot Sword-Fighting

Torsten Kröger, Ken Oslund, Tim Jenkins, Dan Torczynski,  
Nicholas Hippenmeyer, Radu Bogdan Rusu, and Oussama Khatib

**Abstract.** Real-world sword-fighting between human opponents requires extreme agility, fast reaction time, and dynamic perception capabilities. In this paper, we present experimental results achieved with a 3D vision system and a highly reactive control architecture which allows a robot to sword fight against human opponents. An online trajectory generator is used as an intermediate layer between low-level trajectory-following controllers and high-level visual perception. This architecture allows robots to react nearly instantaneously to the unpredictable human motions perceived by the vision system as well as to sudden sword contacts detected by force and torque sensors. Results show how smooth and highly dynamic motions are generated on-the-fly while using the vision and force/torque sensor signals in the feedback loops of the robot motion controller.

## 1 Introduction

Born as a class project [1, 2], the idea of the “JediBot” is a robot that performs sword fighting against a human opponent. The basic requirements for implementing a sword-fighting robot are (i) a reliable visual perception system that detects motions of the opponent and its sword, (ii) a reactive motion generation and control system for the robot to be able to immediately react to the opponent’s motion, and (iii) compliant and reactive motion control capabilities for physical human-robot interaction. Furthermore, appropriate attack and defense strategies are required that make use of the three mentioned aspects.

---

Torsten Kröger · Ken Oslund · Tim Jenkins · Dan Torczynski ·  
Nicholas Hippenmeyer · Oussama Khatib  
Artificial Intelligence Laboratory at Stanford University, Stanford, CA 94305-9010, USA  
e-mail: tkr@stanford.edu

Radu Bogdan Rusu  
Open Perception, Inc. , 68 Willow Road, Menlo Park, CA 94025, USA

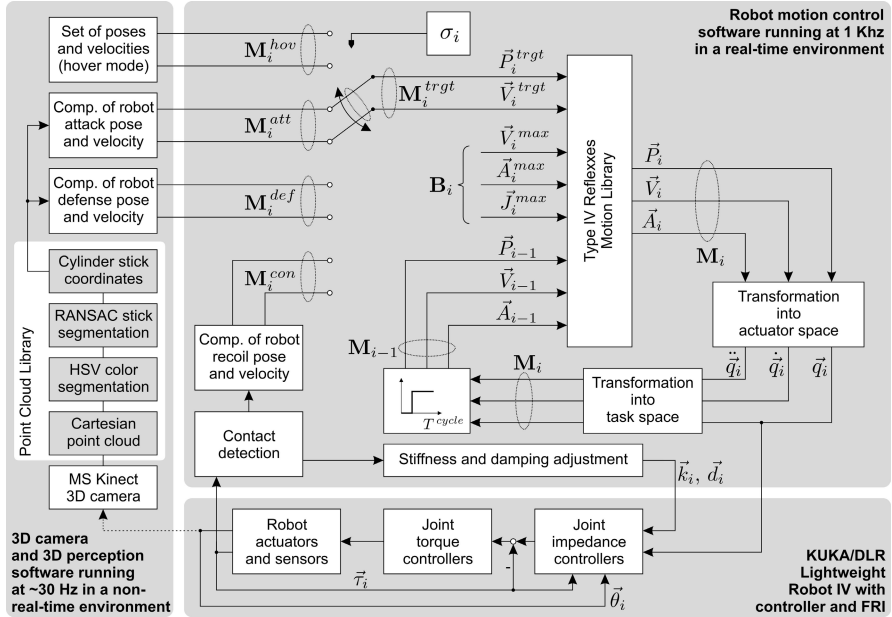
This paper describes (a) the hardware and (b) software system of “JediBot”, (c) experimental results of human-robot interaction during sword-fighting, and (d) how readers can replicate the system and run experiments with their own system set-up. Figure 1 illustrates the suggested setup: A KUKA/DLR Lightweight Robot [3] is equipped with a foamed wooden sword, and a 3D camera (MS Kinect, [4]) is used to perceive the human opponent and its sword. The two most challenging parts that make use of recent research outcomes are *3D Visual Perception* and *Online Trajectory Generation*.



**Fig. 1** Reactive motion generation and control during physical human-robot interaction with “JediBot”

**3D Visual Perception.** Based on underlying technology from PrimeSense [5], 3D cameras such as Microsoft Kinect [4] have started to simplify 3D visual perception procedures in robotics. Our approach to detect the opponents posture and sword in unstructured environments operates on 3D data. One of the most popular descriptors for 3D data is the Spin-image presented by Johnson *et al.* [6], which is a 2D representation of the surface surrounding a 3D point and is computed for every point in the scene. Two of the key technologies for online segmentation operations are provided by FLANN (Fast Library for Approximate Nearest Neighbors, [7]) and RANSAC (Random Sample Consensus, [8]). All required 3D perception methods and algorithms for this task are provided by PCL (Point Cloud Library, [9,10]). How these methods are applied is, for instance, shown in [11].

**Online Trajectory Generation.** Reactive online motion generation is required to immediately react to motions of the human opponent and to contacts between the robot’s and the opponent’s sword. Broquère *et al.* [12] published a method that uses an online trajectory generator for an arbitrary number of independently acting degrees of freedom. The approach is very similar to the one of Liu [13] and is based on the classic seven-segment acceleration profile. The work of Haschke *et al.* [14] presents an online trajectory planner in the very same sense as [15] does. The proposed algorithms generate jerk-limited trajectories from arbitrary states of motion. All required motion generation concepts for this task are provided by the Reflexxes Motion Libraries [16, 17].



**Fig. 2** Overall control scheme of “JediBot”. It can be subdivided into three units: the visual perception hardware and software (left), the real-time motion generation and control unit (top), and the KUKA/DLR Lightweight Robot IV (bottom).

## 2 Technical Approach

Figure 2 gives an overview of the system’s hardware and software components, which consist of three main blocks:

- hardware and software for visual detection of the human opponent its sword,
- a real-time motion generation and control unit, and
- the robot arm with a sword mounted to its hand (cf. Fig. 1).

The following three subsections describe these components, and Sec. 3 discusses the interplay between them.

### 2.1 Human Opponent and Sword Detection

The hardware component of the JediBot vision system is a Microsoft Kinect sensor which provides both a standard RGB color image and a stereo camera derived depth image at up to 30 fps and  $640 \times 480$  pixels in resolution. The software component is based on the Point Cloud Library (PCL) [9]. At startup the system goes through an automatic calibration routine, in which the sword is placed in two perpendicular positions which are known in the robot’s frame and then detected in the Kinect



**Fig. 3** Screen shots showing the precision and robustness of the RANSAC segmentation [8] of the human’s sword (yellow) in different configurations

reference frame. This is done using the same algorithm that detects the opponent’s sword (described below).

Multiple measurements are taken at each position and averaged to reduce noise. Using this data, an affine transformation between the robot and Kinect reference frames is calculated using functions built into PCL. Once the calibration routine is complete, the entire point cloud is transformed into the robot frame as soon as it is captured, so that the results of all further processing are automatically in the robot reference frame.

To detect the sword, the point cloud is first filtered based on depth to eliminate background points. Based on the calibration data, an HSV (hue, saturation, value) color segmentation is performed to select points, which are approximately the same color as the sword. Then a RANSAC algorithm is used to fit a line in three dimensional space to determine end points of the most stick-like object. Finally, the length of the detected sword is compared to the expected length, and the detection discarded if it is too long or too short to further reduce false positives. Figure 3 illustrates three sample results. The sword speed, calculated from its movement between two or more successive frames, is also used because it proved to be the most reliable method of detecting when an opponent begins their swing. This processing pipeline is split across multiple threads to improve performance.

## 2.2 Online Trajectory Generation

The Online Trajectory Generation algorithms of [16] are contained in the Reflexxes Motion Libraries [16, 17]. They let us compute synchronized motions for  $N$  degrees of freedom from any state of motion  $\mathbf{M}_{i-1}$  at instant  $T_{i-1}$  represented by position, velocity, and acceleration vectors with  $N$  elements each,

$$\mathbf{M}_{i-1} = (\vec{P}_{i-1}, \vec{V}_{i-1}, \vec{A}_{i-1}). \quad (1)$$

The algorithm will transfer the system from this state of motion into the desired target state

$$\mathbf{M}_i^{trgt} = (\vec{P}_i^{trgt}, \vec{V}_i^{trgt}, \vec{0}) \quad (2)$$

under consideration of the current maximum values for velocity, acceleration, and jerk,

$$\mathbf{B}_i = (\vec{V}_i^{max}, \vec{A}_i^{max}, \vec{J}_i^{max}) . \quad (3)$$

The output values of the algorithm define the desired state of motion  $\mathbf{M}_i$  at  $T_i = T_{i-1} + T^{cycle}$ , where  $T^{cycle}$  is the value of the control cycle time. After the transformation into actuator space, the values of joint position  $\vec{q}_i$  and its derivatives are used as command variables for the joint controller.

The values of  $\vec{A}_i^{max}$  are permanently updated using the forward dynamic model, for which a constant maximum torque vector  $\vec{\tau}^{max}$  is assumed:

$$\vec{A}_i^{max} = \vec{f}(\vec{q}_i, \vec{q}_i, \vec{\tau}^{max}) . \quad (4)$$

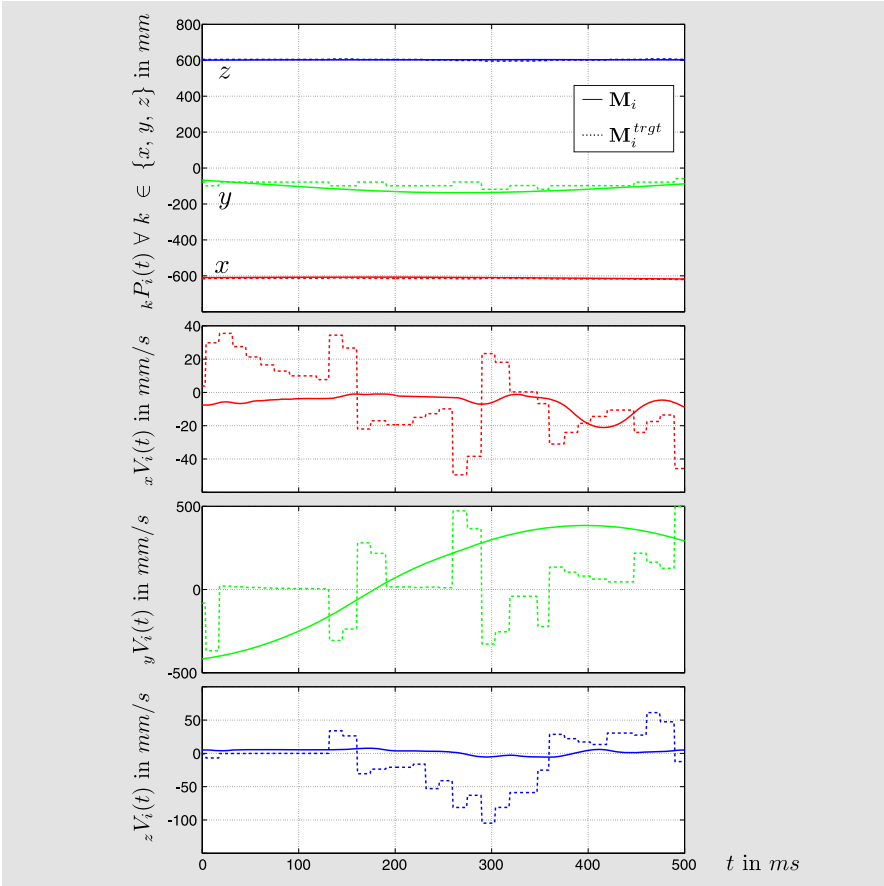
As we will learn in Sec. 3, this will allow for the use of discontinuous input signals  $\mathbf{M}_i^{trgt}$  of a switched system while permanently guaranteeing steady, jerk-limited, synchronized robot motions for all  $N$  degrees of freedom:

$$\begin{aligned} \forall n \in \{1, \dots, N\} : \\ |{}_n P_i - {}_n P_{i-1}| \leq {}_n V_{i-1} T^{cycle} + \frac{1}{2} {}_n A_{i-1} (T^{cycle})^2 \pm \frac{1}{6} {}_n J_i^{max} (T^{cycle})^3 \wedge \\ |{}_n V_i - {}_n V_{i-1}| \leq {}_n A_{i-1} T^{cycle} \pm \frac{1}{2} {}_n J_i^{max} (T^{cycle})^2 \wedge \\ |{}_n A_i - {}_n A_{i-1}| \leq {}_n J_i^{max} T^{cycle} \wedge |{}_n V_i| \leq {}_n V_i^{max} \wedge |{}_n A_i| \leq {}_n A_i^{max} . \end{aligned} \quad (5)$$

Because of this property, the trajectory generation can guarantee that it will send control commands to the robot at perfectly regular intervals (a requirement for the robot to operate properly) even if it receives target states of motion at highly irregular intervals. Thus, the image processing hardware and software does *not* necessarily have to be real-time capable, and its interface with the real-time trajectory generator can be very simple, consisting only of sending the desired states of motion whenever they become available. (cf. Fig. 2). A sample trajectory for three degrees of freedom is shown in Fig. 4.

### 2.3 Robot Hardware

A *KUKA Light-Weight Robot IV* [3, 18] was controlled through the *Fast Research Interface* [19, 20] with a control cycle time of  $T^{cycle} = 1\text{ms}$ . The simplicity of our setup is based on the control scheme of Fig. 2. Its three components as well as their interfaces were implemented with a focus on overall computational efficiency. The robot end-effector only consists of a foamed wooden sword (cf. Fig. 1).



**Fig. 4** Position and velocity progressions of  $\mathbf{M}_i$  and  $\mathbf{M}_i^{trgt}$  during a motion in defense mode. While  $\mathbf{M}_i^{trgt}$  is only updated sporadically, the computed trajectory is provided at a rate of 1 KHz, such that the robot reacts instantaneously to sensor signals.

### 3 Sword Fighting Strategies

In order to implement appropriate sword fighting strategies using the three components described in the previous section, four control modes have been implemented: defense, attack, contact, and hover.



**Defense** (*def*) The robot defends itself by attempting to block attack swings of the human. The pose of the human’s sword, obtained by the vision system, is used to determine the desired position and orientation of the robot’s sword. When the human’s sword is within a specified defense range and the velocity of that sword in the direction of the robot exceeds a certain threshold value, the robot moves to the blocking position.

In the blocking position, the midpoint of the robot’s sword is placed on the defense line, which is a line between the midpoint of the human’s sword and a predefined defense point which represents the center of the robot (cf. Fig. 5). The robot’s sword is oriented such that it is orthogonal to both the defense line and the human’s sword. Mathematically speaking, the robot’s sword is oriented parallel to the direction of the cross product between the directions of the human’s sword and the defense line. As the human’s sword moves in towards the robot’s sword, the robot will push its sword proportionally outward along the defense line until they meet.

**Attack** (*att*) When the opponent’s sword is beyond the defense range, the robot executes periodic attacking motions, swinging toward the opponent from a randomly selected direction.

**Contact** (*con*) Detection of collisions between the human and robot swords is done using the torque sensors in each of the robot’s joints. If the human blocks an attack motion, the robot detects this through the increased torque in its joints, and it immediately recoils, returning to the defensive position. This mode is activated in the same control cycle contact is detected.

**Hover** (*hov*) If no human sword is detected, the robot enters idle mode, where the bot sword just hovers around a specified position and orientation until a human opponent’s sword is detected.

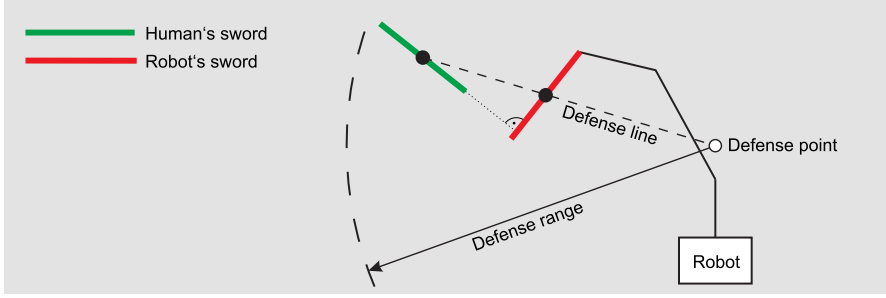
In the scheme of Fig. 2, the discrete value of

$$\sigma_i \in \{def, att, con, hov\} \quad (6)$$

selects the signal source of  $\mathbf{M}_i^{trgt}$  that is used to feed the online trajectory generator:

$$\mathbf{M}_i^{trgt} = \begin{cases} \mathbf{M}_i^{def} & \text{if } \sigma_i = def \\ \mathbf{M}_i^{att} & \text{if } \sigma_i = att \\ \mathbf{M}_i^{con} & \text{if } \sigma_i = con \\ \mathbf{M}_i^{hov} & \text{if } \sigma_i = hov \end{cases} \quad (7)$$

$\sigma_i$  can change spontaneously based on the fight strategy and on sensor signals. For instance, if contact is detected,  $\sigma_i = con$  will be applied in the same control cycle. How the value of  $\sigma_i$  is selected and how the components of Fig. 2 interact, will be described in Sec. 4.



**Fig. 5** During *defense mode*, the midpoint of the robot's sword (red) lies on the defense line, which is a line between the midpoint of the human's sword (green) and a predefined defense point

In order to provide a compliant behavior of the end-effector while both swords are in contact, the stiffness vector  $\vec{k}_i$  is adjusted. Assuming a contact was detected at a time instant  $T_c$ , the following function is used:

$$\vec{k}_i = \begin{cases} \vec{k}^{min} + \left( \frac{T_i - T_c}{T^{recover}} \right) (\vec{k}^{max} - \vec{k}^{min}) & \text{if } T_i \leq T_c + T^{recover} \\ \vec{k}^{max} & \text{if } T_i > T_c + T^{recover} \end{cases} \quad (8)$$

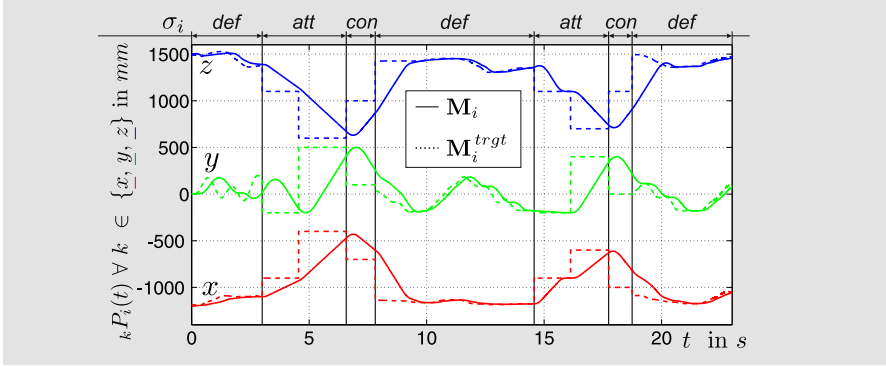
$T^{recover}$  is the time until the maximum stiffness  $\vec{k}^{max}$  is achieved again. As long as the contact detection indicates contact,  $T_c$  is set to  $T_i$ . The damping vector  $\vec{d}_i$  remains constant.

## 4 Robot Sword Fighting Experiments and Results

The key aspect for achieving a good human-robot interaction behavior during the experiments is an appropriate motion strategy. The simplicity and very high reactivity of the proposed control scheme allow the system to freely realize different strategies. Based on the usage of multiple sources for desired states of motion  $\mathbf{M}_i^{trgt}$ , the system switches between them depending on the current state, strategy, and situation. This allows the robot make *instantaneous* use of sensor signals.

For example, if contact is detected at an instant  $T_i$  ( $\sigma_i = con$ ),  $\mathbf{M}_i^{con}$  and an adjusted value of  $\vec{k}_i$  will be applied in the same control cycle already. After the re-coiling motion is completed, the robot can either attack again or switch to defend mode.

The overall strategy defines when and how to set the selection variable  $\sigma_i$ . The most appropriate behavior has been achieved with *defense mode* as default strategy. If the human's sword is beyond the defense radius, then the *attack mode* is initiated. The *contact mode* is only active while the swords are in contact after an attack



**Fig. 6** Position progressions of  $M_i$  and  $M_i^{trgt}$  while switching between different modes over a period of 23 s. Although arbitrary switching procedures are triggered by  $\sigma_i$  (i.e.,  $M_i^{trgt} = M_i^\sigma$  is not steady), a jerk-limited and executable motion  $M_i$  is generated online in each control cycle (cf. Fig. 4). The switching sequence of  $\sigma_i$  is *def-att-con-def-att-con-def*.

motion (cf. eqn. 8). Figure 6 shows the desired and the achieved steady progressions of the robot’s end-effector position that is achieved during arbitrary switchings of  $\sigma_i$ .

The vision processing chain on the left of Fig. 2 typically runs at rates of 10–15 fps, but since some parts of the processing pipeline take place in parallel, the latency is up to 300 ms. The need to calculate sword speed between two successive frames and other delays in the sword command and control further increase the latency. The total system latency between the opponent beginning a swing with their sword and the beginning of robot motion in response is typically around 500 ms. While it is possible for the opponent to move faster than this, it is commonly fast enough for the robot to respond.

As indicated in Fig. 3, the quality of recognition is very high, with very few false positives or dropped frames. If the sword was pointed straight at the Kinect, such that only its tip is visible, it could not be detected, but it typically only needs to be angled about 15 degrees away from the camera to allow detection. By properly positioning the camera it is possible to ensure that the sword would rarely pass through this narrow cone, and when it does, it would only be for a very brief amount of time.

Based on the visual sword detection, the desired defense pose and velocity ( $M_i^{def}$ ) are continuously updated. As indicated in Fig. 4, the image processing loop runs at a different rate than the robot motion controller (10–15 Hz and 1 KHz); as soon as a new value of  $M_i^{def}$  is provided, it will be immediately applied in the next robot control cycle.

## 5 Conclusions

### Target State Switching

Real-world experimental results of a switched-system using a selection variable ( $\sigma_i$ ) to select between different desired target states of motion were shown (cf. Fig. 6). Despite its simplicity, this approach promises to simplify and improve sensor-based robot motion control, because robots can react instantaneously and in different ways to unforeseen sensor signals and events.

### Human-Robot Interaction

Using the approach of target state switching, jerk-limited executable motions are generated online, such that robots can *permanently* respond to human motions deploying sensor signals in the feedback loops.

### Reliability and Robustness

The recognition of the opponent's sword based on camera data of a MS Kinect 3D camera using segmentation procedures of the Point Cloud Library (PCL) works very reliably. If there is a significant translational and/or rotational error or if there is a dropped frame, a jerk-limited and executable motion will always be generated by the Reflexxes Motion Libraries, such that the system is stable despite erroneous sensor signals. The overall control scheme reacts deterministically and runs very robustly because only a joint position or impedance controller is required.

### Replicable Implementations and Experiments

The two main software components, the Point Cloud Library (PCL), the Reflexxes Motion Library, and the interface software for the KUKA/DLR Lightweight Robot are freely available [10, 17, 20], such that the proposed control scheme can be easily duplicated with very reasonable efforts. If other robots are used, only a trajectory tracking controller is required. For visual perception, only a MS Kinect 3D camera is required [4].

### Implementation Time: Three Weeks

The original “JediBot” was entirely created by students of the class CS225A at Stanford University [1] within only three weeks. Despite many iterations to improve the system and to exhibit it at the 2011 IEEE International Conferences on Intelligent Robots and Systems, the control scheme of Fig. 2 remained with relatively few modifications. Using [10, 17], the scheme is very simple and straight-forward to implement even for students with limited experience in robotics. The websites of [10, 17] provide tutorials and examples for the presented matter.

**Acknowledgements.** All authors would like to thank *QNX Software Systems* and *Wind River Systems* for providing free software licenses. The first author would like to thank the *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) that is currently co-funding him.

## References

1. Stanford Artificial Intelligence Laboratory, Stanford University, 353 Serra Mall, Stanford, CA 94305-9010, USA. Course Description: CS225A Experimental Robotics. Internet (2012), <http://cs.stanford.edu/groups/manips/teaching/cs225a> (accessed: June 2, 2012)
2. Fyffe, S.: Students Create 'JediBot'. YouTube Video, Internet (2011), <http://www.youtube.com/watch?v=VuSCErmoYpY> (accessed: June 2, 2012)
3. Bischoff, R., Kurth, J., Schreiber, G., Köppe, R., Albu-Schäffer, A., Beyer, A., Eiberger, O., Haddadin, S., Stemmer, A., Grunwald, G., Hirzinger, G.: The KUKA-DLR lightweight robot arm — A new reference platform for robotics research and manufacturing. In: Proc. of the Joint Conference of ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics), Munich, Germany. VDE Verlag (June 2010)
4. Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052-7329, USA. Microsoft Kinect homepage. Internet (2012), <http://xbox.com/Kinect> (accessed: June 2, 2012)
5. PrimeSense, 28 Habarzel St. Tel-Aviv, 69710, Israel. Homepage, Internet (2012), <http://www.primesense.com> (accessed: June 2, 2012)
6. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 21(5), 433–449 (1999)
7. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: Proc. of the International Conference on Computer Vision Theory and Application, Lisbon, Portugal, pp. 331–340 (June 2009)
8. Fischler, A.M., Bolles, C.R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), 381–395 (1981)
9. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China (May 2011)
10. Point Cloud Library (PCL). Homepage, Internet (2012), <http://www.pointclouds.org> (accessed: June 2, 2012)
11. Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: Point feature extraction on 3d range scans taking into account object boundaries. In: Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China, pp. 2601–2608 (May 2011)
12. Broquère, X., Sidobre, D., Herrera-Aguilar, I.: Soft motion trajectory planner for service manipulator robot. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, pp. 2808–2813 (September 2008)
13. Liu, S.: An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators. In: Proc. of the Seventh International Workshop on Advanced Motion Control, Maribor, Slovenia, pp. 365–370 (July 2002)

14. Haschke, R., Weitnauer, E., Ritter, H.: On-line planning of time-optimal, jerk-limited trajectories. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, pp. 3248–3253 (September 2008)
15. Kröger, T., Wahl, F.M.: On-line trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Trans. on Robotics* 26(1), 94–111 (2010)
16. Kröger, T.: Opening the door to new sensor-based robot applications — The Reflexxes Motion Libraries. In: Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China (May 2011)
17. Reflexxes GmbH, Sandknöll 7, D-24805 Hamdorf, Germany. Homepage, Internet (2012), <http://www.reflexxes.com> (accessed: June 2, 2012)
18. KUKA Laboratories GmbH, Zugspitzstraße 140, D-86165 Augsburg, Germany. Homepage, Internet (2012), <http://www.kuka-labs.com/en> (accessed: June 2, 2012)
19. Schreiber, G., Stemmer, A., Bischoff, R.: The fast research interface for the KUKA lightweight robot. In: Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications — How to Modify and Enhance Commercial Controllers at the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, pp. 15–21 (May 2010)
20. Kröger, T.: Documentation of the fast research interface library, version 1.0. Internet (November 2011), <http://cs.stanford.edu/people/tkr/fri/html> (accessed: June 2, 2012)

# Development of Aerobots for Satellite Emulation, Architecture and Art

Inna Sharf, M.S. Persson, David St-Onge, and Nicolas Reeves

**Abstract.** In this paper, we present two unique aerobots: the spherical blimp used for satellite emulation and the cubic blimp developed for use in floating architecture and visual art. The blimp designs bear a number of similarities, in particular, their construction with an exoskeleton, full actuation to enable six-dof motion and requirement for autonomous localization. Experimental results are presented to demonstrate the closed-loop control for station-keeping, as well as the selected performance statistics such as maximum speeds attained and time the aerobots can remain afloat. Additional qualitative results are presented from the experiments with satellite capture and artistic performances and common challenges with further use in the intended and new applications will be outlined.

## 1 Introduction

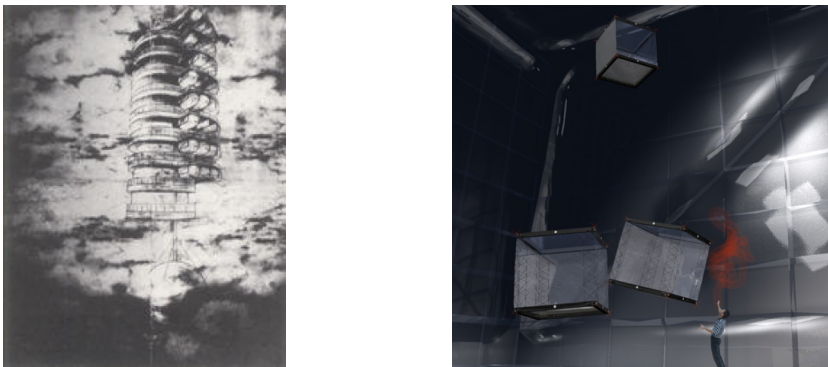
The concept of aerobot, i.e., an autonomous flying robot, has been around for several decades. The use of balloons or airships as aerobots has been explored in-depth in the context of planetary exploration [1]. Indoor applications of such systems, beyond their use for educational purposes, are rare. In this paper, we would like to present two aerobots developed for two very different purposes, yet with the a ? design and features which have much in common. The first platform, developed in the Aerospace Mechatronics Laboratory at McGill (AML) is a spherical airship, the design of which was motivated by one of the authors' research on the problem of robotic grasping of objects in space. In particular, the airship represents a novel concept for emulating gravity-free conditions in a laboratory setting and has been used to develop autonomous algorithms for satellite capture in the context of satellite rescue and on-orbit servicing operation. The second

---

Inna Sharf · M.S. Persson  
Centre for Intelligent Machines, McGill University,  
Montreal, Canada

David St-Onge · Nicolas Reeves  
NXI Gestatio Design Lab, School of Design, University of Quebec,  
Montreal, Canada

platform was designed and constructed at the NXI Gestatio Design Laboratory of the University of Quebec in Montreal. It originates from an architectural myth studied by its creator, professor Reeves, architect and artist. Reeves envisioned more than 10 years ago the possibility of developing flying objects whose shape would be in strong contradiction with the idea of flying or hovering. Such hovering structures as well as the paradox they represent (see Fig. 1) would constitute an architectural statement by themselves: they somewhat materialize the old and mythical dream of an architecture freed from the law of gravity — an image that can be found along the whole history of architecture, in many civilizations [2]. The cubic shape, chosen for the [Voiles!SAILS] aerobot prototypes (see Fig. 1), makes them conceptually similar to bricks, the basic unit of construction, and gives them the potential to assemble into bigger structures. From that conceptual starting point, the first prototypes developed to date show a high potential for visual art installations, as well as for hybrid theatrical performances where aerobots interact with human actors. The cubic aerobot described in this paper, called Tryphon, evolved into a research-creation platform bringing the disciplines of engineering, performing art, architecture and visual art together.



**Fig. 1** [Voiles!SAILS] Concept: a) Krutikov flying cities and b) Simulation of interactively assembled Tryphons

## 1.1 Related Work

The description of related work will be presented in relation to the two applications for which the aerobots were designed. Starting with the application of satellite emulation in the laboratory setting, previously developed experimental facilities for satellite emulation are usually built by using a spherical air bearing [3]. Experimental test-beds for space robotics research typically use one of the following concepts to emulate weightless environment of space on earth: (a) the robot moves on a flat horizontal surface; (b) a neutral buoyancy water tank; (c) complicated gravity compensation systems and (d) a free-fall tower.



To the best of our knowledge, no other aerobot has been used to date for architectural research and exploration. The closest systems somewhat comparable to the concepts used in the [Voiles|SAILS] program for architectural research are the rapid prototyping printers. From a computer generated, or a computer assisted design, the architect can get a small scale version to better visualize the 3D presence of the building. In terms of robots and art, many examples may be found since the Norman White installation “Facing Out Laying Low” in 1977 [4]. Robotic art is a field of media art that is increasingly explored by contemporary artists; this helped by the development of easy to use systems such as the Arduino open-source computing platform [5]. Hybrid performances are still very rare, but among the most known are the “Grace State Machines” of Bill Vorn (University of Concordia) or the “Hexapod” of Stelarc (University of Western Sydney) [6].

This manuscript describes the design, development and experiments conducted with the two aerobots. Where appropriate, similarities between the two aerobot platforms are highlighted and commonality of issues related to the development of autonomous capabilities and autonomous operations are discussed. Results are presented which demonstrate the performance of the two aerobots for each of their intended applications. In particular, for the spherical airship, we showcase its capability to produce general rotational motion and the free-floating nature of its response as a result of interaction with a robotic arm during capture. For the Tryphon robot, we focus on the high reliability and reproducibility of generated interactions with a human, as well as its long lasting autonomy for standalone installations. The geometry is also mandatory to describe as it is key to floating architecture explorations.

## **2 Aerobots Design, Construction and Control**

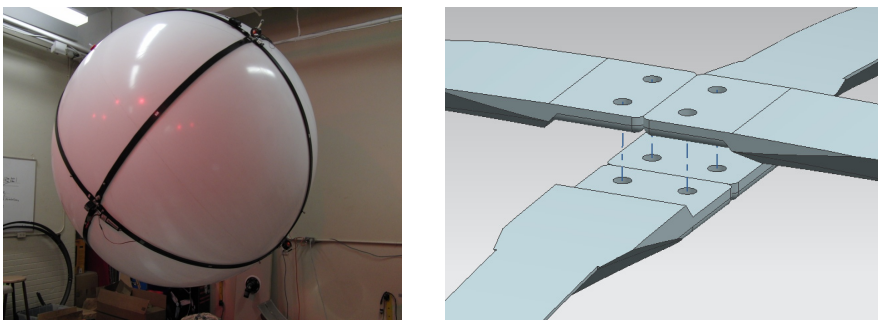
### ***2.1 Design***

The balloon employed for satellite emulation is a custom design spherical airship equipped with six propellers, accompanying control electronics, onboard power, and sensors for pose estimation. The design (see Fig. 2a) was motivated by three principal requirements: 1) the balloon must closely emulate a free-floating object which requires it to be neutrally buoyant and balanced; 2) it must carry a grapple fixture, initially, a simple design and ultimately more sophisticated designs; 3) it has to be capable of a range of motions including rotation about a fixed axis and tumbling to emulate, for example, a spin-stabilized satellite or a spacecraft out of control. Moreover, these motions need to be generated in a controlled manner to allow multiple experimental tests under the same conditions.

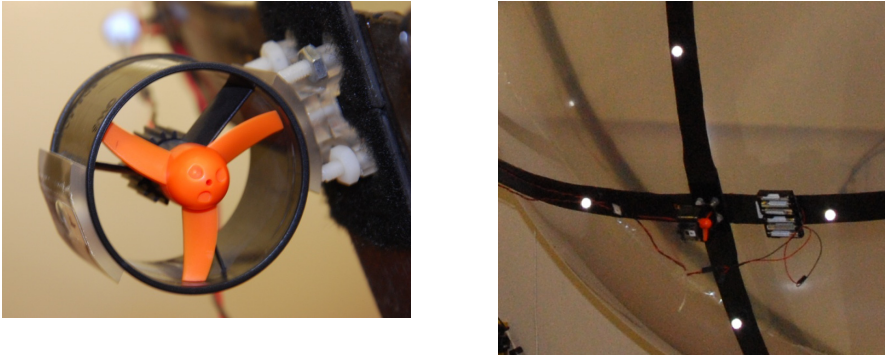
After several design iterations over a period 2003-2007, the current design, shown in Figure 2 incorporates the following main components:

- 1) A light 6-ft diameter spherical bladder bag, made of 2.5 mil thick polyurethane for a maximum net lift of 3.34 kg.

- 2) A rigid frame (Fig. 2), designed and manufactured in-house, made up of three carbon-fiber hoops with light-weight honey-comb cores arranged normal to each other. Each ring in turn made up of quarter-length arcs interconnected at small carbon-fibre extensions (see Figure 2b). The frame allows for easier and more reliable balloon assembly and to eliminate the inaccuracies introduced by the deformable blimp bag on the airship dynamics and control. The balloon bag is inflated inside this structure and supports it through a friction fit.
- 3) Six identical propellers mounted in ducted fans, consisting of DC motors driving 48 mm diameter propellers within 35 mm long plastic cylinders (see Figure 3a). At a nominal voltage of 8.4 VDC, each thruster is capable of producing up to 0.45 N thrust in its primary direction or up to 0.25 N in its reverse direction. The propellers are mounted in custom-made supports, in a symmetrical arrangement on the sphere. With the chosen arrangement of the propellers the balloon is fully actuated and in theory, is capable of producing decoupled motions in all three translations and rotations.
- 4) Six speed controls for the propellers. The ducted fan speed control electronics perform two main functions: signal conditioning and amplification of the control signal. The incoming standard PWM signal is converted to a bipolar PWM signal zeroed around 50% duty cycle, allowing for forward and reverse thrusting of the ducted fans.
- 5) The sensor suite on the airship includes two types of sensors: an Inertial Measurement Unit (IMU) Microstrain GX1 and a laser rangefinder (Hokuyo URG-04LX). The sensors communicate wirelessly with the ground station via two pairs of Bluetooth transceivers.
- 6) The battery used on the balloon to power the propellers and the speed control electronics is an 8.4 VDC, 400 mAh lithium-polymer battery. A second battery powers the IMU, the laser rangefinder and the Bluetooth transceivers.
- 7) A composite-material grapple fixture affixed to the structure for experiments in capture of the airship by the robotic arm.



**Fig. 2** Helium airship for satellite emulation: a) Current airship configuration; b) Joint of two hoops of the rigidizing structure



**Fig. 3** Balloon propellers in the mount on the structure and Vicon marker cluster around one of the propellers

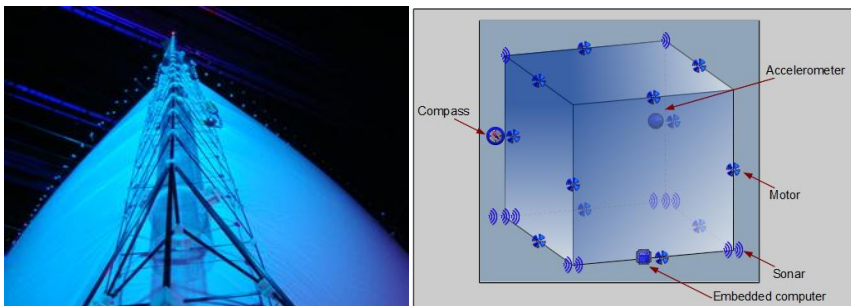
In order for the airship to closely emulate a free-floating object in space it must be both neutrally buoyant and balanced, thereby eliminating the effects of gravity. With these conditions met, the unactuated airship freely floats in air and has no preferred orientation. An additional desirable property is for the airship to have a diagonal inertia matrix in the body-fixed frame, the axes of which are aligned with the three orthogonal propeller thrusts. To meet these requirements, the locations of components which can be placed freely on the ring structure were determined to achieve the center of mass close to the geometric center of the 6-ft sphere and a nearly diagonal inertia matrix. The airship is also equipped with 6 posts affixed to the propeller mounts on which balancing masses can be easily placed to aid with the balancing procedure. The final balancing is carried out manually by the operator, again with the aid of balancing masses.

The Tryphon robot flies thanks to an inflatable cubic blimp, with the side of 2.05 metres, and similarly to the spherical blimp, it is filled with helium. The material used for the blimp is 3.5 mil thick polyurethane, which weighs 3.5 oz per square yard ( $0.12\text{kg/m}^2$ ). The blimp itself is made of six square faces welded together. As with any other balloon shape, the faces become convex when the balloon is inflated and more pressure tends to make it more spherical. To maintain a cubic shape, the blimp has to be constrained by a rigid structure — an exoskeleton. Therefore, unlike the case of the spherical blimp, where the exoskeleton is primarily used for mounting equipment on the blimp, the structure confining Tryphon is there to maintain its cubic shape. The structure is made of carbon fiber tubes, strips and rods (see Fig. 4a). Each edge is a triangular truss of 2.25 metres length and the whole structure weighs approximately 1 kg. Assembly of the cube, including filling it with helium, can be completed in less than two hours by two people.

Similarly to the spherical airship, the structure of Tryphon also supports all the electronics as well as the propulsion system. The actuators consist of small ducted fans, with ducts and propellers made of carbon fiber. Four are located at the mid-point of each bottom edge and oriented in the  $x$  and  $y$  positive directions. Another four motors are similarly placed along the vertical edges (see Fig.4b). Positioning the motors this way allows an independent control of the translations of the robot along the  $x$ ,  $y$  and  $z$  axes. Since there is no motor on the top trusses, and since most of the batteries are fixed to the bottom trusses, the global centre of mass of the system lies about 20 cm below the centroid of the cube. As a consequence, the robot cannot turn upside down and its roll and pitch angles are thus stabilized in a passive way.

To fix the body reference frame, for both the spherical airship and the cubic blimp, the origin of the frame is located at the blimp centroid. This choice is made to simplify the formulation of the dynamics equations since the centre of mass can easily change, depending on the equipment mounted on the robot. For example, the use of textiles to hide the edges of Tryphon, or a change in the sensors' configuration, will modify the mass distribution.

In the current prototype of the cubic blimps, the bladder used is made of thicker material, which allows the blimp to maintain nearly perfect equilibrium for several days: usually 3 to 6 days, depending on the room temperature variations. The 8 batteries allow a soft control of the oscillation when stabilizing, installation known as the “Paradoxal Sleep”, for about 6 to 8 hours in optimal room conditions (no ventilation, and constant temperature). In harsh environment, like a building hall, or with heavy interaction, like in hybrid performances, the airship can operate for approximately 2 to 3 hours with its current set of batteries. Table 1 presents a comparison of the design of the two blimps.



**Fig. 4** Tryphon design: a) Structure of one carbon fiber truss and its polycarbonate ducts; b) Layout of actuators and sensors

## 2.2 Control of Aerobots

The spherical airship is controlled from a ground-station PC that transmits commands to the airship wirelessly over a Futaba radio. The ground-station PC

performs all computations for the controller. The controller resides in the Simulink environment with the QuaRC toolbox and soft real-time target developed by Quanser. Initially [7], a PD controller was implemented on the airship with gains adjusted through simulation and by a trial and error process. The state feedback for the controller in [7] was obtained from the measurements by the Vicon motion-capture system, which is a set of six infrared cameras mounted along the periphery of the lab. They track retro-reflective markers affixed to the spherical airship (see Figure 2b). The system therefore provides position and orientation data for the blimp; velocities were calculated by taking finite differences of the previous 10 samples. Recently [8], we have implemented optimal LQR and LQG controllers on the airship and improved the state estimation from Vicon measurements by using the Unscented Kalman filter with angular velocity measurements from the onboard IMU.

**Table 1** Design comparison of two blimps

	<b>Satellite emulator</b>	<b>Tryphon</b>
Structure	Molded carbon fiber rings	Assembled carbon fiber rods, tubes and strips in 12 triangular sections trusses
Balloon	Spherical white bladder, 2.5mil polyurethane.	Truncated white cube bladder, 3.5mil polyurethane.
Motors	6 GWS fans	8 to 12 Alfa carbon fiber propellers and duct mount on Mega brushless motors
Sensors	IMU, Laser range finder, MoCap external system	16 sonars, 8 light sensors, compass, accelerometer
Batteries	8.4 VDC, 400 mAh LiPo  (there should be two batteries here ?)	8 LiPo 2500mAh
Brain	Computer off board	Gumstix onboard computation
Other		20 hubs to allow different sensor configurations

The control architecture of the [Voiles|SAILS] aerobots evolved over the course of many performances and installations created by actors, visual artists and other artists involved in the project [9]. In 2006, the first autonomous control was reactive to the physical attributes of the space. Compass and sonars were the only

sensor inputs used to stabilize the aerobots relatively to a fixed setup. A simple distributed PID controller (one by sensor, one by motor and one by robot state) was sufficient for these needs at that time. Since then, an accelerometer and a gyroscope were added to provide the aerobots with more information on its state. Light sensors and microphones were also implemented for human interactions. A study of different controller approaches led to the use of a fuzzy controller in completely autonomous and stand-alone installations (without human interaction) while the performances with actors or dancers rely on a PID controller onboard and a trajectory planning algorithm used in parallel with the fuzzy controller.

The development focus was set on embedding the hardware and control in the robot, with only a laptop running a custom designed java interface to allow the technician operating the aerobots in their installations to monitor its battery and mechatronic states. For research and creation purposes, the team is currently exploring the potential of external motion capture systems, such as the Vicon system used for the AML spherical blimp. Such a system could be used to detect visitors, to enhance the interactions as well as to control the motion of the blimp and to better understand the dynamics of its unique shape.

### 3 Closed-Loop Control Experiments with Aerobots

In this section, we present a sampling of experimental results obtained for the aerobots to demonstrate the hovering performance of the two blimps under the PD control. As mentioned earlier, the Tryphon usually relies on its onboard computer and sensors for control. For the experimental results presented here, however, experiments were conducted in a large room equipped with a Vicon tracking system in order to understand the aerobot dynamics in flight and to evaluate the controller performance. Specifically, a PD controller combined with a Kalman filter of Vicon pose measurements was implemented in Matlab for off-board closed-loop control.

The relevant experimental response statistics of the controllers are stated in Table 2 in addition to “application” related statistics, such as the time that the aerobots can remain afloat and the maximum translational and rotational speeds achieved in our laboratory environments.

Fig. 5 displays the hovering performance (position response) of the spherical and Tryphon blimps under PD control, with pose feedback provided by the Vicon motion capture system. The corresponding results for attitude response are shown in Fig. 6. Fig. 7 presents the response of the spherical blimp to light translational and rotational disturbances, demonstrating the rise times of approximately 5 seconds, settling times of 15 seconds, and overshoot of approximately 10%, although the latter particularly is difficult to define because of the poorly defined steady state. Analogously, in Fig. 8, we include the step position response of the Tryphon aerobot, showing the rise time of 15 seconds, settling time of approximately 50 seconds and overshoot of around 25%.

From the results in Table 2, we observe that the station-keeping control of the spherical blimp is better than of Tryphon. In particular, the RMS errors for position regulation of the two blimps are 0.028 m and 0.101 m respectively, while the corresponding errors for attitude regulation are 0.017 rad and 0.122 rad. On the other hand, Tryphon can maintain neutral buoyancy for a significantly longer time period, nearly one day. Tryphon is also able to reach a higher translational speed, although the results obtained for the AML spherical blimp were limited by the size of the laboratory at McGill. At the same time, the spherical blimp can reach a higher rotational speed, because of better aerodynamic characteristics.

**Table 2** Experimental performance comparison of two blimps

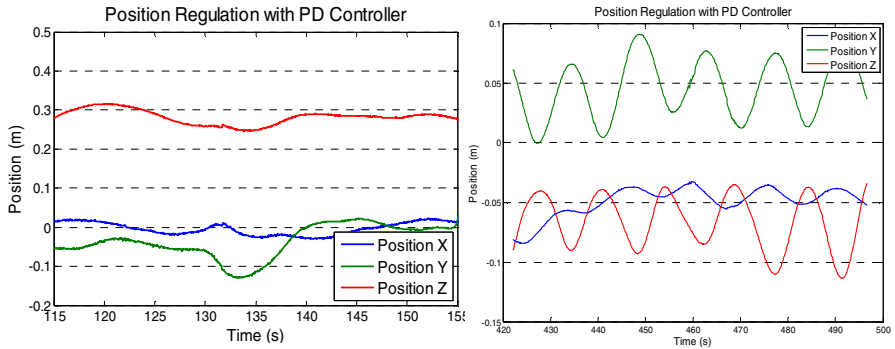
	Satellite emulator	Tryphon
Regulation position error (RMS)	0.028 m	0.101 m
Regulation attitude error (RMS)	0.017 rad	0.122 rad
Neutral buoyancy time	~1 hour	~24 hours
Max. translational speed	0.3 m/s	0.75 m/s (observed) 1m/s (simulated)
Max. rotational speed	2.3 rad/s	1.6 rad/s

## 4 Applications of Aerobots

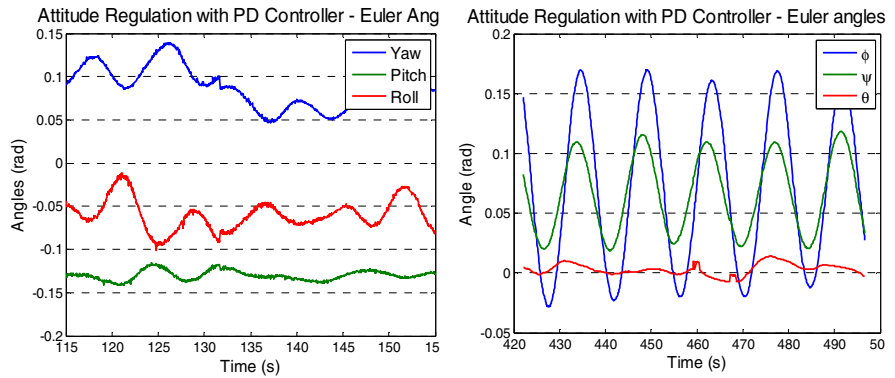
### 4.1 Satellite Emulation Experiments

A number of experiments have been carried out with the spherical airship employed as a free-floating target for capture by the seven-dof robotic arm housed in the laboratory. Snapshots of the satellite capture experiments are shown in Fig. 9 for a successful capture of the slowly translating airship by its grapple fixture. The fiducial three-dot mark on the airship is employed for visual servoing of the robot when its end-effector is sufficiently close to the grapple fixture. The capture also involves the planning of the optimal interception trajectory, as per the algorithm described in [10].

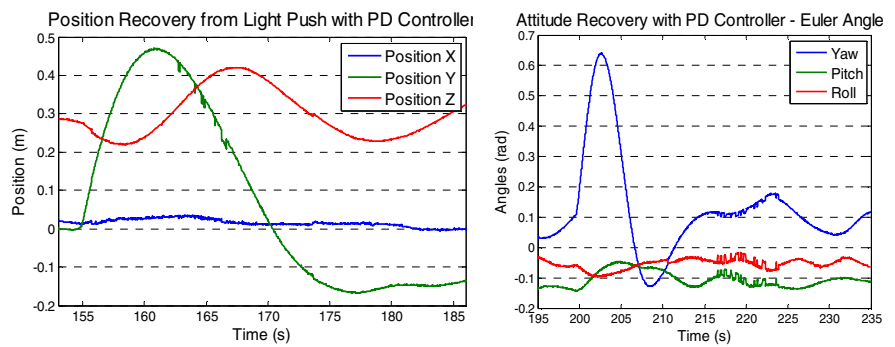
The airship has been recently used as an aerial platform for testing and evaluation of state estimation and localization algorithms that we have developed for an entirely different unmanned aerial vehicle: a quadrotor platform [11]. Indeed, because of its inherent safety and user-friendliness, the airship represents an ideal platform for testing and evaluation of many planning and control aspects of autonomous aerial vehicles.



**Fig. 5** Position regulation of the AML spherical (left) and Tryphon (right) blimps



**Fig. 6** Attitude regulation of the AML spherical (left) and Tryphon (right) blimps



**Fig. 7** Recovery from position disturbance (left) and attitude disturbance (right) of the AML spherical blimp



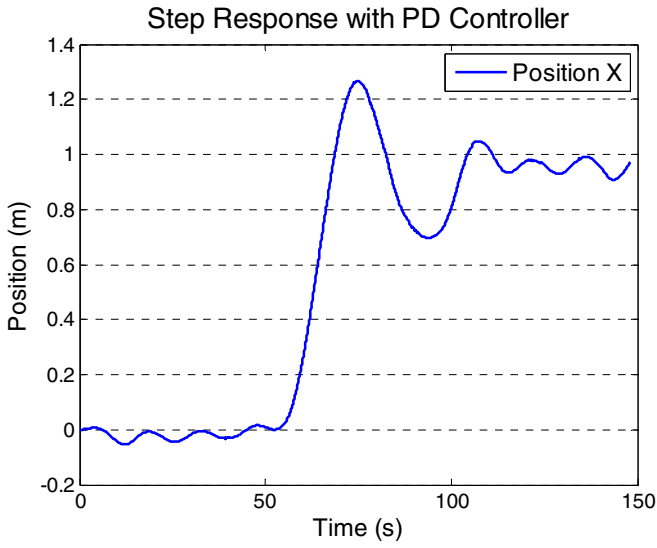


Fig. 8 Step response in X-position of Tryphon

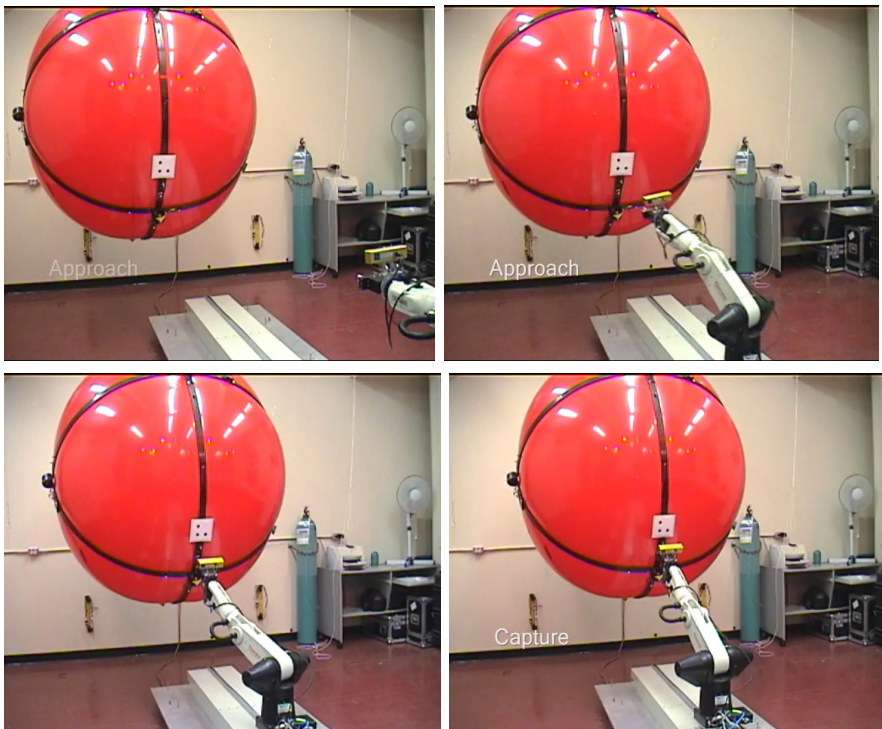


Fig. 9 Snapshots of airship capture maneuver with 7 dof robotic arm

## 4.2 Tryphon Performance Experiments

Since the development of the robotized version of the [Voiles|SAILS] prototypes, numerous performances and installations have been achieved. In Figs. 10 and 11, we include pictures of three performances conducted since 2006 in a variety of venues. The first picture on the left shows an event at the Montreal Science Center in 2008, based on the idea to introduce robots to children. Actress Veronique Daudelin is shown explaining the functionalities of Nestor, the smallest brother of Tryphon. During the performance, she triggered different scripted actions by using various stimuli: a quick movement in front of a sonar initiated a rotation of the aerobot, a powerful beam of light on a light sensor attracted the aerobot to the middle of the space. Children were also invited to start such interactions with Nestor.

Fig. 11a shows a picture of the first performance of the [Voiles|SAILS] aerobots, in 2006 at the Museum of Civilization in Quebec city. Three early blimp versions with a linden structure, called Mascarillons, were hovering in a large room in the dark. When a visitor approached, a real-time projection of an actress' eyes illuminated the closest aerobot's sides and the visitor could start a discussion with a "peculiar intelligence," simulated by an actress hidden behind the scene.

Finally, over the period from 2009 to 2010, numerous workshops were conducted with actors from a theater company, the "Théâtre des 4 coins". During each workshop, lasting for two weeks, the actors visited a room with one or more Tryphons to develop movements, choreography, scenarios and interactions with the blimps. The first public demonstration based on these workshops will be held in Sao Paulo FILE festival in July 2012. The show will be based on a partially improvised choreography and the aerobot will be controlled by the dancer's movements and singing.



**Fig. 10** Interactive Performance in Montreal Science Center. Actress: Véronique Daudelin.



**Fig. 11** Left: discussion with an aerobot at the Quebec Museum of Civilization, 2006; right: art residency with the Theatre des Quatre Coins at Laval University, Quebec, 2010

The above example events illustrate the specific constraints that determined the design of the Tryphons, which are quite different from those that defined the AML spherical aerobot's. The Tryphons were planned from the very beginning with artistic/performance objectives in mind. Thus, their abilities and functionalities were to be used for the sake of conveying expressions and emotions through combinations of translations, rotations, states and behaviours; they were seen as embedding notions of *personality* and *identity*. While the first installations involved only reactions from visitors, the cubes were designed to interact with humans (audience or performers) through a variety of sensors. Scripted interactions with actors have been possible for the past three years, and many relevant observations were made prior to that from the visitors' reactions to the cubes. These observations allowed us to refine the aerobot's sensing abilities in order to create full interactive performances, in which performers and aerobots interact through real hybrid choreographies. In particular, they influenced the number and positioning of sonar sensors on the cube's periphery, and led us to use compared informations from different sensor sources, in order to compensate from the imprecisions inherent to any kind of sensing device.

Interactions with humans can lead to applications in the fields of museology or event design. The cubes "have been invited" to fashion design shows; with a proper sensor configuration, they could be used as individual or group guides for exhibitions or historical places. In addition to the possibility for them to speak through sound transducers, they could display written information on their faces thanks to micro video projectors inserted in the helium bladder.

Similarly to the AML spherical blimp, the cubes can also be used for engineering applications. Profiting from their cubic shape, which allows them to assemble into structures, several horizontal or vertically connected cubes can lift multi-kilogram payloads, like flying cranes. In indoor large spaces, they could be used to carry objects or pieces with excellent degree of precision. The cubes can become test-beds for the development of control algorithms for 6 DoF objects in zero-gravity environment. In this case though, their particular geometry imposes certain limits: because of the cubic shape, their moments of inertia depend on their

rotation axis. However, that very same shape allows for investigation and implementation of autonomous assembly algorithms for applications to space structure assembly and other missions.

## 5 Conclusions and Future Work

We have summarized the development and experiments conducted with two unique indoor aerobots: the spherical blimp developed at McGill and the square Tryphon blimp developed at UQAM. The aerobots represent a significant departure from conventional lighter-than-air vehicles, both in their design and intended applications. Future work holds many more challenges with respect to developing motion planning, state estimation and control strategies for fully autonomous operation of the blimps for the intended applications: accurate trajectory tracking for satellite emulation and for indoor navigation, docking of airships for recharging, self-assembly of Tryphons into free-floating structures and autonomous behaviours in response to artists' commands for hybrid performances.

**Acknowledgements.** Sharf would like to thank former graduate students Joel Robert, Yin Yang and Mikelis Valdmanis for their dedication and perseverance with the development and experiments on the spherical airship. Funding support from NSERC, PRECARN and Canadian Space Agency is also gratefully acknowledged.

The [Voiles|SAILS] project was possible thanks to the work of more than seven graduated student in computer science and engineering. The authors would like to acknowledge the financial support of the NSERC, the Canada Research Chair Program, the MCCCFC, the CALQ, the Canadian Council for the Arts, the FQRSC, the Hexagram|CIAM Institute as well as of the UQAM fund for support to research and creation (PAFARC).

## References

- [1] Kerzhanovich, V.V., Cutts, J.A.: Aerobots in planetary exploration. In: 2000 IEEE Aero-space Conference. Proceedings (Cat. No.00TH8484), vol. 7, pp. 547–555 (2000)
- [2] Cooke, C., et al.: See in particular Krutikov's flying cities. Editions of the Museum of Modern Art, New York, USA (1990)
- [3] Agrawal, B.N., Rasmussen, R.E.: Air-bearing-based satellite attitude dynamics simulator for control software research and development. In: Proceedings of the SPIE - The International Society for Optical Engineering, vol. 4366, pp. 204–214 (2001)
- [4] Wilson, S.: Information Arts. MIT Press, Cambridge (2003)
- [5] Warren, J.-D., Adams, J., Molle, H.: Arduino Robotics. Apress, New York (2011)
- [6] Moura, L., et al.: INSIDE [art and science], LxXL, Lisbon, Portugal (2009)
- [7] Sharf, I., Laumonier, B., Persson, M., Robert, J.: Control of a Fully-actuated Airship for Satellite Emulation. In: Video Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2008, Pasadena, CA, May 19-23 (2008)
- [8] Yang, Y.: Nonlinear Control and State Estimation of Holonomic Indoor Airship, M.Eng., Mechanical Engineering, McGill University (2011)

- [9] St-Onge, D., Reeves, N., Gosselin, C.: [VOILESISAILS]: a modular architecture for fast parallel development in an international multidisciplinary project. In: Proceedings of the ICAR 2011, Tallinn, Estonia (2011)
- [10] Robert, J.: Autonomous Capture of a Free-floating Object Using a Predictive Approach, M.Eng., Mechanical Engineering, McGill University (2008)
- [11] Harmat, A., Sharf, I., Trentini, M.: Parallel Tracking and Mapping with Multiple Cameras on an Unmanned Aerial Vehicle. Submitted to ICIRA 2012 (2012)

# Experimental Multi-Vehicle Path Coordination under Communication Connectivity Constraints

Pramod Abichandani, Kenneth Mallory, and Mong-ying Ani Hsieh

**Abstract.** The main contribution of this paper is the experimental validation of a decentralized Receding Horizon Mixed Integer Nonlinear Programming (RH-MINLP) framework that can be used to solve the Multi-Vehicle Path Coordination (MVPC) problem. The MVPC problem features path-constrained vehicles that begin their transit from a fixed starting point and move towards a goal point along fixed paths so as to avoid collisions with other robots and static obstacles. This framework allows to solve for time optimal velocity profiles for such robots in the presence of constraints on kinematics, dynamics, collision avoidance, and inter-robot communication connectivity. Experiments involving up to five (5) robots operating in a reasonably complex workspace are reported. Results demonstrate the effect of communication connectivity requirements on robot velocity profiles and the effect of sensing and actuation noise on the path-following performance of the robots. Typically, the optimization improved connectivity at no appreciable cost in journey time, as measured by the time of arrival of the last-arriving robot.

**Keywords:** Multi-Vehicle, Motion Planning, Communication connectivity constraints, Mixed Integer Non-Linear Programming, Path Coordination, Receding Horizon.

## 1 Introduction

The problem of Multi-Vehicle Path Coordination (MVPC) is formally defined as follows: *Given a group of vehicle robots that have fixed and known paths connecting an initial and a goal location, generate time-optimal velocity*

---

Pramod Abichandani · Kenneth Mallory · Mong-ying Ani Hsieh  
Drexel University,  
3141 Chestnut Street, Philadelphia, PA, 19104  
e-mail: {pva23, km374, mhsieh1}@drexel.edu

*profiles that satisfy kinematic, dynamic, collision avoidance, and communication connectivity constraints.*

While a significant body of work has been devoted to path planning for mobile robots (e.g., [1], [2]), we focus our work on the relatively untouched area of communication-centric velocity planning along predetermined routes. More often than not, one does not get the liberty of planning an arbitrary path around sparse obstacles, and rather, must follow a prescribed route. This is especially true in situations where multiple driverless car-like vehicles operate in an urban environment. The formulations and results provided in this paper address the communication-centric motion planning challenges associated with such unmanned operations.

Several approaches have been used to address the problem of path coordination of multiple robots [1], wherein multiple robots with fixed paths coordinate with each other so as to avoid collisions and reach destination points. These approaches include the use of coordination diagrams [3], constrained configuration space roadmaps [4], and grouping robots with shared collision zones into subgroups [5]. In [6], mixed integer linear programming (MILP) formulations were used to generate continuous velocity profiles for a group of robots that satisfy kinodynamics constraints, avoid collisions and minimize task completion time. This body of work was extended in [7], [8], and [9], where the authors address the communication requirements of the problem by incorporating physical layer communication connectivity models and using state-of-the-art interior-point methods to solve the resulting non-linear programming (NLP) and MINLP formulations. Readers are referred to [10] and the exhaustive list of references therein for Mathematical Programming (MILP, NLP, MINLP) based motion planning formulations and solution techniques. One of the main advantages of Mathematical Programming (MP) based frameworks is that they allow incorporation of a rich set of constraints that represent the nuances of a typical motion planning problem. For the work described in this paper, the MP-based RH-MINLP framework facilitates a systematic study of the interplay between inter-robot wireless communication and robot speed profiles.

Most closely related to the work presented in this paper is [11], where the authors present a decentralized receding horizon formulation for multiple aircraft path planning using MILP to generate provably safe trajectories. Similar to [11], the solution algorithm presented here features a sequential decision ordering mechanism. Numerical simulation results for the RH-MINLP framework experimentally verified here were reported in previous work [9].

Only a handful of studies have documented practical implementations of MP-based motion planning (e.g. [11], [12]) and to the best of our knowledge, this paper presents the first experimental results for MVPC under communication connectivity constraints using RH-MINLP.

## 2 Technical Approach and Problem Formulation

Figure 1 describes the technical approach adopted in this work. A group of two-wheeled differential drive mobile robotic vehicles travel along fixed and known piecewise cubic spline paths in their workspace while maintaining communication connectivity with  $n_{conn}$  neighboring vehicles. The feasibility criteria for trajectories require that the robots' kinematic and dynamic constraints be satisfied, along with the imperatives of avoiding collisions and obeying the communication connectivity constraints. The distributed decision-making and computations are simulated on a central computer using MATLAB, which interfaces with the MINLP solver MILANO [13]. This computer communicates the optimal speed profile information to each robot via a wireless communication link. The computer receives real-time robot location and speed information. The three main elements of the problem - robot paths, inter-robot communication - and receding horizon planning are discussed in the following sections.

### 2.1 Robot Motion and Fixed Paths

Consider a group of  $N$  two wheeled differential drive mobile robots shown in Figure 1. The robots move in a global (X, Y) Cartesian co-ordinate plane and are represented by the following kinematic model with associated non-holonomic constraints (that disallow the robot from sliding sideways).

$$\dot{x} = s \cos(\theta); \quad \dot{y} = s \sin(\theta); \quad \dot{\theta} = \omega \quad (1)$$

$$\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0. \quad (2)$$

Here  $s$  and  $\omega$  are the linear and angular speeds of the robot, respectively;  $x$ ,  $y$  and  $\theta$  are the coordinates of the robot with respect to the global (X, Y) coordinate system.

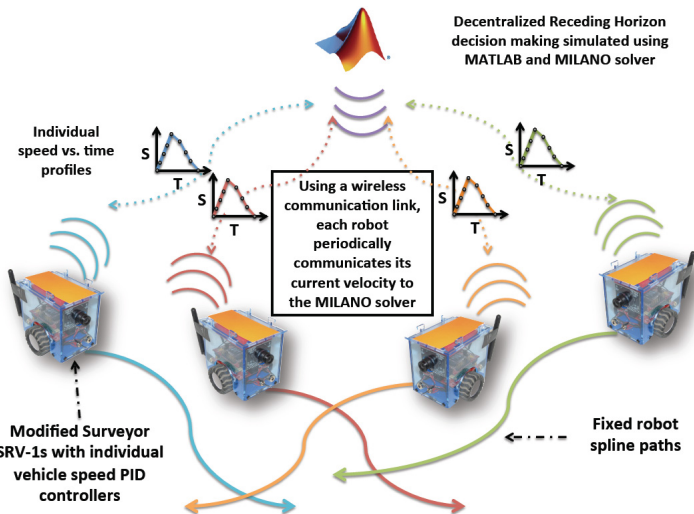
Each robot  $i = 1, \dots, N$  has a given start (origin) point  $o^i$  and a given end (goal) point  $e^i$ .  $O$  is the set of all start (origin) points.  $o^i \in O, \forall i = 1, \dots, N$ .  $E$  is the set of all end points.  $e^i \in E, \forall i = 1, \dots, N$ . The Euclidean distance between two robots  $i$  and  $j$  is denoted by  $d^{ij}$ . The robots are required to maintain a minimum safe distance  $d_{safe}$  from each other in order to avoid collisions. At any given discrete time step, the distance between the current location and the goal point for robot  $i$  is given by  $d_{goal}^i$ .  $s^i$  and  $\omega^i$  denote the speed and angular velocity, respectively, of robot  $i$  along its path at a given time.

Each robot  $i$  follows a fixed path represented by a two dimensional piecewise cubic spline curve of length  $U^i$ , which is obtained by combining two one dimensional piecewise cubic splines  $x(u)$ , and  $y(u)$ , where the parameter  $u$  is arc length along the curve. Let  $\kappa(u)$  be the curvature along the spline curve.

For each robot  $i$ ,

$$\omega^i(u) = s^i(u)\kappa^i(u) \quad (3)$$





**Fig. 1** Experimental approach for implementing the RH-MINLP algorithm for MVPC under communication connectivity constraints

These piecewise cubic splines have continuous first derivatives (slope) and second derivatives (curvature) along the curve. This property makes the path kinematically feasible. Furthermore, upper and lower bounds on the speed, acceleration and angular speed (turning rate) are enforced, thereby taking the robot dynamics into account. The paths represented by the two dimensional piecewise cubic splines along with the constraints on speed, accelerations and turn rates result in a kinodynamically feasible trajectory. For a detailed discussion on spline curve design and analysis, see [14] and its references. Other path primitives that result in twice continuously differentiable functions in our constraints such as quintic curves, polar splines, and cubic spirals can easily be accommodated using this framework.

## 2.2 Communication Model

Each robot is equipped with a wireless transceiver node. Consider two robots that try to communicate with each other at a given point in time. The Euclidean distance between them is denoted by  $d$ . The signal transmission power of the wireless node placed on the transmitter robot is denoted by  $P_{tr}$ . The received signal power of the wireless node placed on the receiver robot is denoted by  $P_r$ . The power experienced by the receiver robot node is calculated using Friis's equation [15]

$$P_r = P_{tr} G_t G_r \left( \frac{\lambda}{4\pi d} \right)^\alpha \quad (4)$$

where  $\alpha$  is the path loss exponent. The noise  $\sigma$  is assumed to be thermal ( $kTBF$ ).  $\lambda$  is the wavelength and is equal to  $c/f$ , where  $c = 3 \times 10^8$  m/s and  $f = 2.4 \times 10^9$  Hz. The values of  $G_t$  and  $G_r$  (antenna gains) are assumed here to be 1. The values of the path-loss exponent  $\alpha$  range from 1.6 (indoor with line of sight) to 6 (outdoor obstructed) depending on the environment.

The Signal to Noise Ratio (SNR) experienced by the receiver robot is calculated using the relationship  $\text{SNR} = P_r/\sigma$  to determine whether the robots are in communication range of each other. If the SNR experienced by a receiver node placed on a robot is above a predefined threshold  $\eta_c$ , the two robots are considered to be in communication range of each other. For a known  $P_{tr}$  and  $\alpha$ , the condition  $\text{SNR} \geq \eta_c$  can be modified appropriately as  $d \leq \eta_d$ , where  $\eta_d$  defines the maximum communication range beyond which path-loss results in loss of communication.

### 2.3 Receding Horizon

The parameter  $t$  represents steps in time.  $T_{hor}$  is the receding horizon time. At each time step  $t$ , each robot must calculate its plan for the next  $T_{hor}$  time steps, and communicate this plan with other robots in the network. While the robots compute their trajectory points and corresponding input commands for the next  $T_{hor}$  time steps, only the first of these solutions is implemented, and the process is repeated at each time step.  $T_{max}$  is the time taken by the last arriving robot to reach its end point. At  $t = T_{max}$  the scenario is completed. If a robot reaches the goal point before  $T_{max}$ , it continues to stay there until the mission is over. However, if required, it can still communicate with other robots.

Each robot plans its own trajectory by taking into account the plans of all other robots at each discrete time step. For a given time step  $t$ , each robot determines its speed for the next  $T_{hor}$  time steps starting at time  $t$  i.e.  $s^i(t), \dots, s^i(t + T_{hor})$  and implements the first speed  $s^i(t + 1)$  out of all these speeds. In this way the plan starting at time step  $t + 1$  must be computed during time step  $t$ . Thus during each time step  $t$ , each robot communicates the following information about its plan  $\mathcal{P}^i(t)$  to other robots:  $\mathcal{P}^i(t) = [\mathbf{p}^i(t) \dots \mathbf{p}^i(t + T_{hor})]$ , where  $\mathbf{p}^i(t) = (x^i(t), y^i(t))$  is the location of the robot  $i$  on its path at time  $t$  calculated based on the optimal speed  $s^i(t)$ .

### 2.4 Decision Ordering

The robots are assigned a pre-determined randomized decision order. The decentralized algorithm presented here sequentially cycles through each robot thereby allowing each robot to solve its planning problem in the order  $ord(i)$ ,  $i \in \{1, \dots, N\}$ .

### 3 Optimization Model Formulation

Each robot  $i$  solves the optimization problem  $\mathcal{O}^i(t)$  indicated by (5)-(17) in the order  $ord(i)$  that it has been assigned.

$$\text{minimize } \sum_{k=t}^{t+T_{hor}} d_{goal}^i(k) \quad (5)$$

$$\text{subject to } \forall j \in \{1, \dots, N\}, j \neq i$$

$$\forall k \in \{t, \dots, t + T_{hor}\}$$

$$(x^i(0), y^i(0)) = o^i \quad (6)$$

$$u^i(0) = 0 \quad (7)$$

$$u^i(k) \leq U^i \quad (8)$$

$$u^i(k) = u^i(k-1) + s^i(k)\Delta t \quad (9)$$

$$(x^i(k), y^i(k)) = ps^i(u^i(k)) \quad (10)$$

$$s_{min} \leq s^i(k) \leq s_{max} \quad (11)$$

$$a_{min} \leq a^i(k) \leq a_{max} \quad (12)$$

$$d_{goal}^i(k) = U^i - u^i(k) \quad (13)$$

$$d^{ij}(k) \geq d_{safe} \quad (14)$$

$$d^{ij}(k) \leq M(1 - C^{ij}(k)) + \eta_d \quad (15)$$

$$\sum_{j:j \neq i} C^{ij}(k) \geq n_{conn} \quad (16)$$

$$C^{ij}(k) \in \{0, 1\} \quad (17)$$

#### 3.1 Decision Variables

In (5)-(17), the main decision variables are the speeds,  $s^i(t), \dots, s^i(t + T_{hor})$ , for robot  $i$  at time  $t$ . The values of the remaining variables are dependent on the speeds.

#### 3.2 Objective Function

Equation (5) represents the objective function to be minimized. This formulation forces the robots to minimize the total distance between their current location and the goal position over the entire receding horizon. Constraint (13) defines the distance to goal  $d_{goal}^i(k)$  for each robot  $i = 1 \dots N$  at time-step  $k$ ,  $\forall k \in \{t, \dots, t + T_{hor}\}$  as the difference between its path length  $U^i$

and the total arc length travelled  $u^i(k)$ . The choice of this objective function results in the robots not stalling and moving to their goal position as fast as possible (minimum time solution).

### 3.3 Path (Kinematic) Constraints

Constraints (6)-(10) define the path of each robot. The constraints (6), (7), and (8) form the boundary conditions. Constraint (6) indicates that each robot  $i$  has to start at a designated start point  $o^i$ . Constraint (7) initializes the arc length travelled  $u$  to zero value. Constraint (8) provides the upper bound on the arc length travelled. Constraint (9) increments the arc length at each time step based on the speed of the robot ( $\Delta t = 1$ ). Constraint (10) ensures that the robots follow their respective paths as defined by the cubic splines. The function  $ps^i(u^i(k))$  denotes the location of robot  $i$  at time step  $k$ ,  $\forall k \in \{t, \dots, t + T_{hor}\}$  after travelling an arc length of  $u^i(k)$  along the piecewise cubic spline curves. It should be noted that the constraint (10) is a non-convex nonlinear equality constraint.

### 3.4 Speed and Acceleration (Dynamic) Constraint

Constraints (11)-(12) are dynamic constraints and ensure that the speed  $s^i(k)$  (and hence, angular velocity) and the acceleration  $a^i(k)$  for each robot  $i = 1, \dots, N$  at each time-step  $k$ ,  $\forall k \in \{t, \dots, t + T_{hor}\}$  are bounded from above (by  $s_{max}$  and  $a_{max}$  respectively) and below (by  $s_{min}$  and  $a_{min}$  respectively). These constraints are determined by the capabilities of the robot and the curvature  $\kappa(u)$  of the paths represented by the spline curve. Here we assume that the curvature of the paths is within the achievable bounds of the angular speed and radial acceleration of the robots. Hence the angular speed required by the robots corresponding to the optimal speed is always achievable, and can be determined by (3).

### 3.5 Collision Avoidance Constraint

The non-convex constraint (14) ensures that there is a sufficiently large distance  $d_{safe}$  between each pair of robots to avoid a collision at all times.

### 3.6 Communication Connectivity Constraint

Constraints (16) and (17) state that vehicle  $i$  should be in communication range of at least  $n_{conn}$  vehicles. This means that, for at least  $n_{conn}$  values of  $j = 1, \dots, N$ ,  $j \neq i$ , the condition  $d^{ij} \leq \eta_d$  should be satisfied. The remaining vehicles may or may not be in communication range of  $i$ . In order to express this requirement, we introduce a constant  $M$  and formulate constraint (15),

which states that if  $C^{ij}(k) = 1$  then vehicles  $i$  and  $j$  are within communication range. If  $C^{ij}(k) = 0$ , then the constraint will be trivially satisfied for a sufficiently large  $M$ . Constraint (15) is an example of a big-M constraint [16].

In the numerical implementation, an equivalent form of constraint (16) as  $\frac{(d^{ij}(k))^2}{(M(1 - C^{ij}(k)) + \eta_d)} \leq (M(1 - C^{ij}(k)) + \eta_d)$  is used, in order to avoid the nondifferentiability of a Euclidean distance calculation within the nonlinear solver. The nondifferentiability is not going to occur at the optimal solution due to the collision avoidance constraint keeping  $d^{ij}$  sufficiently large, but during the initial iterations of the MILANO solver, it may cause numerical difficulties. The reformulation removes the potential of such an occurrence and provides numerical stability.

### 3.7 RH-MINLP Algorithm

All robots are initially assumed to be in communication range of each other. The general outline of the algorithm is as follows:

For any time step  $t$ , let each robot  $i$  implement the following algorithm:

**Start:** Start at time  $t$

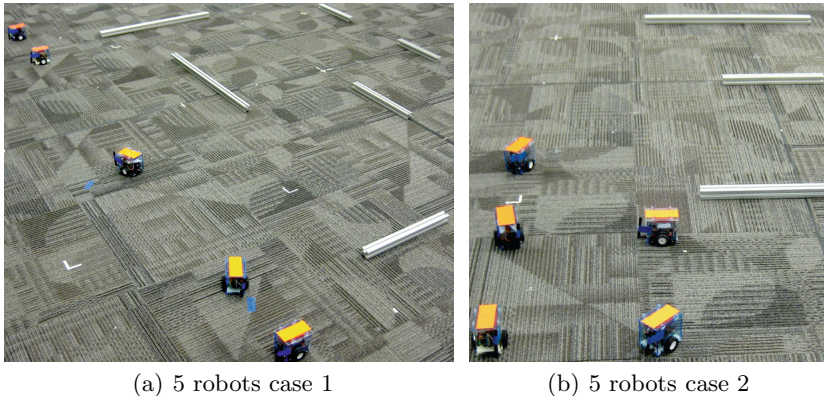
- **Step 0** - An order is enforced in terms of which robot plans its trajectories first. The ordering can be randomly assigned or can be assigned a priori.
- **Step 1** - Based on its decision order  $ord(i)$ , each robot  $i$  solves the problem  $\mathcal{O}^i(t+1)$  at time  $t$  by taking into account the following plans:
  - 1.1 Plans  $\mathcal{P}^j(t+1)$  for robots  $j$ ,  $\forall j \in \{1, \dots, N\}$ ,  $j \neq i$  whose  $ord(j) < ord(i)$  - these robots have already calculated their new plans, and
  - 1.2 Plans  $\mathcal{P}^\zeta(t)$  for robots  $\zeta$ ,  $\forall \zeta \in \{1, \dots, N\}$ ,  $\zeta \neq i$  whose  $ord(i) < ord(\zeta)$  - these robots are yet to calculate their new plans.
- **Step 2**
  - 2.1 If a feasible solution is found, the new plan is  $\mathcal{P}^i(t+1)$ .
  - 2.2 If  $\mathcal{O}^i(t+1)$  is infeasible then use the previously available plan  $\mathcal{P}^i(t)$  for the next  $T_{hor} - 1$  time steps i.e. the new plan

$$\mathcal{P}^i(t+1) = \mathcal{P}^i(t) \setminus \mathbf{p}^i(t) \quad (18)$$

where  $\mathbf{p}^i(t) = (x^i(t), y^i(t))$

- **Step 3** - Broadcast this plan to the other robots.

**End:** End by  $t+1$ , and repeat



**Fig. 2** Five mSRV-1 robots at their starting points in their workspace with static obstacles. Two cases studied during the experimentation are shown above.

## 4 Experiments

Experimental validation of the proposed strategies was conducted using our multi-robot testbed which consists of 5 modified SRV-1 robots (mSRV-1) in a 4.5x4.5 meter workspace shown in Figure 2. The mSRV-1 is a modified version of the Surveyor, Inc. SRV-1 robot that is equipped with a 600 MHz Blackfin embedded processor, 802.11b wireless communication, wheel encoders, and a color camera. Localization for the individual robots was provided by a network of overhead cameras. Optimal speeds for each robot, determined by the proposed optimization framework, is achieved via a low-level trajectory-following PID controller. Table 1 provides the parameters of the experimental setup. The decentralized RH-MINLP algorithm is simulated on a computer using the MATLAB-MILANO combination. MILANO is a MATLAB-based solver for MILP and MINLP problems. It uses a branch-and-bound method for handling integer variables, and an interior-point method for solving the nonlinear relaxations. Source code for MILANO has been made available online [17].

**Table 1** Parameter values for experiments

$T_{hor}$	3	$s_{min}$	0	$a_{min}$	-0.33 m/s <sup>2</sup>
$M$	10	$s_{max}$	0.33 m/s	$a_{max}$	0.33 m/s <sup>2</sup>

### 4.1 Experimental Scenarios

To understand the practical effects of inter-robot communication on motion planning, the following two sets of experiments were studied:

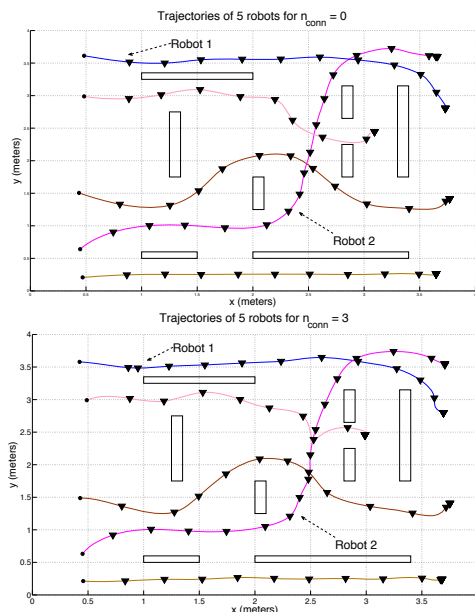
1. **Pre-planned speed profiles:** In the first set of experiments, the MVPC problem was pre-solved and the pre-planned optimal linear and angular speeds were communicated to the robots, one step at a time.
2. **Real-time speed profile generation:** In the second set of experiments, real-time location feedback provided by the motion-tracking system was incorporated into the optimization problem. At each discrete time-step, the real-time location feedback was utilized to calculate the speed profile for the next  $T_{hor}$  time steps. The linear and angular speeds for the next time time step were communicated to the robots.

## 4.2 Experimental Results and Insights

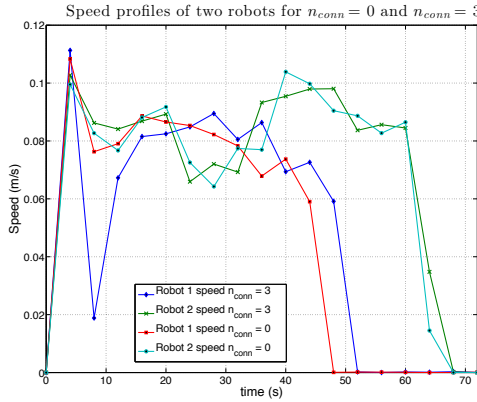
Several aspects of the MVPC problem were analyzed during the experiments documented in this section. Specifically, the focus was on studying

- The effect of communication constraints on the velocity profiles of the robots and on the solution computation times, and
- The effects of sensing and actuation noise on the performance of the robots.

For an experiment where  $\eta_d = 2.9\text{m}$ , Figure 3 shows the trajectories for first case of the 5 robot scenario shown in Figure 2(a) for  $n_{conn} = 0$  (no communication connectivity requirement) and  $n_{conn} = 3$ . While the optimization



**Fig. 3** Effect of change in  $n_{conn}$  on the trajectory of Robot 1 for the first case of the 5 robot scenario shown in Figure 2(a)



**Fig. 4** Speed profiles of Robots 1 and 2 for the first case of the 5 robot scenario shown in Figure 2(a)

allows for more stringent communication connectivity requirements without significantly degrading the scenario completion time, the speed profiles of the individual robots are affected. The most visible changes are observed in Robot 1 trajectory as evidenced by the triangular markings that represent the position of the robots along their paths while following their optimal speed profiles.

Figure 4 shows the velocity profiles of both robots for these two scenarios. Typically, the robots whose times of arrival at their respective destinations are less than the scenario completion time change their speed profiles to comply with the new communication constraint. For this experiment, Robot 2 happens to be the last arriving robot. As seen in Figure 4, the arrival to destination time of Robot 2 does not change.

$T_{max}$  for this experiment increases from 61.6 seconds to 62.4 seconds as  $n_{conn}$  goes from 0 to 3. Table 2 enlists  $T_{max}$  for a variety of experiments. It is observed that even with more stringent communication connectivity requirements, there is no appreciable increase in  $T_{max}$  values. Again, this is attributed to the fact that the faster robots slow down to accommodate the more stringent communication connectivity requirements.

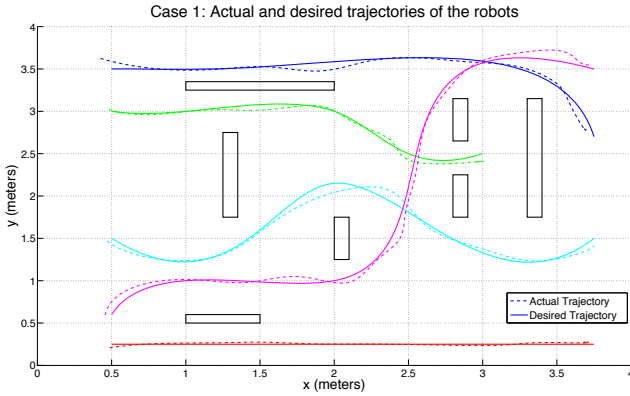
**Table 2**  $T_{max}$  in seconds for various pre-planned scenarios

$N$	$n_{conn}$			
	0	1	2	3
2 ( $\eta_d = 1.65$ m)	51.22	51.27	-	-
3 ( $\eta_d = 1.65$ m)	51.44	52.22	51.66	-
4 ( $\eta_d = 2$ m)	52.56	52.83	52.39	52.42
5 ( $\eta_d = 2.9$ m)	61.60	61.60	64	62.40

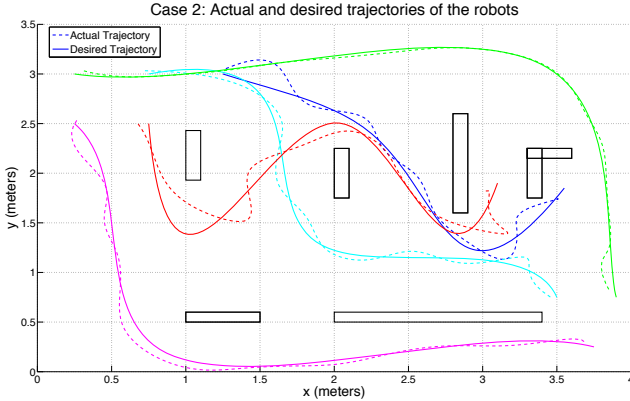


**Table 3** Average *RMSE* in meters for two 5-robot scenarios (3 trials per scenario)

Case	Pre-planned	Real-time feedback
Figure 2(a)	0.0612	0.0507
Figure 2(b)	0.0873	0.0836



(a) Actual and desired paths of the robots in the 5 robots case 1



(b) Actual and desired paths of the robots in the 5 robots case 2

**Fig. 5** Effect of sensor and actuation noise as demonstrated by the *RMSE* between the desired and the actual paths. Paths whose curvature changes quickly result in high *RMSE* values.

Post-processing the overhead localization data allows to quantify the effects of sensing and actuation noise on the performance of the robots by calculating the *RMSE* between the desired robot paths and the actual paths traversed by them. The actual paths traversed by the robots are ascertained

by recreating cubic splines using position feedback provided by the overhead cameras. For the 5-robot scenarios depicted in Figure 5, it was observed that in general the real-time feedback reduced the *RMSE* as evidenced in the Table. 3. The actual and desired robot paths for this case are shown in Figure 5(a). The RMSE becomes particularly pronounced when the curvature of the fixed paths changes rapidly between successive discrete time steps as shown in 5-robot scenario of Figure 5(b).

During the experiments, it was found the decision order  $ord(i)$  of the robots  $i = 1, \dots, n$  can qualitatively affect the solution of each robot depending on the geometry of the paths. Due to the inherent decentralized decision making, certain robots' decisions may render the coordination problem difficult to solve for other robots. In some cases, reassigning a different decision order  $ord(i)$  of the robots  $i = 1, \dots, n$  helped improve overall solutions. Also in some cases, certain robots' decisions can render the coordination problem infeasible for other robots regardless of the decision ordering used. In such cases, the robots may use their plans from the previous time steps as indicated by Step 2.2 of the RH-MINLP algorithm.

## 5 Conclusion and Lessons Learned

A decentralized RH-MINLP formulation for solving multi-vehicle path coordination problems under communication constraints was presented and experimentally verified. The effect of communication connectivity constraints on the robot velocity profiles and experimental runtimes were reported. The effects of sensor and actuation noise were quantified in terms of the RMSE between the desired path and actual path traversed by the robots.

Mathematical Programming (MP) provides a powerful framework for solving communication -centric multi-vehicle motion planning problems. The proposed framework and associated experimental results facilitate the practical study of the trade-offs between mobile robot trajectories and inter-vehicle communication connectivity requirements. The results provide an insight into the technical challenges associated with the implementation of MP, specifically MINLP, based motion planning. While these results are encouraging, there are several challenges associated with the practical implementation of an MP-based motion planning framework that still need to be addressed. In addition to the kinodynamic, collision avoidance, and communication connectivity constraints, it is important to account for the constraints imposed from noise in the sensors and the errors of the actuators in implementing the optimal commands perfectly. While the RMSE values reflect the performance of lower-level robot controllers and allow for better gain selection, ultimately the control issues need to be reflected in the MP formulations.

While MP-based frameworks prove to be effective for a small number of robots, in order for them to be used with a large number of robots, the computational bottlenecks resulting from non-convexities and/or the size of

the problem should be handled effectively. In recent work [18], highly efficient code has been written to implement convex optimization solution algorithms in real time. The effectiveness of these algorithms in solving multi-vehicle motion planning problems remains to be seen. Future work will focus on developing formulations that can capture lower-level robot control issues and that are amenable to fast solution times for a large number of robots.

## References

1. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Norwell (1991)
2. Todt, E., Raush, G., Sukez, R.: Analysis and Classification of Multiple Robot Coordination Methods. In: 2000 IEEE International Conference on Robotics and Automation (ICRA), pp. 3158–3163. IEEE Press, San Francisco (2000)
3. O’Donnell, P.A., Lozano-Perez, T.Z.: Deadlock-free and collision-free coordination of two robot manipulators. In: 1989 IEEE International Conference on Robotics and Automation (ICRA), Scottsdale, pp. 484–489. IEEE Press (1989)
4. LaValle, S., Hutchinson, S.: Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation* 14, 912–925 (1998)
5. Simeon, T., Leroy, S., Laumond, J.: Path coordination for multiple mobile robots: a resolution-complete algorithm. *IEEE Transactions on Robotics and Automation* 18, 42–49 (2002)
6. Peng, J., Akella, S.: Coordinating Multiple Robots with Kinodynamic Constraints along Specified Paths. *The Int. J. of Robotics Research* 24, 295–310 (2005)
7. Abichandani, P., Benson, H.Y., Kam, M.: Multi-Vehicle Path Coordination under Communication Constraints. In: 2008 American Control Conference (ACC), Seattle, pp. 650–656. IEEE Press (2008)
8. Abichandani, P., Benson, H.Y., Kam, M.: Multi-Vehicle Path Coordination in Support of Communication. In: 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, pp. 3237–3244. IEEE Press (2009)
9. Abichandani, P., Benson, H.Y., Kam, M.: Decentralized Multi-Vehicle Path Coordination under Communication Constraints. In: 2011 International Conference on Robotic Systems (IROS), San Francisco, pp. 2306–2313 (2011)
10. Abichandani, P., Ford, G., Benson, H.Y., Kam, M.: Mathematical Programming for Multi-Vehicle Motion Planning Problems. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), St. Paul, pp. 3315–3322. IEEE Press (2012)
11. Schouwenaars, T., How, J., Feron, E.: Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees. In: 2004 AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence (2004)
12. Mellinger, D., Kushleyev, A., Kumar, V.: Mixed-Integer Quadratic Program Trajectory Generation for Heterogeneous Quadrotor Teams. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), St. Paul, pp. 477–483. IEEE Press (2012)

13. Benson, H.Y.: Using Interior-Point Methods within an Outer Approximation Framework for Mixed Integer Nonlinear Programming. In: Lee, J., Leyffer, S. (eds.) *Mixed Integer Nonlinear Programming, The IMA Volumes in Mathematics and its Applications 2012*, vol. 4128, pp. 225–243. Springer, New York (2006)
14. Lepetic, M., Klancar, G., Skrjanc, I., Matko, D., Potocnik, P.: Time optimal path planning considering acceleration limits. *Robotics and Autonomous Systems* 45, 199–210 (2003)
15. Friis, H.T.: A Note on a Simple Transmission Formula. In: *Proceedings of the 1946 IRE*, vol. 34, pp. 254–256 (1946)
16. Bemporad, A., Morari, M.: Control of Systems Integrating Logic, Dynamics, and Constraints. *Automatica* 35, 407–427 (1999)
17. Benson, H.Y.: MILANO: Mixed-Integer Linear and Nonlinear Optimizer, <http://www.pages.drexel.edu/~hvb22/milano/>
18. Mattingley, J., Boyd, S.: Automatic Code Generation for Real-Time Convex Optimization. In: Eldar, Y., Palomar, D. (eds.) *Convex Optimization in Signal Processing and Communications*. Cambridge University Press (2009)

# Proactively Approaching Pedestrians with an Autonomous Mobile Robot in Urban Environments

Daniel Carton, Annemarie Turnwald, Dirk Wollherr, and Martin Buss

**Abstract.** This paper presents a trajectory planning method enabling autonomous robots to approach people in dynamic environments to initiate a conversation proactively. It is shown how integrating human inspired parameters in optimal control based motion planning enables people to predict and read the purpose of a motion more easily.

Experimental evaluations in literature propose to incorporate human-like aspects since the intended action becomes more comprehensible for humans. Therefore, factors like approach speed, distance to the person, positioning near the person, trajectory shape, and the avoidance method are adopted from human behavior to generate motions. The presented trajectory planner is designed to improve the human-like appearance of an approach motion implementing these aspects. Human-likeness is evaluated according to naturalness and comfort of the approach behavior. By executing corresponding trajectories, the approach movement appears more natural and the intended action is easier to predict for humans.

This paper formulates the motion planning procedure as an optimal control problem. Human-like behavior is generated through specific constraints and cost. In order to achieve correct timing, appropriate trajectory shape and the desired behavior for collision avoidance in a dynamic environment, the optimization is split into three consecutive steps. An implementation of a planning algorithm for dynamic environments, capable of online replanning, is proposed. Experiments conducted with this system showed the appropriateness of speed and distance parameters. Further statistical results

---

Daniel Carton · Annemarie Turnwald · Dirk Wollherr · Martin Buss  
Institute of Automatic Control Engineering (LSR),  
Technische Universität München, D-80290 Munich, Germany  
e-mail: {dc,a.turnwald,dw,mb}@tum.de

Dirk Wollherr · Martin Buss  
Institute for Advanced Study, Technische Universität München, D-85748 Garching,  
Germany

confirmed that the shape of a trajectory significantly affects the naturalness of an approach motion.

## 1 Introduction

Latest developments in service robotics, increase the necessity for information exchange between human collaborators and robot assistants due to the complexity of collaborative tasks. There are two scenarios where an initiation of a conversation is needed: to provide/to retrieve information to/from humans. While issuing information would usually leave the initiative to humans, a robot is required to approach humans proactively to gather information from interaction. Thus, instead of exclusively reacting, a robot has to commence an interaction in the latter case. This imposes a trajectory planning problem which is tackled in this paper. A robot has to approach a person in order to get close enough for an interaction while simultaneously drawing the person's attention.

It is shown in literature [4] that humans understand more reliably what a robot intends when it executes human-like motions. It is inconvenient for nearby humans if motions executed by a robot are not easily predictable and interpretable. This gains special importance in crowded areas and is termed readability in [30]. Accordingly, the goal is to improve the readability (predictability) of motions executed by a robot for humans in its vicinity. Following this, aspects listed in literature for human navigation in a social context are adopted: smooth trajectory shapes, specified approach speed, appropriate human-robot distance, positioning in the field of view and the behavior for human-like dynamic obstacle avoidance [18]. These social parameters help drawing the person's attention without creating an obtrusive or obliging situation. Thus, the approached person decides whether the interaction request is accepted. In order to assess the readability of an approach motion, conditions from social psychology like naturalness, comfort, or sensation are used.

In this paper we expand optimal control based motion planning by social elements to model intuitively readable human-like movements. Resulting trajectories let the approach movement look more natural thereby showing the robot intention more apparently. Integrated on a robotic platform the system facilitates the ability to approach walking persons applying spatio-temporal planning with incorporation of motion prediction methods. Dynamic obstacle avoidance is realized during the planning step by slowing down or speeding up.

The remainder of this paper is organized as follows. The subsequent Sec. 2 describes the use case for the proposed approach. Sec. 3 discusses related work by pointing out parallels and contributions. An optimal control based problem formulation is given in Sec. 4. Details regarding our exemplary implementation are presented in Sec. 5. Experimental results based on this

realization are shown in Sec. 6. Conclusions are drawn in Sec. 7 where future work is proposed as well.

## 2 Experimental Application Scenario: The IURO Project

Outside of experiment-taylorred laboratory settings the world is constantly changing its appearance and requirements for robotic systems. Therein, an autonomous robot is confronted with a large variety of situations that challenge it in performing its task. In order to interpret the environment and select appropriate actions, robots require knowledge. Many approaches favor pre-programmed or learned knowledge to cope with the vitality of dynamic environments. However, as situations and objectives may change unpredictably it is inevitable that an autonomous system is confronted with a situation the robot designer has not foreseen. Learning technologies are often too time consuming to resolve such situations. Yet, overcoming this problem requires the robot to deal with this knowledge gap. One viable resource for acquiring the missing information is asking human passers-by. Requesting help from a person raises the question what abilities robots need to successfully retrieve missing information from humans and how they are supposed to act. This is the central question investigated in the Interactive Urban RObot (IURO) project.

Within IURO the paradigm of missing prior information that has to be gathered from human-robot interactions is the focal point. A robot without map information, internet access, or GPS support is placed in an urban environment with the task of reaching a designated place within the city. Human passers-by are the only source of information available to establish a hypothesis about the goal location. Generating this hypothesis from human-robot interactions demands perception abilities, proactive behavior, modalities for internal state representation to carry the intention, as well as communication capabilities. More specifically the IURO robot is facing the following multi-disciplinary challenges: Firstly a specialized mechatronic platform optimized for outdoor performance is required. Environment perception algorithms allow for understanding scenes. Communication in natural dialogs is used to query specific information. Aspects of nonverbal communication let the interaction appear more natural and pleasant. Finally navigation in dynamic environments is the key ability to reach another location.

This paper elaborates the specific navigation problem of approaching a human in an urban environment. This ability is crucial for the IURO robot as it is the entry point for every interaction which the system heavily relies on. For this reason, the motion planning is construed for proactive and socially acceptable approaching behavior as well as good readability of the intention as these aspects are assumed to increase the number of successful interaction initiations.

### 3 Classification within the State of the Art

Approaching humans requires a fusion of multiple research areas to generate well readable motions. Fundamentals in trajectory planning are applied to reach a goal pose in front of the approached target. Motion prediction is used to estimate the position of this goal pose. Dynamic obstacle avoidance is necessary because in a dynamic environment the target person as well as passers-by have an impact on the robot trajectory. Lastly, to achieve human-like motions that increase the readability of an approach, user studies with a focus on human approach are taken into account. Subsequently, similarities and differences between the work at hand and publications in the mentioned fields are presented.

Trajectory planning includes the generation of time profiles along with the path which is required to navigate in dynamic environments. This field features a wide range of publications, so that we only address the most related ones. Latombe [19] and LaValle [20] are often referred to as groundwork for motion planning. These standard algorithms are also applied in [35] which focuses on shortest/fastest path search, neglecting the path shape and curvature constraints. Indeed, [26] indicates that smooth and jerk limited paths are considered as more convenient by humans. These trajectory properties are satisfied by Bézier curves as Choi et al. show in [6] and [8]. The approach of optimizing Bézier curve parameters is adopted in this work. Fraichard [13] includes timing by means of a state-time space for planning optimal paths in a dynamic environment with moving obstacles. In [11] a library of pre-defined velocity profiles for different purposes is applied to an optimization based trajectory planner. Similarly, our work uses suitable velocity profiles for readability improvement but generates them during an optimization process. This allows for adjusting velocities during path execution in order to react to dynamic changes in the environment or to slow down when the target person is close.

Many researchers employ the velocity obstacle space for path planning and dynamic obstacle avoidance [34]. Fiorini [12] uses this space to optimize conservative trajectories with respect to time. In [27] velocity obstacles are utilized in an adaptive time horizon developing a representation for absolutely safe velocities. Masehian et al. [21] intercept a target and avoid static and moving obstacles by evaluating the set of all collision free directions. The approach proposed in this paper resembles these methods but also considers social acceptance parameters.

Reaching an appropriate final pose at about the same time as a moving target person requires movement prediction. Besides, the planned trajectory needs to take crossing passers-by into account. In [3] Bennewitz et al. realize motion prediction for humans using Gaussian-Mixture-Models. Another probabilistic approach is shown in [14] where Partially Observable Markov Decision Processes (POMDPs) are employed. Statistical data association combined with a particle filter predict motions in [23]. A grid model



containing motion probabilities is shown in [31]. Prediction of future poses based on Kalman filtering is described in [9, 29]. In favor of its computational efficiency a Kalman filter is implemented in our approach that assumes humans to be walking on a straight line and at constant velocity.

User studies yield insight into the effects of robot behavior on human attitude towards the machine. A variety of research groups run experiments to reveal the underlying aspects. Edward Hall [15] introduces the proxemics model which is considered a groundwork for many interaction scenarios. Walters et al. [32, 33] analyze the comfortable approach distance and estimate it to be between 0.5 m and 1.2 m. This was reconfirmed in [5] which also examined favorable approach poses, along with [16, 24, 32, 36], for static and moving persons. Preferable approach directions are frontal left and right with no significant difference for walking people. In [5] it is further shown that appropriate approach speed and slowing down from 0.6 m/s to 0.4 m/s nearby the approached person has a positive effect on the comfort level. Autonomous approach behaviors implemented on a real robot are less common. The human aware motion planner (HAMP) [25] incorporates social aspects that consider a person's position, posture and field of view. A planner based on the Dynamic Window Approach is proposed in [17] which addresses social acceptance by considering proxemics. Yet, the implemented approach is not capable of approaching moving persons. In [22, 28] a moving person is approached in a shopping mall using path prediction. It is shown that proactive behavior improves the success rate for interaction initiations but without integrating further human-like aspects. The shopping mall setting, however, is prepared to be fully observable in contrast to our setting where outdoor navigation without map knowledge is considered.

## 4 Problem Statement

This section defines the robot trajectory planning problem for human approach in dynamic environments. From a given initial pose the robot has to plan its trajectory to reach a goal pose (position and orientation). The goal pose is estimated from the position and orientation of the target person or from the path prediction step for moving persons. The path planner has to find a valid path from the initial to the final pose while avoiding static objects. In order to keep the trajectory shape fixed and avoid dynamic obstacles the velocity profile has to be adapted. Concurrently, the overall timing constraint of reaching the goal pose simultaneously with the person has to be taken into account. If emerging objects lead to a blocked path or, for the dynamic situation, the robot can not reach the goal at approximately the same time as the target, the path needs to be replanned to a reachable predicted goal pose.

In order to meet the timing constraint, a superordinate optimization is defined that is solved iteratively after every successful generation of a trajectory

with the associated velocity profile. From these steps a minimum time  $t_{\min}$ , required by the robot to reach the goal pose, is received if the actual goal pose is not reachable within the maximum time horizon  $t_E$ , where  $t_E$  is defined as the time available to the robot to reach the currently selected goal pose. The possible approach timings are discretized in absolute time. Then a minimization of the needed time between the robot start position  $\mathbf{x}_R(\lambda(0)) = (x(0), y(0))^T$  and the goal position  $\mathbf{x}_G = \mathbf{x}_R(\lambda(t_E)) = (x(t_E), y(t_E))^T$  is aimed for. Basically, a search for the fastest approach is performed taking into account multiple solutions. This search is restarted if a new  $t_{\min}$  is provided. Formulated as a nonlinear optimization problem with the objective to find a minimum for  $t_E$  minimizing  $J_{\text{time}}$  while subject to the respective constraints:

$$\begin{aligned}
 & \text{find } t_E \\
 & \min J_{\text{time}}(t_E) = \int_{t_{\min}}^{t_E} 1 \, dt \\
 & \text{s.t. } t_E = v_H s_H, \quad t_E > t_{\min}, \quad t_{\min, \text{init}} = 0, \quad \lambda(0) = 0, \quad \lambda(t_E) = 1, \\
 & \quad \mathbf{x}_R(\lambda(t)) = B(\lambda(t), \mathbf{p}_n), \quad \mathbf{x}_G(\lambda(t_E)), \quad \mathcal{O} = \bigcup_i \mathcal{O}_i
 \end{aligned} \tag{1}$$

where  $v_H \in \mathbb{R}$  is the estimated speed of the target person in the velocity space  $\mathcal{V}$ ,  $s_H$  is the distance the person has traversed during  $t_E$  and  $\lambda(t)$  is a monotonically increasing function that maps the time horizon  $[0, t_E]$  to the curve parameter  $0 \leq \lambda \leq 1$ . The Bézier solution  $B(\lambda(t), \mathbf{p}_n)$  is a special parametrized solution subspace to the path planning optimal control problem. This subspace is utilized in reference to the attributes of Bézier curves depicted later in Sec. 5. Bézier curves of degree  $n$  are defined by their control points  $\mathbf{p}_n = (x, y)^T$ . The workspace for the robot is denoted as  $\mathcal{W}$  and the occupied space corresponds to the trajectory  $B(\lambda(t), \mathbf{p}_n) \subset \mathcal{W}$ . Objects  $\mathcal{O}_i$  and moving objects  $\mathcal{O}_i(t)$  are summed up as a unified occupancy  $\mathcal{O} = \bigcup_i \mathcal{O}_i$ . The starting point  $\mathbf{x}_R(\lambda(0)) \subset \mathcal{W}$  and the human's position  $\mathbf{x}_H \subset \mathcal{W}$  are known from robot localization and target tracking. The term  $t_A$  depicts The starting time,  $t$  the actual time and  $t_E$  is the maximum time to the currently selected goal pose  $\mathbf{x}_G \subset \mathcal{W}$  in front of the person at  $\mathbf{x}_H$ .

As a basic concept in trajectory planning a non-holonomic robot is used to let the movement appear natural as proposed in [1] and to avoid trajectories where a turning on spot is part of the approach. Moreover we split the optimization for motion planning into two consecutive steps. Firstly, a path of minimum length in the static environment is generated using Bézier curves avoiding objects by shaping the curve accordingly. Secondly, velocity and parameters constituting social aspects are optimized and generate the velocity profile on the trajectory.

The trajectory generation is formulated as a minimization of the path length:

$$J_{\text{path}}(\mathbf{p}_n) = \int_{\mathbf{x}_R(\lambda(0))}^{\mathbf{x}_R(\lambda(t_E))=\mathbf{x}_G} ds + \theta_1 \int_{\lambda(0)}^{\lambda(t_E)} \left( \frac{\mathbf{x}_R(\lambda(t)) - \mathcal{O}_i}{\|\mathbf{x}_R(\lambda(t)) - \mathcal{O}_i\|} \right)^{-1} dt \quad (2)$$

with the respective constraints:

$$\mathbf{x}_R(\lambda(t)) = B(\lambda(t), \mathbf{p}_n), \quad d_{\min} \leq \|\mathbf{x}_G(\lambda(t_E)) - \mathbf{x}_H(t_E)\| \leq d_{\max}, \quad |\kappa| \leq \kappa_{\max}$$

where  $\mathbf{x}_R$  is the robot position on a Bézier curve  $B(\lambda(t), \mathbf{p}_n)$  of degree  $n = 4$  with  $0 \leq \lambda \leq 1$ , the adjustable control point  $\mathbf{p}_2$  and the defined control points  $\mathbf{p}_0 = \mathbf{x}_R(\lambda(0))$ ,  $\mathbf{p}_1 = f(\mathbf{p}_0)$ ,  $\mathbf{p}_3 = f(\mathbf{p}_4, x_H(t_E))$  and  $\mathbf{p}_4 = \mathbf{x}_R(\lambda(t_E)) = \mathbf{x}_G(\lambda(t_E))$ .

$$\mathbf{x}_R(\lambda(t)) = B(\lambda(t), \mathbf{p}_n) = \sum_{i=0}^n \binom{n}{i} (1 - \lambda(t))^{n-i} t^i \mathbf{p}_i \quad (3)$$

The goal position  $\mathbf{x}_G(\lambda(t_E))$  on the predicted linear trajectory, is assumed to be fixed for the current time step. This pose will be modified over time if the second optimization step fails to meet the timing constraints. The curvature  $\kappa$  is denoted as:

$$\kappa = \frac{\dot{x}(\lambda(t))\ddot{y}(\lambda(t)) - \dot{y}(\lambda(t))\ddot{x}(\lambda(t))}{[\dot{x}^2(\lambda(t)) + \dot{y}^2(\lambda(t))]^{\frac{3}{2}}} \quad (4)$$

and  $\theta_1$  is an arbitrary parameter weighting the distance the robot should keep from obstacles. The nonlinear optimization problem with the objective to find a free control point  $\mathbf{p}_2$  in the Bézier curve that minimizes  $J_{\text{path}}$  under the respective constraints is defined as follows:

$$\begin{aligned} & \text{find } \mathbf{p}_2 \\ & \min J_{\text{path}}(\mathbf{p}_2) \\ & \text{s.t. } \mathbf{x}_R(\lambda(t)) = B(\lambda(t), \mathbf{p}_2) \\ & \quad d_{\min} \leq \|\mathbf{x}_G(\lambda(t_E)) - \mathbf{x}_H(t_E)\| \leq d_{\max}, \quad \kappa_{\min} \leq \kappa \leq \kappa_{\max} \end{aligned} \quad (5)$$

In order to generate velocity profiles that support the acceleration and deceleration for dynamic obstacle avoidance, the obstacles  $\mathcal{O}$  are transformed into the Bézier curve space  $\lambda(t)$  yielding  $\mathcal{O}_i(\lambda(t))$ . An obstacle occupies a certain area in this space depending on its dimensions and the time it takes to cross the trajectory. A two-point-boundary-value problem is formulated resulting in a velocity profile. The following costs apply:

$$J_{\text{vel}}(\dot{\lambda}(t)) = \int_{t_A=0}^{t_E} \dot{\lambda}^2(t) dt + \theta_2 \int_{t_A=0}^{t_E} \frac{1}{\lambda(t) - \mathcal{O}_i(\lambda(t))} dt + \theta_3 \frac{\dot{\lambda}(t)}{1 - \lambda(t)} \quad (6)$$

With the constraints:

$$\begin{aligned} t_E &= v_H s_H, \quad \lambda(0) = 0, \quad \lambda(t_E) = 1 \\ 0 \leq v \leq v_{\max}, \quad t &= t_E \text{ if } \lambda(t) = 1, \quad v = 0 \text{ if } \lambda(t) = 1 \end{aligned}$$

where a new ending time  $t_E$  has to be found if  $\lambda(t) < 1$  and  $t \geq t_E$ .  $v_R(t) = \dot{\lambda}(t) \subset \mathcal{V}$  is the velocity profile followed by the robot and  $\theta_2$  as well as  $\theta_3$  are arbitrary parameters weighting the distance  $v_R(t)$  should keep from obstacles in the curve space and the deceleration close to the target person.

The approach is formulated as a nonlinear optimization problem with the objective to find a velocity profile  $\dot{\lambda}(t)$  minimizing  $J_{\text{vel}}$  subject to its constraints:

$$\begin{aligned} &\text{find } \dot{\lambda}(t) \\ &\min J_{\text{vel}}(\dot{\lambda}(t)) \\ &\text{s.t. } t_E = v_H s_H, \quad \lambda(0) = 0, \quad \lambda(t_E) = 1 \\ &\quad 0 \leq v \leq v_{\max}, \quad t = t_E \text{ if } \lambda(t) = 1, \quad v = 0 \text{ if } \lambda(t) = 1 \end{aligned} \tag{7}$$

By applying this method, the first step will generate a collision free path constrained in curvature within the static environment. Since the shape is fixed, the velocity mapping in step two will force the robot to slow down or speed up for moving obstacle avoidance. Given that the timing constraint does not hold, the overall time optimization will provide a new goal pose to the planning method.

Constraints enhancing social acceptance of the approach path exist in all optimization steps. At first, the timing confinement to preserve the perceived proactivity. Further, the distance related cost in step one will keep the trajectory away from objects and moving humans as passing closely is usually perceived as uncomfortable. Selecting a Bézier curve with curvature constraints opts for natural and smooth paths. Velocity profiles generated in step two allow for moving obstacle avoidance in spite of the fixed path which is necessary to enhance the readability of the intention. The second term keeps the robot from passing shortly behind a moving obstacle that just crossed the path. The last part of  $J_{\text{vel}}$  influences the velocity profile close to the approached human by reducing the speed gradually. A sidewise or frontal approach position with appropriate orientation is implicitly defined within the goal pose  $\mathbf{x}_G$ , given the person's orientation is known.

## 5 Implementation

This chapter presents an integrated real time capable system that approximates the developed optimal control problem in Chap. 4 by applying a rule based brute force search. This system allows conducting user studies to confirm parameter effects or to develop further aspects that affect readability.

For motion planning Bézier curves are used since they feature properties which are beneficial for the readability as well [10]. Firstly, the starting point  $\mathbf{p}_0$  and the endpoint  $\mathbf{p}_n$  of a Bézier curve are freely controllable. For human approach these points are fixed to the robot and target person position and change since both move. Secondly, tangents at  $\mathbf{p}_0$  and  $\mathbf{p}_n$  that connect  $\mathbf{p}_0, \mathbf{p}_1$  and  $\mathbf{p}_{n-1}, \mathbf{p}_n$  allow the definition of the final position and orientation. Thirdly, as the  $k$ -th derivative of a Bézier curve is still continuous [7], the curve has continuous curvature. Accordingly, trajectories are consistent continuations of each other if their respective starting and ending point are the same which is the case when continuous online replanning is applied. Lastly, due to the  $k$ -fold differentiability [7] Bézier curves supply smoothness and continuous jerk.

A maximum time frame for the trajectory construction is given by the fact that a person is only approachable until she has reached a distance which the robot is unable to catch up with. The trajectory origin is always set at the robot position whereas the final pose depends on the predicted movement (position, speed) for the person. Without loss of generality the person’s orientation is assumed to be known. Given the mentioned attributes a cubic Bézier curve is used for static scenarios as one can see in Fig. 1.

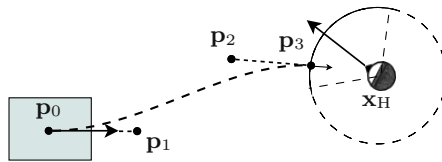


Fig. 1 Trajectory for Bézier curve of degree three in free space

In the collision case the degree is increased to four such that one control point, here  $\mathbf{p}_2$ , pulls the curve away from the obstacle as shown in Fig. 2. By checking for discontinuities in the laser scan, object dimensions are assessed and extremal points found. Searching for a collision free curve,  $\mathbf{p}_2$  is shifted iteratively. The control points  $\mathbf{p}_0, \mathbf{p}_1$  remain on the line defined by

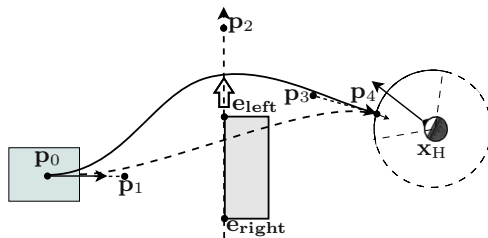


Fig. 2 Trajectory for Bézier curve of degree four in the collision case

the initial orientation, the starting point  $\mathbf{p}_0 = \mathbf{x}_R(\lambda(0))$  and  $\mathbf{p}_1$  while  $\mathbf{p}_2$ ,  $\mathbf{p}_3$  or alternatively  $\mathbf{p}_3$ ,  $\mathbf{p}_4$  are defined through the position of the human  $\mathbf{x}_H$ , the ending point  $\mathbf{p}_n = \mathbf{x}_G$  and the final orientation. Therefore, turning on spot never results from this planning process. The point positions can be defined as optimization constraints:

$$\mathbf{p}_1 = f(\mathbf{p}_0) = \mathbf{p}_0 + \eta_1 \begin{pmatrix} x \\ y \end{pmatrix} \quad (8)$$

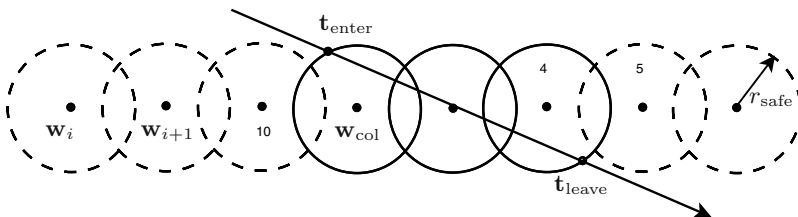
$$\mathbf{p}_2 = f(\mathbf{p}_3, x_H) = \mathbf{p}_3 + \eta_2 (\mathbf{p}_3 - \mathbf{x}_H) \quad (9)$$

where  $\eta_1$  and  $\eta_2$  are arbitrary parameters and  $\mathbf{p}_3 = \mathbf{x}_G$  depicts the goal pose that employs social aspects like human-robot distance, positioning in the field of view and preference for sidewise approach. For degree four curves the constraint on  $\mathbf{p}_2$  and  $\mathbf{p}_3$  is dependant on the obstacle dimension and its extremal points  $\mathbf{e}_{\text{left}}$  and  $\mathbf{e}_{\text{right}}$ . Pulling the curve out on the side of  $\mathbf{e}_{\text{right}}$  leads to the following:

$$\mathbf{p}_2 = \mathbf{e}_{\text{right}} + \eta_3 (\mathbf{e}_{\text{right}} - \mathbf{e}_{\text{left}}) \quad (10)$$

$$\mathbf{p}_3 = f(\mathbf{p}_4, x_H) = \mathbf{p}_4 + \eta_4 (\mathbf{p}_4 - \mathbf{x}_H) \quad (11)$$

where  $\eta_3$  and  $\eta_4$  are arbitrary parameters. Given the planned curve, it is discretized in time and space forming a trajectory. The primary velocity profile assumes that maximum speed is possible due to absence of collision with a deceleration nearby the goal pose as proposed in [5]. The velocity profile is then adapted to provide the slow down or speed up for avoidance of moving obstacles. At first a safety region with radius  $r_{\text{safe}}$  is assumed around every discrete position on the trajectory. For a moving obstacle crossing the trajectory, the entrance time  $t_{\text{enter}}$  and the emission time  $t_{\text{leave}}$  are calculated applying a constant velocity model. Based on this concept the robot has to execute the trajectory up to the colliding position  $w_{\text{col}-1}$  or to the position  $w_{\text{end}+1}$  after the critical point to avoid the obstacle as it leaves the zone or before it crosses. This is realized in the velocity profile by accelerating or decelerating. Fig. 3 illustrates the concept.



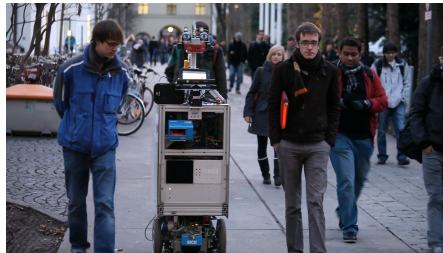
**Fig. 3** Collision zones indicated by circles with a continuous line around a waypoint with high collision potential

Finally the arrival time at each trajectory point is estimated by the distance of two discrete points and the according velocity. After that, the velocity profile of the whole trajectory has to be considered in order to assess if the final goal pose is reached in time and the robot arrives simultaneously with the target person. This leads to an iterative algorithm capable of adapting the trajectory, velocities and goal positions online.

The separation of spacial and temporal planning entails that the trajectory shape does not change with the velocity profiles making the movement more predictable. Obstacle avoidance therefore follows a human-inspired approach [18]. The real-time capability enables the system to adapt in case of tracking errors or dynamic changes in the environment.

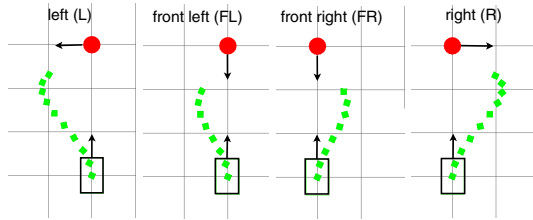
## 6 Experiment

For assessing the human perception of the approach behavior, we set up an experiment where a robot moves towards a standing person. We asked the probates to rate the convenience of the robot velocity, the distance where it stopped, how natural the motion seemed and how comfortable they felt during the approach.



**Fig. 4** ACE platform moving outdoors with mounted Eddie emotional display

This user study was conducted on the ACE (Autonomous City Explorer) platform [2] displayed in Fig. 4. The design of ACE, the predecessor of IURO, is less human-like and pleasing but since only the movement should be in the focus, an elaborate design can induce distractive effects. In order to enforce the focusing on the movement the emotional display is unmounted as well. Thus, further studies are possible to evaluate the influence of human-like design features on the naturalness of the approach. As free space for an approach is needed the experiment was set up in the lobby of a public building. The robot is started by an instructor for each of the four different approaches from a position in 4 m distance opposite to the probate. The probate takes three different orientations: facing the robot or  $\pm 90^\circ$  looking to the left or right. The according trajectories are shown in Fig. 5. Subsequent to each scenario the probate rates naturalness, approach speed, approach distance and



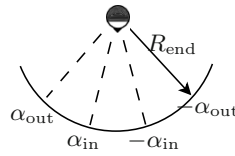
**Fig. 5** Paths executed during the four different approaches of a static person

sensation on a questionnaire using a 10-point Likert scale. Randomization of the approach scenario succession was applied to exclude bias effects.

Table 6a shows the parameters for this study with further illustration in Fig. 6b, where  $v_{mid}$  is the mean robot speed,  $v_{end}$  the velocity at a distance below 1.5 m to the person,  $R_{end}$  the final approach distance,  $\alpha_{out}$  the angle describing the part of the field of view where the robot stops in and  $\alpha_{in}$  defines an area which provides the goal positions for sidewise approaches.

**Parameter Value**

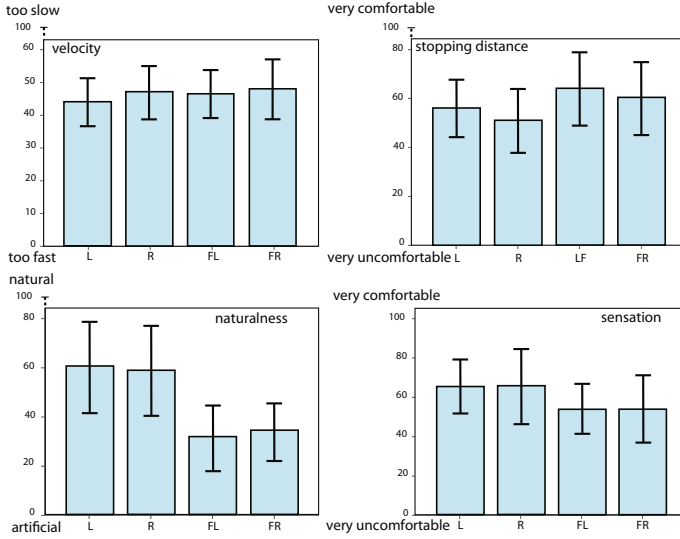
$v_{mean}$	0.45 m/s
$v_{end}$	0.23 m/s
$R_{end}$	0.55 m
$\alpha_{in}$	30°
$\alpha_{out}$	35°



**Fig. 6** Parameters for the user study and possible robot goal poses in front of a human within  $\varphi \in [-\alpha_{out}, \alpha_{out}]$

The experiment was carried out with 10 probates in the age group from 21 to 38 years and academic high school diploma or higher degree who were rather experienced in the field of robotics. Descriptive statistical results were acquired from an analyses of repeated measures using a parametric test (ANOVA). The velocity ratings were principally optimal with a trend to slow. Figure 7 top left shows mean and variance where 0 indicates "too slow", 50 "optimal" and 100 "too fast". The stopping distance (within the personal space) was rated comfortable showing differences between sidewise and frontal approach. Mean and variance in Fig. 7 top right are scaled from 0 "very uncomfortable" to 100 "very comfortable". Significant results were found in the naturalness condition, Fig. 7 bottom left. The scaling ranges from 0 "artificial" to 100 "natural". The outcome shows significant differences comparing left and front right approach with  $\sigma_{L,FR} = 0.019$  as well as right and front left/right approach with  $\sigma_{R,FL} = 0.046$  and  $\sigma_{R,FR} = 0.018$ . The left and front left scenario diverge almost significantly with  $\sigma_{L,FL} = 0.058$ . The mean effect size assuming sphericity is medium to high with  $F(3, 27)$ ,





**Fig. 7** Descriptive statistics with mean and standard deviation. Results are given for the approach directions L=left, R=right, FL=frontal left, FR=frontal right

$p < 0.0001$ ,  $\eta^2_{\text{partial}} = 0.586$ . According to that, the power of 0.999 indicates that the naturalness condition is a strong benchmark for path shape taking into account the small sample size. Considering comments of the probates, the less natural rating for the frontal approach originates from the fact that the robot slightly turns away and then comes back towards the person, Fig. 5. Probates referred to this movement as hard to interpret. The sensation rating is scaled from 0 "very uncomfortable" to 100 "very comfortable" showing a slight difference between frontal and sidewise approaches, as shown in Fig. 7 bottom right. Over all the approach behavior is rated comfortable showing that people were not intimidated by the robot.

Follow up studies will investigate naturalness by e.g. employing various shapes for frontal or sidewise approaches. By integrating an approach from behind it is further assessable whether comfort and naturalness follow the assumed characteristics. Currently running user studies compare our method to a simple movement to a position in front of the person that also allows turning on spot. As of now, results for velocity, distance and comfort remain stable. Indeed, the ratings for naturalness show a strong trend for the simple approach behavior to be perceived as less natural.

## 7 Conclusions and Future Work

The problem of increasing the readability for optimal control based trajectory planning in the setting of human approach is discussed in this paper. An

analytical formulation of the problem is provided along with an integrated planning algorithm enabling a mobile robot to approach a moving person autonomously in a dynamic environment by planning in space and time consecutively. The human-inspired avoidance mechanism further enforces the natural appearance. The applied parameters for approach speed and distance are rated positively. Frontal approaches come too close presumably because the maneuverability of the person is constrained in this case. We identified that trajectory shape is an important factor when evaluating the naturalness of the approach behavior.

In this paper it is shown how human-like approach behaviors improve the readability and social acceptance of planned trajectories. We approximate the optimization problem by a rule based implementation. The integrated system facilitates a motion planner for proactive human approach in dynamic environments that avoids static and moving obstacles. It is capable of on-line replanning for limited numbers of dynamic obstacles and a constrained spatial horizon enabling the planner to adapt to a dynamically changing environment. The Bézier curve based trajectories feature smooth shapes to further enhance the natural appearance of the motion. Due to this simplicity the amount of avoidable static objects is constrained such that the algorithm is not complete. Yet, this paper shows that complicated trajectory shapes provide lower naturalness and are less readable compared to a direct approach.

Future work will concentrate on experiments to confirm and find social parameters. Large scale experiments within the IURO setting will serve as further benchmarks. Studies comparing multiple planning methods with respect to resulting trajectory shapes will yield cues on readability enhancements. As the performance of the system relies on predicted paths a more advanced approach is to be integrated. Finally, improving the system by taking into account the mutual interactions between humans, the robot and obstacles will further raise social acceptance of the motion.

**Acknowledgement.** This work is part of the IURO project ICT-248314 supported by the 7th Framework Programme of the European Union, by the ERC Advanced Grant SHRINE Agreement No. 267877 under the 7th Framework Programme, by the DFG excellence initiative research cluster Cognition for Technical Systems CoTeSys, and by the Technische Universität München Institute for Advanced Study, funded by the German Excellence Initiative. The authors gratefully thank Daniel Althoff, Andreas Lawitzky and Antonia Glaser for the lively discussions that strongly supported the developments shown.

## References

1. Arechavaleta, G., Laumond, J., Hicheur, H., Berthoz, A.: The nonholonomic nature of human locomotion: a modeling study. In: Proc. Int. Conf. on Biomedical Robotics and Biomechanics, pp. 158–163 (2006)

2. Bauer, A., Klasing, K., Lidoris, G., Mühlbauer, Q., Rohrmüller, F., Sosnowski, S., Xu, T., Kühnlenz, K., Wollherr, D., Buss, M.: The autonomous city explorer: Towards natural human-robot interaction in urban environments. *Int. Journal of Social Robotics* 1(2), 127–140 (2009)
3. Bennewitz, M., Burgard, W., Thrun, S.: Learning motion patterns of persons for mobile service robots. In: *Proc. Int. Conf. on Robotics and Automation*, pp. 3601–3606 (2002)
4. Breazeal, C., Kidd, C.D., Thomaz, A.L., Hoffman, G., Berlin, M.: Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In: *Proc. Int. Conf. on Intelligent Robots and Systems*, pp. 708–713 (2005)
5. Buss, M., Carton, D., Gonsior, B., Kuehnlenz, K., Landsiedel, C., Mitsou, N., de Nijs, R., Zlotowski, J., Sosnowski, S., Strasser, E., Tscheligi, M., Weiss, A., Wollherr, D.: Towards proactive human-robot interaction in human environments. In: *Proc. Int. Conf. on Cognitive Infocommunications*, pp. 1–6 (2011)
6. Choi, J.-W., Curry, R., Elkaim, G.H.: Path planning based on bezier curve for autonomous ground vehicles. In: *Advances in Electrical and Electronics Engineering - IAENG Special Edition of WCECS*, pp. 158–166 (2008)
7. Choi, J.-W., Curry, R., Elkaim, G.H.: Smooth path generation based on bezier curves for autonomous vehicles. In: *Lecture Notes in Engineering and Computer Science: Proc. of WCECS*, pp. 668–673 (2009)
8. Choi, J.-W., Curry, R., Elkaim, G.H.: Piecewise bezier curves path planning with continuous curvature constraint for autonomous driving. In: *Machine Learning and Systems Engineering. LNEE*, vol. 68 (2010)
9. Elnagar, A.: Prediction of moving objects in dynamic environments using kalman filters. In: *IEEE Int. Symp. on Computational Intel. in Robotics and Automation*, pp. 414–419 (2001)
10. Farin, G.E.: *Curves and surfaces for CAGD: a practical guide*, 5th edn. Morgan Kaufmann Publishers - Academic Press (2002)
11. Ferguson, D., Howard, T.M., Likhachev, M.: Motion planning in urban environments. In: *The DARPA Urban Challenge*, pp. 61–89 (2009)
12. Fiorini, P., Shiller, Z.: Time optimal trajectory planning in dynamic environments. In: *Proc. Int. Conf. on Robotics and Automation*, pp. 1553–1558 (1996)
13. Fraichard, T.: *Trajectory Planning in Dynamic Workspace: a ‘State-Time Space’ Approach*. Technical Report RR-3545, INRIA (1998)
14. Foka, A.F., Trahanias, P.E.: Predictive autonomous robot navigation. In: *Proc. Int. Conf. on Intelligent Robots and Systems*, pp. 490–495 (2002)
15. Hall, E.T.: *The Hidden Dimension: Man’s Use of Space in Public and Private*. The Bodley Head Ltd., London (1966)
16. Huttenrauch, H., Severinson Eklundh, K., Green, A., Topp, E.A., Christensen, H.I.: What’s in the gap? Interaction transitions that make HRI work. In: *Int. Symp. on Robot and Human Interactive Communication*, pp. 123–128 (2006)
17. Kessler, J., Schröter, C., Gross, H.-M.: Approaching a person in a socially acceptable manner using a fast marching planner. In: *ICIRA*, pp. 368–377 (2011)
18. Kruse, T., Basili, P., Glasauer, S., Kirsch, A.: Legible robot navigation in the proximity of moving humans. In: *Proc. Int. Workshop ARSO*, pp. 83–88 (2012)
19. Latombe, J.-C.: *Robot Motion Planning*. Kluwer Academic Publishers (1991)
20. LaValle, S.M.: *Planning Algorithms*. Cambridge University Press (2006)
21. Masehian, E., Katebi, Y.: Robot motion planning in dynamic environments with moving obstacles and target. *Int. Journal of Mechanical, Industrial and Aerospace Engineering*, 20–25 (2007)

22. Satake, S., Kanda, T., Glas, D., Imai, M., Ishiguro, H., Hagita, N.: How to approach humans? - strategies for social robots to initiate interaction. In: Proc. Int. Conf. on Human Robot Interaction, pp. 109–116 (2009)
23. Schulz, D., Burgard, W., Fox, D., Cremers, A.B.: Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In: Proc. Int. Conf. on Robotics and Automation, pp. 1665–1670 (2001)
24. Sisbot, E.A., Alami, R., Simeon, T., Dautenhahn, K., Walters, M., Woods, S., Koay, K.L., Nehaniv, C.: Navigation in the presence of humans. In: Proc. Int. Conf. on Humanoid Robots, pp. 181–188 (2005)
25. Sisbot, E., Marin-Urias, L., Alami, R., Simeon, T.: A human aware mobile robot motion planner. *Trans. on Robotics*, 874–883 (2007)
26. Sisbot, E., Marin-Urias, L., Broquire, X., Sidobre, D., Alami, R.: Synthesizing robot motions adapted to human presence. *Int. Journal of Social Robotics* 2, 329–343 (2010)
27. Shiller, Z., Gal, O., Raz, A.: Adaptive time horizon for on-line avoidance in dynamic environments. In: Proc. Int. Conf. on Intelligent Robots and Systems, pp. 3539–3544 (2011)
28. Shiomi, M., Kanda, T., Glas, D., Satake, S., Ishiguro, H., Hagita, N.: Who will be the customer?: A social robot that anticipates people’s behavior from their trajectories. field trial of networked social robots in a shopping mall. In: Proc. Int. Conf. on Intelligent Robots and Systems, pp. 2846–2853 (2009)
29. Svenstrup, M., Tranberg, S., Andersen, H.J., Bak, T.: Pose estimation and adaptive robot behaviour for human-robot interaction. In: Proc. Int. Conf. on Robotics and Automation, pp. 3571–3576 (2009)
30. Takayama, L., Dooley, D., Ju, W.: Expressing thought: improving robot readability with animation principles. In: Proc. Int. Conf. on HRI, pp. 69–76 (2011)
31. Thompson, S., Horiuchi, T., Kagami, S.: A probabilistic model of human motion and navigation intent for mobile robot path planning. In: Proc. Int. Conf. on Autonomous Robots and Agents, pp. 663–668 (2009)
32. Walters, M., Dautenhahn, K., Koay, K., Kaouri, C., te Boekhorst, R., Nehaniv, C., Werry, I., Lee, D.: Close encounters: Spatial distances between people and a robot of mechanistic appearance. In: Proc. Int. Conf. on Humanoid Robots, pp. 450–455 (2005)
33. Walters, M., Koay, K., Woods, S., Syrdal, D., Dautenhahn, K.: Robot to human approaches: preliminary results on comfortable distances and preferences. In: Symp. Multidisciplinary Collaboration for Socially Assistive Robotics, pp. 103–109 (2007)
34. Wilkie, D., van den Berg, J., Manocha, D.: Generalized velocity obstacles. In: Proc. Int. Conf. on Intelligent Robots and Systems, pp. 5573–5578 (2009)
35. Woo, H.J., Park, S.B., Kim, J.H.: Research of the optimal path planning methods for unmanned ground vehicle in darpa urban challenge. In: Proc. Int. Conf. on Control, Automation and Systems, pp. 586–589 (2008)
36. Woods, S., Walters, M., Koay, K., Dautenhahn, K.: Methodological issues in HRI: A comparison of live and video-based methods in robot to human approach direction trials. In: Proc. Int. Symp. on Robot and Human Interactive Communication, pp. 109–116 (2006)

# Multitask Humanoid Control with a Brain-Computer Interface: User Experiment with HRP-2

Pierre Gergondet, Abderrahmane Kheddar, Christoph Hintermüller,  
Christoph Guger, and Mel Slater

**Abstract.** In this paper, we present our approach to design a brain-computer interface (BCI) that allows the user to perform multitask humanoid control. We efficiently integrate techniques from computer vision and the task-function based control together with the brain-computer interface into an immersive and intuitive control application despite the well-known shortcomings of BCI. This approach is assessed in a user experiment involving 4 subjects who successfully controlled the HRP-2 humanoid robot in a scenario involving both grasping tasks and steering. The user experiences and the interface performances are presented and give a rich insight into future research that can be made to improve and extend such interface.

## 1 Introduction

Brain-computer interfaces (BCI) [1] allow bypassing the usual communication channels between a human and a computer such as hand or voice input interfaces. Instead, they allow the users to communicate his *intentions* to the

---

Pierre Gergondet · Abderrahmane Kheddar  
UM2-CNRS LIRMM UMR5506, Montpellier, France  
e-mail: pierre.gergondet@aist.go.jp

Pierre Gergondet · Abderrahmane Kheddar  
CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan

Christoph Hintermüller · Christoph Guger  
g.tec Guger Technologies OG, Austria

Mel Slater  
ICREA - University of Barcelona, Spain

Mel Slater  
University College London, United Kingdom

computer. In return, the user is able to control different application (software or device) systems connected to the BCI. Recent work has already demonstrated impressive capability for controlling mobile robots, virtual avatars or humanoid robots [2].

However, the use of BCI in those previous works has been limited to the accomplishment of a single task, for example: steering a robot. Our work attempts to make more tasks available to the user. It originates from the VERE project, which aims at embodying the user's conscience into a virtual avatar or a physical robot. Therefore, we aim at allowing the user to perform a wide range of actions with an emphasis on liberty and reactivity. The scenario that is demonstrated in this paper illustrates the approach we have taken and combines both locomotion and manipulation tasks, that is whole-body motion.

Using the well-known brain pattern: steady-state visually evoked potentials (SSVEP), we allow the user to perform humanoid whole-body control. We efficiently integrate techniques from computer vision and the task-function based control. We propose to use task-function primitives as affordances on objects of interest detected through the robot's embedded cameras. The user is fed with direct vision feedback from those cameras. Our main contributions and novelties are the following:

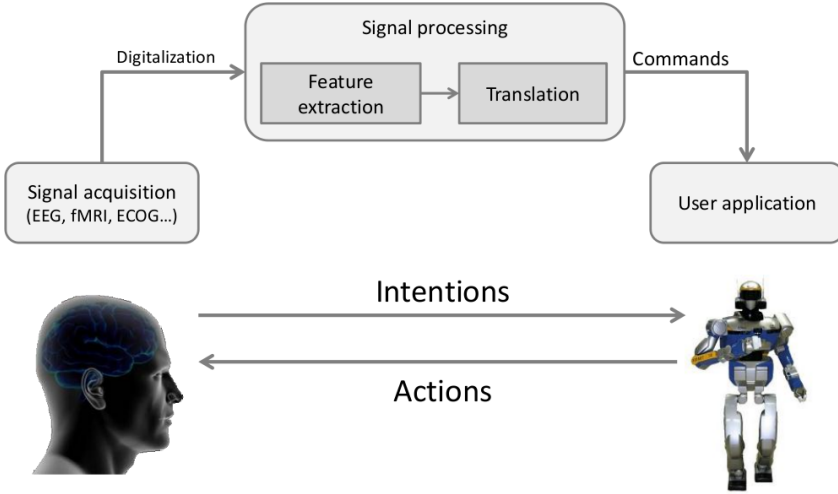
- By using well-known techniques from image processing, objects of interest are detected within the scene and automatically blinked at different frequencies. SSVEP allows identifying which object is of the user's interest.
- Integrating BCI and task-based control allowing instant and smooth task integration in the controller.
- A transition state machine proposes switching between whole body manipulation and locomotion tasks. During locomotion, SSVEP is also used to choose direction and speed of locomotion during which visual feedback is continuously displayed to the user.
- Our approach is assessed in real experiments using the HRP-2 robot controlled from an electroencephalography (EEG) cap and g.BCIsys (g.tec medical engineering GmbH, Austria). The scenario of these experiments allows the user to achieve multiple tasks.

## 2 Technical Approach

This section introduces three major components of our system: (i) the brain-computer interface, (ii) the stack-of-tasks (SoT) controller for the robot, and (iii) the robot visual perception system. The integration of these components is also a key to the extended capacities of our system. This integration occurs at two levels of interaction: between the visual system and the BCI on the one hand, and between the BCI and the SoT controller on the other hand.

## 2.1 Brain-Computer Interface

In recent years, several frameworks such as OpenViBE or BCI2000 have introduced a similar three-layer model to produce BCI application as shown in Figure 1.



**Fig. 1** General design of a BCI system

The signal acquisition layer monitors the physiological signals from the brain through one or several physical devices and digitizes these signals to pass them onto the signal-processing unit. The signal-processing unit is in charge of extracting features — e.g. power spectrum, signal energy — from the raw signals, and pass them onto a classification algorithm to distinguish the *intentions* of the user. Finally, these decoded intentions are passed onto the user application.

In this work, we employ the widely used electroencephalography (EEG) technique for signal acquisition because it is non-invasive, cheap and allows for real-time acquisition even though it suffers from poor spatial localization accuracy and a poor signal to noise ratio.

The brain features we decided to exploit are the steady-state visually evoked potentials (SSVEP). The SSVEP describe the activities that the brain generates when the user observes a flickering stimulus. The method relies uniquely on the user’s attention to the stimulus. It also allows detecting that the user is maintaining his attention on a given stimulus and to detect a shift of attention in a few seconds. The process we used to extract the SSVEP is based upon the minimum energy classifier approach introduced in [3]. It provides a zero-class implementation that allows to detect that the user is not interested in interacting with the system. After a short training, about 6

minutes, it is able to operate at an 80% recognition rate [4] for 4 classes and can provide a new command every 200ms. This is a satisfactory performance for an SSVEP-based BCI system [5].

## Recursive and Enforced SSVEP Selection

As mentioned previously, our SSVEP extraction process can reach an 80% successful recognition rate with a short training phase. However, given the nature of the SSVEP stimuli, errors are bound to happen over the course of the experiment due to distraction or fatigue [6]. Therefore, the 100% mark would be difficult to reach. This unreliability of the decision outcome becomes an important problem when using the BCI to control an avatar, especially if the decision that was taken cannot be reversed. Moreover we aim at minimizing the number of frequencies that we can detect to ensure high accuracy while keeping the training time as short as possible. Therefore we devised an SSVEP-based selection paradigm that allows for a large number of commands and puts an emphasis on accuracy.

To increase the number of commands we used a simple recursive selection algorithm. For example, if we have sixteen commands available but only trained the system to detect four different frequencies we split each command into four groups. The user then selects one of these groups and finally selects a command among the four commands in this group.

To enforce the selection we ask the user to maintain his attention on the command he wants to select for a certain time. We consider this command as the actual intention of the user only if he was able to maintain his attention "long enough", which in our case means three seconds, i.e. fifteen successive classifications.

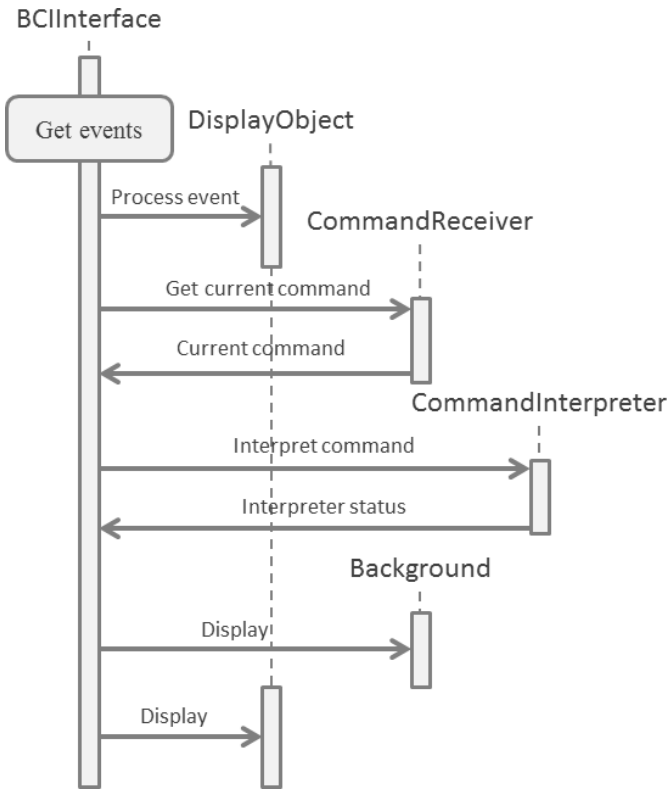
## User Interface

We also developed an application framework for the user interface that allows practical switching between different BCI paradigms, thus allowing multiple tasks control with our BCI system.

The framework relies on a core class: `BCIInterface`. Its role is to create the graphical interface, handle and distribute events and get the different primitives of the library working together. As illustrated in Figure 2, the main graphical loop executes the following operations for each frame:

1. Get and handle events, pass unhandled events to `DisplayObjects` and `CommandInterpreter`.
2. Receive the current command from the BCI thanks to a `CommandReceiver` instance.
3. Give this command to a `CommandInterpreter` instance that will update the controlled system status and/or the `DisplayObjects` to give feedback to the user according to its implementation.
4. Display a `Background` instance and then display all `DisplayObjects`.





**Fig. 2** A display loop iteration by the BCIInterface core

Additionally, the elements that require synchronous operations to work - e.g. a *CommandReceiver* that receives commands over the network - can implement another loop; this loop is run in a separate thread to have no impact on the display loop performance and it does not require any extra work from the developer. A paradigm switch can be triggered by the owner of the BCI-Interface instance or by the *CommandInterpreter*.

## 2.2 Stack of Tasks Controller

The task-function based control is a powerful control paradigm to design complex behaviors for robots without explicit trajectory planning at the joint or Cartesian level. A task can be seen as motion primitives or constraints that can be defined directly or indirectly in the robots sensory space. The desired task can be defined simply as a state or a state error vector in the sensory space, which is mapped into the robots motor space (joint actuators) using an appropriate projection operator, e.g. the robot’s jacobian in kinematics. A complex behavior can then be planned as a succession of tasks, which

can in turn be structured into a hierarchical way: a stack of tasks. This formalism proved to be particularly suited for the control of highly redundant robots; we have demonstrated several complex scenarios using this powerful control tool with our HRP-2 humanoid robot [7]. We have also extended this controller, called simply stack-of-tasks (SoT), to address issues such as tasks scheduling and the fundamental issue of control continuity under discrete tasks scheduling operations in the SoT, such as tasks insertion, removal and swapping. Our controller is able to insert or remove tasks components, on the fly, and we can adapt it to be used in a BCI control context.

### *2.3 Vision System*

Our vision system is built around a lightweight core that operates different cameras and plugins aimed at realizing a specific task such as transmitting images over the network or recognizing objects within the scene.

#### **Objects Recognition**

The object recognition method is based on the work presented in [8] and its extension to account for color properties of objects in [9]. This method relies on the construction of a vocabulary set of texture-related features and color-related features. These features are trimmed down to a restricted set through a k-means clustering algorithm, associated to the relevant objects and organized in a kd-tree for efficient closest neighbor research needed by the recognition algorithm. This allows this method to scale very well as when the objects database grows, the vocabulary itself does not grow but evolves and enriches itself.

The recognition algorithm then consists in (i) extracting interest points in the scene, (ii) computing color and texture features at those interest points, (iii) match those features with the ones from the vocabulary, (iv) each feature from the vocabulary will then cast a vote for the relevant object - this voting mechanism is further explained afterward - (v) the object presence is decided based upon its best score value. The score each object will give is determined from a training set where the algorithm knows the objects present in the scene. The votes are computed so that the more specific to an object a feature is the bigger vote it will cast. For example, if the same feature is extracted from 10 images with 10 different objects in the scene it will contribute a 0.1 vote to each object. However, among the same set, if a feature is found only twice, it will cast a 0.5 vote for the two objects involved in these two scenes. Finally, a second pass over the training set allows us to define threshold score above which the object presence is assumed. Thanks to the sharing of features among different objects, the recognition algorithm can operate very efficiently, above 15 Hz even in rich scenes, and the features selected for classification permits a consistent detection of the objects.

## Shape Extraction

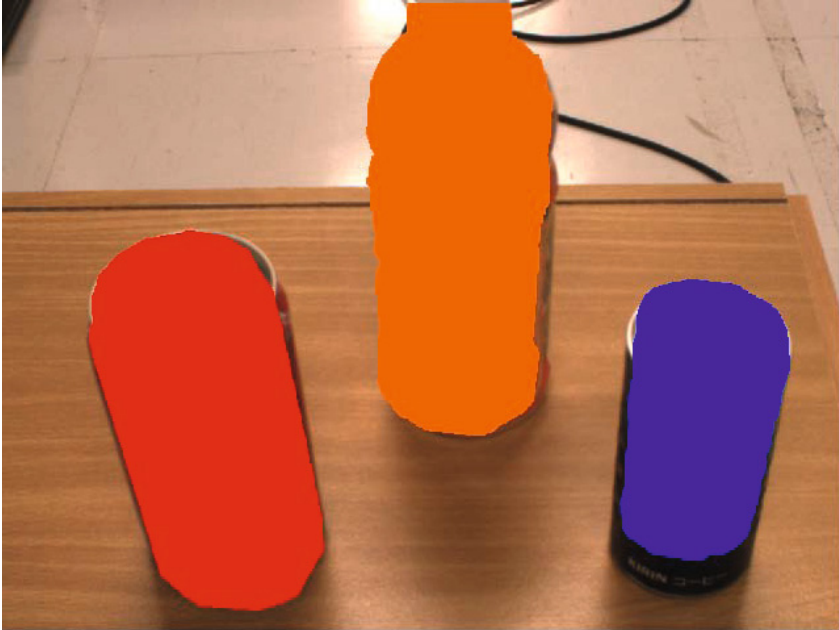
In order to allow a smooth integration of the detected objects within the user interface, we extended this method to detect the objects' shapes.

To do so, we need to collect more data when we build the vocabulary as well as during the course of the algorithm. During the training run, when we create a link between a feature and the object, we register two information from the training image: the shape of the object and the position of the feature point. During the recognition process, we maintain a list of all voters for each object. For each voter, we register the point of interest position in the image being processed, the link it is associated with and where the vote is casted.

Once an object and its center have been detected, we extract the voters that voted for the object around the object's center. Each voter is ultimately linked to an object's mask from the vocabulary set through the link it holds. For each of these masks, we compute an homography from the points in the vocabulary set to the matched points in the processed image. The points are filtered before the homography computation. If multiple points in the processed image match the same point in the vocabulary set, we keep the one for which the distance between the point and the center of the object is closest to the same distance for the vocabulary point. The same selection criterion is applied when the same point in the processed image matches different points in the vocabulary set, which can happen when the texture match and the color match are different points from the same image of the vocabulary set. The vocabulary object that retains the most matching points in the end is selected and its shape is deformed thanks to the computed homography which allows us to match the shape of the object in the processed image.

## 2.4 *BCI and Visual System Integration*

When the user is interacting with the world through the robot he needs to be informed about the capacities of the robot and the interaction it can operate with the world. To do so we integrate visual stimuli on top of the video feedback fed to the user. We can distinguish two kinds of integrated stimuli: static ones and dynamic ones. On the one hand, the static stimuli are relative to the current operation mode of the robot. In visual exploration mode they will consist in flickering arrows to control the robot's gaze, while in steering mode a similar set of arrows will control the robot's speed. On the other hand, the dynamic stimuli are controlled by the visual system. This allows presenting the user with the objects he will be able to interact with. A flickering stimulus is shown at the detected center of the object, communicated by the vision system. Their color is chosen to fit the recognized object. The result can be observed in Figure 3 and in the experiment video linked below.



**Fig. 3** Interface for object grasping: images are streamed from the vision server, object-related stimuli are positioned thanks to the objects detection plugin

## *2.5 Tasks Selections with a BCI*

Contrary to other interfaces used to control a robot, EEG-based BCI is a rather unreliable and very slow interface. In other words, the user will not be able to guide efficiently an arm to grasp an object in a certain position and in a particular way. What s/he will be able to request however is an action that results from an association of the recognized object of intention and the afforded task (or tasks). For example, “grasp object A, given that this “object A has been detected and localized in the real world via the visual system and presented to the user as explained previously. This naturally translates into considering tasks as affordances on the environment objects.

This approach is reminiscent of the shared-control approach that has been often used in BCI-based control since it allows the user to perform the same task better with less input commands. However, the system does not only rely on its own intelligence to perform the task based on a limited input, it also shares its knowledge of the environment with the user to present affordable tasks to him, that is what we refer to as shared intelligence.

### 3 Experiment Scenario

The scenario is designed to illustrate multitask control, i.e. locomotion and manipulation, via BCI.

At first, the user is presented, through the robot's eyes, with multiple objects - known from the recognition system - put on a cupboard. In this phase, the user selects the object he wishes to interact with. The selection relies on an SSVEP paradigm: the objects are blinked as discussed in the previous section. Once this selection happens, the robot grasps the object and the second phase of the experiment begins.

In the second phase, the user steers the robot freely in its environment to a location of his choice. The steering is done through SSVEP. Three stimuli allow the user to control the robot orientation and make it move forward, a fourth one allows him to stop the robot. Once the robot is stopped, the final phase begins.

In the final phase, the user recursively selects a position within the visual field of the robot. The selection of a position makes the robot drop the object above this position thus achieving the experiment.

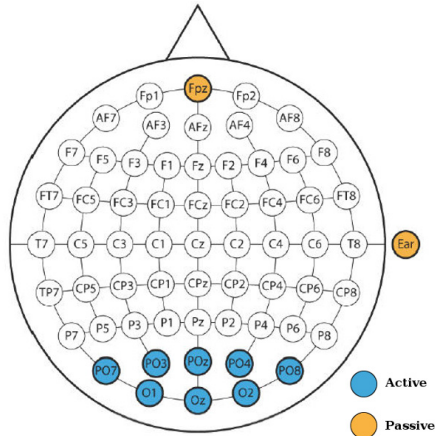
To confirm the usability of the interface, the user is given two instructions that define his mission: (a) which object to pick up from the cupboard and (b) which sixteenth of the screen to select at the end. To evaluate the objective performances of the interface we measure the time taken to achieve each phase and the success rate of the missions.

## 4 Experiments

### 4.1 Material and System Setup

We use a g.USBamp (24 Bit biosignal amplification unit, g.tec Medical Engineering GmbH, Austria) to acquire the EEG data from the user's brain at a sampling frequency of 256 Hz, bandpass filtered between 0.5 and 30 Hz with a notch filter at 50 Hz to get rid of the power line noise. The electrodes positioning is shown in Figure 4. We use 8 Ag/AgCl active electrodes. The electrodes are placed on the POz, PO3, PO4, PO7, PO8, O1, O2 and Oz positions of the international 10-20 system [10], Fpz is used as the ground electrode and the earlobe as a reference.

The experiment was carried out using the HRP-2 humanoid robot. The subject, equipped with an EEG cap, is comfortably seated in an armchair, about 1 meter away from a 17" LCD screen. In such setup the accurate display of SSVEP stimuli is ensured thanks to the method proposed in [11]. The SSVEP stimuli frequencies that were used in this work are: 6, 8, 9 and 10 Hz. Those were carefully selected to have neither common first or second harmonics and are below 20 Hz to minimize the risk of eliciting an epileptic crisis in healthy subjects as advised in [12].



**Fig. 4** Electrodes positions for the experiment

## 4.2 Results

A video showing the interface in action as well as the robotic counterpart can be retrieved at the following URL: <http://pbs.dinauz.org/Videos/ISER-2012.avi>.

Four users performed the scenario that we described. Each of them performed the scenario five times with different conditions, i.e. different objects and different drop locations. Over the 20 trials, the users' selections were consistently accurate thanks to the enforcing of SSVEP selection we setup. In Figure 5 we report their average performance over the multiple trials.

The phase 1 and phase 3 times are consistent with the performance of our SSVEP classification method and the adoption of the enforced SSVEP selection process. The system operates on a 3 seconds window of EEG data and we require the user to maintain his attention on the stimulus for 3 seconds before we make a conclusion about his intention. The phase 3 involves the recursive SSVEP paradigm described earlier and thus the time needed to reach a conclusion in phase 3 is about twice the time needed in phase 1.

The enforced SSVEP selection paradigm also proved its usefulness during the trials as no misinterpretation of the user's intentions occurred during these phases for all subjects across all trials. However, we chose to enforce the selection a 3 seconds period which covers 15 decisions by the SSVEP extraction process. To achieve a more reactive experience, this activation time could be tuned down according to the classifier performance with the user. This would allow the system to reach a conclusion more rapidly while keeping a very high-level of accuracy.

Finally, the phases 2 times illustrate interesting effects of training and motivation on the scenario performance. The performance of each subject over each trial can be seen in Figure 6. It shows that the user's performance

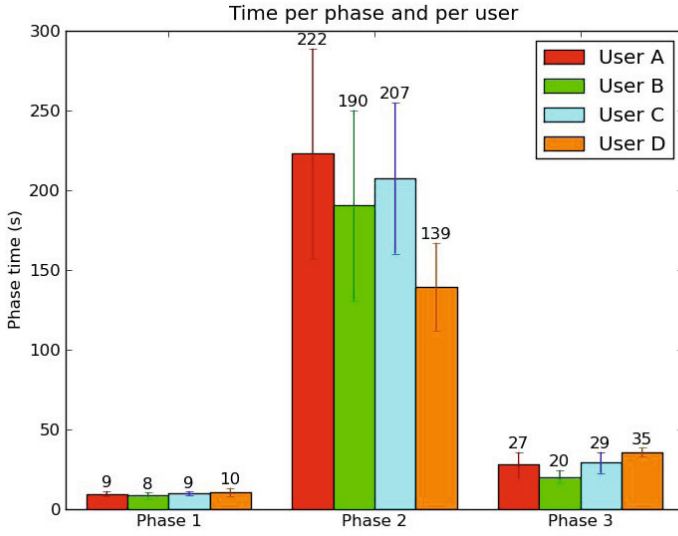


Fig. 5 Average performance of each user over the experiment trials

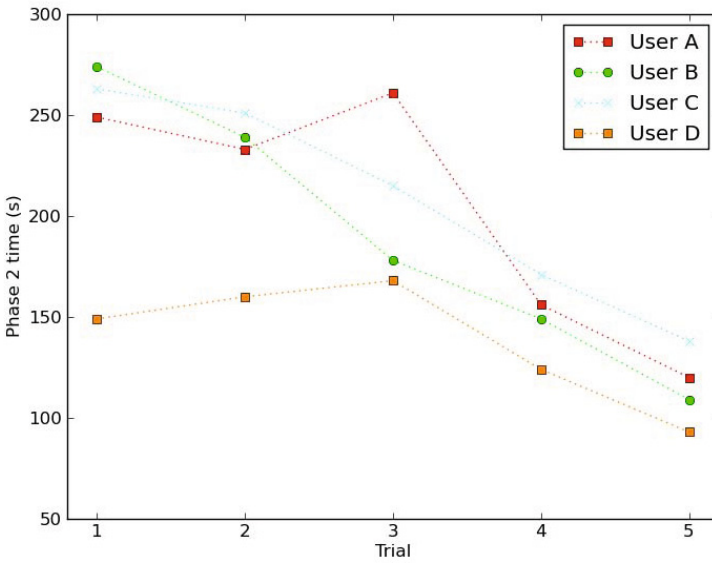


Fig. 6 Walking phase performance for each subject and trial

in the navigation phase improve after each trial. Two factors, reported by the users, can explain this phenomenon. On the one hand, the users felt more and more comfortable with the interface. They progressively learned how to compensate for the lag of the SSVEP extraction process and they also acquainted themselves with the environment and the way it is perceived through the robot's camera. Indeed, this camera has a rather constrained field of view which is well suited for the stereo application it is used for but makes it particularly difficult for the user to understand the robot's location in space. On the other hand, the user also used more 'aggressive' strategy to reach their destination because they felt more comfortable steering the robot but most importantly because they wished to 'beat' their previous time which illustrates the importance of motivation in the context of BCI applications [13].

## 5 Future Works

The success of the experiment and positive feedback from the users confirm the viability of the concepts presented throughout this paper. This section discusses possible improvements and issues left unaddressed in this work regarding the problem of task selection and parameterization through BCI, and the navigation in unknown environment.

### 5.1 BCI and Tasks Selection

This work addresses the issue of mapping tasks to environment objects by conceiving tasks as affordances on objects. From the control feedback and visual perception we can infer a feasible task among those attached to the objects. However, two main issues are left to be dealt with: task selection among many and task parameterization.

We are now able to understand that the user wants to interact with an object. If we wish to pursue the embodiment of one's conscience into an avatar, we also have to be able to understand *why* the user wants to interact with this object as the visual and control information may not be enough to conclude which afforded task should be selected. A good example to illustrate this issue is the one of a bottle of water. If we want to pour some of its content into a glass we will not grasp it like if we want to put this bottle in a bag. However, in the current state of this work, such a distinction cannot be done, as we are only able to understand that the user wants to "interact with the bottle". This is the problem of task selection among many.

The second unaddressed issue regarding the task selection concerns the task parameterization. The problem here is to select the appropriate gain, i.e. speed of execution, and variant of the task, e.g. grasp with left or right hand. Addressing these problems is difficult, as it requires a much deeper understanding of the user's intention than current EEG-based BCI permits.



A possible solution may rely on the measurement of user's satisfaction as the robot executes the demanded task. Measuring the stress of the user for example could be used to modulate the gain of the task while understanding the EEG features related to user's satisfaction will allow to cancel the current task and try another one to fit the user's expectation. However, from the robot's control perspective, task backtracking is also an open and complex problem that will have to be tackled to allow this kind of parameterization.

## ***5.2 Landmark-Assisted Navigation***

The phase 1 and phase 3 of the experiment's scenario are good examples of the benefits of shared-control in BCI control application. Within a few second of selection, the user can command the execution of complex tasks by the robot while 'relaxing'. In phase 2, the user continuously commands the robot to guide to its final position. The benefit of such method is that it allows the user to reach any position he wants in the robot's environment. However, it is also tiresome for the user, especially over long period of control, and the speed of the robot has to be limited to allow a fine control by the user.

To improve this situation, we will investigate an hybrid scheme of navigation control that mixes full control approach and shared control approach. In this navigation scheme, the object recognition module presented in this paper is used during navigation to detect known objects in the robot's environment, these objects are then proposed for selection to the user as in the first phase of the scenario. If the user selects an object, the 3D model of this object can be tracked in the scene [14] to approach it efficiently without further input from the user. Once the object has been reached, the user once again controls the robot either to reach its final destination or to find another landmark to go to.

## **6 Conclusion**

We presented our method to control a humanoid robot through a brain-computer interface within a multitask scenario. The key of this work is to not rely on a pre-defined set of task and thus differs from classical approach in BCI control application. In place, the tasks are dynamically constructed from the tight collaboration of the visual system, control architecture and user's intention extraction process and are presented to the user in a user-friendly way together with the visual feedback of the robot. In a scenario we designed to illustrate these concepts, four users were able to successfully control a HRP-2 humanoid robot in a multitask scenario.

Future works will focus on resolving the many issues raised during the development of this new architecture in the three different fields it involves. We believe that, extending upon this work, we can provide strong solutions

to the problems of user embodiment and whole-body control through brain-computer applications.

**Acknowledgement.** This research is supported by the European Union FP7 Integrated Project VERE (No. 257695) [www.vereproject.eu](http://www.vereproject.eu).

## References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. *Clinical Neurophysiology* 113, 767–791 (2002)
2. del R. Millán, J., Rupp, R., Müller-Putz, G.R., Murray-Smith, R., Giugliemma, C., Tangermann, M., Vidaurre, C., Cincotti, F., Kübler, A., Leeb, R., Neuper, C., Müller, K.-R., Mattia, D.: Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. *Frontiers in Neuroprosthetics* 4, 161 (2010)
3. Friman, O., Volosyak, I., Gräser, A.: Multiple Channel Detection of Steady-State Visual Evoked Potentials for Brain-Computer Interfaces. *IEEE Transactions on Biomedical Engineering* 54(4), 742–750 (2007)
4. Prueckl, R., Guger, C.: A brain-computer interface based on steady state visual evoked potentials for controlling a robot. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) *IWANN 2009, Part I. LNCS*, vol. 5517, pp. 690–697. Springer, Heidelberg (2009)
5. Vialatte, F.-B., Maurice, M., Dauwels, J., Cichockia, A.: Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. *Progress in Neurobiology* 90, 418–438 (2010)
6. Zhu, D., Bieger, J., Molina, G.G., Aarts, R.M.: A Survey of Stimulation Methods Used in SSVEP-Based BCIs. *Computational Intelligence and Neuroscience* 2010, 12 (2010)
7. Mansard, N., Stasse, O., Evrard, P., Kheddar, A.: A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In: *Int. Conf. on Advanced Robotics, ICAR* (2009)
8. Murphy-Chutorian, E., Triesch, J.: Shared Features for Scalable Appearance-Based Object Recognition. In: *IEEE Workshop on Applications of Computer Vision* (2005)
9. Murphy-Chutorian, E., Aboutalib, S., Triesch, J.: Analysis of a Biologically-Inspired System for Real-time Object Recognition. *Cognitive Science Online* 3.2, 1–14 (2005)
10. Sharbrough, F., Chatrian, G.-E., Lesser, R.P., Lüders, H., Nuwer, M., Picton, T.W.: American Electroencephalographic Society guidelines for standard electrode position nomenclature. *Journal of Clinical Neurophysiology* 8, 200–202 (1991)
11. Wang, Y., Wang, Y.-T., Jung, T.-P.: Visual stimulus design for high-rate SSVEP BCI. *Electronic Letters* 46, 1057–1058 (2010)

12. Fisher, R.S., Harding, G., Erba, G., Barkley, G.L., Wilkins, A.: Photic- and pattern-induced seizures: a Review for the Epilepsy Foundation of America Working Group. *Epilepsia* 46, 1426–1441 (2005)
13. Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., Pfurtscheller, G.: Brain-Computer Communication: Motivation, aim and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15, 473–482 (2007)
14. Comport, A.I., Marchand, E., Pressigout, M., Chaumette, F.: Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transaction on Visualization and Computer Graphics* 12(4), 615–628 (2006)

# Coordination Strategies for Multi-robot Exploration and Mapping

John G. Rogers III, Carlos Nieto-Granda, and Henrik I. Christensen

**Abstract.** Situational awareness in rescue operations can be provided by teams of autonomous mobile robots. Human operators are required to teleoperate the current generation of mobile robots for this application; however, teleoperation is increasingly difficult as the number of robots is expanded. As the number of robots is increased, each robot may interfere with one another and eventually decrease mapping performance. Through careful consideration of robot team coordination and exploration strategy, large numbers of mobile robots be allocated to accomplish the mapping task more quickly and accurately.

## 1 Motivation

Projects like the Army Research Laboratory's Micro-Autonomous Systems Technology (MAST) [1] seek to introduce the application of large numbers of inexpensive and simple mobile robots for situational awareness in urban military and rescue operations. Human operators are required to teleoperate the current generation of mobile robots for this application; however, teleoperation is increasingly difficult as the number of robots is expanded. There is evidence in human factors research which indicates that the cognitive load on a human operator is significantly increased when they are asked to teleoperate more than one robot [18].

Autonomy will make it possible to manage larger numbers of small robots for mapping. There is a continuum of options as to the degree of shared autonomy between robot and human operator [11]. Current robots employed in explosive ordnance disposal (EOD) missions are fully tele-operated. At the other extreme, robots can be given high-level tasks by the operator, while autonomously handling low-level tasks [3] such as obstacle avoidance or balance maintenance. In this paper, our

---

John G. Rogers III · Carlos Nieto-Granda · Henrik I. Christensen  
Center for Robotics and Intelligent Machine  
Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332  
e-mail: {jgrogers, carlos.nieto, hic}@gatech.edu

robot teams occupy the latter end of the spectrum; we imagine that the operator has tasked the robot team to autonomously explore and map an unknown environment while focusing on the high level task of looking for survivors.

In the multi-robot scenario, resources are distributed amongst a team of robots instead of concentrated on one large and expensive machine. This distribution offers a number of advantages and disadvantages over the single robot case. The distributed team is able to continue its mission even if some of the robots are disabled or destroyed. A single robot can only explore or monitor at one location at a time; however, the multi-robot team can provide situational awareness in many locations at once. Unless the single robot is able to move much faster than the multi-robot agents, the lone robot will be slower in performing the exploration and mapping task. These advantages are taken for a multi-robot team at the cost of increased complexity in communication and coordination.

As the number of robots is increased, each robot may interfere with one another and eventually decrease the performance of the mapping task. Careful consideration of exploration strategy and coordination of large numbers of mobile robots can efficiently allocate resources to perform the mapping task more quickly and more accurately.

Mobile robot simultaneous localization and mapping (SLAM) has been thoroughly addressed in the literature, see [2] and [6] for a detailed review of the history and state-of-the-art in SLAM research. The specific techniques used in this paper are based upon the Square Root SAM algorithm [4] [5] which uses the well-known algorithms of linear algebra least-squares system solving to compute the map and robot trajectory based on a set of measurements.

Multi-robot mapping and exploration was addressed in [9] and [17]. These papers build a map using up to 3 robots with a decision-theoretic planner which trades off robot rendezvous operations with frontier exploration. These robots rendezvous to determine their relative pose transforms to provide constraints to recover the final map. In contrast, our approach does not require this rendezvous step because landmarks are globally data associated between each robot on a central map coordinator. The exploration strategy used is similar to our strategy called *Reserve*; however, we will not use a rendezvous step and do not require a decision-theoretic planner.

## 2 Technical Approach

We use the Robot Operating System (ROS) from [12]. ROS provides interprocess communication as well as coordination of sensor data with pose information. Our robot algorithms are implemented as a distributed set of programs which run in the ROS system. In addition, we make use of several implementations of common mobile robot software components which are provided in the ROS distribution such as motion planning, obstacle avoidance, platform control, and IMU and odometry filtering.

### 2.1 Mapping System

Our mapping system is based upon the *GTsam* library developed at Georgia Tech. This library extends the Square Root SAM technique in [5] with sparse linear algebra in a nonlinear optimization engine. We have extended the *GTsam* library with a framework based upon the M-space formulation of Folkesson and Christensen [8] called *OmniMapper*. *OmniMapper* is a map library based upon a system of plugins which handle multiple landmark types simultaneously. We have used the *OmniMapper* in the past to build maps using multiple types of landmarks such as walls, doors, and objects [14] [13] [16]. This implementation builds maps of planar regions corresponding to walls and tables from [15].

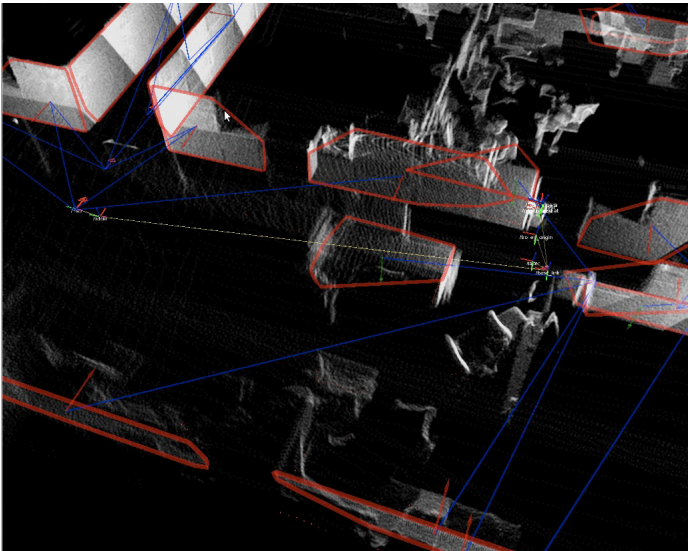


Fig. 1 OmniMapper

Each robot in the team builds a map locally with the *OmniMapper* and sends map data to the map coordinator. Each robot can incorporate new landmark measurements whenever it has moved far enough from the last pose where measurements were made. In the current implementation this is set to 10cm. When a robot finishes optimizing its local map with new landmark measurements, all relevant information needed by the map coordinator is packaged and transmitted.

The information which is needed by the map coordinator to incorporate a new piece of information from a team member consists of many components. First, the sensor measurement data is needed. In the current implementation, this consists of the extracted plane information consisting of a plane equation along with a convex hull of points along the perimeter of the plane. This represents a significant compression over an alternative scheme where all point-cloud data could be transmitted

and processed at the master node. Secondly, the team member's integrated odometry is transmitted. This allows the master node to compute the odometric relative pose since the prior landmark measurement data was incorporated; this is used to insert a relative pose factor and also give initial conditions for data association. Finally, the team member's local map pose is transmitted. This is used by the master node to compute a map pose correction. This correction is sent back to the team member so that it knows its relative pose in the global map frame. This knowledge is needed so that the team member can interpret exploration goals correctly.

The map coordinator maintains trajectories for each of the robots in the team. Measurements from each robot are merged into one global view of the landmarks. This is realized through a simple modification to the standard *OmniMapper* through duplication of data structures tracking indexing data and pose information used for interaction with GTsam into arrays. This implementation potentially allows for an unlimited number of team members to build a map together.

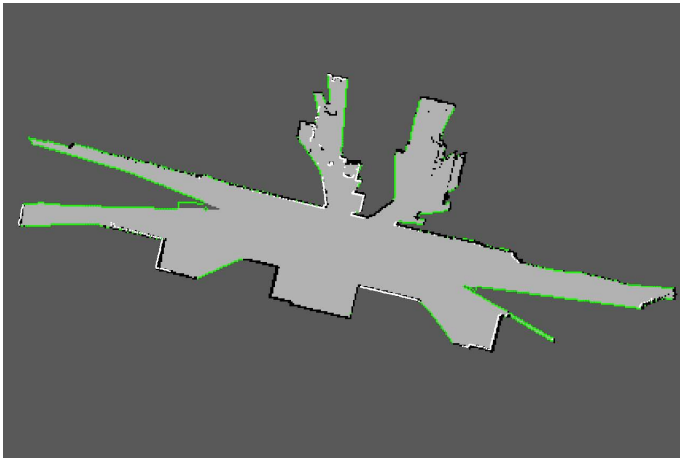
Most modern SLAM approaches use a *pose graph* [10] which is generated via laser scan matching in 2D or point-cloud ICP in 3D. This approach is effective for single robot mapping; however, it has some drawbacks for larger multirobot mapping. Scan matching and ICP algorithms are computationally intensive and matching across many robots would rapidly become intractable. Also, point cloud representations are large and their transport over a wireless link could be prohibitive if the link is limited in capacity due to mesh network routing or environmental interference. To address these limitations, our robots extract relevant, parsimonious features from the environment and transmit them to the master node.

Each turtlebot in these experiments maps planar wall structures using a Microsoft Kinect sensor. Planar segments corresponding to walls are extracted from point clouds via a RANSAC [7] based algorithm [15]. Points are uniformly sampled from the point cloud and any sufficiently large set of points coplanar with these three points are selected as a plane and are removed from the point cloud. This process is repeated until up to four planes are extracted or a fixed number of iterations is reached. To improve the speed of plane extraction, the Kinect point cloud is computed at QVGA ( $\frac{1}{8}$ ) resolution, which achieves 1Hz frame rate.

The Kinect sensor on each robot has a narrow field-of-view which is not ideal for detecting exploration frontiers. To alleviate this problem, we incorporated a strategy by which each robot will rotate periodically to get a 360 degree view of its surroundings. This data is synchronized with robot odometry to synthesize a 360 degree laser scan. This synthesized laser scan is sent to the local mapper and forwarded to the global mapper. At the global mapper, it is linked to a trajectory pose element and used to populate an occupancy grid. This occupancy grid is re-computed after every map optimization so that a loop closure will result in a correct occupancy grid map. The frontier based exploration strategies detailed below use this occupancy grid to find the boundary between clear and unknown grid cells.

## 2.2 Exploration Strategy

Each robot team leader uses a frontier based exploration strategy similar to the one used in [17]. An exploration frontier is defined on a costmap cellular decomposition where each cell has one of three labels: *Clear*, *Obstacle*, and *Unknown*. The costmap is initialized as *Unknown*. Costmap cells are set to *Obstacle* corresponding to locations where the Kinect sensor detects an obstacle in the environment. The cells on a line between the obstacle cell and the robot’s current location are set to *Clear*. Exploration frontiers are defined as *Clear* cells which are adjacent to at least one neighbor where the label is *Unknown*.



**Fig. 2** Global maps using the *Reserve* coordination algorithm described in this paper

The high level robot exploration goal allocation is centrally planned on the same workstation where the global map is constructed. There are many choices which can be made by the exploration planner when choosing which robot or group of robots should move towards an exploration goal. We have chosen to employ a greedy strategy by which the nearest robot or team is allocated to a goal instead of a more sophisticated traveling-salesman type of algorithm. We believe that this is appropriate because the exploration goals will change as the robots move through the environment; re-planning will be required after each robot or team reaches an exploration goal.

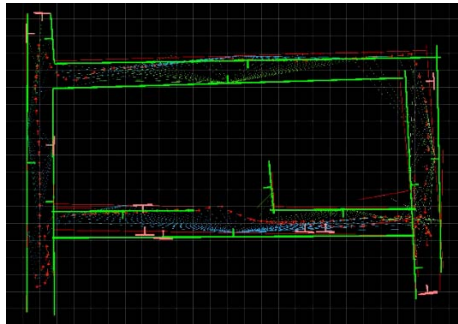
## 2.3 Coordination Strategy

The coordination strategy used between robot agents as well as the number of robots are the independent variables in the experiments performed in this paper. The coordination strategy refers to the proportion of robots which are dispatched to each exploration goal. On one extreme, a single robot can be sent to explore a new goal;



at the other extreme all available robots can be sent to a new goal. Larger robot teams sent to a new exploration goal will improve availability of new agents at the location of new exploration goals are discovered. The larger group has spare robots which can be quickly allocated to explore new goals, such as those discovered when the team moves past a corridor intersection or t-junction. If the group of robots allocated to a navigation goal is too large, then the robots can interfere with each other due to local reactive control of multiple agents with respect to dynamic obstacles and limited space in corridors. The strategies selected for testing trade off *availability* (robots are close and able to explore branching structure quickly) with *non-interference* (robots do not get in each other's way).

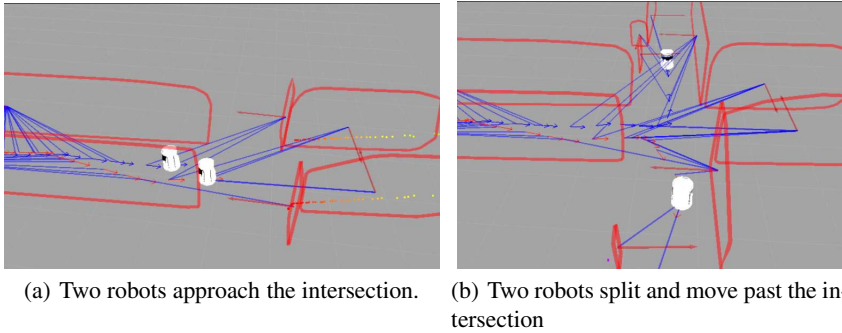
The first coordination algorithm is called *Reserve*. In this algorithm, all unallocated robots remain at the starting locations until new exploration goals are uncovered. When a branching point is detected by an active robot, the closest reserve robot will be recruited into active status to explore the other path. This strategy has low availability because all of the reserve robots remain far away at the entrance; however, it has minimal interference because the exploring robots will usually be further away from other robots.



**Fig. 3** A map built by three robots using the *Reserve* cooperative mapping strategy

The second coordination algorithm is *Divide and Conquer*. In this strategy, the entire robot group follows the leader until a branching point is detected. The group splits in half, with the first  $\frac{n}{2}$  robots following the original leader, robot  $\frac{n}{2} + 1$  is selected as the leader of the second group, and robots  $\frac{n}{2} + 2$  through  $n$  are now members of its squad. Once there are  $n$  squads with one robot, no further divide operations can be made and new exploration goals will only be allocated once a robot has reached a dead-end or looped back into a previously explored area. This algorithm maximizes availability, but potentially causes significant interference between robots.

An example 3D map built by two robots as they approach a branch point can be seen in figure 4(a). At this point, the robot team splits and each team member takes a separate path, as seen in figure 4(b). The map shown is built concurrently with local maps built on each robot. The global map is used to establish a global frame of reference for robot collaboration message coordinates.



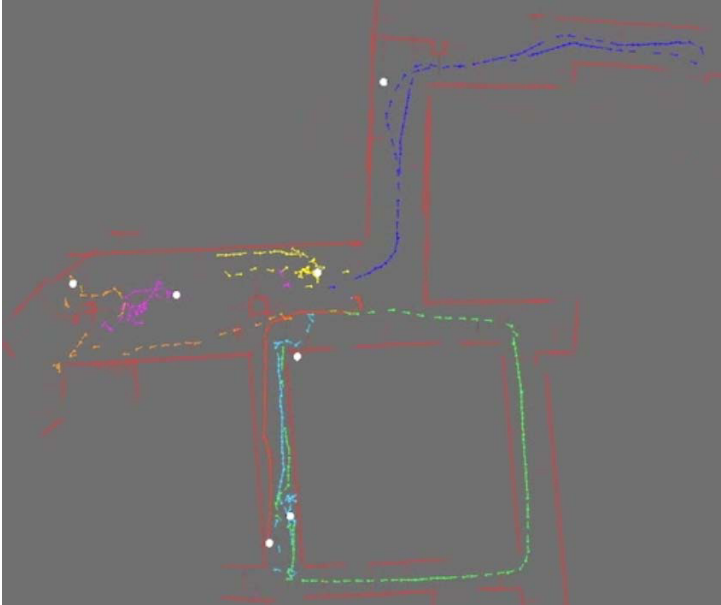
**Fig. 4** An illustration of the *Divide and Conquer* exploration strategy. As the robots approach an intersection, the team must split and recruit new partner robots from the reserved units.

### 3 Experiments

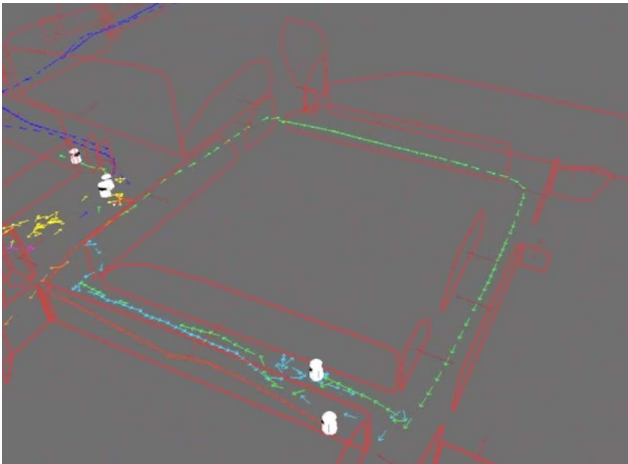
The setting for the multi-robot mapping task for this series of experiments consists of a team of robots being introduced into a single entrance in an unknown environment. Each robot is an inexpensive Willow Garage *TurtleBot*; a team of nine of these robots is shown in figure 3. The *TurtleBot* was chosen for this application due to its low cost and the ease of integrating large numbers of robots through ROS. The *TurtleBot* platform is based on the iRobot *Create* base. The robots make measurements of planes with a Kinect sensor, and use an onboard IMU together with odometry to estimate ego-motion.

We evaluated the performance of various robot coordination strategies in the multi-robot exploration and mapping task. An example scenario for the *Divide and Conquer* cooperative mapping strategy can be seen in the panorama image in figure 3.

We performed a series of experiments to demonstrate the performance of our two cooperative mapping strategies. A total of 6 runs were performed for each cooperation strategy, team size, and starting location. For each experiment run, the *TurtleBot* team explored the environment from a wedge-shaped starting configuration, which can be seen in figure 3. These experiments were performed in an office environment. In order to measure the exploration and mapping performance in each location, we chose specific starting locations which are labeled *Base1* and *Base2* in figure 3. These starting locations were chosen because the area around the robot teams could be blocked off so there is only one initial exploration frontier, directly in front of the lead robot. This initial configuration was chosen to represent a breaching behavior which would be needed for implementation of collaborative mapping in a hostile environment.



(a) A map built by seven robots in an experiment using the *Reserve* cooperative mapping strategy.



(b) The same map shown from a different angle to demonstrate 3D plane features which are used for map landmarks.

**Fig. 5** Global maps gathered by a team of seven mobile robots



**Fig. 6** Our nine TurtleBots used in these experiments



**Fig. 7** An example scenario for the experiments described in this paper. Three teams of two robots are exploring the branching hallway structure in an office environment. In this illustration, the robots are using the *Divide and Conquer* cooperative mapping strategy.

## 4 Results

We performed a series of experiments for this paper which demonstrate team performance based upon coverage in a mapping task on an unknown office environment. Robot team sizes were varied from 2 to 9 robots. An map built with 7 robots at TurtleBots using the *Reserve* strategy is seen in Figure 5(a). An image showing the same final global map from a side view demonstrates the 3D plane features in figure 2.3.

Each of the collaboration strategy and robot team size experiments were performed from two starting locations. These starting locations are labeled *Base1* and *Base2* in figure 3. A series of interesting locations was determined in advance by examining the building floor-plan; these points of interest are also marked in figure 3. Each experiment run gets a score based on how many of these points of interest are



**Fig. 8** Our office environment where the experiments were performed. The areas labeled Base1 and Base2 are the initial position of the robots. Red lines indicate artificial barricades to restrict the initial exploration of the robot teams to simulate a breach entrance into a hostile environment. Blue squares indicate the position of points-of-interest. Results are reported on the number of these points-of-interest visited by the robot team.

visited and mapped before a time limit is reached. This score represents the effectiveness of that algorithm and team size at providing coverage while exploring an unknown map.

In the first experiment series from *Base1* in figure 3, both strategies achieve reduced exploration coverage per robot as the team size is increased, as can be seen in the graphs in figure 9. In this starting location, there is limited space to maneuver, so both strategies generate significant interference between robots trying to move to their goals. In several instances, pairs of robots even crashed into each other due to the limited field-of-view of their sensors. We believe that the *Divide and Conquer* strategy results in figure 9(b) indicate that the team was slightly more effective than the *Reserves* strategy in figure 9(a). At the largest team size of 9 robots, the *Divide and Conquer* strategy usually visited one additional point-of-interest more than the *Reserves* strategy. Additional qualitative impressions are that the *Divide and Conquer* strategy explored the points-of-interest that it reached more quickly than with the *Reserves* strategy. For both strategies, the best team size appears to be 6 robots in this starting location.

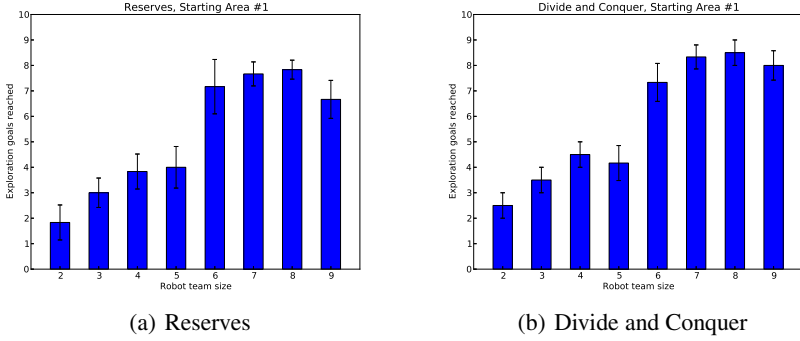


Fig. 9 Results from the first starting area

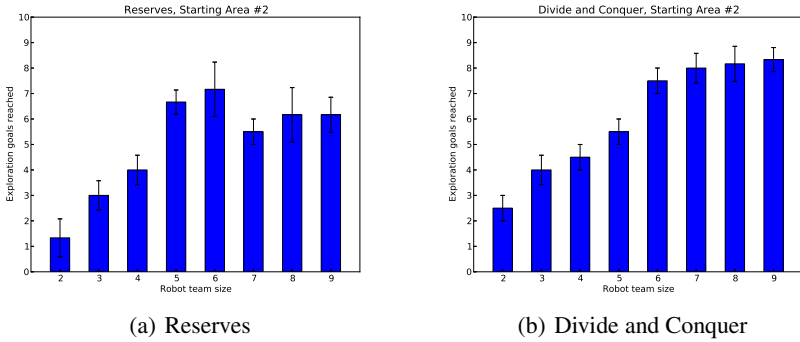


Fig. 10 Results from the second starting area

In the second set of experiments, the robot teams were placed in the starting area labeled *Base2* in figure 3. As in the first experiment, the per-robot performance of both strategies decreased as the number of robots were increased. This series of experiments demonstrates a marked improvement of the *Divide and Conquer* strategy over the *Reserves* strategy as can be seen in figure 10. The *Divide and Conquer* strategy causes more robots to be making observations of exploration frontiers due to the fact that groups contain more than one robot. These additional observations of the frontier allow the *Divide and Conquer* strategy to find exploration frontiers faster than the *Reserves* strategy, and therefore explore more points-of-interest. The second experiment started from an area where there is more room to maneuver. This allowed the *Divide and Conquer* strategy to have less interference since the entire team moved together out of the starting area into the larger area before any divide operations were performed. The *Reserves* strategy still had to initially maneuver from the cramped starting location. As in the first experiment, the *Divide and Conquer* strategy qualitatively explored the environment faster than the

*Reserves* strategy. The best value for the number of robots is 6, which is the same value found in the first experiment.

## 5 Discussion

We have presented experiments which evaluate two collaboration strategies which can be used by teams of mobile robots to map and explore an unknown environment. We have also evaluated the impact of the number of robots on coverage in the exploration and mapping task.

The first collaboration strategy, called *Reserves* keeps a pool of unallocated robots at the starting location. A new robot is activated when there are more exploration frontiers than currently active robots. This strategy was intended to minimize the amount of interference between robot agents since robots would be far away from each other during exploration. The results from our experiments do not indicate that this strategy results in less interference than other strategies since performance decreases more when more robots are added in some environments. The *Reserves* strategy is significantly slower at exploring the environment than other strategies.

The second collaboration strategy, called *Divide and Conquer* has all available robots proceed in one large group. Once there are two exploration frontiers, at a corridor t-junction for example, the team will divide in half and each sub-team will follow one of the exploration frontiers. This process will be repeated with teams dividing in half each time they see branching structure in the environment. It was anticipated that this strategy would result in higher interference since robots would be maneuvering close together; however, the increased availability of robots near new exploration frontiers offsets this phenomenon.

*Divide and Conquer* appears to be a more effective strategy than *Reserves* for exploring and mapping an unknown environment. There are additional hybrid strategies which could now be considered such as the *Buddy System*, which modifies the *Reserves* strategy with teams of 2 robots instead of 1. We believe that this strategy will mitigate much of the slowness of the *Reserves* strategy while still minimizing interference.

**Acknowledgements.** This work was made possible through the Army Research Lab (ARL) MAST CTA project, and the Boeing corporation.

## References

1. ARL: Army Research Lab Micro Autonomous Systems and Technology Collaborative Technology Alliance MAST CTA (2006), <http://www.arl.army.mil/www/default.cfm?page=332>
2. Bailey, T., Durrant-Whyte, H.: Simultaneous localisation and mapping (SLAM): Part II state of the art. *Robotics and Automation Magazine* (September 2006)

3. Chipalkatty, R., Daepf, H., Egerstedt, M., Book, W.: Human-in-the-loop: MPC for shared control of a quadruped rescue robot. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4556–4561. IEEE (2011)
4. Dellaert, F.: Square root SAM: Simultaneous localization and mapping via square root information smoothing. *Robotics: Science and Systems* (2005)
5. Dellaert, F., Kaess, M.: Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research* (2006)
6. Durrant-Whyte, H., Bailey, T.: Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine* (June 2006)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), 381–395 (1981)
8. Folkesson, J., Christensen, H.: Graphical SLAM - a self-correcting map. In: IEEE International Conference on Robotics and Automation (2004)
9. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., Stewart, B.: Distributed multi-robot exploration and mapping. *Proceedings of the IEEE* 94(7), 1325–1339 (2006)
10. Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P., Burgard, W.: Efficient estimation of accurate maximum likelihood maps in 3d. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007, pp. 3472–3478. IEEE (2007)
11. Heger, F., Singh, S.: Sliding autonomy for complex coordinated multi-robot tasks: Analysis and experiments. In: *Proceedings of Robotics: Science and Systems, Philadelphia, USA* (August 2006)
12. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: *ICRA Workshop on Open Source Software* (2009)
13. Rogers, J., Trevor, A., Nieto, C., Cunningham, A., Paluri, M., Michael, N., Dellaert, F., Christensen, H., Kumar, V.: Effects of sensory perception on mobile robot localization and mapping. In: *International Symposium on Experimental Robotics, ISER* (2010)
14. Rogers III, J.G., Trevor, A.J.B., Nieto-Granda, C., Christensen, H.I.: Simultaneous localization and mapping with learned object recognition and semantic data association. In: *IEEE International Conference on Intelligent Robots and Systems, IROS* (2011)
15. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China* (2011)
16. Trevor, A.J.B., Rogers III, J.G., Nieto-Granda, C., Christensen, H.I.: Tables, counters, and shelves: Semantic mapping of surfaces in 3D. In: *IROS Workshop on Semantic Mapping and Autonomous Knowledge Acquisition* (2010)
17. Vincent, R., Fox, D., Ko, J., Konolige, K., Limketkai, B., Morisset, B., Ortiz, C., Schulz, D., Stewart, B.: Distributed multirobot exploration, mapping, and task allocation. *Annals of Mathematics and Artificial Intelligence* 52(2), 229–255 (2008)
18. Zheng, K., Glas, D., Kanda, T., Ishiguro, H., Hagita, N.: How many social robots can one operator control? In: *Proceedings of the 6th International Conference on Human-Robot Interaction*, pp. 379–386. ACM (2011)



# Experiments Comparing Precision of Stereo-Vision Approaches for Control of an Industrial Manipulator

John-David Yoder, Jeffrey West, Eric Baumgartner, Mathias Perrollaz, Michael Seelinger, and Matthew Robinson

## 1 Motivation, Problem Statement, Related Work

Despite years of research in the area of robotics, the vast majority of industrial robots are still used in “teach-repeat” mode. This requires that the workpiece be in exactly the same position and orientation every time. In many high-volume robotics applications, this is not a problem, since the parts are likely to be fixtured anyway. However, in small to medium lot applications, this can be a significant limitation. The motivation for this project was a corporation who wanted to explore the use of visual control of a manipulator to allow for automated teaching of robot tasks for parts that are run in small lot sizes.

Since the 1970s, researchers have been proposing ways to use vision in order to solve this problem. While the purpose of this paper is not to provide a complete review of vision-based robotic manipulation, an excellent overview of early work is found in [1]. There has been success in the application of such technologies, especially in 2-D and 2.5-D problems [2]. Despite the fact that the theory for the solution of this problem has been established, there are limited implementations of full 3-D applications. Basically, the reasons for this come down to “the devil is in the details” and, in real 3-D applications, the accuracy, robustness, and cost-effectiveness of vision-based systems have been insufficient to justify widespread use.

---

John-David Yoder · Jeffrey West · Eric Baumgartner  
Ohio Northern University, Ada, OH, USA  
e-mail: j-yoder@onu.edu

Mathias Perrollaz  
Inria Rhône-Alpes, Saint Ismier, France

Michael Seelinger  
Yoder Software Inc., South Bend, IN, USA

Matthew Robinson  
Jet Propulsion Laboratory, Pasadena, CA, USA

A variety of efforts have been put forth to solve the problem of vision-based robotic manipulation. Visual Servoing offers promise by “closing the loop” in the image plane [3]. However, in many assembly tasks, the robot end effector obscures the target when it gets close to task completion. Several methods have been proposed to get past this limitation, such as outfitting both the robot and the workpiece with fiducials that remain visible [4], or “eye-in-hand” in cases where it is possible to place the camera(s) in the robot end effector. Another limitation of visual servoing is that in general it provides the least accuracy in the direction of the focal axis. Camera Space Manipulation [5][6] offers an alternative methodology and has been shown to allow completion of high-precision 3-D tasks. Unfortunately, this method does not allow standard stereo cameras to be used, as it requires widely separated and highly vergent cameras and utilizes the simplified orthographic model (though later work created accuracy similar to the pinhole model) [7].

The Mars Exploration Rovers used stereo vision with calibration for placing instruments on rock and soil targets [8], but this showed some limitations in accuracy [9]. Recently, in space robotics applications, two additional approaches have been offered, HIPS [9] and AGATE [10][11]. Both of these approaches have shown promise on space-related platforms to produce high-precision, vision-based manipulation using stereo cameras, with an application of instrument placement. These papers also reported large numbers of experiments. In particular, [9] showed the improved accuracy using HIPS compared to precalibrated stereo on NASA testbeds. However, these testbeds typically have inaccurate kinematics (backlash, inconsistent zero offsets, lack of rigidity, etc.).

## 2 Technical Approach

In response to a request from a local manufacturer, several of the authors were involved in a project to use computer vision to automate the teaching of a robot task for parts with small lot sizes. The details of that task are found in [12]. While the corporation was pleased with the results of the project, the accuracy obtained was not sufficient for all of their parts. As such, the authors have begun a detailed investigation of the accuracy obtained, and how that compares to other approaches. It should be noted that the industry-focus of this project has influenced the technical approach.

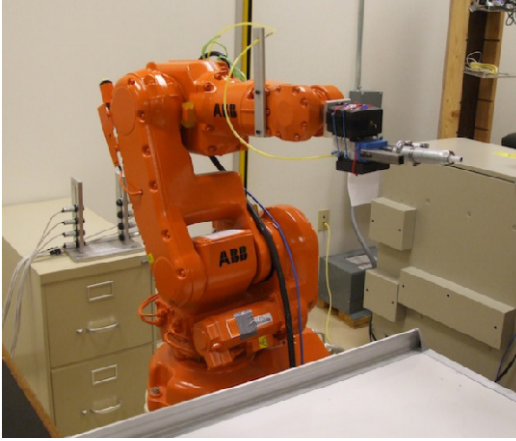
The goal of this paper is to answer the question of whether techniques such as AGATE can create significant improvement in accuracy when applied to an industrial robot by using local data to modify the parameters of the camera-robot model. Also, while AGATE was developed for mobile manipulation (as it can control the mobile base and the manipulator relative to a visual target), this paper will use AGATE techniques for a fixed-base manipulator. The paper further includes the first direct comparison of AGATE with HIPS. Future work will compare these methods directly to visual servoing and traditional camera calibration techniques.

HIPS and AGATE both make use of the CAHVOR camera model [13]. In both cases, the cameras can be calibrated using a standard checkerboard, but are typically calibrated by moving the manipulator through a series of poses in camera-space. In summary, a least-squares minimization is completed to compute the CAHVOR parameters based on data sets made up of 3-D robot positions and the 2-D image-plane appearance of the end effector in each camera. This minimization is completed after the manipulator moves through a pre-programmed set of positions designed to cover a significant portion of the robot's workspace and view of the camera(s). The CAHVOR parameters are then updated with additional measurements when they are available. Such measurements can be obtained while the end effector is approaching the target point. Addition of the localized samples results in a mapping between image space and physical space that is not globally accurate, but is very accurate relative to the manipulator in the region of the target point. This general approach has been shown to work well even in the presence of large kinematic errors [9][11]. Both HIPS and AGATE have been shown to provide the following advantages compared to traditional calibration methods: robustness to poor kinematics, the ability to deal with changes to the internal and external parameters of the camera, and the ability to achieve high precision manipulation relative to visually specified points. In addition, AGATE has been shown to be able to control mobile manipulators, and use more than two cameras [11].

Robinson [9] showed the advantage of HIPS (which finds the CAHVOR model parameters based on manipulator observations of fiducials on the manipulator) over pre-calibrated stereo, and the advantage of adding the local information – both in simulation and in a large number of experiments with a mobile manipulator. However, the manipulator in question was not as accurate as a typical industrial manipulator – it was less rigid and had significant kinematic errors (it was a prototype on a mobile manipulator testbed). The primary goal of this study is to see if a similar improvement takes place using an industrial manipulator, or whether the improved rigidity and kinematic accuracy negates any improvement due to on-the-fly CAHVOR model updates. For consistency with [9], we will call the version of the code that does not update the CAHVOR camera models during approach “Static AGATE” and the version that does update the camera models “Dynamic AGATE”.

### 3 Experiments

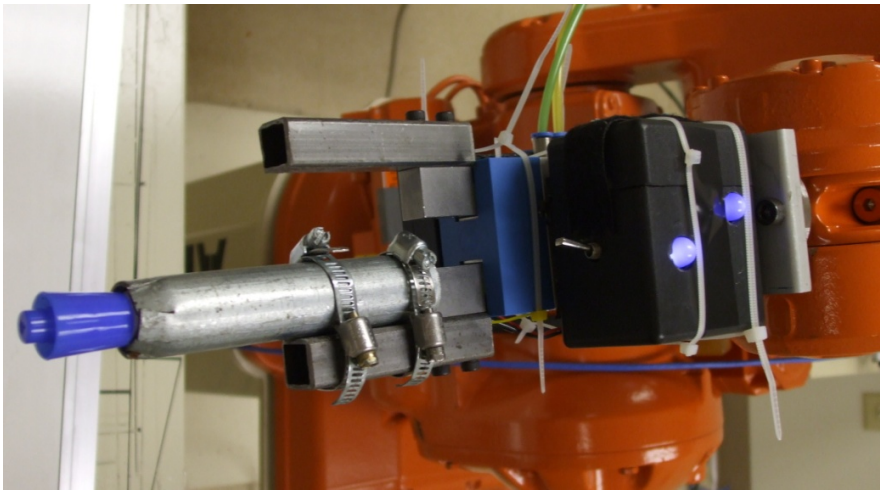
Experiments were conducted with the 6-axis ABB IRB-140 robot shown in Figure 1. The vision sensor is a pair of Point Grey Flea2 cameras, delivering 640 x 480 RGB images at 15FPS. As shown in Figure 2, Cameras are placed on a common portable support with a baseline separation of 15cm, and are placed approximately 2.5m from the workspace. While the cameras are effectively parallel, no particular efforts were taken to ensure a proper rectified configuration.



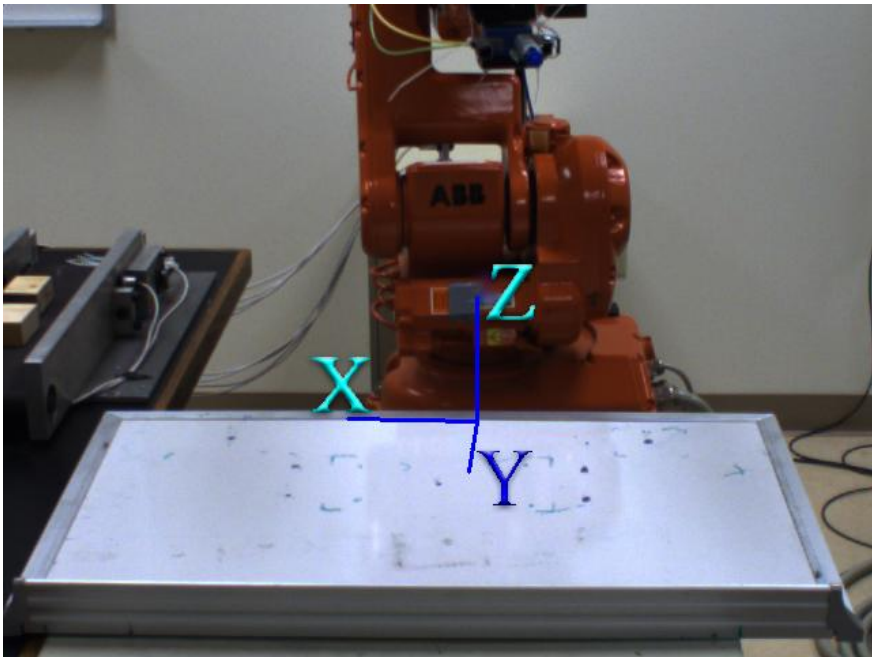
**Fig. 1** Experimental Setup



**Fig. 2** Camera System



**Fig. 3** Close up of marker attachment and LEDs on robot end-effector



**Fig. 4** Robot Coordinate System

The communication between the robot and the PC is performed through the serial port, with a message-based protocol. The C++ program for perception, control and command runs on a desktop PC equipped with an INTEL XEON W3505 (2009), dual core 2.54GHz, with 4GB of RAM.

Central to HIPS or AGATE is the ability to locate the end effector of the robot as it moves through the workspace. These image-plane locations, along with the corresponding location of the robot, are used to update the CAHVOR camera parameters. To simplify this task, the robot's end-effector is equipped with a LED blinking system as shown in Figure 3. The blinking frequency is roughly a third of the frequency of the cameras to avoid possible aliasing. We chose a rather simple design of the LED lamp, which does not require synchronization between the emitter (LED) and the receiver (camera). This is possible because the arm can be static during the acquisition of a set of images. Once the images are acquired, two images, respectively representing the mean value and the standard deviation over time of the intensity of each pixel, are computed. The cyan component of both images is computed using the green and blue planes (because our LEDs produce cyan rather than blue light). Then connected components of high standard deviation are extracted. These regions are detected as LEDs if their mean value over the cyan image is also sufficiently high.

All control is done by the ABB's controller. A directly-connected serial connection is used to transfer commands and coordinates to the ABB. More

details about the system, and the implementation of a precise manipulation task, can be found in [12].

A typical experiment consists of placing a fiducial in the workspace. For Dynamic AGATE, the robot is commanded to move half the distance to the target, and an additional measurement is made. If the results of this measurement are within normal bounds, this point is used to update the CAHVOR parameters. The process is repeated several times, until the robot is less than 4mm from the final goal, at which point it is moved to the final goal position. For Static AGATE, once the fiducial is found, the robot is moved directly to the target.

## 4 Results

To provide a measure of accuracy, the robot end effector was equipped with a standard whiteboard marker. The marker was placed in a holding device that provides approximately 1cm of compliance along the axis of the marker (shown in Figure 3). Circular fiducials were placed in the workspace, and the robot was instructed to approach the center of the fiducial and stop 5mm above the fiducial (above corresponds to the Z axis) of the base frame of the robot.

The jog mode of the robot is then used to move straight down to mark the fiducial. The vertical travel during the jog is used to measure the error in the Z direction. The distance from the mark to the center of the fiducial circle can be used to measure the error in the X-Y plane in the robot base frame, since the fiducials are placed on this plane. The X-Y-Z coordinate frame of the robot is shown in Figure 4. A sample of the marked fiducial is given in Figure 5. To provide a sense of scale, the outer diameter of the black circle is 76mm.

Thirty tests were conducted using Static AGATE and Dynamic AGATE. The tests consisted of 20 different physical locations throughout a region of the workspace (ten of the positions are reached twice). Table 1 shows a summary of these measurements, under "TEST SET ONE". In summary, the mean error with Dynamic AGATE was 11.0mm, while the mean error with Static AGATE was 14.6mm. A t-test showed this to be significant at the 95% confidence level ( $t=-2.169$ , significance =0.038). Dynamic AGATE showed a standard deviation of 7.8mm, reduced from 9.9mm with Static AGATE.

Looking at the data in Table 1, it is clear that most of the difference between the two is due to the XY-plane error. Comparing only this error, Dynamic AGATE resulted in a mean error (distance from the center of the cue) of 9.6mm compared to 13.5mm for Static AGATE. A t-test showed this to be significant at the 95% confidence level ( $t=-2.135$ , significance =0.041). It is also worth noting that the standard deviation of the errors was reduced by over 20% with Dynamic AGATE as well.



Fig. 5 Marked fiducial

Using the data sets obtained during the calibration process and during the tests, a direct comparison was completed between AGATE and HIPS. Both methods were used to project the observed locations of the LEDs from camera-space into physical space. The methods use different approaches for this process. While full details are outside the scope of this paper (see [9] and [11]), HIPS find the 3-D point at the center of the shortest line between the projecting rays. AGATE finds the 3-D point that minimizes the image-plane error between the measured and projected points. Despite these differing approaches, the results were very similar. The average difference (as measured by the norm of the vector between the 3-D projections) was 0.021mm, and the standard deviation was 0.025mm. Given the scale of the other errors involved, the conclusion is that AGATE and HIPS would produce the same accuracy of control.

**Table 1** Summary of Results

		STATIC AGATE (mm)	DYNAMIC AGATE (mm)	% change	Statistically Significant?
TEST SET 1					
	Mean X-Y error	13.5	9.6	29	YES
	Standard dev. X-Y	9.9	7.8	22	
	Mean total error	14.6	11.0	25	YES
	Standard dev. Total	9.4	7.3	23	
TEST SET 2					
	Mean X-Y error	8.0	7.1	11	NO
	Standard dev. X-Y	5.0	3.1	39	
	Mean total error	8.7	8.2	6	NO
	Standard dev. Total	5.2	2.4	53	

## 5 Understanding the Source of Positioning Errors

To draw any conclusions from the data presented, there must be an understanding of the sources of error that are included in the errors reported in Table 1. The basic sources of error will be enumerated here:

- 1) Errors due to the limitations on the mechanical system. That is, a motor can only be commanded to within one encoder count – this limitation on the precision of control will manifest itself as an error in the final measurement.
- 2) Error due to inaccuracies in the models and the limitations of the algorithms.
- 3) Error in making the measurement.
- 4) Error in the detection of the target.
- 5) Error in the detection of the LEDs.

- 6) Error due to the kinematic modeling of the LEDs and the marker.
- 7) Error in ensuring that the same physical point is selected in both cameras (correspondence error).

While these are all sources of error, error #1 is very small for an industrial robot. ABB reports repeatability, not accuracy, of their robots. The repeatability of the ABB 140 used in these tests is  $\pm 0.03\text{mm}$  [14], which is insignificant compared to all other errors.

Error #2 is related to the approach, and will be discussed later.

Error #3 has been minimized by physically marking the fiducial for later measurement. However, since the marker makes a point that has a diameter of over 1mm, it is likely that an error on the order of 1mm is reasonable for this item. A detailed examination of this could be undertaken by having multiple people measure the location of each mark, but this was not done for the experiments described here.

Using a fiducial limits the magnitude of error #4, the detection of the target. Multiple measurements were taken (though not a statistically significant number), and it was found that the standard deviation on the location of the center of the fiducial was approximately 0.2 pixels. While the volume of a voxel varies throughout the workspace, it is approximately  $1.67 \times 5.00 \times 1.67\text{mm}$  (in X, Y, Z, respectively). Thus, simply due to the resolution, one should expect errors with a standard deviation of approximately 1mm.

Regarding error #5, the LEDs are detected using image-differencing in order to reduce the likelihood of false positives. A center-of-gravity approach is used to find the center of each LED, and the two LED locations are combined into an “average” of their locations in order to reduce errors and eliminate the ambiguity between the two. Multiple tests were run and the standard deviation for the detection of the LEDs was found to be negligible.

Error #6 would always be present – the kinematics of any tool that is added on to the robot must be modeled and would always include error. Note that when used in teach-repeat mode, this error would not cause any positioning errors. Because of the plan to use this in industry, the standard ABB functions were used to “teach” the ABB the location of the points of interest (the tip of the marker, and the point between the two LEDs). This involves taking the point being taught to the same point in physical space from four different directions. The ABB then completes a least-squares minimization to locate the point relative to the robot coordinate system. For the data set shown in Table 1, the mean error for the marker was 0.73mm, while the mean error for the LED center position was 0.41mm.

Error #7, the correspondence error, is very important, and is related to the detection errors. Essentially this error is due to the fact that a given item (LED, fiducial, etc.) in X-Y-Z space will produce 4 values in camera space (assuming two cameras). If both cameras detect EXACTLY the same physical point, then there is no inconsistency between the camera-space measurements and the



physical space measurements. However, this is impossible to do (as stated earlier, finding the center of the fiducial, for example, has a standard deviation of 0.2 pixels). An error of 1 pixel in the horizontal location of an item in one of the cameras, for instance, will create an error of over 20mm when it is projected back into physical space using the AGATE method. This was found to be identical using HIPS.

When computing the parameters of the CAHVOR camera model, many of these errors are included. Recall that this computation is based on the image-plane location of the LEDs and the corresponding physical location of the robot. Thus, errors #1-3, and #5-7 are included (all errors other than the marker kinematics and the fiducial errors). In our experiments over multiple calibrations, these lead to a mean reconstruction error in the CAHVOR model of approximately 5mm. The reconstruction error is found by using the CAHVOR parameters to project the LED locations back into 3-D space and comparing those locations to the actual, recorded 3-D locations of the robot.

Given this understanding of the errors, some changes were made to the experimental procedure. To mitigate the effects of the error of detection of the target (#4), and the related error in correspondence of the target (#7), the center of the fiducial was found ten consecutive times, and the average position of the center was used. The system was taught the positions of the LED and marker again in order to mitigate error #6. Thresholds were updated in order to take into account the large effects of correspondence error. Finally, the CAHVOR parameters were updated again.

After these changes, another 30 tests were conducted using both Static AGATE and Dynamic AGATE. These results are shown in Table 1 as "TEST SET 2". There is a clear improvement compared to TEST SET 1. In summary, the mean total error was 8.2mm for Dynamic AGATE, and 8.7mm for Static AGATE, a 6% improvement using Dynamic AGATE. The mean X-Y error was reduced to 7.1mm for Dynamic AGATE, and 8.0mm for Static AGATE, an 11% improvement using Dynamic AGATE. These differences are not statistically significant at a 95% confidence level. It is worth noting, however, that the standard deviation of the total error was reduced over 50% using Dynamic AGATE.

## 6 Main Experimental Insights

There are three major experimental insights from this work.

- 1) For TEST SET 1, Dynamic AGATE produced a statistically significant improvement compared to Static AGATE.
- 2) For TEST SET 2, in which the system had reduced errors in terms of target identification and the kinematic model of the LED and Marker, Dynamic AGATE still produced better results, but not at a level to show statistical significance. The variation in positioning errors, however, was greatly reduced.
- 3) For the system described here, AGATE and HIPS produce effectively the same results.

It should be noted that the Dynamic AGATE discussed to this point in the paper typically resulted in six to eight model updates. Experiments were conducted to see if additional updates would further improve the accuracy. In order to do this, the approach was modified so that the robot moved one third of the way to the target each time rather than half way. This resulted in, on average, five additional updates per experiment. Another 30 experiments were conducted in this manner. However, no statistically significant change was observed. Thus it was concluded that this number of updates slowed the system without a significant advantage in accuracy.

The accuracy is not as high as that shown in previous work with AGATE and HIPS. Examining the sources of error described above, there are a variety of reasons for this increased error. Error #1 is much smaller than other systems tested, and errors #2 and #3 are comparable. Error #6, however, is larger. In previous work with HIPS and AGATE, highly accurate kinematic models of the end effectors (specifically the relationship between the end effector and the fiducials) were used. It is more realistic for an industrial tool which will have to be changed regularly to have an operator use the tool-teaching algorithms built-in to the ABB. However, this introduces more error. Taking the mean of errors #1, #3, and #6 would predict a mean error of approximately 2mm.

Errors #4, #5, and #7, however, should be expected to be larger in the experiments described in this paper than in previous work using HIPS or AGATE. This is because the distance from the cameras to the workspace is much greater (2.5m in this case compared to approximately 0.5m in previous studies). This leads to increased error in several ways. First, it increases the error in locating the target and the LEDs in the image plane. Second, it increases the size of a voxel. Thirdly, and most importantly, it increases the importance of stereo correspondence error. The importance of this effect was shown in [15]. But in that set of experiments, the error shows was only 1.5mm/pixel. In the experiments shown in this paper, the stereo correspondence error was over 20mm/pixel. Given the magnitude of these errors, showing an positioning overall error on the order of 10mm is consistent with expectations.

It is also worth noting that a major advantage of this methodology is that it allows the placement of the stereo rig on a simple tripod. This tripod can be moved relative to the robot without impacting the performance of the system, since the system can re-estimate the CAHVOR camera model at any time. In industrial settings, the cameras would certainly be fixed – but this would provide the ability to adjust if the cameras are bumped or moved during operation or maintenance activities. The method furthermore does not require that the images are rectified, and the alignment of one camera relative to the other is arbitrary, as long as the workspace can be seen by both cameras.

## 7 Continuing Work

While the work to date has shown that Dynamic AGATE shows improvement over Static AGATE, and that AGATE and HIPS produce effectively the same result, much work remains to be done. In particular:

- Build a more accurate tool holder to improve the accuracy of measurements. It is unclear how much of the current error is due to the methodology, and how much is due to inaccuracy in the tool holder. As can be seen in Figure 3, the current setup is simply clamped on, in a manner that is neither accurate nor repeatable.
- Comparison with traditional CAHVOR stereo (using a checkerboard for calibration).
- The use of more than two cameras (which is directly supported in AGATE).
- Comparison with an “off the shelf” disparity-based stereo system.
- Comparison with visual servoing.

When complete, it is expected that this will provide a rich dataset of direct comparison of four different, previously-published approaches to the problem of visually-guided manipulation. The one approach not specifically examined is Camera-Space Manipulation. While this approach has shown very good results, it requires widely separated and highly vergent cameras, and therefore would require a completely different experimental setup than the other methods. This would make a direct comparison difficult.

**Acknowledgment.** The authors would like to thank our industrial partner, American Trim, for their support of this project. Furthermore, thanks are due to Amber Cool and Patrick Whitten, undergraduate students without whose assistance we could not have gathered sufficient data for this paper. Further thanks to Dr. Sami Khorbotly, who developed the hardware for the LED system, and Dr. David Mikesell and undergraduate student Kyle Simmons who were responsible for the installation and testing of the robot.

## References

- [1] Corke, P.: Visual Control of Robot Manipulators – A review. In: Hashimodo, K. (ed.) Visual Servoing. Robotics and Automated Systems, pp. 1–31 (1993)
- [2] Morel, M.K.: System helps prepare sintered metal parts. Vision Systems Design (January 2004)
- [3] Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation 12(5) (October 1996)
- [4] Hammer, B., Koterba, S., Shi, J., Simmons, R., Singh, S.: An autonomous mobile manipulator for assembly task. Autonomous Robots 28(1) (2010)
- [5] Skaar, S., Brockman, W., Jang, W.: Three-dimensional camera space manipulation. Int. Journal Robotic Research 9 (August 1990)
- [6] Gonzalez-Galvan, E., Skaar, S., Korde, U., Chen, W.: Application of a Precision-Enhancing Measure in 3D Rigid-Body Positioning Using Camera-Space Manipulation. Int. Journal Robotic Research 16 (April 1997)
- [7] Gonzalez-Galvan, E.J., Skaar, S.B.: Efficient camera-space manipulation using moments. In: Proc. IEEE Intl. Conf. on Robotics and Automation, pp. 3407–3412 (1996)

- [8] Baumgartner, E.T., Bonitz, R.G., Melko, J.P., Shiraishi, L.R., Leger, P.C.: The Mars Exploration Rover Instrument Positioning System. In: Proceedings of the 2005 IEEE Aerospace Conference, Big Sky, MT (March 2005)
- [9] Robinson, M., Baumgartner, E., Nickels, K., Litwin, T.: Hybrid image plane/stereo (HIPS) manipulation for robotic space applications. *Autonomous Robots* 23 (2007)
- [10] Yoder, J.-D., Seelinger, M.: Long-range autonomous instrument placement. In: Proc. ISER, Rio de Janeiro, Brazil (2006)
- [11] Seelinger, M., Yoder, J.-D., Baumgartner, E.: Autonomous Go-And-Touch Exploration (AGATE). *Journal of Field Robotics* 29(3) (May/June 2012)
- [12] Perrollaz, M., Khorbotly, S., Cool, A., Yoder, J.-D., Baumgartner, E.: Teachless Teach-Repeat: Toward Vision-based Programming of Industrial Robots. In: IEEE International Conference on Robotics and Automation (2012)
- [13] Gennery, D.B.: Least-Squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points. In: Grun, A., Huang, T. (eds.) *Calibration and Orientation of Cameras in Computer Vision*. Springer Series in Information Sciences, vol. 34, pp. 123–136 (2001)
- [14] ABB Corporation, IRB 140,  
<http://www.abb.com/product/seitp327/7c4717912301eb02c1256efc00278a26.aspx?productLanguage=us&country=US>  
(accessed May 22, 2012)
- [15] Baumgartner, E.T., Klymyshyn, N.A.: Sensitivity Analysis for a Remote Vision-Guided Robot Arm under Imprecise Supervisory Control. In: *Sensor Fusion and Distributed Robotic Agents*, SPIE Proc., Boston, MA, vol. 2905, pp. 218–226 (October 1996)

# Part IV: ISER Session Summary on “Aerial Robotics”

Gaurav S. Sukhatme

University of Southern California

## Session Summary

Aerial robotics is a rapidly growing area. Recent advances in lightweight and durable airframes, low-power processors, compact batteries, and miniature low-power sensors have driven the development of small, ready-to-fly platforms that are excellent testbeds for research in advanced motion control, planning, multi-robot coordination, and sensor fusion, to name a few. At the other end of the size spectrum, military acquisition and deployment of unmanned drones has gone up nearly four orders of magnitude in a little over a decade. In spite of these dramatic advances, the level of autonomy of aerial robots is still relatively low and much remains to be done to develop systems that can function reliably in shared airspace over long periods of time in tandem with human operators while enduring significant environmental perturbation. The four papers in the session on Aerial Robotics highlight recent progress in various aspects of these problems. Das et al. in *‘Environmental Sensing using Land-based Spectrally-selective Cameras and a Quadcopter’*, address the problem of planning a survey by an aerial vehicle that can best utilize high-fidelity but incomplete information from sensors deployed on the ground. Shen and Michael in *‘State Estimation for Indoor and Outdoor Operation with a Micro-Aerial Vehicle’* described an algorithm for mapping using an aerial vehicle that rectifies the drift accumulated indoors when GPS is unavailable, thus enabling near-seamless indoor-outdoor operation. In *‘Influence of Aerodynamics and Proximity Effects in Quadrotor Flight’*, Powers et al. describe a mechanical model for a small quadrotor that accounts for ground (and ceiling!) effect. An interesting application is the ability to use the difference between the predicted and actual rotor speeds as a sensor that measures proximity to the environment. Finally, Kottas et al. in *‘On the Consistency of Vision-aided Inertial Navigation’* give an analysis (and a practical demonstration) of estimator consistency when an inertial system is aided by a vision system observing features in the environment. Notably, this work is promising not only for aerial platforms but also to human-worn augmentation systems (e.g., for the visually impaired).

# Environmental Sensing Using Land-Based Spectrally-Selective Cameras and a Quadcopter

Jnaneshwar Das, William C. Evans, Michael Minnig,  
Alexander Bahr, Gaurav S. Sukhatme, and Alcherio Martinoli

**Abstract.** We investigate the reconstruction of an environmental scalar field using robotic mobility and heterogeneous sensing. Using two land-based, immobile, co-located spectrally selective cameras, and a non-contact infrared-based temperature sensor on a quadcopter, we study the problem of reconstructing the surface temperature of the ground under survey. Both land units — a thermographic camera for low-resolution thermal images and a commercial digital camera for high resolution truecolor images — are mounted on an elevated camera rig. We explore methods for field reconstruction using a combination of the three imaging sensors. First, we show that the quadcopter data is correlated with the synoptic snapshots obtained by the thermal imaging camera. Next, we demonstrate upsampling of the low-resolution thermal camera data with truecolor images. This results in high-resolution reconstruction of the temperature field. Finally, we discuss adaptive sampling techniques that utilize the mobility of the quadcopter to ‘fill the gaps’ in data acquired by the thermal imaging camera. Our work experimentally demonstrates the feasibility of heterogeneous sensing and mobility to effectively reconstruct environmental fields.

## 1 Introduction

Fast sampling of terrestrial environmental fields is of importance for various studies. In this work, we address rapid sampling of environmental fields

---

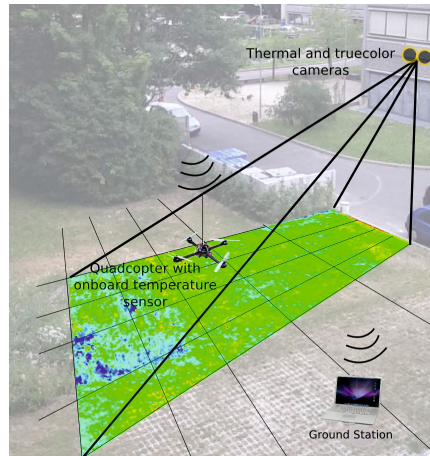
William C. Evans · Michael Minnig · Alexander Bahr · Alcherio Martinoli  
Distributed Intelligent Systems and Algorithms Laboratory,  
École Polytechnique Fédérale de Lausanne, Switzerland  
e-mail: {william.evans,michael.minnig,  
alexander.bahr,alcherio.martinoli}@epfl.ch

Jnaneshwar Das · Gaurav S. Sukhatme  
Robotic Embedded Systems Laboratory, Dept. of Computer Science,  
University of Southern California, Los Angeles, CA 90089 USA  
e-mail: {jnaneshd,gaurav}@usc.edu

using static and mobile imaging sensors. Specifically, we demonstrate effective reconstruction of the surface temperature field of a patch of vegetation. Accurate monitoring of surface temperature is desirable for atmospheric boundary layer studies over complex terrain [1], to cite one example. We describe an experiment where the surface temperature of a region is observed with images from a land-based thermal imaging camera augmented with truecolor images from a parallel-mounted commercial digital camera. In addition, a downward-facing non-contact infrared temperature sensor mounted on a quadcopter serves as a mobile sensing platform. The land-based thermal imaging camera is mounted at an elevation along with the truecolor camera, providing snapshots of the surface temperature and high resolution true-color images respectively. The quadcopter serves as a fast aerial observation platform, allowing rapid sampling of surface temperature using its downward-facing temperature sensor. This provides both speed and flexibility compared to land-based observation platforms such as robotic rovers. This work has three goals:

a) investigate upsampling of the thermal camera data using high resolution truecolor images from the digital camera, b) compare the temperature data acquired by the quadcopter with the synoptic thermal image captured from the thermal camera, and c) to explore adaptive sampling strategies that use the synoptic data from the thermal camera to guide the quadcopter to regions of high prediction uncertainty. Our goal is to demonstrate synergistic use of mobile and static sensors for rapid characterization of environmental phenomena. Such a capability is necessary when there are constraints on the use of land-based imaging sensors resulting in sparse data. This can happen due to long distance between test site and the land-based camera, or insufficient elevation of the camera rig. By using *mixed sensing*, we can reconstruct the temperature field at a resolution higher than that provided by the individual sensors.

The paper is organized as follows. In Section 2 we briefly describe related work. In Section 3, we lay the groundwork for the analysis of the experimental data by describing our technical approach. In Section 4, we describe our field setup followed by analysis of the data from the field trials. We conclude with a summary and discussion of future work in Section 5.



**Fig. 1** Illustration of the experimental setup to sample the surface temperature of a patch of land

## 2 Related Work

Ecological monitoring of large farmlands using UAVs has been studied for rapid mapping and classification of vegetation [2]. Adaptive sampling for environmental monitoring has been investigated in the context of intelligent placement of static sensor nodes [3], and informative paths for mobile aquatic platforms [4, 5]. Upsampling of multimodal remote sensing images has been explored to fuse low-resolution hyperspectral images with high-resolution truecolor images [6]. Our work presents an agile setup that provides a quick reconstruction of the environmental field in a region by use of selective spectral-cameras operating at different resolutions, aided by the mobility of a quadcopter.

## 3 Technical Approach

Our goal is to investigate the use of mixed sensing in the form of static land-based cameras and a quadcopter to rapidly sample the surface temperature of a terrestrial patch of vegetation. We will first describe the land-based camera rig and the quadcopter, followed by a discussion of unwarping and correction of the thermal and truecolor images. We then describe the three contributions of this work for mixed sensing field reconstruction: a) upsampling of thermal images using high-resolution truecolor images (Subsection 3.3), b) comparison of quadcopter data with land-based camera images (Subsection 3.4), and c) an adaptive sampling scheme for the quadcopter to augment land-based sensors (Subsection 3.5).

### 3.1 Sensing Apparatus

The sensing apparatus consists of three imaging sensors, two mounted on a land-based camera rig, and one mounted on a quadcopter. The camera rig consisted of an FLIR A320 thermographic camera with a resolution of 640x480 and a CANON 300D digital camera, both mounted on a tripod head, triggered by a computer to simultaneously capture truecolor and thermal images of the survey site. The aerial platform was an Ascending Technologies Hummingbird quadcopter. It used its onboard computer to log data from a



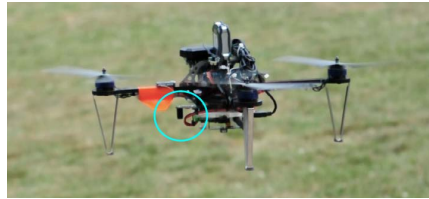
**Fig. 2** The camera rig consisting of a thermographic camera and a digital true-color camera mounted on a pan-tilt head



non-contact IR temperature sensor at a rate of 5 Hz along with GPS data at 1 Hz. We chose an IR-based temperature sensor so that the quadcopter would share a similar modality with the thermographic camera. A human operator maneuvered the quadcopter via remote control. The IR sensor captured the emitted IR radiance from a 10 degree view of field. Hence the temperature captured by the sensor was dependent on the height from the target. However, in this work, we did not model the sensor properties, and ignored the height of the quadcopter in flight.

### 3.2 Image Unwarping and Correction

We first explore fusion of the thermal and truecolor images for quick inspection of the scene. Since the thermal and truecolor cameras are mounted at an elevation, generating a perspective view of the scene, we compute perspective transforms to unwrap the truecolor and thermal images. First, we manually marked four landmarks in the thermal image and the truecolor image. We selected corners of man-made structures such as metal electric poles and concrete slabs because these were easily recognizable in both the thermal and truecolor images.



**Fig. 3** The quadcopter in-flight with the downward looking infra red temperature sensor

Next, we must ensure that all data shares a common frame of reference. We proceed by transforming all images to the Earth’s coordinate frame, with an approach similar to that used during the image unwarping step. Landmarks in the thermal and truecolor images are used along with landmarks in a satellite truecolor image of the scene obtained from commercial map servers (e.g., Google Earth). The transformation was computed using the OpenCV library. Once a perspective transformation matrix is computed, we obtain the unwrapped data points  $z = [x, y, t]^T$ , where  $x$  is the longitude,  $y$  is the latitude, and  $t$  is the color value of the pixel that was unwrapped.

### 3.3 Upsampling

To demonstrate upsampling of sparse thermal data using dense truecolor data, we subsample a sparse set of points from the unwrapped thermal camera image along with the corresponding truecolor pixel values<sup>1</sup>. Our goal is to learn a model that predicts surface temperature at unobserved locations using

<sup>1</sup> Computed using a nearest neighbor search with the thermal camera data points.

the truecolor data. The underlying assumption is that surface patches with similar color will have similar temperature.

## Gaussian Process Regression

We use Gaussian process regression (GPR) [7], a nonlinear Bayesian regression technique commonly used in geostatistics under the name ‘Kriging’. It assumes that the samples from the function to be estimated are normally distributed with the covariance between samples given by a ‘kernel’ or covariance function. We consider the case where the observations are unbiased, that is, the mean of the joint Gaussian distribution is zero. This can be simply satisfied by ‘demeaning’ the observed data. As a result of its formulation, GPR automatically achieves model regularization from data only, without having to choose model complexity parameters a priori. Additionally, GPR is defined completely by a kernel function that controls how quickly the input space becomes decorrelated. This enforces smoothness constraints in the trained function, ideal for spatial models where usually the observed values for nearby input samples are more correlated than the ones farther apart.

Assume we have training data given by  $D = \langle x_1, y_1 \rangle, \dots, \langle x_n, y_n \rangle$ , drawn from the noisy process,

$$y_i = f(x_i) + \epsilon \quad (1)$$

where  $\epsilon$  is a Gaussian noise term.

Given the training data, posterior mean and covariance for a test data point  $x_*$  is given by the following equations,

$$GP_\mu = k_*(K + \sigma_n^2 I)^{-1} y \quad (2)$$

$$GP_\Sigma = k(x_*, x_*) + k_*(K + \sigma_n^2 I)^{-1} k_* \quad (3)$$

The kernel function  $k$  is usually chosen to be a squared-exponential function given by,

$$k(x_p, x_q) = e^{-\frac{1}{2\lambda^2} |x_p - x_q|^2} \quad (4)$$

where  $\lambda$  is the decorrelation length scale. The hyperparameters for the kernel function can be learned using iterative methods such as conjugate gradient descent.  $K$  is the Gram matrix with its elements given by  $K_{pq} = k(x_p, x_q)$ ,  $I$  is the identity matrix, and  $k_*$  is the vector of covariances between the test data point and the training data points.

To apply the GPR model to upsample thermal image data, let us consider the unwarped pixels from the truecolor camera given by the vector  $X_{tc} = \langle Lon_{tc}, Lat_{tc}, R_{tc}, G_{tc}, B_{tc} \rangle$ . We use nearest neighbor search to obtain a training dataset of thermal camera image data points and their corresponding truecolor pixel data. This is given by  $X_{train} = \langle Lon, Lat, R, G, B \rangle$ , and  $Y_{train} = T$ . Now, we use GPR to learn a function  $f$  that maps prediction points  $X_{test} = \langle Lon_{test}, Lat_{test}, R_{test}, G_{test}, B_{test} \rangle$  where  $Lon_{test}$  and

$Lat_{test}$  are the longitude and latitude of unobserved locations, and  $R_{test}$ ,  $G_{test}$ , and  $B_{test}$  are the RGB pixel values from the high-resolution images captured by the truecolor camera corresponding to the query points.

### 3.4 Comparison of Quadcopter and Thermal Camera Data

To investigate the feasibility of using a quadcopter along with a land-based thermal camera, we need to compare the data obtained by the two sensors. Linear interpolation on the quadcopter sensor data on a regular grid immediately reveals visual similarity with the thermal imaging camera data. However, for a quantitative comparison of the two datasets, we compute the Pearson correlation coefficient for quadcopter data and the co-located thermal camera data calculated using nearest-neighbor search on each quadcopter data point. The Pearson correlation coefficient is given by,

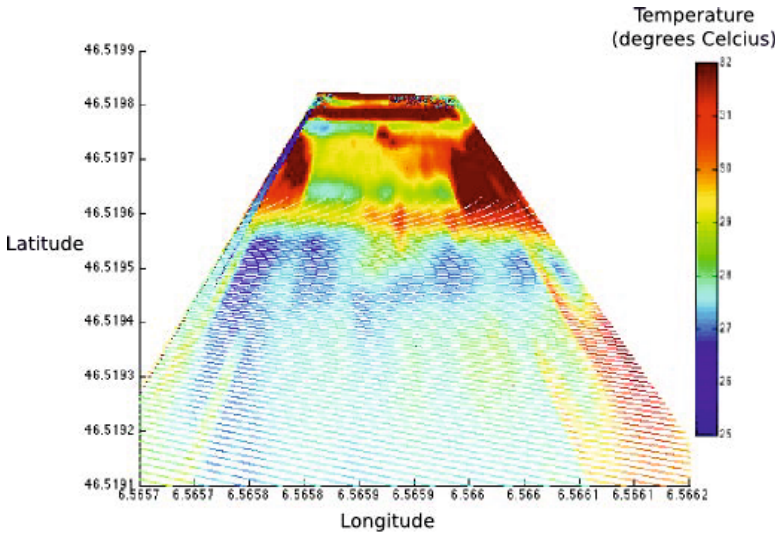
$$R = \frac{\sum_{i=0}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^n (x_i - \bar{x})^2 \sum_{i=0}^n (y_i - \bar{y})^2}} \quad (5)$$

where  $x$  and  $y$  are the two sensor data streams being compared. Higher values of  $R$  indicate a stronger correlation.

The imaging sensors on the thermographic camera and the IR sensor on the quadcopter are not cross-calibrated at the outset. For calibration, we choose a 440 sample data window (90 seconds) of the quadcopter data that is highly correlated with the thermal camera data ( $R > 0.8$ ) and use it to learn a linear mapping from raw quadcopter data to corrected quadcopter data, given by,  $t_{corr} = a_1 t + a_2$ , where  $t$  is raw quadcopter data point,  $t_{corr}$  is the corrected data point, and  $a_1$  and  $a_2$  are regression coefficients.

### 3.5 Adaptive Sampling with Quadcopter

Fig. 4 illustrates the sparsity of data away from the thermal camera once the acquired image is unwarped. This effect is more pronounced when the camera rig is farther away from the test site, or not highly elevated. This scenario will be common in unstructured environments. Also, there are sections of the thermal image (the vertical corners) without any data points. Our goal is to investigate field reconstruction that takes into account the uncertainty of estimates from the thermal camera data as a result of data sparsity. We propose greedily collecting data from regions with high variance. We use a sparser version of the thermal camera data to build a probabilistic spatial model of the temperature field using GPR as described earlier for the upsampling task. Then, we greedily add data points and analyze how many points are needed from the quadcopter data to reduce uncertainty in the reconstructed thermal image.



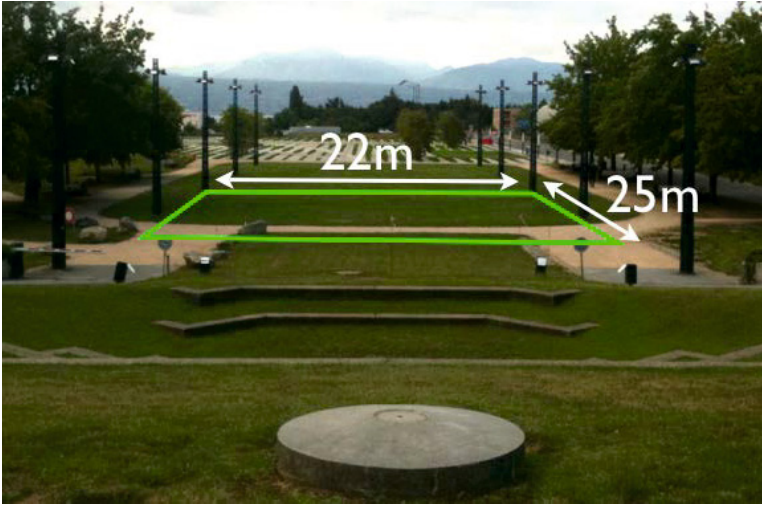
**Fig. 4** Perspective corrected (top-view) thermal camera data highlighting sparsity of measurements farther away from the camera (bottom of the image), and complete lack of data in the top right and left corners

## 4 Experiments and Results

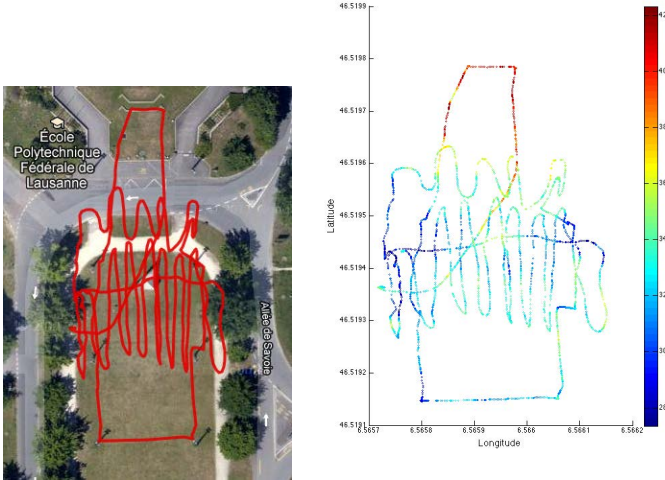
We carried out a series of field experiments on the EPFL campus in a 25m x 22m area with a significant portion being vegetation (grass), and a section of gravel path (Fig. 5). This provided a natural environmental field for the measurement of surface temperature. The land based camera-rig was mounted on a tripod at an elevation of 5m and at a distance of 20m from the experiment site. Images were captured simultaneously by both cameras every minute during the course of the experiment. For our analysis, we use one such concurrent snapshot. The quadcopter was operated manually at a mean height of 3.8m by a human pilot for a period of 10 minutes to capture the surface temperature over a lawnmower pattern as shown in Fig. 6. We show results from one of the field trials.

### 4.1 Image Unwarping and Correction

Fig. 7 shows the result of using visual landmarks in the truecolor and thermal images to find the transformation between the two cameras, which was then used to generate an overlaid image of the scene showing both truecolor and thermal images. This is useful as an initial overview of the scene and can be obtained in realtime. Next, we obtained a remotely-sensed truecolor image of the scene from a commercial map-server. From this, we used visual landmarks to compute a perspective transform to unwarp the images from the thermal

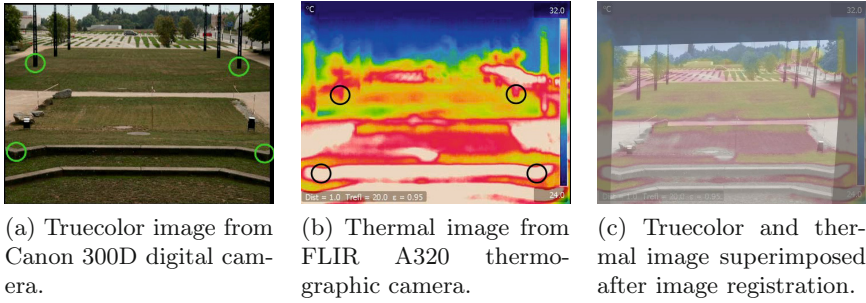


**Fig. 5** The experiment site was a 25m X 22m patch of land with varying density of grass, and a gravel path. The camera-rig was at a height of 9m from the test site, at a distance 30 m from the nearest edge of the test site.

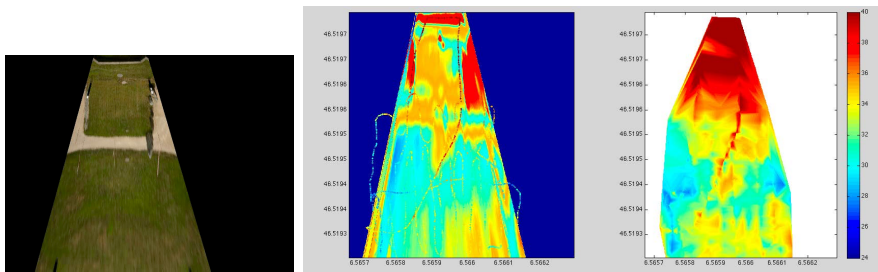


(a) The quadcopter path (solid red path) from one of the experiment runs. A human operator maneuvered the quadcopter to carry out a 'lawnmower' pattern. (b) Quadcopter temperature data points from the field trial.

**Fig. 6** Data from quadcopter field trial



**Fig. 7** Landmarks in the truecolor and thermal images used to generate an overlaid image for quick survey of the scene



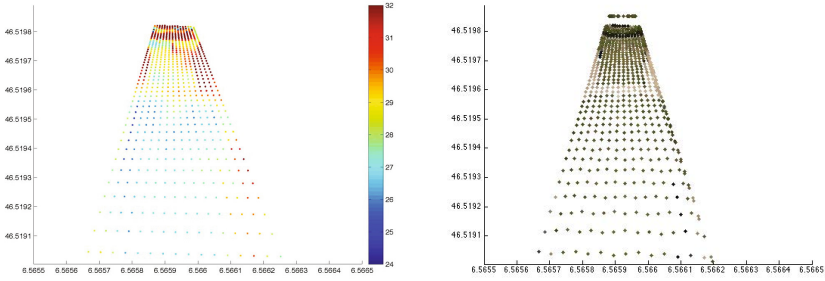
**Fig. 8** Visual comparison of corrected truecolor image, thermal image, and interpolated quadcopter data

camera and the digital truecolor camera. Fig. 8 shows the unwarped top view of the truecolor and thermal images. We performed interpolation on the GPS-tagged surface temperature data collected by the quadcopter for initial visual comparison of the two datasets. Fig. 8b shows this image.

### 4.2 Upsampling

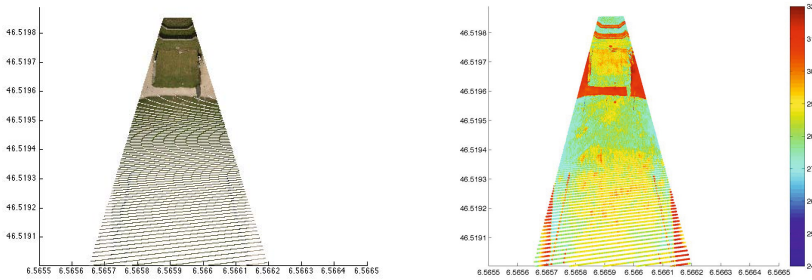
As described in Section 3, we use GPR on thermal camera data augmented with truecolor data to predict temperature at unobserved locations where truecolor data are available. The result of the upsampling analysis is shown in Fig. 9.





(a) Sparse thermal camera data.

(b) Sparse truecolor data corresponding to thermal camera data.



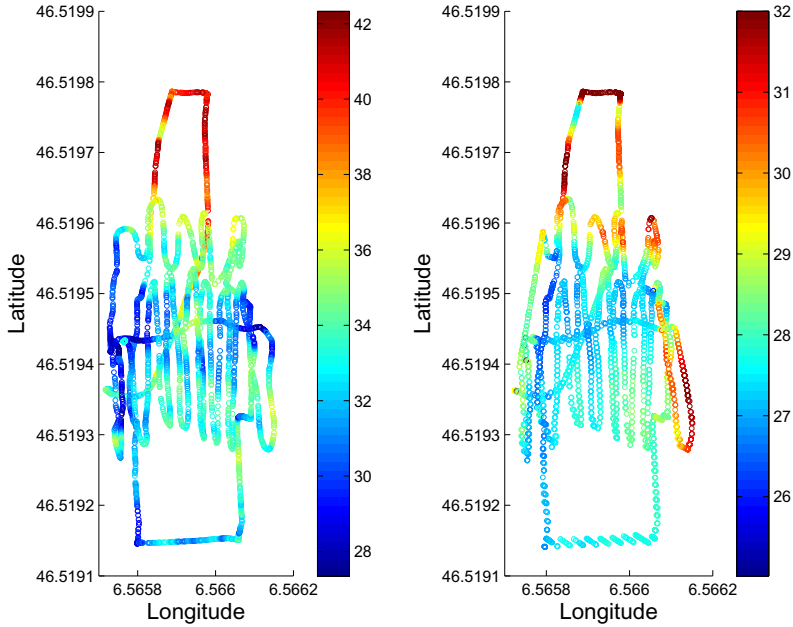
(c) Dense prediction points, with true-color data.

(d) Estimated thermal field.

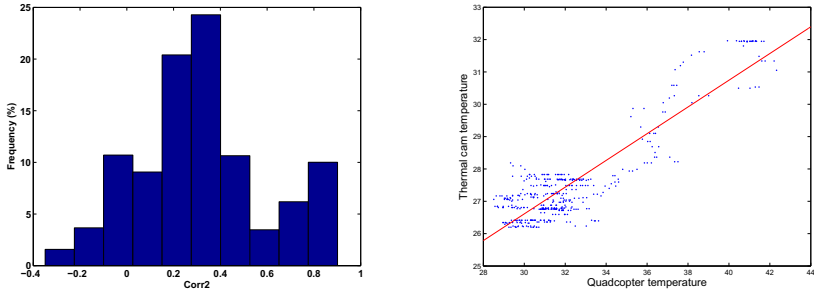
**Fig. 9** Upsampling thermal camera data with high-resolution truecolor data using Gaussian process regression with input augmented with RGB data

### 4.3 Comparison of Quadcopter and Thermal Camera Data

We computed the Pearson correlation coefficient between the quadcopter data points and the perspective corrected data points from the thermal imaging camera. Since the thermal camera data density is much higher than the quadcopter data, we found the Euclidean nearest-neighbor thermal data points to the quadcopter data. Fig. 10 shows the quadcopter data alongside nearest-neighbor thermal camera data. The two vectors were of length 3607 data-points each, and showed  $R = 0.477$ , demonstrating a statistically significant correlation between quadcopter data and thermal camera data. Additionally, to analyze the effect of outliers in the quadcopter data (for example, due to unfavorable altitude), we computed the Pearson coefficient on a sliding window of 440 data points (corresponding to 90 second of quadcopter flight time). The resulting distribution of correlation coefficient is shown in Fig. 11a.



**Fig. 10** Quadcopter data points (left) and nearest-neighbor thermal camera data points (right)



(a) Pearson correlation coefficient distribution for a sliding window of 440 datapoints over quadcopter and nearest-neighbor thermal camera data.

(b) Cross-calibration fit between quadcopter and thermal camera data.

**Fig. 11** Correlation between quadcopter and thermal camera data, and plot of linear cross-calibration fit



Next, we cross-calibrate the quadcopter data with the thermal camera data by finding a linear fit to a section of the quadcopter and thermal camera data with  $R > 0.8$ . Fig. 11b shows the the data points and the resulting linear fit.

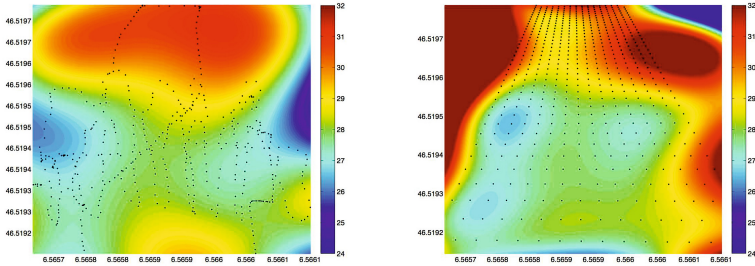
#### 4.4 Adaptive Sampling with Quadcopter

We flew the quadcopter remotely to carry out lawnmower surveys of the test area capturing samples of temperature with the downward looking IR temperature sensor. From this, we can choose points to emulate adaptive collection of data. This approach allows us to try various techniques adaptively without having to perform multiple experiments. We use a subsampled version of the thermal camera image as the pilot data to learn a GPR-based probabilistic regression model of the temperature field. We then use the variance of the field to greedily choose new sample points. As a reference, we show the reconstructed field from the quadcopter data in Fig. 12a. Fig. 12b shows the initial reconstruction of the temperature field from the thermal camera data. The top left and right corner of the reconstructed field are regions that exhibit extrapolation, with high associated uncertainty, as showed in Fig. 12c. Data from the quadcopter is used to fill these gaps, and in Fig. 12d, we see twenty additional data points added greedily to the reconstruction from the cross-calibrated quadcopter dataset. Each addition of a data point is followed by relearning of the temperature field. As seen in this figure, as a result of addition of the new samples, the top left corner of the field now exhibits moderate temperature.

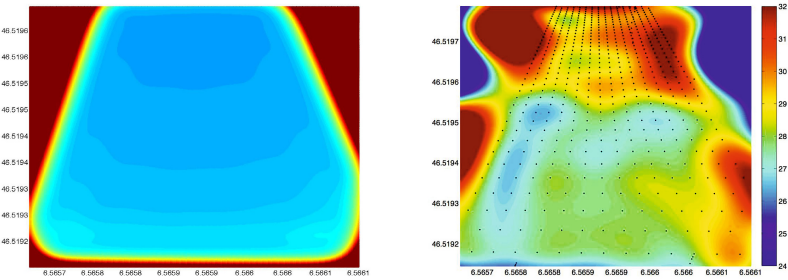
## 5 Discussion and Conclusions

In this paper, we presented an experiment using mixed sensing to reconstruct the surface temperature of a patch of land. A land-based camera-rig consisting of a thermographic camera and a high-resolution truecolor camera was used to generate upsampled thermal images of the experiment site. A quadcopter equipped with a downward looking IR temperature sensor measured surface temperature during flight. We compared the data from quadcopter with the thermal camera and found they are correlated (Pearson coefficient of 0.47). Finally, we investigated adaptive sampling strategies to fill gaps in thermal camera data using the quadcopter. We achieved this by using data from a quadcopter run offline.

This work has limitations that merit future work. First, we have not included the quadcopter altitude in the estimation of the temperature field. Since the IR temperature sensor has a relatively large field of view, the altitude has an impact on the measured data. This likely this has an impact on the correlation between the quadcopter data gathered during the field trial and the corresponding thermal camera data. Fig. 13 shows the distribution of altitude for the 10 minute flight at the experiment site. In the future, we

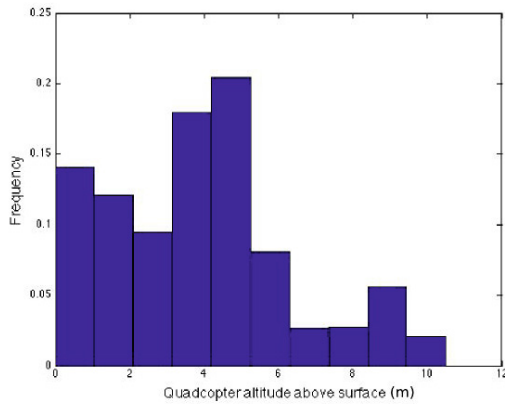


(a) GPR reconstruction on a regular grid for quadcopter data. Black dots show training data locations. (b) GPR reconstruction on a regular grid for thermal data. Black dots show training data locations.



(c) Prediction variance for reconstructed thermal image. Red regions show high uncertainty. (d) GPR reconstruction on a regular grid for thermal data after 20 adaptive samples from quadcopter dataset.

**Fig. 12** GPR reconstructed thermal and quadcopter data



**Fig. 13** The distribution of quadcopter altitude during the field trial

plan to model the IR-based temperature sensor as a pixel-average sensor and take into account the effect of height while reconstructing the temperature field. Second, we have used RGB data from the truecolor camera to augment the thermal camera image to perform upsampling. We plan to explore other characteristics of the land patch in addition to RGB for this task. Finally, we have not carried out online adaptive sampling experiments with the quadcopter. Instead, we collected data using a lawnmower survey and used the data offline to emulate new samples. In future, we will carry out online experiments to validate our approach.

**Acknowledgements.** This work was supported in part by the National Science Foundation under award CCF-0120778 and IIS-1107011. We thank the organizers of the Twenty-second International Joint Conference on Artificial Intelligence (IJCAI) Doctoral Consortium, and the Swiss National Center of Competence in Research in Robotics (NCCR Robotics) for facilitating the extended research visit by the first author to EPFL that made this work possible. William C. Evans and Alexander Bahr were partially supported by "The Swiss Experiment" of the Competence Center Environment and Sustainability of the ETH Domain (CCES), and by the NCCR Transfer project "Tamperproof Monitoring Solution for Weather Risk Management" sponsored by the Swiss National Science Foundation and managed by the National Center of Competence in Research in Mobile Information and Communication Systems (NCCR-MICS). We thank Holly Oldroyd and Daniel Nadeau of the Environmental Fluid Mechanics Laboratory at EPFL for lending us the thermographic camera and helping us with the science motivation.

## References

1. Nadeau, D., Parlange, M.: Atmospheric Boundary Layer Dynamics of Transitional Flows over Complex Terrain. Ph.D. dissertation, EPFL (2011)
2. Bryson, M., Reid, A., Hung, C., Ramos, F., Sukkarieh, S.: Cost-Effective Mapping Using Unmanned Aerial Vehicles in Ecology Monitoring Applications. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) *Experimental Robotics*. STAR, vol. 79, pp. 509–523. Springer, Heidelberg (2012)
3. Krause, A., Singh, A., Guestrin, C.: Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *J. Mach. Learn. Res.* 9, 235–284 (2008)
4. Zhang, B., Sukhatme, G.S.: Adaptive Sampling for Estimating a Scalar Field using a Robotic Boat and a Sensor Network. In: *IEEE International Conference on Robotics and Automation*, pp. 3673–3680 (2007)
5. Binney, J., Krause, A., Sukhatme, G.S.: Informative Path Planning for an Autonomous Underwater Vehicle. In: *IEEE International Conference on Robotics and Automation*, pp. 4791–4796 (2010)
6. Thomas, C., Ranchin, T., Wald, L., Chanussot, J.: Synthesis of Multispectral Images to High Spatial Resolution: A Critical Review of Fusion Methods Based on Remote Sensing Physics. *IEEE T. Geoscience and Remote Sensing* 46(5), 1301–1312 (2008)
7. Rasmussen, C.E.: *Gaussian Processes for Machine Learning*. MIT Press (2006)

# State Estimation for Indoor and Outdoor Operation with a Micro-Aerial Vehicle

Shaojie Shen and Nathan Michael

**Abstract.** In this work, we detail a methodology for estimating the state of a micro-aerial vehicle (MAV) as it transitions between different operating environments with varying applicable sensors. We ensure that the estimate is smooth and continuous throughout and provide an associated quality measure of the state estimate. We address the challenge of maintaining consistency between local and global measurements and propose a strategy to recursively estimate the transform between different coordinate frames. We close with experiments that validate the approach and the resulting performance as a MAV navigates between mixed indoor and outdoor environments.

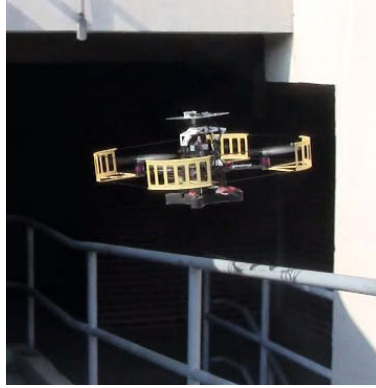
## 1 Introduction

In this paper, we focus on the problem of estimating the state of a micro-aerial vehicle (MAV) while operating in and transitioning between indoor and outdoor environments. Aerial vehicles offer mobility and perspective advantages over ground platforms and can transition through small openings such as windows and doors. This fact makes MAVs particularly applicable to problems such as search-and-rescue and surveillance. In recent work, we focused on the problem of autonomous indoor navigation in multi-floor environments using only onboard sensing and computation [1]. Through our previous work, we developed a methodology that enables mapping, localization, planning, and control of a MAV in complex indoor environments. This paper builds upon this previous work to permit operation in both indoor and outdoor environments, including transitions between these mixed environments.

The problem of autonomous navigation with an aerial vehicle in outdoor or indoor environments is thoroughly studied in the literature. Julier and Uhlmann [2] and Merwe et al. [3] propose state estimation methods using IMU and GPS sensors

---

Shaojie Shen · Nathan Michael  
GRASP Laboratory  
University of Pennsylvania  
Philadelphia, PA 19104, USA  
e-mail: {shaojie, nmichael}@grasp.upenn.edu



**Fig. 1** We address the estimation of the state of a micro-aerial vehicle as it transitions between indoor and outdoor environments. While the robot is equipped with a laser, GPS, magnetometer, and IMU, only a subset of those sensors may be operational at any time given the robot's state and the sensor operating conditions.

for navigation in outdoor environments. Recent developments toward autonomous MAV operation in indoor environments using onboard sensors (IMU, laser, and camera) include the works of Grzonka et al. [4], Bachrach [5], and Blosch et al. [6] as well as our own work [1].

A challenge in operating in both indoor and outdoor environments is the fact that the performance of sensors can vary between environments; sensors can provide meaningful data in one environment while becoming compromised in another environment. In this work, we consider a vehicle equipped with a standard GPS unit, laser range finder, magnetometer, and IMU (Fig. 1). The GPS provides relatively inaccurate and latent information compared to the other sensors and is only operational when outdoors and in view of satellites. Similarly, the magnetometer only provides accurate information when outdoors. The laser scanner provides accurate information indoors but is compromised by direct sunlight and provides limited to no information in large open spaces. The IMU is consistent and operational in both indoor and outdoor environments but is too inaccurate to enable stable feedback control as the only source of information. Conditions that challenge the sensor capabilities extend beyond just indoor and outdoor transitions and include negotiating GPS-shadowing due to buildings and trees. A summary of sensor characteristics and limitations is provided in Table 1.

**Table 1** Characteristics and limitations of the onboard sensors on our MAV platform

Sensor	Indoor	Transition	Outdoor	Frame	Accuracy	Relevant Failure Modes
GPS	×	√	√	Global	Low	Obstructed view of satellites
Magnetometer	×	√	√	Global	Low	Magnetic interference
Laser	√	√	×	Body	High	Direct sunlight, open spaces
IMU	√	√	√	Body	Low	None

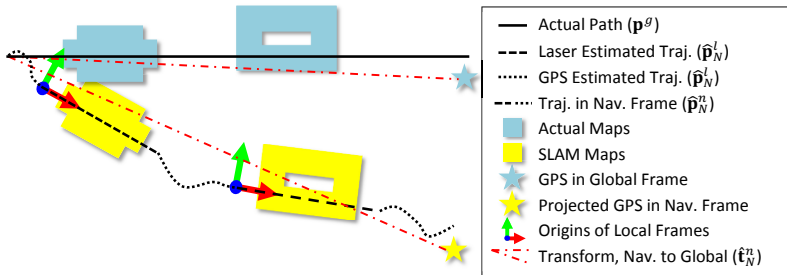
Consistency between local and global reference frames also becomes a challenge when operating in environments where the primary position sensors provide either relative or global state information. The fusion of global observations from GPS and relative observations from sensors such as lasers and cameras is addressed by Carlson [7] and Schleicher et al. [8]. These authors propose strategies to introduce additional global constraints to the existing simultaneous localization and mapping (SLAM) formulation to handle the correspondence between local and global frames. Although these methods yield reasonable results, there is no guarantee that the resulting pose estimate is low-latency and smooth, which is of vital importance for stable feedback control of MAVs. Moore et al. [9] investigate the problem of maintaining smoothness in the state estimate and handling position discontinuity due to GPS interference, but these methods do not directly extend to a MAV as the approach relies on information only available to a ground vehicle (wheel odometry). Additionally, we require a notion of the quality of the estimate as we wish to leverage robust control methods that require both a mean and covariance measure of the state estimate [10].

While there are similarities in terminology in our presentation to submap-based SLAM approaches [11, 12], there are fundamental differences in the problem definition and approach. For submap-based SLAM, local maps are created via a sparse sampling of the robot trajectory and local maps are assumed to be accurate. Local maps are linked by incremental motion constraints or loop-closure correspondence. Pose graph optimization techniques can be applied to obtain globally consistent maps [13, 14]. However, for this work, local maps are created based on sensor availability rather than distance, we do not limit the size of local maps, and we allow for the existence of estimation errors within each local map. Further, our local maps are not linked by any incremental motion constraints, and thus the above optimization techniques are no longer applicable.

## 2 Approach

In this section we discuss our approach to estimating the state of the MAV during indoor and outdoor operations. We discuss the necessity of using different reference frames for state estimation in Sect. 2.1. In Sect. 2.2, we detail an Unscented-Kalman Filter (UKF) formulation that includes process and measurement models for the sensors under consideration. We discuss globally consistent map building and management, as well as loop closure detection and map correction in Sect. 2.3. We conclude by integrating planning and control modules into our proposed approach to form a complete autonomous aerial navigation system (Sect. 2.4).

We build upon our previous work [1] and therefore assume the existence of a laser-based SLAM solution running in real-time onboard the robot. The output of the laser-based SLAM is a pose estimate of the robot in three dimensions with respect to the map generated via the laser observations.



**Fig. 2** Graphical illustration of the use of different coordinate frames in our work. The robot travels along a straight line (black) between two buildings (cyan) in the global frame. When navigating outdoors and away from buildings, only GPS is available as the primary position sensor, and the resulting pose estimate is inaccurate (dotted line). Laser-based SLAM provides an accurate position estimate (dashed line) in a local map (yellow) when navigating indoors or close to the buildings. We address the problem of ensuring consistency between global and local information sources by creating a navigation frame and estimate its transform to the global frame using a recursive formulation.

## 2.1 Reference Frames

We define four reference frames for this work.

- Body frame: defined with respect to the body of the robot and denoted by  $(\cdot)^b$ . All laser and IMU measurements are made in the body frame.
- Global frame: defined with respect to the inertial frame and denoted by  $(\cdot)^g$ . All global measurements (GPS and magnetometer) are made in the global frame.
- Local frame: defined with respect to the origin of the laser-based localization solution when transitioning between regions of mixed sensor information (clarified below) and denoted by  $(\cdot)^l$ .
- Navigation frame: defines the reference frame used by the onboard feedback control (clarified below) and denoted by  $(\cdot)^n$ .

Multiple local frames may be defined through the course of an experiment (see Fig. 2). A new local frame is created when the covariance matrix associated with the laser-based localization, which is computed via an inversion of the Fisher's Information Matrix of the laser scan [15], changes from singular to nonsingular. Such events correspond to when the laser sensor provides sufficiently salient information that we may again include this information in the state estimate (i.e. the sensor is no longer in a failure mode). In experimentation, this event usually occurs when the robot starts to observe strong geometric structures (e.g. buildings) or escapes from strong direct sunlight. Note that even if the laser-based localization fails to provide a valid solution, the state of the robot in the current local frame can still be estimated via sensor measurements made in other frames (IMU, GPS and magnetometer) and transformed into the local frame.

As noted above, maintaining a consistent, smooth, and continuous state estimate for use by the onboard feedback control is of particular importance when working with a MAV and considering operation in mixed environments. We define the frame associated with this state estimate as the navigation frame and now detail its definition. For the sake of brevity, we only consider the pose of the robot,  $\mathbf{p}$ , in this section and simplify the estimation problem to consider a 2D system ( $x$  and  $y$ ) where the vertical position and velocity of the robot is estimated by downward facing laser beams and a pressure sensor. Therefore, the GPS and magnetometer sensors provide 2D pose and velocity observations while the laser-based localization system provides a 2D pose estimate. Roll and pitch angles are obtained directly from the IMU.

Denote the true global pose of the robot as  $\mathbf{p}^g$  and the global pose estimate of the robot as  $\hat{\mathbf{p}}^g$ . Similarly, the pose in the  $k^{\text{th}}$  local frame is given by  $\mathbf{p}_k^l$ . The direct output of the laser-based localization system, which is a noisy estimate of  $\mathbf{p}_k^l$ , is defined as  $\hat{\mathbf{p}}_k^l$ . To simplify the presentation, we use the pose compounding ( $\oplus$ ) and inverse ( $\ominus$ ) operators to capture the transformation between different coordinate frames [16].

Consider a robot, initialized at the global origin, that moves into and out of regions with available laser-based localization information such that it now considers the definition  $N$  local frames. The true global pose of the robot is:

$$\mathbf{p}^g = \mathbf{p}_1^{l_f} \oplus \mathbf{p}_2^{l_f} \oplus \cdots \oplus \mathbf{p}_{N-1}^{l_f} \oplus \mathbf{p}_N^l \quad (1)$$

where  $\mathbf{p}_k^{l_f}$  is the final pose of the vehicle in the  $k^{\text{th}}$  local frame. As  $\mathbf{p}_k^{l_f}$  is unavailable, we consider the estimated value:

$$\begin{aligned} \hat{\mathbf{p}}_k^{l_f} &= \mathbf{p}_k^{l_f} \oplus \mathbf{e}_k^{l_f} \quad k = \{1, \dots, N-1\} \\ \hat{\mathbf{p}}_N^l &= \mathbf{p}_N^l \oplus \mathbf{e}_N^l \end{aligned}$$

where  $\mathbf{e}_k^{l_f}$  and  $\mathbf{e}_N^l$  are estimation errors. We define the pose in the navigation frame as follows:

$$\begin{aligned} \hat{\mathbf{p}}_N^n &= \hat{\mathbf{p}}_1^{l_f} \oplus \hat{\mathbf{p}}_2^{l_f} \oplus \cdots \oplus \hat{\mathbf{p}}_{N-1}^{l_f} \oplus \hat{\mathbf{p}}_N^l \\ &= \left( \mathbf{p}_1^{l_f} \oplus \mathbf{e}_1^{l_f} \oplus \mathbf{p}_2^{l_f} \oplus \mathbf{e}_2^{l_f} \oplus \cdots \oplus \mathbf{p}_{N-1}^{l_f} \oplus \mathbf{e}_{N-1}^{l_f} \oplus \mathbf{p}_N^l \right) \oplus \mathbf{e}_N^l \quad (2) \\ &= \mathbf{p}_N^n \oplus \mathbf{e}_N^l \end{aligned}$$

where the subscript  $(\cdot)_N$  in  $\mathbf{p}_N^n$  indicates the number of local frames integrated into the navigation frame. Hence, the navigation frame serves a similar role as wheel odometry for ground robots. Clearly  $\mathbf{p}_N^n \neq \mathbf{p}^g$  as  $\mathbf{p}_N^n$  incorporates the accrual of error in pose estimates and is therefore suitable for applications that mainly require only local accuracy, such as feedback control and obstacle avoidance. Thus, we define  $\mathbf{p}_N^n$  as the robot pose in the navigation frame (see Fig. 2). The estimated pose in the navigation frame,  $\hat{\mathbf{p}}_N^n$ , follows a similar error model as the pose estimate in the current local frame  $\hat{\mathbf{p}}_N^l$ .



We must now consider the transform between the global frame and the navigation frame. For notational convenience, we first define  $\mathbf{q}_k$  as:

$$\mathbf{q}_k = \hat{\mathbf{p}}_1^{l_f} \oplus \hat{\mathbf{p}}_2^{l_f} \oplus \cdots \oplus \hat{\mathbf{p}}_k^{l_f}. \quad (3)$$

Therefore, if there are  $N$  local frames, then  $\mathbf{q}_1, \dots, \mathbf{q}_{N-1}$  are all known and readily computed constant transforms.

Define the time-varying rigid body transform from the navigation frame to the global frame as  $\mathbf{t}_N^n$  such that:

$$\mathbf{p}^g = \mathbf{t}_N^n \oplus \mathbf{p}_N^n. \quad (4)$$

As  $\mathbf{t}_N^n$  is not directly accessible, we estimate this transform (with error  $\mathbf{e}$ ):

$$\hat{\mathbf{t}}_N^n = \mathbf{t}_N^n \oplus \mathbf{e}.$$

The resulting global pose estimate may be written as:

$$\hat{\mathbf{p}}^g = \hat{\mathbf{t}}_N^n \oplus \hat{\mathbf{p}}_N^n = \mathbf{t}_N^n \oplus \mathbf{e} \oplus \mathbf{p}_N^n \oplus \mathbf{e}^{l_N}.$$

For this work, we consider the state estimation problem in the navigation frame, transforming all global sensor information into this frame via  $\mathbf{t}_N^n$ . We now discuss a recursive filtering formulation that permits estimation of this transform along with the state of the robot.

## 2.2 UKF-Based Sensor Fusion

We employ a UKF framework with delayed measurement compensation to estimate the pose and velocity of the robot, hidden sensor bias parameters, and the transformation between the global and navigation frames [3]. All quantities, unless otherwise specified, are defined with respect to the navigation frame. We also assume that  $N$  local frames have been created since the beginning of the experiment. The subscript and superscript  $(\cdot)_N^n$  will be omitted for the remainder of the paper. The system state is defined as:

$$\mathbf{x} = [\mathbf{r}, \dot{\mathbf{r}}, \Phi, \mathbf{a}_b, \Psi_b, \mathbf{t}_N^l]^T \quad (5)$$

where  $\mathbf{r} = [x, y, z]^T$  is the position of the robot and  $\Phi = [\phi, \theta, \psi]^T$  is the roll, pitch, and yaw Euler angles that represent the 3D orientation of the robot.  $\mathbf{a}_b = [a_{b_x}, a_{b_y}, a_{b_z}]^T$  is the bias of the 3D accelerometer in the body frame and  $\Psi_b = [\phi_b, \theta_b]^T$  is the bias of the roll and pitch estimate from the IMU.  $\mathbf{t}_N^l = [\Delta x^l, \Delta y^l, \Delta \psi^l]^T$  is a 2D rigid body transformation between the global frame and navigation frame and related to  $\mathbf{t}_N^n$ . The definition of  $\mathbf{t}_N^l$  is discussed later in this section. As the UKF formulation follows a similar structure to existing works [3], we only briefly discuss the process model and provide specific details pertinent to the measurement update step.

## Process Model

We consider an IMU-based process model:

$$\begin{aligned}\mathbf{x}_{t+1} &= f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t) \\ \mathbf{u} &= [\boldsymbol{\omega}, \mathbf{a}]^T = [\omega_x, \omega_y, \omega_z, a_x, a_y, a_z]^T \\ \mathbf{v} &= [\mathbf{v}_\omega, \mathbf{v}_\mathbf{a}, \mathbf{v}_{\mathbf{a}_b}, \mathbf{v}_{\Psi_b}]^T\end{aligned}$$

where  $\mathbf{u}$  is the body frame angular velocities and linear accelerations from the IMU.  $\mathbf{v}$  represents additive Gaussian noise associated with the gyroscope, accelerometer, accelerometer bias, and IMU attitude bias.

## Measurement Model – Laser-Based SLAM

The measurement from the laser-based localization relative to the local frame is:

$$\tilde{\mathbf{z}}_N^l = [\tilde{x}_N^l, \tilde{y}_N^l, \tilde{\psi}_N^l]^T.$$

We transform this measurement into the navigation frame prior to the measurement update. The transformed laser measurement is:

$$\tilde{\mathbf{z}}_N^n = \mathbf{q}_{N-1} \oplus \tilde{\mathbf{z}}_N^l$$

where  $\mathbf{q}_{N-1}$  is defined in (3). The measurement model is linear and may be written as:

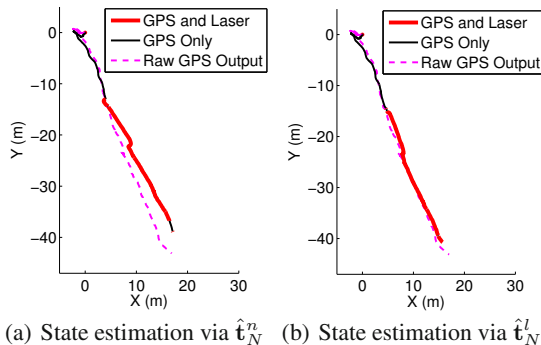
$$\mathbf{z}_N^n = H^n \mathbf{x} + \mathbf{n}_N^n, \quad H^n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \mathbf{0}_{3 \times 6} & 0 & \mathbf{0}_{3 \times 8} \\ 0 & 0 & 1 \end{bmatrix}$$

where  $H^n$  extracts the 3-DOF pose in the state and  $\mathbf{n}_N^n$  is additive Gaussian noise.

Note that the laser-based localization error covariance matrix is obtained in the local frame, not the navigation frame. We must transform this covariance matrix to the navigation frame. Assuming independence between the noise for position and heading, the covariance matrix of  $\mathbf{n}_N^n$  is:

$$\Sigma_N^n = \begin{bmatrix} \Sigma_{xy}^n & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & \sigma_\psi^{n2} \end{bmatrix}, \quad \Sigma_{xy}^n = R_{\psi_{q_{N-1}}} \Sigma_{xy}^l R_{\psi_{q_{N-1}}}^T$$

where  $R_{\psi_{q_{N-1}}}$  is the 2D rotation matrix obtained from the heading component in  $\mathbf{q}_{N-1}$ , and  $\Sigma_{xy}^l$  is a laser-based covariance measure and computed using the methods proposed in [15]. The standard deviation in heading remains the same in both the local frame and the navigation frame (e.g.  $\sigma_\psi^n = \sigma_\psi^l$ ). The measurement update for the laser-based localization is linear and can be performed via a KF update step.



**Fig. 3** The effects of accrued error in the estimate of the transformation from the navigation frame into the global frame,  $\mathbf{t}_N^n$ , are amplified as the robot moves away from the global origin. We choose to estimate  $\mathbf{t}_N^l$  rather than  $\mathbf{t}_N^n$  directly in our recursive filter formulation. Here, we show experimental data collected when transitioning between local frames. The red and black lines show the measurement prediction,  $\mathbf{z}^g$ , and the dashed magenta lines show the raw GPS position measurement,  $\tilde{\mathbf{z}}^g$ . In Fig. 3(a), we try to estimate  $\mathbf{t}_N^n$  directly. As the robot moves away from the origin, the UKF is unable to directly estimate  $\mathbf{t}_N^n$ , resulting in inconsistency between the measurement prediction and the actual GPS measurement. Figure 3(b) shows the result of the proposed approach (estimating  $\mathbf{t}_N^l$ ), where the measurement prediction is consistent with the actual measurement.

### Measurement Model – GPS and Magnetometer

GPS provides information about the horizontal vehicle position and velocity in the global frame. We ignore GPS altitude information as it is generally inaccurate. The magnetometer provides a measure of global orientation. The assembled measurement vector is:

$$\tilde{\mathbf{z}}^g = [\tilde{x}^g, \tilde{y}^g, \tilde{x}^g, \tilde{y}^g, \tilde{\psi}^g]^T.$$

We now motivate the estimation of  $\mathbf{t}_N^l$  in the system state (5). Recall that  $\mathbf{t}_N^n$  represents the transformation from the navigation frame into the global frame and includes the accrual of estimation errors during the evolution of the vehicle operation. As the robot moves away from the global origin, the effects due to error in the estimate of the transformation are amplified. Consequently, the estimator performance decreases. Therefore, rather than estimate the transform directly, we treat the transform as a compound operation on the current and prior transforms and pose estimates:

$$\mathbf{t}_N^n \oplus \mathbf{q}_{N-1} = \hat{\mathbf{t}}_{N-1}^{n_f} \oplus \mathbf{q}_{N-1} \oplus \mathbf{t}_N^l$$

where  $\hat{\mathbf{t}}_{N-1}^{n_f}$  is the estimated transformation between the global frame and the navigation frame at the time of the initialization of the current local frame. This approach moves the origin of the frame transform from the global origin to  $\mathbf{q}_{N-1}$ .

Based on the above discussion, the GPS measurement model is defined as:

$$\mathbf{t}_N^n = \hat{\mathbf{t}}_{N-1}^{n_f} \oplus \mathbf{q}_{N-1} \oplus \mathbf{t}_N^l \ominus \mathbf{q}_{N-1} = [\Delta x, \Delta y, \Delta \psi]^T$$

$$\mathbf{z}^g = h^g(\mathbf{x}) + \mathbf{n}^g = \begin{bmatrix} R_{\Delta\psi} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ R_{\Delta\psi} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \\ \psi + \Delta\psi \end{bmatrix} + \mathbf{n}^g$$

and can be performed via a nonlinear UKF measurement update procedure. The consequence of estimating  $\mathbf{t}_N^l$  and indirectly computing  $\mathbf{t}_N^n$  is a significant improvement in overall estimator accuracy and consistency (see Fig. 3).

### Measurement Model – Altitude, Roll, and Pitch

The vertical position and velocity of the robot is observed by downward facing laser beams and a pressure sensor while the roll and pitch angles are obtained from the IMU. Therefore, the third measurement is assembled as:

$$\tilde{\mathbf{z}}^a = [\tilde{z}, \tilde{\dot{z}}, \tilde{\phi}, \tilde{\theta}]^T.$$

As  $\mathbf{z}^a$  is a subset of the system state  $\mathbf{x}$ , this measurement model follows a linear form:

$$\mathbf{z}^a = H^a \mathbf{x} + \mathbf{n}^a, \quad H^a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \mathbf{0}_{4 \times 2} & \mathbf{0}_{4 \times 2} & 0 & 1 & 0 & \mathbf{0}_{4 \times 4} & 1 & 0 & \mathbf{0}_{4 \times 3} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

where  $H^a$  extracts the required elements in the system state and  $\mathbf{n}^a$  is additive Gaussian noise.

## 2.3 3D Map Generation

We associate a 3D local map with each local frame. If a valid laser-based localization solution is available, the laser scan is transformed into the local frame and the 3D map is created and updated via a multi-volume occupancy grid data structure [17]. We may transform the local map associated with local frame  $k$  into the global frame via its origin in the navigation frame  $\mathbf{q}_{k-1}$  and the corresponding estimated frame transform  $\hat{\mathbf{t}}_k^n$ .

We match the current local map against all previous local maps using a multi-resolution scan matching algorithm [18]. Detection of correspondences between the current and previous local maps results in the alignment and merger of these maps.

## 2.4 Planning and Control

While the focus of this work is on estimation, we validate our approach by integrating it with the planning and control components of the autonomous aerial navigation system. All experimental results consider the overall performance of the navigation system.

We use an RRT-based planner for online trajectory generation and obstacle avoidance given the current 3D map. The desired goals for the planner are provided by high-level commands from the human operator.

We employ a gain-scheduled LQR controller based on pre-computed optimal gains at various levels of state estimate accuracy [10]. To establish the expected state estimate accuracy across different environments, we experimentally obtain a set of representative covariance matrices by flying the robot in different environments and determine LQR gains offline. In practice, we find that this step is only necessary once and the pre-computed gains are applicable to all other experimental environments.

All high-level goals are provided by the operator in the global frame and transformed into the navigation frame prior to being sent to the planner and controller. Note that the transform  $\mathbf{t}_N^n$  is time varying and therefore the desired waypoint in the navigation frame changes over time. This subtle point highlights a key contribution and goal of this work, that despite the drift and non-smooth nature of GPS and magnetometer information along with changing reference frames, we defer these considerations to the goal definition in the navigation frame, not the vehicle state estimate.

## 3 Experimental Results

### 3.1 Experiment Design and Implementation Details

We present experimental results to demonstrate the performance of the proposed algorithm in mixed indoor and outdoor environments. We first detail a representative experiment that investigates the validity of the methodology detailed in this work. We then briefly outline experimental trials in other environments to demonstrate repeatable performance.

The robot platform is sold by Ascending Technologies, GmbH [19] and equipped with an IMU (accelerometer, gyroscope), magnetometer, and pressure sensor. We developed custom firmware to run at the embedded level to address feedback control and estimation requirements. The other computation unit onboard is a 1.6 GHz Atom processor with 1 GB of RAM. All sensing and processing is done onboard the robot without need for external infrastructure. We outfitted the robot with a Hokuyo UTM-30LX (laser) and a uBlox LEA-5T GPS module. A custom 3D printed mount is attached to the laser that houses mirrors and redirects a small number of laser beams upward and downward. Communication with the robot for monitoring experiment progress is via 802.11n networking. Figure 1 shows a picture of our robot

platform. All algorithm development is in C++ using ROS [20] as the interfacing robotics middleware.

There are two modes of operation for our robot (1) high-level waypoint control (discussed above) and (2) kinematic control using a desired velocity command provided by the operator. Switching between control modes can be done seamlessly without aborting the experiment. In either operation mode, the robot is autonomously controlled using the onboard state feedback and operator guidance (via desired waypoint or velocity inputs). As operator input, planning, and control are not the focus of this work, we only note that we switch between these modes during the experiments based on operator preference.

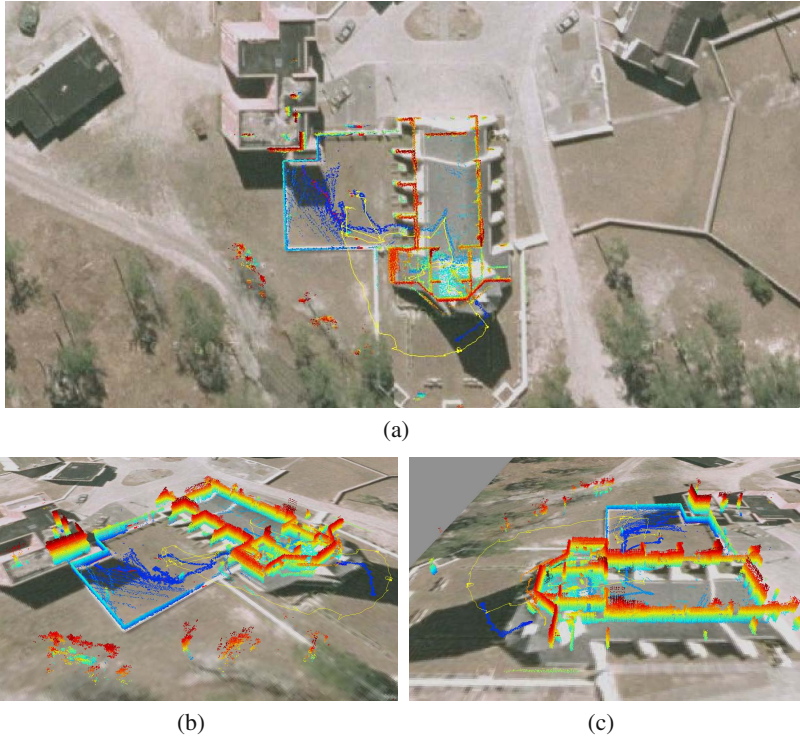
### 3.2 *Autonomous Flight with Indoor/Outdoor Transitions*

We now detail a representative field experiment conducted at an urban operations (UO) testing facility. Aerial satellite imagery of the experimental site and the resulting 3D map generated online by our system are shown in Fig. 4(a). The experimental site contains indoor environments, outdoor environments with building structure, and open outdoor spaces.

The experiment begins as the robot takes off in an open courtyard where laser-based localization fails to provide any valid solution (Fig. 5(a)). The robot flies inside a nearby building via an open window (Fig. 5(b)) and navigates the various rooms and corridors of the building (Fig. 5(c)) prior to exiting the building through a door located on the second story of the building (Fig. 5(d)). The robot then flies near a sparse wooded area (Fig. 5(e)). The robot is intentionally oriented away from the building so that the laser scanner does not observe any strong geometric structures. The experiment concludes as the robot returns to the courtyard, yielding loop closure across multiple local maps (Fig. 5(f)). Three local maps are created during the experiment and aligned into a globally consistent map when the robot returns to the courtyard and detects loop closure. An overlay of the resulting globally consistent 3D map with the corresponding satellite imagery is detailed in Figs. 4(b) and 4(c).

The estimated trajectory of the robot in the navigation frame is shown in Fig. 6(a). We are always able to obtain the position estimate in the global frame,  $\hat{\mathbf{p}}^g$ , via the current transform estimate  $\hat{\mathbf{t}}_N^n$  (according to (4)) as shown in Fig. 6(b). A comparison between the position estimate in the navigation frame and the global frame (with reference GPS measurements) is shown in Fig. 6(c). It is clear that when valid GPS measurements are available, the global position estimate,  $\hat{\mathbf{p}}^g$ , is consistent with these global measurements, showing the validity of the recursively estimated frame transform  $\hat{\mathbf{t}}_N^n$ . While the navigation frame drifts over the length of the vehicle trajectory, the role of the navigation frame is to maintain a locally consistent state estimate. The smoothness and continuity of the position and velocity estimate in the navigation frame is presented in Fig. 7.

To highlight the performance of the position estimate and associated estimator quality resulting from switching sensors, we provide the position estimate and corresponding  $3\sigma$  error bounds for a subset of the trajectory in Fig. 8. We can clearly

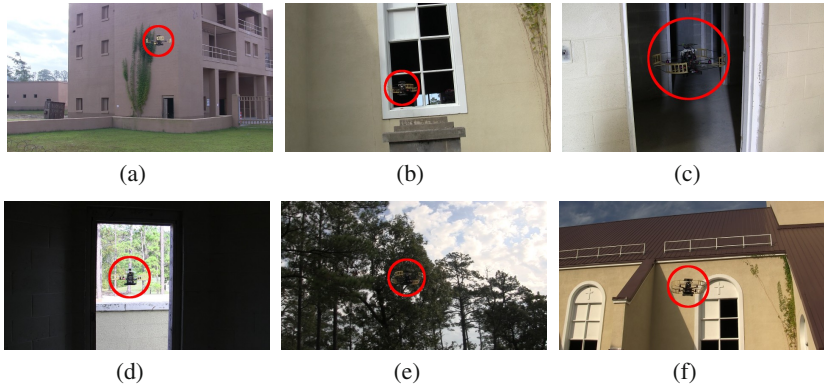


**Fig. 4** Aerial imagery of the experiment site with an overlay of the 3D map generated online during the experiment. Both large open spaces and confined indoor spaces are observed in the experimental setting. Perspective views of the map are visible in Figs. 4(b)-4(c). Videos of the experiments are available at <http://mrsl.grasp.upenn.edu/shaojie/ISER2012.m4v>.

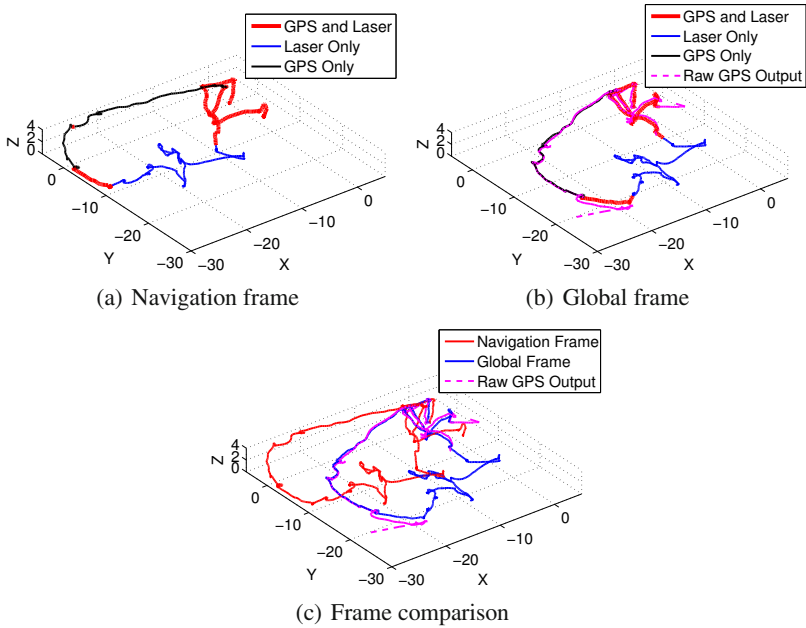
see changes in the estimation quality when adding or removing the laser-based localization from the system. However, note that even during rapid sensor addition and removal, the position estimate remains smooth and continuous.

### 3.3 Operation in Different Environments

To demonstrate that our approach is applicable in a variety of environments, we pursued two additional field tests at the Franklin Field at the University of Pennsylvania (Fig. 9(b)) and a training building at the Philadelphia Fire Academy (Fig. 9(c)). In both cases, the experiment considered at least one outdoor to indoor transition and in the latter case, also included operation in a multi-story building. Due to space constraints, we do not detail the full analysis of the experimental results here and only show an overlay of the resulting global trajectories on the corresponding satellite imagery. Although there is no ground truth information available, we can empirically

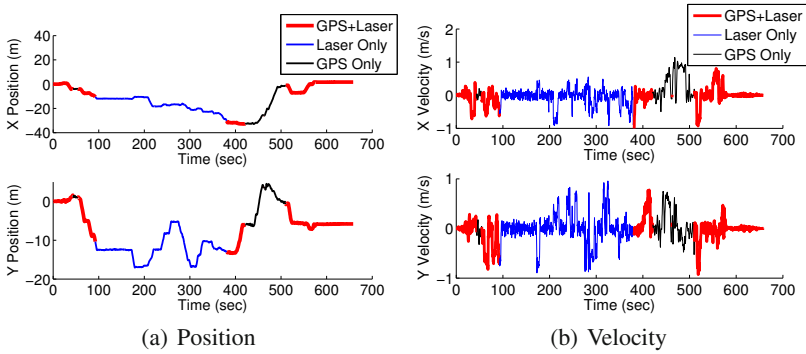


**Fig. 5** The MAV flies between indoor and outdoor environments during an experimental trial. We highlight the position of the robot with a red circle.

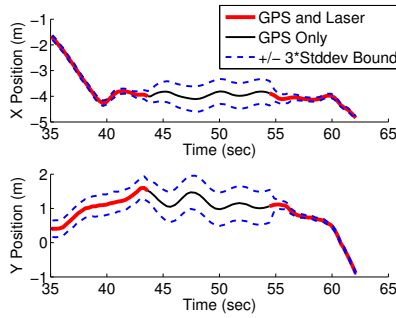


**Fig. 6** Trajectory of the robot in the navigation frame (Fig. 6(a)) and the global frame (Fig. 6(b)). Figure 6(c) shows a comparison between the state estimate in the navigation and global frame.

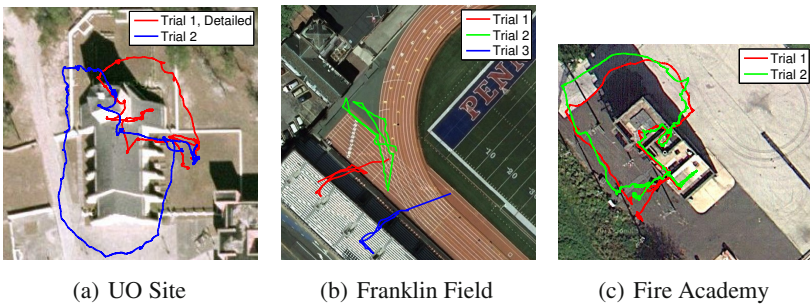




**Fig. 7** Position and velocity estimate in the navigation frame in X-Y directions. Note that there are no discontinuities or jumps in the position estimate despite adding and removing sensors.



**Fig. 8** A detail of the position estimate with associated error bounds corresponding to Fig. 7



**Fig. 9** Autonomous flight experiments in multiple environments. The red trajectory in Fig. 9(a) is detailed in Sect. 3.2.

verify that there is no conflict between the estimated trajectories of the robot and the satellite imagery. All analysis of the resulting experimental data yields similar commentary to the discussion in Sect. 3.2.

## 4 Conclusion and Future Work

In this work, we detail a methodology for estimating the state of a micro-aerial vehicle as it transitions between different operating environments with varying applicable sensors. We propose using a separate coordinate frame, called the navigation frame, to ensure smoothness in the state estimate and provide robust handling of sensor failures. We address the challenge of maintaining consistency between local and global measurements and propose a strategy to recursively estimate the transform between different coordinate frames. We present field experiments of a MAV navigating between mixed indoor and outdoor environments using the onboard state estimate for feedback control. We close by analyzing experimental results, showing local smoothness, global consistency, as well as repeatability of the proposed approach.

We are interested in moving forward by enhancing our current methods with vision-based state estimation methods so that our system can handle more general environments where laser and GPS information may be unavailable or unusable. We are also interested in adapting our current map alignment and loop closure methods to incorporate information from multiple robots.

## References

1. Shen, S., Michael, N., Kumar, V.: Autonomous multi-floor indoor navigation with a computationally constrained MAV. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Shanghai, China, pp. 20–25 (May 2011)
2. Julier, S.J., Uhlmann, J.K.: A new extension of the kalman filter to nonlinear systems. In: Kadar, I. (ed.) Proc. of SPIE, vol. 3068, pp. 182–193 (July 1997)
3. Merwe, R.V.D., Wan, E.A., Julier, S.I.: Sigma-point kalman filters for nonlinear estimation: Applications to integrated navigation. In: Proc. of AIAA Guidance, Navigation, and Controls Conf., Providence, RI (August 2004)
4. Grzonka, S., Grisetti, G., Burgard, W.: Towards a navigation system for autonomous indoor flying. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Kobe, Japan, pp. 2878–2883 (May 2009)
5. Bachrach, A.G.: Autonomous flight in unstructured and unknown indoor environments. Master's thesis, MIT, Cambridge, MA (September 2009)
6. Blösch, M., Weiss, S., Scaramuzza, D., Siegwart, R.: Vision based MAV navigation in unknown and unstructured environments. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Anchorage, AK, pp. 21–28 (May 2010)
7. Carlson, J.: Mapping large urban environments with GPS-aided SLAM. Ph.D. dissertation, CMU, Pittsburgh, PA (July 2010)
8. Schleicher, D., Bergasa, L.M., Ocaña, M., Barea, R., López, E.: Real-time hierarchical GPS aided visual SLAM on urban environments. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Kobe, Japan, pp. 4381–4386 (May 2009)

9. Moore, D.C., Huang, A.S., Walter, M., Olson, E.: Simultaneous local and global state estimation for robotic navigation. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Kobe, Japan, pp. 3794–3799 (May 2009)
10. Weinmann, A.: *Uncertain Models and Robust Control*. Springer, New York (1991)
11. Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., Teller, S.: An atlas framework for scalable mapping. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Taipei, Taiwan, vol. 2, pp. 1899–1906 (September 2003)
12. Estrada, C., Neira, J., Tardos, J.D.: Hierarchical SLAM: Real-time accurate mapping of large environments. *IEEE Trans. Robot.* 21(4), 588–596 (2005)
13. Dellaert, F., Kaess, M.: Square root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. Robot. Research* 25(12), 1181–1203 (2006)
14. Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: Incremental smoothing and mapping. *IEEE Trans. Robot.* 24(6), 1365–1378 (2008)
15. Censi, A.: On achievable accuracy for pose tracking. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Kobe, Japan, pp. 1–7 (May 2009)
16. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Rayleigh, NC, vol. 4, p. 850 (March 1987)
17. Dryanovski, I., Morris, W., Jizhong, X.: Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst., Taipei, Taiwan, pp. 1553–1559 (October 2010)
18. Olson, E.: Real-time correlative scan matching. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Kobe, Japan, pp. 4387–4393 (May 2009)
19. Ascending Technologies, GmbH (February 2012), <http://www.asctec.de/>
20. Robot Operating System (February 2012), <http://www.ros.org/>

# Influence of Aerodynamics and Proximity Effects in Quadrotor Flight

Caitlin Powers, Daniel Mellinger, Aleksandr Kushleyev,  
Bruce Kothmann, and Vijay Kumar

**Abstract.** The dynamic response and performance of a micro UAV is greatly influenced by its aerodynamics which in turn is affected by the interactions with features in the environment in close proximity. In the paper we address the modeling of quadrotor robots in different flight conditions that include relative wind velocity and proximity to the ground, the ceiling and other robots. We discuss the incorporation of these models into controllers and the use of a swarm of robots to map features in the environment from variations in the aerodynamics.

## 1 Introduction

With recent advances in aerial robotics it is now possible to design and build vehicles that can fly autonomously in three-dimensional, cluttered, indoor environments. However, flying close to such obstacles as pillars, through windows, close to the ground or near ceilings changes the dynamics of the vehicle because of changes in the aerodynamics. For autonomous flight, it is essential to understand and anticipate these changes and develop controllers that are responsive and adapt to these changes. Indeed, we want to be able to exploit the asymmetry introduced by features in the environment the same way birds are able to do during flight [8].

Nature also offers many benefits of flying in flocks in close proximity. Experimental evidence suggests benefits of higher efficiency [7], superior localization [6], better decision making [5], and survival [15]. Similar benefits can be potentially realized in groups of aerial robots. Once again it is essential to be able to model the effects of flight in close proximity and develop controllers that are informed by these models.

We are interested in autonomous micro helicopters, exemplified by coaxial rotor crafts [3], ducted fans [16], quadrotors [4] and hexarotors, and specifically in

---

Caitlin Powers · Daniel Mellinger · Aleksandr Kushleyev · Bruce Kothmann · Vijay Kumar  
GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104 USA  
e-mail: {cpow, dmel, akushley, kothmann, kumar}@seas.upenn.edu

small quadrotors such as the ones recently demonstrated in [11] that weigh less than 100 grams, are less than 0.25 meters, and are agile in three-dimensional flight. In quadrotors, the lift generated by the aircraft depends on the flow conditions near the propellers and the propeller speeds. These are coupled with the rigid body dynamics, the dynamics of the propeller and the motor dynamics. In a quadrotor, an onboard motor controller is used to control the motor current with feedback loops that incorporate information from onboard gyros and accelerometers and position relative to either features in the environment or with respect to a global coordinate system. However, the actual motor speed and the effective forces and moments acting on the airframe are a complex function of the aerodynamics. It is this function that must be properly understood and modeled in order to produce agile flight.

Previous investigation of quadrotor aerodynamics has shown the importance of considering aerodynamics in quadrotor control [9]. We take a different approach here that incorporates blade element theory. We choose this approach in order to frame our considerations in terms of rotor speeds.

In this paper, we describe the modeling of a single rotor and the robot (Sec. 2) as well as our testbed (Sec. 3). We investigate the effect of proximity to surfaces on thrust produced (Sec. 4) and measure thrust versus speed in different flow conditions with our experimental test rig and in-flight data (Sec. 5). We show that the lift produced by a propeller is affected by relative wind velocity and by proximity to the ground and ceiling and develop empirical models that lend themselves to control. Finally, we demonstrate applications to control and sensing (Sec. 7).

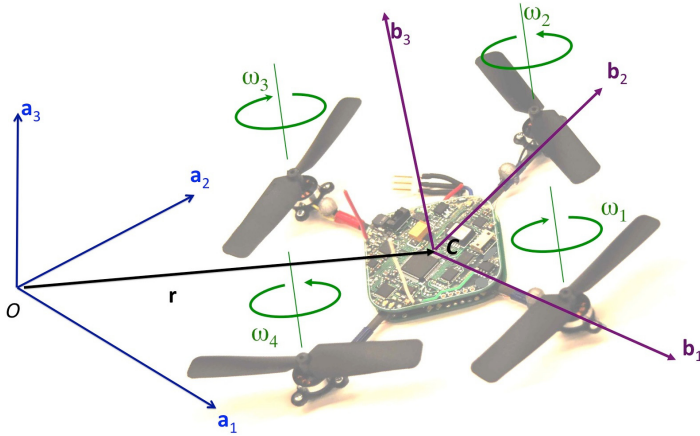
## 2 Dynamics and Control

The dynamic model and control for the micro quadrotor is based on the approach in [13]. As shown in Figure 1, we consider a body-fixed frame  $\mathcal{B}$  aligned with the principal axes of the quadrotor (unit vectors  $\mathbf{b}_i$ ) and an inertial frame  $\mathcal{A}$  with unit vectors  $\mathbf{a}_i$ .  $\mathcal{B}$  is described in  $\mathcal{A}$  by a position vector  $\mathbf{r}$  to the center of mass  $C$  and a rotation matrix  $R$ . In order to avoid singularities associated with parameterization, we use the full rotation matrix to describe orientations. The angular velocity of the quadrotor in the body frame, given by  $\hat{\omega}_B = R^T \dot{R}$ , where  $\hat{\cdot}$  denotes the skew-symmetric matrix form of the vector.

As shown in Fig. 1, the four rotors are numbered 1-4, with odd numbered rotors having a pitch that is opposite to the even numbered rotors. The angular speed of the rotor is  $\omega_i$ . The resulting thrust,  $T_i$ , and the reaction moment,  $M_i$ , are given by:

$$T_i = k_T \omega_i^2, \quad M_i = k_M \omega_i^2 \quad (1)$$

where the constants  $k_T$  and  $k_M$  are empirically determined for a propeller in still air. For our micro quadrotor, the motor dynamics have a time constant less than 10 msec and are much faster than the time scale of rigid body dynamics and aerodynamics. Thus we neglect the dynamics and assume  $T_i$  and  $M_i$  can be instantaneously changed. Therefore the control input to the system,  $\mathbf{u}$ , consists of the net thrust in the  $\mathbf{b}_3$  direction,  $u_1 = \sum_{i=1}^4 T_i$ , and the moments in  $\mathcal{B}$ ,  $[u_2, u_3, u_4]^T$ , given by:



**Fig. 1** The reference frames and propeller numbering convention

$$\mathbf{u} = \begin{bmatrix} k_T & k_T & k_T & k_T \\ 0 & k_T L & 0 & -k_T L \\ -k_T L & 0 & k_T L & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2)$$

where  $L$  is the distance from the axis of rotation of the propellers to the center of the quadrotor.

The Newton-Euler equations of motion are given by:

$$m\ddot{\mathbf{r}} = -mg\mathbf{a}_3 + u_1\mathbf{b}_3 \quad (3)$$

$$\dot{\boldsymbol{\omega}} = \mathcal{I}^{-1} \left[ -\boldsymbol{\omega} \times \mathcal{I} \boldsymbol{\omega} + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} \right] \quad (4)$$

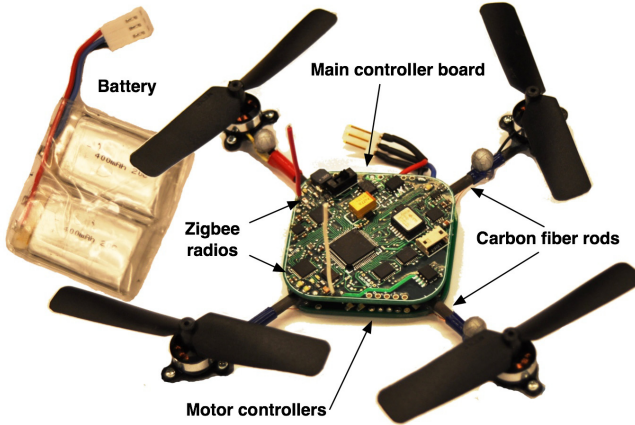
where  $\mathcal{I}$  is the moment of inertia matrix along  $\mathbf{b}_i$ . The main error in this model comes from the simplistic assumption underlying (1) that the lift force and drag moments are directly proportional to the square of the motor speed. In reality they are complex functions of the motor speed and environmental conditions.

Given a desired trajectory,  $\mathbf{r}_T$ , the controller derives the input  $u_1$  based on position and velocity errors:

$$u_1 = (-K_p \mathbf{e}_p - K_v \mathbf{e}_v + mg\mathbf{a}_3) \cdot \mathbf{b}_3 \quad (5)$$

where  $\mathbf{e}_p = \mathbf{r} - \mathbf{r}_T$  and  $\mathbf{e}_v = \dot{\mathbf{r}} - \dot{\mathbf{r}}_T$ . From the desired acceleration and a chosen yaw angle the total desired orientation can be found. As described in [13], the desired moments are expressed as a function of an orientation error,  $\mathbf{e}_R$ , and an angular velocity error,  $\mathbf{e}_\omega$ :

$$[u_2, u_3, u_4]^T = -K_R \mathbf{e}_R - K_\omega \mathbf{e}_\omega, \quad (6)$$



**Fig. 2** The kQuadNano [1]

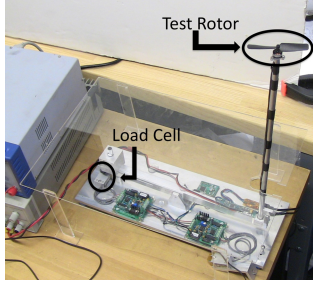
where  $K_R$  and  $K_\omega$  are diagonal gain matrices. Finally we compute the desired rotor speeds to achieve the desired  $\mathbf{u}$ .

### 3 Experiment Testbed

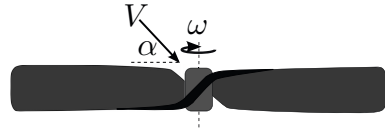
For the experiments presented in this work we use the kQuadNano developed by KMeI Robotics [1] shown in Figure 2. The vehicle uses four 8 cm diameter fixed-pitch propellers. The vehicle propeller-tip-to-propeller-tip distance is 21 cm and the total weight with a battery is about 76 grams.

We use a Vicon motion capture system [2] to sense the position of each vehicle at 100 Hz. This data is streamed over a gigabit ethernet network to a desktop base station. High-level control is done in MATLAB on the base station which sends commands to each quadrotor at 100 Hz. The base station sends, via custom radio modules, the desired commands, containing orientation, thrust, angular rates and attitude controller gains to the individual quadrotors. The onboard rate gyros and accelerometer are used to estimate the orientation and angular velocity of the craft. The main microprocessor runs an attitude controller and sends the desired propeller speeds to each of the four motor controllers at 600 Hz. A more detailed description of the experimental setup is presented in [12].

Experiments were also performed using a custom thrust test rig. The test rig includes a load cell from Transducer Technologies rated for 100 g for measuring thrust and electronic instrumentation for data collection. The rotor for testing is mounted on a rod and attached to a lever which transmits the thrust to the load cell. The test rig is pictured in Fig. 3a.



(a) Thrust Test Rig



(b) Diagram of Rotor

**Fig. 3** The test rig (left) and a schematic showing the relative wind speed and the angle of attack,  $\alpha$

### 4 Effect of Proximity to Horizontal Surfaces

It is known that rotorcraft operating near surfaces experience a “ground effect” where the rotors produce more thrust per unit power compared to flight at a large distance from the ground. One proposed mathematical description of ground effect [10], based on the method of images, is

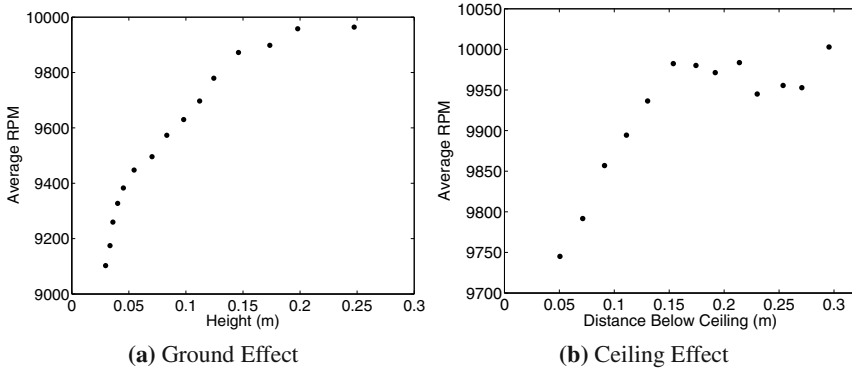
$$\frac{T}{T_\infty} = \frac{1}{1 - (\frac{R}{4z})^2} \tag{7}$$

Here  $R$  is the radius of the rotor,  $z$  is the vertical distance from the ground,  $T$  is the thrust produced by the propeller in ground effect, and  $T_\infty$  is the thrust produced at the same power outside of ground effect. Note that for  $\frac{z}{R} = 2$  the predicted ratio between  $T$  and  $T_\infty$  is just 1.016. Therefore, this formula predicts that ground effect is negligible when the rotor is more than one diameter off the ground,  $\frac{z}{R} > 2$  [10].

In order to test the ground effect in flight, the quadrotor was commanded to hover at a range of distances above the ground for 10 seconds and the average rotor speed required to maintain hover was recorded. This test provides a relationship between height and angular speed required to produce a constant thrust. The data for this test is shown in Fig. 4a. The general trend predicted by (7) is observed in this data since the propeller requires a lower rotor speed to maintain hover at lower heights. However, in this data the ground effect can be observed up to about 20 cm ( $\frac{z}{R} = 5$ ) which is significantly higher than (7) predicts.

Unlike larger helicopters, micro quadrotors can be used for indoor missions. This motivates the investigation of a “ceiling effect” which is present when the vehicle is close to an overhead plane. Performing a similar experiment, we found data for the ceiling effect as shown in Fig. 4b. The ceiling effect is not as large as the ground effect as observed by the smaller change in average rotor speed for the same separation. Note that while the ground effect pushes the vehicle away from the ground,





**Fig. 4** Effect of proximity to horizontal surfaces. Each data point is based on 10 seconds of hover data for a vehicle with propellers with  $R = 4$  cm. Note that noise could likely be reduced by averaging data for a longer time period.

the ceiling effect pulls the vehicle towards the ceiling which can cause a crash in the worst case.

### 5 Aerodynamics of Vertical and Forward Flight

The thrust produced by a rotor in flight is in general a function of the relative velocity between the rotor and the surrounding air,  $V$ , as well as the angle of attack  $\alpha$  as shown in Fig. 3b. For the situation where  $\alpha = \frac{\pi}{2}$ , the craft is said to be in climb. The velocity of the slipstream increases as it passes through the rotor. We refer to this additional velocity imparted by the rotor as the induced velocity,  $v$ . Momentum theory analysis relates the additional kinetic energy of the air at an infinite distance from the rotor to the thrust, and provides us with an expression for the thrust in climb or descent [10]:

$$T = \dot{m}w = 2\rho Av |V + v| \tag{8}$$

Here  $T$  is the rotor thrust,  $\rho$  is the density of air,  $A$  is the area swept by the rotor,  $\dot{m}$  is the mass flow through a disc of area  $A$  and  $w$  is the velocity of the air an infinite distance after it has passed through the rotor. In order to have a complete system of equations, we turn to blade element theory to find another expression for the thrust in climb from a rotor [10]:

$$T = \frac{\rho abc\omega^2 R^3}{4} \left( \theta_{tip} - \frac{V + v}{\omega R} \right) \tag{9}$$

Here  $\omega$  is the rotor speed,  $R$  is the radius of the rotor,  $\theta_{tip}$  is the pitch angle at the blade tip,  $a$  is the lift curve slope,  $b$  is the number of blades on the rotor, and  $c$  is the blade chord.  $a$ ,  $b$ ,  $c$ , and  $\theta_{tip}$  are functions of the rotor geometry alone. Note that

if  $V$  is 0, the induced velocity is equal to  $\sqrt{\frac{T}{2\rho A}}$ , and this expression becomes the familiar  $T = k\omega^2$  in (1).

Given (8) and (9), a climb velocity  $V$  and angular speed  $\omega$  we can solve a quadratic equation for  $v$  and then use either expression to find the thrust. In (9) we have two groups of terms describing the geometry of the rotor which depend on assumptions about the rotor construction, e.g., that it is ideally twisted and has a constant chord. By grouping the terms we can rewrite (9) as

$$T = k_1 \omega^2 + k_2(V + v)\omega \tag{10}$$

where  $k_1$  and  $k_2$  are determined by the rotor geometry and the density of air. One approach for finding these constants would be to measure the physical parameters of the propeller, however, this approach suffers from reliance on many assumptions about the blade geometry. Instead we choose to empirically determine these constants. We collect test rig and in-flight data to find the constants which best describe the experimental data. The in-flight test was performed by commanding a micro quadrotor to ascend or descend at a constant velocity. We then measured the steady state rotor speed to determine  $\omega$  required to produce a thrust of  $mg$  at the given vertical velocity. Several data points were collected for each velocity by adding small amounts of weight to the quadrotor to vary the required thrust. In addition, measurements from the thrust test rig provide data for a variety of rotor speed and thrust combinations. We use the thrust from all these trials to compute the induced velocity from (8). Next we note that the unknowns appear linearly in (10). We combine the data from all trials and use a linear least squares method to determine the constants which best fit the data. The experimental data and the fitted model is shown in Fig. 5. The maximum error between the fit and the experimental data is .64 g over all trials.

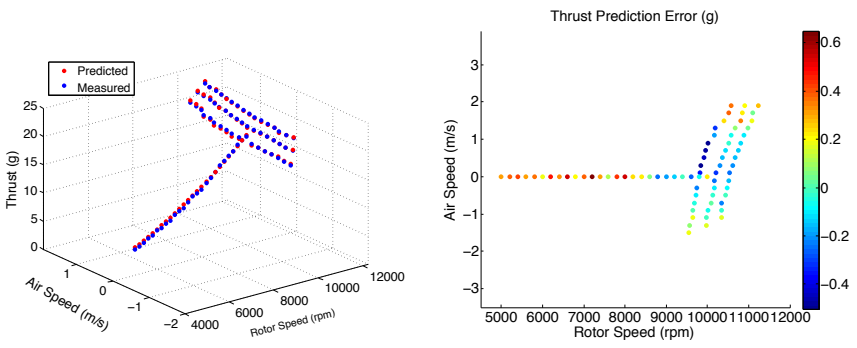


Fig. 5 Experimental Data and Fitted Model for Thrust as a Function of Air and Rotor Speed

Once we have constants  $k_1$  and  $k_2$ , we can predict the thrust at a given  $V$  and  $\omega$  using our model following a two step process:

1. First, solve for  $v$  by finding the roots of the quadratic equation formed from (8) and (9). Choose the positive root as the reasonable physical induced velocity.
2. Substitute  $v$  into equation (9) to find the predicted thrust.

We can implement this procedure in reverse to find the  $\omega$  which produces a given thrust at the current vertical velocity. The inclusion of this approach in the control law improves tracking performance as shown in Sec. 7.

In forward flight, we must add angle of attack,  $\alpha$ , to our model as shown in Fig. 3b. The equations for thrust become

$$T = 2\rho A v \sqrt{V^2 + 2Vv \sin \alpha + v^2} \quad (11)$$

and

$$T = \frac{\rho abc \omega^2 R^3}{2} \left( \frac{\theta}{3} + \frac{V^2 \cos^2 \alpha \theta}{2\omega^2 R^2} + \frac{V \sin \alpha + v}{2\omega R} \right) \quad (12)$$

where  $\theta$  is the pitch angle and is a function of rotor geometry alone as we are considering fixed pitch rotors [10]. The equation for  $v$  is now a fourth order polynomial instead of a quadratic. We characterized the dependence of thrust on rotor speed using the thrust test rig for five angles of attack and four wind speeds at each angle. The total thrust produced by the rotor decreases notably with increased wind speed in a given direction. As the angle of attack increases, the thrust variation due to wind speed decreases, as the component of wind velocity perpendicular to the rotor increases more slowly. At  $\alpha = 30^\circ$ , the effect of wind speed has decreased significantly, such that the largest observed difference between the no wind condition and the highest wind speed condition is 1 g.

While the data from the test rig is consistent with our model, it is not completely consistent with the data collected in free flight. We suspect this is primarily due to our inability to reproduce the flow conditions during free flight. The data from the test rig in the no wind condition is consistent with the in flight data and so was used in fitting the model. This is a subject of ongoing investigation as we continue to determine the best model for wind effects on small aerial vehicles.

## 6 Effect of Neighboring Vehicles

Micro quadrotors can be used in teams to cooperate to accomplish tasks that they individually cannot perform. If we are to reliably control formations of vehicles, we should be aware of the effects that neighboring quadrotors have on one another. It is known that vehicles have difficulty flying in the downwash of other vehicles [14]. Here we investigate the effect of nearby quadrotors in the same plane. We flew quadrotors along the same 3 meter long trajectory but in several scenarios as

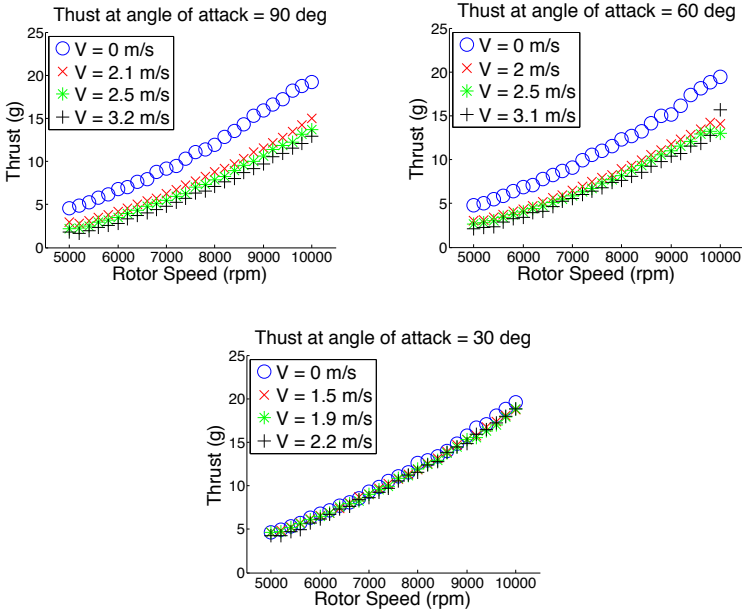
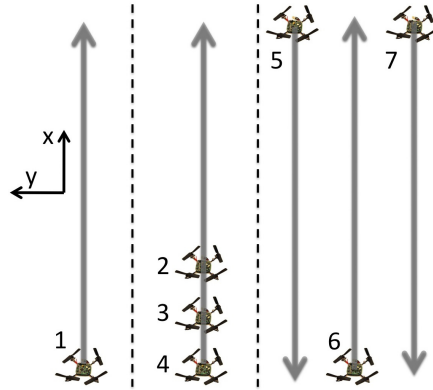


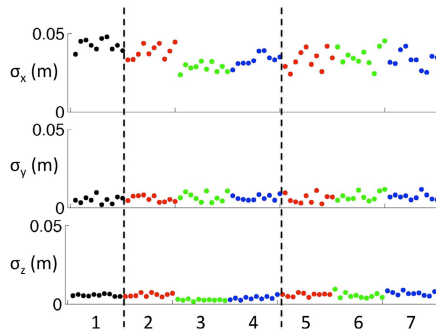
Fig. 6 The dependence of thrust on rotor speed and wind speed

illustrated in Fig. 7. For the first scenario (quadrotor 1) we evaluated the performance of a single quadrotor flying by itself without neighbors. For the second scenario (quadrotors 2, 3 and 4) we investigated the effect of quadrotors traveling in close proximity in the same direction. And for the final scenario (quadrotors 5, 6 and 7) we tested the effect of quadrotors traveling close together but in different directions. For each scenario the command to an individual quadrotor was the same. Each quadrotor was commanded to start at hover, accelerate at  $1 \text{ m/s}^2$  to a speed of  $1.5 \text{ m/s}$  and then decelerate to a stop at the same acceleration all at a constant height of  $1 \text{ m}$ . Note that for the scenarios with multiple vehicles the desired separation between vehicles was set to  $40 \text{ cm}$ .

Each scenario was run for 10 trials. The standard deviations in position error for each quadrotor for each trial, labeled 1-7, are shown in Fig. 8. As expected, the error in the direction of travel is the worst. There does not appear to be a significant change in any error from one configuration to another. One interesting observation is that the leading quadrotor (number 2) in the second scenario exhibits a slightly larger error in the direction of travel than the vehicles that follow it (numbers 3 and 4). We hypothesize that this is because quadrotors 3 and 4 are “drafting” quadrotor 2 and experience less aerodynamic drag which leads to slightly increased tracking performance.



**Fig. 7** Three scenarios for testing the effect of neighboring vehicles. Dotted lines separate the scenarios and arrows illustrate the commanded trajectories for the vehicles. Left: A single quadrotor. Middle: Three quadrotors moving in the same direction. Right: Three quadrotors moving in opposite directions.



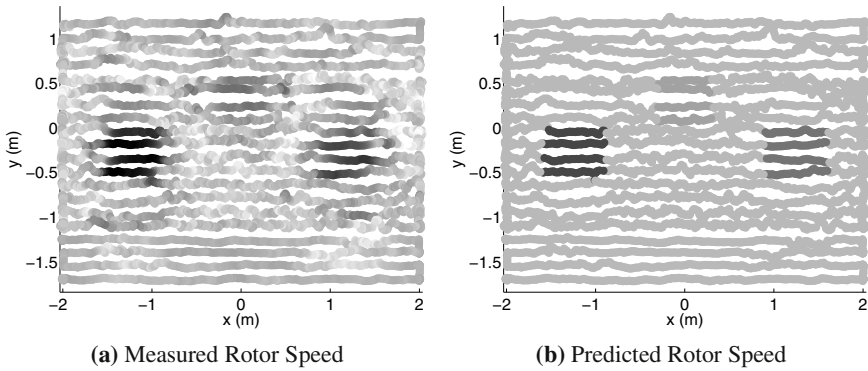
**Fig. 8** Standard Deviations in Position Errors for 10 trials for each of the 3 scenarios shown in Fig. 7. The number along the  $x$  axis indicates that the 10 data points above it correspond to the trials for the vehicle number shown in Fig. 7. The dotted lines separate the three scenarios.

## 7 Applications

### 7.1 Blind Terrain Mapping

Here we exploit the ground effect discussed in Sec. 4 to enable a team of quadrotors to build a height map of the terrain over which they fly. No sensors are required to build this map, only knowledge of the average rpm required to hover. At slow forward velocities, we can assume that the thrust produced is approximately thrust required to hover. We used a team of five micro quadrotors to sweep an area of varying height in the  $z$  direction at a lateral speed of 6 cm/s at a constant height

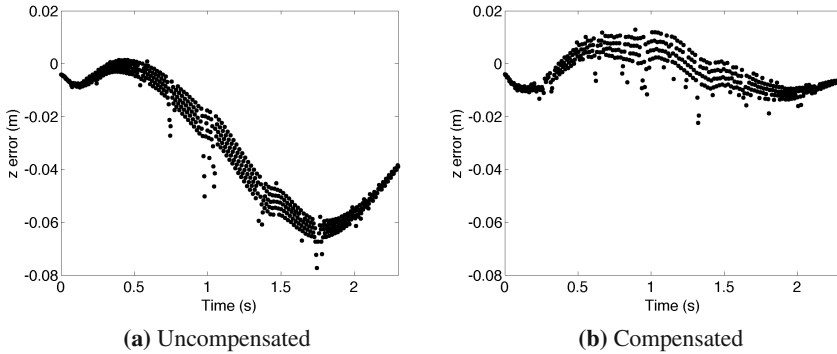
of 20 cm. Vehicles flew at a separation distance of 60 cm and made four equally spaced passes through the environment. Three different 60 cm × 60 cm blocks with heights of 15 cm, 10 cm, and 6 cm were placed in the environment. We recorded the low-pass filtered rotor speed required to hover along each trajectory. This data can be used with the ground effect data shown in Fig. 4a to create a map of the terrain height underneath the vehicles. The rotor speed required to hover collected during the experiment is compared to the required rotor speed predicted from our model and the actual terrain map is shown in Fig. 9. Here darker colors represent smaller rotor speeds (taller terrain). From the data we can clearly identify the location of the tallest two blocks in the environment and somewhat identify the presence of the shortest block. Note that this approach is not likely useful for developing high-resolution ground maps with a high degree of accuracy but it is useful for getting a rough estimate of the terrain map. In addition, it may be useful in detecting sudden changes in height during mapping for verification of other sensors.



**Fig. 9** Terrain Map - darker colors represent smaller rotor speeds and taller terrain

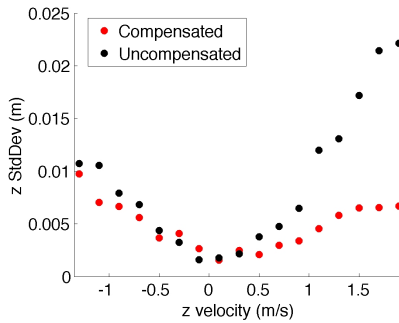
### 7.2 Control Compensation

Our standard quadrotor control law assumes the relationship between thrust and rotor speed is simply  $T = k_T \omega^2$ . However, as described in Sec. 5 this is only true when a propeller is in a stationary air field. Here we show that we can improve our controller performance by replacing the standard mapping with our model fit to the data for a quadrotor described in Sec. 5. As shown in Fig. 10a, when commanded to follow a trajectory along the  $z$  axis ascending at a velocity of 1.9 m/s, the standard uncompensated controller causes the quadrotor to lag behind the desired position up to 6 cm. This is because the propeller produces less thrust when ascending than it does when stationary for the same rotor speed. In Fig. 10b, we use a compensated controller that takes into account the vertical velocity of the vehicle in order to calculate the desired rotor speed from the desired thrust. This compensated controller reduces the maximum error to 2 cm for the same trajectory.



**Fig. 10**  $z$  error when following an ascending vertical trajectory at 1.9 m/s. The trajectory is 2.7 m long and the commanded acceleration when starting and stopping is  $2 \text{ m/s}^2$ .

As shown in Fig. 11, the controller compensation improved performance to a greater extent with increasing ascent velocity as expected. The compensation did not have an appreciable effect on performance in descent. This is likely because the increase in thrust during descent is not as great as the decrease in thrust during ascent for the same speed.



**Fig. 11** Compensated and Uncompensated Controller Performance. The deviation from the trajectory decreases with added compensation.

## 8 Conclusions

In this paper, we address the modeling of aerodynamic effects for a small quadrotor flying through three-dimensional environments in close proximity to environmental features and neighboring quadrotors. We develop models from first principles that explain the dependence of thrust generated by the propellers on different parameters and empirically determine the coefficients for this model. We present data collected

for a single rotor on a test rig as well as data collected from quadrotors in flight. We show how such models can be used to improve performance in flight and to infer the presence of ground features without additional sensors for blind mapping. A logical future step is to test the rotor performance in a larger variety of wind conditions, from strictly laminar flow to turbulent or time varying flow, in order to characterize the performance.

In the future, we plan to apply the results of this work to more complicated estimation and control problems, which will allow more robust operation of quadrotors in diverse environments. Further investigation of the influence of aerodynamics on the operation of small multi-rotor vehicles will allow engineers to make better decisions when designing vehicles and their controllers. In addition, experimentally verifiable models of quadrotor aerodynamics will help enable UAVs to detect and correct for the fluctuating environments that they encounter while in operation.

**Acknowledgements.** We gratefully acknowledge support from ONR Grants N00014-07-1-0829 and N00014-09-1-1031, and ARL Grant W911NF-08-2-0004.

## References

1. KMel Robotics, <http://www.kmelrobotics.com>
2. Vicon Motion Systems, Inc., <http://www.vicon.com>
3. Bermes, C.: Design and dynamic modeling of autonomous coaxial micro helicopters. Ph.D. dissertation, ETH Zurich, Zurich, Switzerland (March 2010)
4. Bouabdallah, S.: Design and Control of Quadrotors with Applications to Autonomous Flying. Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland (February 2007)
5. Couzin, I.D., Krause, J., Franks, N.R., Levin, S.A.: Effective leadership and decision-making in animal groups on the move. *Nature* 433(7025), 513–516 (2005)
6. Dall'Araccia, G., Dell'Omo, G., Wolfer, D.P., Lipp, H.-P.: Flock flying improves pigeons' homing: GPS track analysis of individual flyers versus small groups. *Anim. Beh.* 76, 1165–1172 (2008)
7. Hainsworth, F.R.: Precision and dynamics of positioning by canada geese flying in formation. *J. Exp. Biol.* 128, 445–462 (1987)
8. Hedrick, T.L., Usherwood, J.R., Biewener, A.A.: Wing inertia and whole-body acceleration: an analysis of instantaneous aerodynamic force production in cockatiels (*nymphicus hollandicus*) flying across a range of speeds. *J. Exp. Biol.* 207, 1689–1702 (2004)
9. Huang, H., Hoffmann, G.M., Waslander, S.L., Tomlin, C.J.: Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 3277–3282 (May 2009)
10. Johnson, W.: *Helicopter Theory*. Princeton University Press, Princeton (1980)
11. Kumar, V., Mellinger, D., Kushleyev, A.: *Autonomous, agile, aerial robots* (2012)
12. Kushleyev, A., Mellinger, D., Kumar, V.: Towards a swarm of agile micro quadrotors. In: *Robotics: Science and Systems, RSS* (2012)



13. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA (May 2011)
14. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The grasp multiple micro-uav testbed. *IEEE Robotics Automation Magazine* 17(3), 56–65 (2010)
15. Parrish, J., Hamner, W.: *Animal Groups in Three Dimensions*. Cambridge University Press, New York (1997)
16. Pines, D., Bohorquez, F.: Challenges facing future micro air vehicle development. *AIAA Journal of Aircraft* 43(2), 290–305 (2006)

# On the Consistency of Vision-Aided Inertial Navigation

Dimitrios G. Kottas, Joel A. Hesch, Sean L. Bowman, and Stergios I. Roumeliotis

**Abstract.** In this paper, we study estimator inconsistency in Vision-aided Inertial Navigation Systems (VINS). We show that standard (linearized) estimation approaches, such as the Extended Kalman Filter (EKF), can fundamentally alter the system observability properties, in terms of the number and structure of the unobservable directions. This in turn allows the influx of spurious information, leading to inconsistency. To address this issue, we propose an Observability-Constrained VINS (OC-VINS) methodology that explicitly adheres to the observability properties of the true system. We apply our approach to the Multi-State Constraint Kalman Filter (MSC-KF), and provide both simulation and experimental validation of the effectiveness of our method for improving estimator consistency.

## 1 Introduction

Many estimation problems in robotics, and in particular localization, involve non-linear process and measurement models. Existing estimators, such as the Extended Kalman Filter (EKF), often suffer from inconsistency when applied to such tasks. As defined in [2], a state estimator is consistent if the estimation errors are zero-mean and have covariance smaller than or equal to the covariance calculated by the filter. In other words, an inconsistent estimator is overconfident in the accuracy of its estimates and its errors grow over time, possibly even causing divergence.

Julier and Uhlmann [14] first reported EKF inconsistency for 2D Simultaneous Localization and Mapping (SLAM). Since then, others have developed methods that seek to mitigate inconsistency in 2D SLAM (e.g., [1, 5]). However, little was known about the causes of inconsistency until recently. Specifically, in [9, 11] we have shown that a main cause of inconsistency is the mismatch between the observability properties of the linearized system used by the EKF [or the Unscented Kalman

---

Dimitrios G. Kottas · Joel A. Hesch · Sean L. Bowman · Stergios I. Roumeliotis  
Dept. of Computer Science and Engineering, Univ. of Minnesota,  
Minneapolis, MN 55455, USA  
e-mail: {dkottas, joel, bowman, stergios}@cs.umn.edu

Filter (UKF)] and the underlying (true) nonlinear system. As a remedy, after identifying the unobservable directions either analytically [9] or numerically [10], we enforced them, either indirectly, by appropriately selecting the linearization points where the Jacobians are evaluated [11], or directly, by projecting the Jacobians onto the observable portion of the state space [10].

While most of the emphasis has been on 2D, very little is known about the inconsistency of 3D localization. This is primarily due to the complexity of the motion and measurement models involved in estimating a 15 (instead of 3) dimensional state. In this paper, we focus on Vision-aided Inertial Navigation Systems (VINS) that fuse data from a camera and an Inertial Measurement Unit (IMU) to track the six degrees-of-freedom (d.o.f.) pose of a sensing platform. Numerous VINS approaches have been presented in the literature, including methods based on the EKF [4, 22], UKF [6], and Batch-least Squares (BLS) [26]; however, these have not investigated the issue of VINS inconsistency.

The main contributions of this paper are the following:

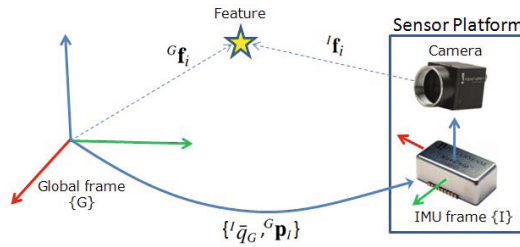
- We study the observability properties of VINS and analytically determine the four unobservable directions (i.e., rotation about the gravity vector and global translation).
- We identify and solve the conditions that the VINS propagation and measurement Jacobians need to satisfy in order to ensure that the observability properties of the estimator match those of the true linearized system.
- We validate the proposed approach and demonstrate its capability to improve consistency through Monte-Carlo simulations and real-world experiments.

Although the proposed methodology is general enough to be applicable to any linearized estimator (e.g., EKF, UKF), in regular or inverse filter form, and regardless of the number of robot poses considered (smoother vs. filter), due to space limitations, we hereafter focus on Visual-Inertial Odometry (VIO) using the Multi-State Constraint Kalman Filter (MSC-KF) [22].

The remainder of this paper is organized as follows: We first review the related work (Sect. 2), followed by a description of the VINS model and the observability properties of VINS (Sect. 3). In Sect. 4, we introduce our Observability Constrained (OC)-VINS methodology for mitigating inconsistency, which we apply to the MSC-KF (Sect. 5). We validate our approach with simulation trials (Sect. 6) and real-world experiments (Sect. 7). Lastly, we provide our concluding remarks and discuss our future research directions (Sect. 8).

## 2 Related Work

For the task of IMU-camera extrinsic calibration, Mirzaei and Roumeliotis [19], as well as Kelly and Sukhatme [15], have analyzed the system observability using Lie derivatives [7] to determine when the IMU-camera transformation is observable. Jones and Soatto [13] studied VINS observability by examining the indistinguishable trajectories of the system [12] under different sensor configurations (i.e., inertial only, vision only, vision and inertial). Martinelli [18] utilized the concept of



**Fig. 1** Sensor platform comprising an IMU and a camera.  $\{{}^I\bar{q}_G, {}^\sigma\mathbf{p}_I\}$  are the quaternion of orientation and position vector describing the pose of the sensing frame  $\{I\}$  with respect to the global frame  $\{G\}$ . The feature's 3D coordinates in  $\{G\}$  and  $\{I\}$  are denoted as  ${}^G\mathbf{f}_i$  and  ${}^I\mathbf{f}_i$ , respectively.

continuous symmetries to show that the IMU biases, 3D velocity, and absolute roll and pitch angles are observable for VINS.

VINS inconsistency was recently investigated by Li and Mourikis [16]. Specifically, they studied the link between the VINS observability properties and estimator inconsistency for the bias-free case, and leveraged the First-Estimates Jacobian (FEJ) methodology of [9] to mitigate inconsistency in VIO. In contrast to their work, our approach has the advantage that any linearization method can be employed (e.g., computing Jacobians analytically, numerically, or using sample points) by the estimator. Additionally, we show that our approach is flexible enough to be applied in a variety of VINS problems such as VIO or SLAM.

In this work, we study the observability properties of the ideal linearized VINS model (i.e., the one whose Jacobians are evaluated at the true states), and show it has four unobservable d.o.f., corresponding to three-d.o.f. global translations and one-d.o.f. global rotation about the gravity vector. Due to linearization errors, the number of unobservable directions is reduced in a standard EKF-based VINS approach, allowing the estimator to gain spurious information and leading to inconsistency. To address this problem, we introduce a modification of the EKF-based VINS where its estimated Jacobians are updated so as to ensure that the number of unobservable directions is the same as when using the true Jacobians. In this manner, the global rotation about the gravity vector remains unobservable (as it should) and the consistency of the VINS EKF is significantly improved.

### 3 VINS Model

In what follows, we present an overview of the propagation and measurement models that describe the general case of a VINS, along with the main results of our observability analysis. For simplicity, we consider the case of a single physical feature being observed over multiple time steps; however, the analysis is straightforward to extend to the multiple feature case.

### 3.1 State Vector and Propagation Model

The  $19 \times 1$  system state includes the IMU pose and linear velocity together with the time-varying IMU biases and the 3D coordinates of the feature (see Fig. 1), i.e.,

$$\mathbf{x} = [{}^I\bar{q}_G^T \ \mathbf{b}_g^T \ {}^G\mathbf{v}_I^T \ \mathbf{b}_a^T \ {}^G\mathbf{p}_I^T \ | \ {}^G\mathbf{f}^T]^T \quad (1)$$

where  ${}^I\bar{q}_G(t)$ ,  ${}^G\mathbf{p}_I(t)$ , and  ${}^G\mathbf{v}_I(t)$  are the orientation, position, and velocity of the IMU frame  $\{I\}$  with respect to the global frame  $\{G\}$ ,  $\mathbf{b}_g(t)$  and  $\mathbf{b}_a(t)$  are the gyroscope and accelerometer biases, and  ${}^G\mathbf{f}$  is the feature's position expressed in  $\{G\}$ .

The system model describing the time evolution of the state is (see [27]):

$${}^I\dot{\bar{q}}_G(t) = \frac{1}{2}\Omega({}^I\omega(t)){}^I\bar{q}_G(t), \quad {}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t), \quad {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t) \quad (2)$$

$$\dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad \dot{{}^G\mathbf{f}}(t) = 0, \quad (3)$$

where  ${}^I\omega$  and  ${}^G\mathbf{a}$  are the rotational velocity and linear acceleration, and

$$\Omega(\omega) \triangleq \begin{bmatrix} -[\omega \times] & \omega \\ \omega^T & 0 \end{bmatrix}, \quad [\omega \times] \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (4)$$

Here,  $[\omega \times]$  denotes the skew-symmetric matrix parameterized by  $\omega$ . The time-varying biases are modeled as random-walk processes driven by white zero-mean Gaussian noise  $\mathbf{n}_{wg}(t)$  and  $\mathbf{n}_{wa}(t)$ , respectively, with autocorrelations  $\mathbb{E}[\mathbf{n}_{wg}(t)\mathbf{n}_{wg}^T(\tau)] = \mathbf{Q}_{wg}\delta(t-\tau)$  and  $\mathbb{E}[\mathbf{n}_{wa}(t)\mathbf{n}_{wa}^T(\tau)] = \mathbf{Q}_{wa}\delta(t-\tau)$ . The gyroscope and accelerometer measurements,  $\omega_m$  and  $\mathbf{a}_m$ , are:

$$\omega_m(t) = {}^I\omega(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \quad (5)$$

$$\mathbf{a}_m(t) = \mathbf{C}({}^I\bar{q}_G(t))({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t). \quad (6)$$

The noise terms,  $\mathbf{n}_g$  and  $\mathbf{n}_a$ , are modeled as zero-mean white Gaussian random processes. The gravitational acceleration  ${}^G\mathbf{g}$  is known with respect to the global frame  $\{G\}$ . The matrix  $\mathbf{C}(\bar{q})$  is the rotation matrix corresponding to the quaternion  $\bar{q}$ . Lastly, the time derivative of  ${}^G\mathbf{f}$  is zero, since the camera observes a static scene.

Linearizing at the current estimates and applying the expectation operator on both sides of (2)-(3), we obtain the state estimate propagation model

$${}^I\dot{\hat{q}}_G(t) = \frac{1}{2}\Omega({}^I\hat{\omega}(t)){}^I\hat{q}_G(t), \quad {}^G\dot{\hat{\mathbf{p}}}_I(t) = {}^G\hat{\mathbf{v}}_I(t), \quad {}^G\dot{\hat{\mathbf{v}}}_I(t) = \mathbf{C}^T({}^I\hat{q}_G(t))\hat{\mathbf{a}}(t) + {}^G\mathbf{g} \quad (7)$$

$$\dot{\hat{\mathbf{b}}}_g(t) = \mathbf{0}_{3 \times 1}, \quad \dot{\hat{\mathbf{b}}}_a(t) = \mathbf{0}_{3 \times 1}, \quad \dot{\hat{{}^G\mathbf{f}}}(t) = 0 \quad (8)$$

where  $\hat{\mathbf{a}}(t) = \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t)$ , and  ${}^I\hat{\omega}(t) = \omega_m(t) - \hat{\mathbf{b}}_g(t)$ .

The  $18 \times 1$  error-state vector is defined as

$$\tilde{\mathbf{x}} = [{}^I\delta\theta_G^T \ \tilde{\mathbf{b}}_g^T \ {}^G\tilde{\mathbf{v}}_I^T \ \tilde{\mathbf{b}}_a^T \ {}^G\tilde{\mathbf{p}}_I^T \ | \ \tilde{{}^G\mathbf{f}}^T]^T. \quad (9)$$

For the IMU position, velocity, biases, and the observed feature, an additive error model is utilized (i.e.,  $\tilde{\mathbf{y}} = \mathbf{y} - \hat{\mathbf{y}}$  is the error in the estimate  $\hat{\mathbf{y}}$  of a quantity  $\mathbf{y}$ ), while for the quaternion we employ a multiplicative error model [27]. The main advantage of such an error model is that it allows us to represent the attitude uncertainty by a  $3 \times 3$  covariance matrix, which is a minimal representation.

The linearized continuous-time error-state equation is

$$\dot{\tilde{\mathbf{x}}}(t) = \begin{bmatrix} \mathbf{F}_s(t) & \mathbf{0}_{15 \times 3} \\ \mathbf{0}_{3 \times 15} & \mathbf{0}_3 \end{bmatrix} \tilde{\mathbf{x}}(t) + \begin{bmatrix} \mathbf{G}_s(t) \\ \mathbf{0}_{3 \times 12} \end{bmatrix} \mathbf{n} = \mathbf{F}_c(t) \tilde{\mathbf{x}}(t) + \mathbf{G}_c(t) \mathbf{n} \quad (10)$$

where  $\mathbf{n} = [\mathbf{n}_g^T \ \mathbf{n}_{wg}^T \ \mathbf{n}_a^T \ \mathbf{n}_{wa}^T]^T$ ,  $\mathbf{F}_c$  is the error-state transition matrix, and  $\mathbf{G}_c$  is the input noise matrix, with

$$\mathbf{F}_s = \begin{bmatrix} -[\hat{\omega}(t) \times] & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\mathbf{C}^T({}^l\hat{q}_G(t))[\hat{\mathbf{a}}(t) \times] & \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^l\hat{q}_G(t)) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix}, \quad \mathbf{G}_s = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{C}^T({}^l\hat{q}_G(t)) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}.$$

The discrete-time state transition matrix from time  $t_1$  to  $t$ ,  $\Phi(t, t_1)$ , is computed in analytical form [8] as the solution to the matrix differential equation  $\dot{\Phi}(t, t_1) = \mathbf{F}_c(t) \Phi(t, t_1)$ , with initial condition  $\Phi(t_1, t_1) = \mathbf{I}_{18}$ . As we show in [8], the structure of  $\Phi(t_{k+1}, t_k) = \Phi(t_{k+1}, t_1) \Phi(t_k, t_1)^{-1}$  when the state contains the IMU pose, velocity, biases, and a single landmark is given by

$$\Phi_{k+1} = \Phi(t_{k+1}, t_k) = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{31} & \Phi_{32} & \mathbf{I}_3 & \Phi_{34} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \Phi_{51} & \Phi_{52} & \delta t \mathbf{I}_3 & \Phi_{54} & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}. \quad (11)$$

In the ensuing analysis, we consider  $k \geq 1$  and define  $\Phi_1 := \Phi(t_1, t_1) = \mathbf{I}_{18}$ .

### 3.2 Measurement Model

As the sensor platform moves in the environment, the camera observes point features, which are tracked across images. Generally, in a VINS [21], these measurements are exploited to concurrently estimate the motion of the sensing platform and, optionally, the structure of the environment.

We employ the pinhole camera model to describe the perspective projection of the 3D point  $\mathbf{f}$  on the image plane and model the measurement  $\mathbf{z}_k$  at time step  $t_k$ , i.e.,

$$\mathbf{z}_k = 1/z [x \ y]^T + \eta_k, \quad [x \ y \ z]^T = {}^l\mathbf{f} = \mathbf{C}({}^l\bar{q}_G) ({}^G\mathbf{f} - {}^G\mathbf{p}_I), \quad (12)$$

where  $\mathbf{C}({}^l\bar{q}_G)$  is the rotation matrix from  $\{G\}$  to  $\{I\}$  and  $\eta_k$  follows a Gaussian distribution with  $\mathbb{E}[\eta_k] = \mathbf{0}_{2 \times 1}$  and  $\mathbb{E}[\eta_k \eta_k^T] = \sigma_\eta^2 \mathbf{I}_2$ . Note also that, without loss of generality, we express the image measurement in normalized pixel coordinates, and consider the camera frame to be coincident with the IMU<sup>1</sup>. By differentiating the nonlinear measurement model (12), we obtain the measurement Jacobian:

$$\begin{aligned} \mathbf{H}_k &= \mathbf{H}_{cam} [\mathbf{H}_{\theta_G} \mathbf{0}_{3 \times 9} \mathbf{H}_{\mathbf{p}_I} \mid \mathbf{H}_{\mathbf{f}}] \\ \mathbf{H}_{cam} &= \frac{1}{z} \begin{bmatrix} 1 & 0 & \frac{-x}{z} \\ 0 & 1 & \frac{-y}{z} \end{bmatrix}, \quad \mathbf{H}_{\theta_G} = [{}^l\mathbf{f} \times], \quad \mathbf{H}_{\mathbf{p}_I} = -\mathbf{C}({}^l\bar{q}_G), \quad \mathbf{H}_{\mathbf{f}} = \mathbf{C}({}^l\bar{q}_G). \end{aligned} \quad (13)$$

### 3.3 System Observability Analysis

In order to compute the analytical expressions of the four unobservable directions, we form the observability matrix  $\mathbf{M}(\mathbf{x}^*) = \mathbf{M}$ , as a function of the linearization point  $\mathbf{x}^*$ , for a system observing a feature over time steps  $t_1 \dots t_N$ , i.e.,

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_1 \Phi_1 \\ \mathbf{H}_2 \Phi_2 \Phi_1 \\ \vdots \\ \mathbf{H}_N \Phi_N \cdots \Phi_1 \end{bmatrix}. \quad (14)$$

**Lemma:** In VINS, when using the true state to evaluate  $\mathbf{H}$  and  $\Phi$ , each block row of the observability matrix has the following form

$$\mathbf{M}_k = \mathbf{H}_{cam,k} \mathbf{C}({}^l\bar{q}_{G,k}) \left[ [{}^G\mathbf{f} - {}^G\mathbf{p}_{I,1} - {}^G\mathbf{v}_{I,1} \delta t_{k-1} + \frac{1}{2} {}^G\mathbf{g} \delta t_{k-1}^2 \times] \mathbf{C}({}^l\bar{q}_{G,1})^T \mathbf{D}_k - \mathbf{I} \delta t_{k-1} \mathbf{E}_k - \mathbf{I}_3 \mathbf{I}_3 \right],$$

where  $\delta t_{k-1} = (k-1)\delta t$ , and  $\mathbf{D}_k$  and  $\mathbf{E}_k$  are time-varying matrices.

**Proof:** See [8].

**Theorem:** The right nullspace of the observability matrix of a VINS spans four directions, i.e.,

$$\mathbf{M}\mathbf{N}_1 = \mathbf{0}, \quad \mathbf{N}_1 = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}({}^l\bar{q}_{G,1}) {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_3 & -[{}^G\mathbf{v}_{I,1} \times] {}^G\mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{I}_3 & -[{}^G\mathbf{p}_{I,1} \times] {}^G\mathbf{g} \\ \mathbf{I}_3 & -[{}^G\mathbf{f} \times] {}^G\mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{R,1} \\ \mathbf{N}_{f,1} \end{bmatrix} \quad (15)$$

where the  $15 \times 4$  matrix  $\mathbf{N}_{R,1}$  comprises the nullspace elements corresponding to the sensor platform (robot) state, and the  $3 \times 4$  matrix  $\mathbf{N}_{f,1}$  corresponds to the feature. We note that, the first three columns of the matrix  $\mathbf{N}_1$  correspond to global translations, while its fourth column corresponds to global rotations about  ${}^G\mathbf{g}$ .

<sup>1</sup> We perform both intrinsic and extrinsic camera/IMU calibration off-line [3, 19].

**Proof:** See [8].

In the preceding analysis, the system is linearized at the true state, hence, the aforementioned unobservable directions correspond to the true linearized system model. In practice, we do not have access to the true state and hence we typically linearize at the current state estimate. However, this causes the observability matrix of the estimated system  $\widehat{\mathbf{M}} = \mathbf{M}(\widehat{\mathbf{x}})$  to have higher rank (a fact that can be easily verified by numerically evaluating  $\widehat{\mathbf{M}}$  during any experiment).

## 4 Observability-Constrained VINS (OC-VINS)

Ideally, we would like to design a filter that adheres to the true unobservable directions of the system. However, this would require knowledge of the *true* sensor pose and landmark position, which is clearly unrealizable in practice. Alternatively, we require that the estimator adheres to the *number* and *structure* of nullspace directions by ensuring that  $\widehat{\mathbf{M}}\widehat{\mathbf{N}}_1 = \mathbf{0}$  is satisfied for every block row of  $\widehat{\mathbf{M}}$ , i.e.,

$$\widehat{\mathbf{H}}_k \widehat{\Phi}_k \dots \widehat{\Phi}_1 \widehat{\mathbf{N}}_1 = \mathbf{0}, \quad k \geq 1. \quad (16)$$

We do so by appropriately modifying  $\widehat{\Phi}_k$  and  $\widehat{\mathbf{H}}_k$  at each time step so that<sup>2</sup>

$$\widehat{\mathbf{N}}_{k+1} = \widehat{\Phi}_{k+1} \widehat{\mathbf{N}}_k, \quad \widehat{\mathbf{H}}_k \widehat{\mathbf{N}}_k = \mathbf{0}, \quad k \geq 1 \quad (17)$$

where  $\widehat{\mathbf{N}}_k$ ,  $k \geq 1$  is computed analytically based on (15). We hereafter present our method for initializing the nullspace (Sect. 4.1), and employing the nullspace to preserve the system observability properties during the propagation (Sect. 4.2) and update (Sect. 4.3) steps of the filter.

### 4.1 Nullspace Definition at Time Step $k$

At each time step, we compute the nullspace  $\widehat{\mathbf{N}}_k$  as a function of the state estimate, and use it to enforce the unobservable directions. For the robot state, the initial nullspace as well as the nullspace at all subsequent times are [8]

$$\widehat{\mathbf{N}}_{R,1} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C} \left( {}^I \widehat{q}_{G,1|1} \right)^G \mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_3 & -[{}^G \widehat{\mathbf{v}}_{I,1|1}] \times ]^G \mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{I}_3 & -[{}^G \widehat{\mathbf{p}}_{I,1|1}] \times ]^G \mathbf{g} \end{bmatrix}, \quad \widehat{\mathbf{N}}_{R,k} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C} \left( {}^I \widehat{q}_{G,k|k-1} \right)^G \mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_3 & -[{}^G \widehat{\mathbf{v}}_{I,k|k-1}] \times ]^G \mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{I}_3 & -[{}^G \widehat{\mathbf{p}}_{I,k|k-1}] \times ]^G \mathbf{g} \end{bmatrix}, \quad (18)$$

<sup>2</sup> Although this could also be accomplished by appropriate selection of the linearization points (as in [9]), we instead choose to employ information projections of the Jacobians  $\widehat{\mathbf{H}}_k$  and  $\widehat{\Phi}_k$  since this allows more freedom in the way that we compute these matrices (e.g., analytically, using sample points as in the UKF, or through numerical integration for  $\widehat{\Phi}_k$ ).



where the notation  $\hat{\mathbf{x}}_{i|j}$  denotes the estimate of quantity  $\mathbf{x}$  at time-step  $i$  computed using measurements up to time-step  $j$ . For each feature, the corresponding nullspace block element is

$$\hat{\mathbf{N}}_{f,k} = [\mathbf{I}_3 - \text{[}^G\hat{\mathbf{f}}_{\ell|k} \times \text{]}^G \mathbf{g}], \quad (19)$$

where  $^G\hat{\mathbf{f}}_{\ell|k}$  is the first estimate of the feature's position, initialized at time step  $\ell$ .

## 4.2 OC Propagation: Modification of the State Transition Matrix $\Phi$

During each propagation step, we must ensure that  $\hat{\mathbf{N}}_{k+1} = \hat{\Phi}_{k+1} \hat{\mathbf{N}}_k$ . We note that the first block-column of this constraint is automatically satisfied by the structure of  $\hat{\Phi}_{k+1}$  [see (20)], so we focus on the fourth column of  $\hat{\mathbf{N}}_{k+1} = \hat{\Phi}_{k+1} \hat{\mathbf{N}}_k$ , which we write element-wise as:

$$\begin{bmatrix} \mathbf{C} (^t\hat{q}_{G,k+1|k})^G \mathbf{g} \\ \mathbf{0}_{3 \times 1} \\ -\text{[}^G\hat{\mathbf{v}}_{I,k+1|k} \times \text{]}^G \mathbf{g} \\ \mathbf{0}_{3 \times 1} \\ -\text{[}^G\hat{\mathbf{p}}_{I,k+1|k} \times \text{]}^G \mathbf{g} \\ -\text{[}^G\hat{\mathbf{f}}_{\ell|k} \times \text{]}^G \mathbf{g} \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_{11} & \hat{\Phi}_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \hat{\Phi}_{31} & \hat{\Phi}_{32} & \mathbf{I}_3 & \hat{\Phi}_{34} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \hat{\Phi}_{51} & \hat{\Phi}_{52} & \delta t \mathbf{I}_3 & \hat{\Phi}_{54} & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{C} (^t\hat{q}_{G,k|k-1})^G \mathbf{g} \\ \mathbf{0}_{3 \times 1} \\ -\text{[}^G\hat{\mathbf{v}}_{I,k|k-1} \times \text{]}^G \mathbf{g} \\ \mathbf{0}_{3 \times 1} \\ -\text{[}^G\hat{\mathbf{p}}_{I,k|k-1} \times \text{]}^G \mathbf{g} \\ -\text{[}^G\hat{\mathbf{f}}_{\ell|k} \times \text{]}^G \mathbf{g} \end{bmatrix}. \quad (20)$$

This results in constraints on the block-elements  $\hat{\Phi}_{11}$ ,  $\hat{\Phi}_{31}$ , and  $\hat{\Phi}_{51}$  (see [8]). Specifically, for  $\hat{\Phi}_{11}$  we require that:

$$\mathbf{C} (^t\hat{q}_{G,k+1|k})^G \mathbf{g} = \hat{\Phi}_{11} \mathbf{C} (^t\hat{q}_{G,k|k-1})^G \mathbf{g} \Rightarrow \hat{\Phi}_{11} = \mathbf{C} (^{t,k+1|k}\hat{q}_{I,k|k-1}). \quad (21)$$

The constraints for  $\hat{\Phi}_{31}$  and  $\hat{\Phi}_{51}$  are

$$\hat{\Phi}_{31} \mathbf{C} (^t\hat{q}_{G,k|k-1})^G \mathbf{g} = \text{[}^G\hat{\mathbf{v}}_{I,k|k-1} - ^G\hat{\mathbf{v}}_{I,k+1|k} \times \text{]}^G \mathbf{g} \quad (22)$$

$$\hat{\Phi}_{51} \mathbf{C} (^t\hat{q}_{G,k|k-1})^G \mathbf{g} = \text{[}\delta t ^G\hat{\mathbf{v}}_{I,k|k-1} + ^G\hat{\mathbf{p}}_{I,k|k-1} - ^G\hat{\mathbf{p}}_{I,k+1|k} \times \text{]}^G \mathbf{g} \quad (23)$$

both of which are in the form  $\mathbf{A}\mathbf{u} = \mathbf{w}$ , where  $\mathbf{u}$  and  $\mathbf{w}$  are nullspace elements that are known. We seek to find a perturbed  $\mathbf{A}^*$ , for  $\mathbf{A} = \hat{\Phi}_{31}$  and  $\mathbf{A} = \hat{\Phi}_{51}$  that fulfills the constraints (22) and (23). We formulate this as a minimization problem:

$$\min_{\mathbf{A}^*} \|\mathbf{A}^* - \mathbf{A}\|_{\mathcal{F}}^2, \quad \text{subject to } \mathbf{A}^* \mathbf{u} = \mathbf{w} \quad (24)$$

where  $\|\cdot\|_{\mathcal{F}}$  denotes the Frobenius matrix norm. Applying the method of Lagrange multipliers, we solve (24) in closed form as  $\mathbf{A}^* = \mathbf{A} - (\mathbf{A}\mathbf{u} - \mathbf{w})(\mathbf{u}^T \mathbf{u})^{-1} \mathbf{u}^T$ .

We compute  $\hat{\Phi}_{11}$  from (21), and  $\hat{\Phi}_{31}$  and  $\hat{\Phi}_{51}$  from (24) and construct the observability constrained discrete-time propagation Jacobian matrix.

### 4.3 OC Update: Modification of the Measurement Matrix $\mathbf{H}$

During each update step, the measurement Jacobian must satisfy  $\widehat{\mathbf{H}}_k \widehat{\mathbf{N}}_k = \mathbf{0}$ , i.e.,

$$\widehat{\mathbf{H}}_{cam} \left[ \widehat{\mathbf{H}}_{\theta_G} \mathbf{0}_{3 \times 9} \widehat{\mathbf{H}}_{\mathbf{p}_I} \mid \widehat{\mathbf{H}}_{\mathbf{f}} \right] \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}({}^I \hat{q}_{G,k|k-1}) {}^G \mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_3 & -[{}^G \hat{\mathbf{v}}_{I,k|k-1} \times] {}^G \mathbf{g} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{I}_3 & -[{}^G \hat{\mathbf{p}}_{I,k|k-1} \times] {}^G \mathbf{g} \\ \mathbf{I}_3 & -[{}^G \hat{\mathbf{f}}_{\ell} \times] {}^G \mathbf{g} \end{bmatrix} = \mathbf{0}. \quad (25)$$

The first block column of  $\widehat{\mathbf{N}}_k$  dictates that  $\widehat{\mathbf{H}}_{\mathbf{f}} = -\widehat{\mathbf{H}}_{\mathbf{p}_I}$ . We substitute this equality, and rewrite the fourth column of (25) in a compact form as

$$\widehat{\mathbf{H}}_{cam} \left[ \widehat{\mathbf{H}}_{\theta_G} \widehat{\mathbf{H}}_{\mathbf{p}_I} \right] \begin{bmatrix} \mathbf{C}({}^I \hat{q}_{G,k|k-1}) {}^G \mathbf{g} \\ [{}^G \hat{\mathbf{f}}_{\ell} \times] {}^G \mathbf{g} - [{}^G \hat{\mathbf{p}}_{I,k|k-1} \times] {}^G \mathbf{g} \end{bmatrix} = \mathbf{0}. \quad (26)$$

This is a constraint of the form  $\mathbf{A}\mathbf{u} = \mathbf{0}$ , where  $\mathbf{u}$  is a fixed quantity determined by elements in the nullspace, and  $\mathbf{A}$  comprises elements of the measurement Jacobian. We compute the optimal perturbed matrix  $\mathbf{A}^*$  that fulfills (26), by solving a problem of the same form as (24) to obtain the modified elements of the measurement Jacobian. Specifically, after computing  $\mathbf{A}^* = \mathbf{A} - \mathbf{A}\mathbf{u}(\mathbf{u}^T \mathbf{u})^{-1} \mathbf{u}^T$ , we recover the measurement Jacobian elements as

$$\widehat{\mathbf{H}}_{cam} \widehat{\mathbf{H}}_{\theta_G} = \mathbf{A}_{1:2,1:3}^* \quad , \quad \widehat{\mathbf{H}}_{cam} \widehat{\mathbf{H}}_{\mathbf{p}_I} = \mathbf{A}_{1:2,4:6}^* \quad , \quad \widehat{\mathbf{H}}_{cam} \widehat{\mathbf{H}}_{\mathbf{f}} = -\mathbf{A}_{1:2,4:6}^* \quad (27)$$

where the subscripts (i:j, m:n) denote the submatrix spanning rows i to j, and columns m to n.

## 5 Application: Observability-Constrained MSC-KF (OC-MSK-KF)

The MSC-KF [20] is a VINS that performs tightly-coupled visual-inertial odometry over a sliding window of  $m$  poses, while maintaining linear complexity in the number of observed features. The key advantage of the MSC-KF is that it utilizes all the constraints for each feature observed by the camera over  $m$  poses, without requiring to build a map or estimate the features as part of the state vector. We hereafter describe how to apply our OC-VINS methodology to the MSC-KF.

Each time the camera records an image, the MSC-KF creates a stochastic clone [24] of the sensor pose. This enables the MSC-KF to utilize delayed image measurements; in particular, it allows all of the observations of a given feature  $\mathbf{f}_i$  to be processed during a single update step (when the first pose that observed the feature is about to be marginalized). Whenever the current pose is cloned, we also clone the corresponding nullspace elements to obtain an augmented nullspace, i.e.,

$$\widehat{\mathbf{N}}_k^{aug} = \begin{bmatrix} \widehat{\mathbf{N}}_k \\ \widehat{\mathbf{N}}_{k,clone} \end{bmatrix}, \quad \text{where } \widehat{\mathbf{N}}_{k,clone} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}({}^t\hat{q}_{G,k|k-1})^G \mathbf{g} \\ \mathbf{I}_3 & -[{}^G\hat{\mathbf{p}}_{t,k|k-1} \times]^G \mathbf{g} \end{bmatrix}. \quad (28)$$

During propagation, the current state estimate evolves forward in time by integrating (7)-(8), while the current clone poses are static. We employ (21)-(23) to compute the observability-constrained discrete-time state transition matrix  $\widehat{\Phi}_k$ , and propagate the covariance as

$$\mathbf{P}_{k+1|k}^{aug} = \begin{bmatrix} \widehat{\Phi}_k & \mathbf{0}_{15 \times 6m} \\ \mathbf{0}_{6m \times 15} & \mathbf{I}_{6m} \end{bmatrix} \mathbf{P}_{k|k}^{aug} \begin{bmatrix} \widehat{\Phi}_k^T & \mathbf{0}_{15 \times 6m} \\ \mathbf{0}_{6m \times 15} & \mathbf{I}_{6m} \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_k & \mathbf{0}_{15 \times 6m} \\ \mathbf{0}_{6m \times 15} & \mathbf{0}_{6m} \end{bmatrix} \quad (29)$$

where  $\mathbf{P}_{i|j}^{aug}$  denotes the covariance of the augmented state (corresponding to  $m$  cloned poses, along with the current state).

During the MSC-KF update step, we process all measurements of the features observed by the  $m$ -th clone (i.e., the one about to be marginalized from the sliding window of poses). We utilize (26) to compute the observability-constrained measurement Jacobian,  $\widehat{\mathbf{H}}_k$ , for each measurement and stack all observations of the  $i$ -th feature across  $m$  time steps into a large measurement vector

$$\begin{bmatrix} \tilde{\mathbf{z}}_k \\ \vdots \\ \tilde{\mathbf{z}}_{k-m} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{H}}_k \\ \vdots \\ \widehat{\mathbf{H}}_{k-m} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{aug} \\ \tilde{\mathbf{f}} \end{bmatrix} + \begin{bmatrix} \eta_k \\ \vdots \\ \eta_{k-m} \end{bmatrix} = \widehat{\mathbf{H}}_x \tilde{\mathbf{x}}^{aug} + \widehat{\mathbf{H}}_f \tilde{\mathbf{f}} + \boldsymbol{\eta} \quad (30)$$

where  $\widehat{\mathbf{H}}_x$  and  $\widehat{\mathbf{H}}_f$  are the Jacobians corresponding to the augmented state vector  $\tilde{\mathbf{x}}^{aug}$ , and to the feature, respectively. To avoid including  $\tilde{\mathbf{f}}$  into the state, we marginalize it *on-the-fly* by projecting (30) onto the left nullspace of  $\widehat{\mathbf{H}}_f$ ,  $\mathbf{W}$ . This yields

$$\mathbf{W}^T \tilde{\mathbf{z}} = \mathbf{W}^T \widehat{\mathbf{H}}_x \tilde{\mathbf{x}}^{aug} + \mathbf{W}^T \boldsymbol{\eta} \quad \Leftrightarrow \quad \tilde{\mathbf{z}}' = \widehat{\mathbf{H}}_x' \tilde{\mathbf{x}}^{aug} + \boldsymbol{\eta}', \quad (31)$$

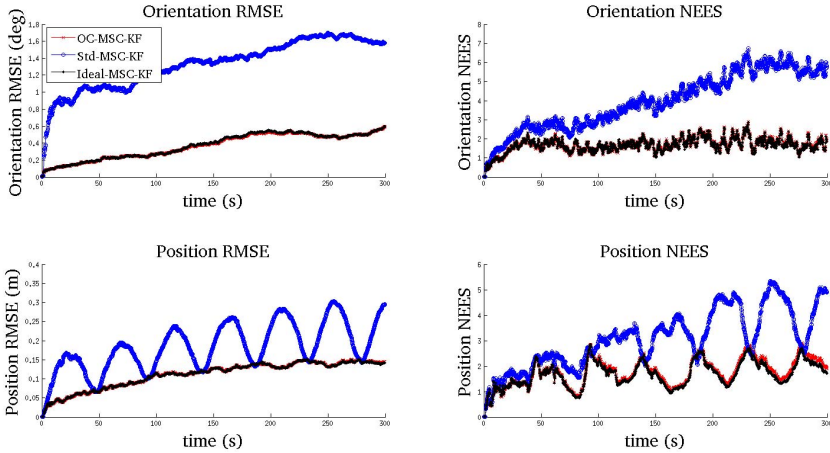
which we employ to update the state estimate and covariance using the standard EKF update equations.<sup>3</sup>

## 6 Simulation Results

We conducted Monte-Carlo simulations to evaluate the consistency of the proposed method applied to the MSC-KF [22]. Specifically, we compared the standard MSC-KF (Std-MSC-KF) with the Observability-Constrained MSC-KF (OC-MSC-KF) (see Sect. 4 and Sect. 5). We employed the Ideal-MSC-KF, whose Jacobians are linearized at the true states, as a benchmark, since it fulfills the observability properties of the true linearized system.

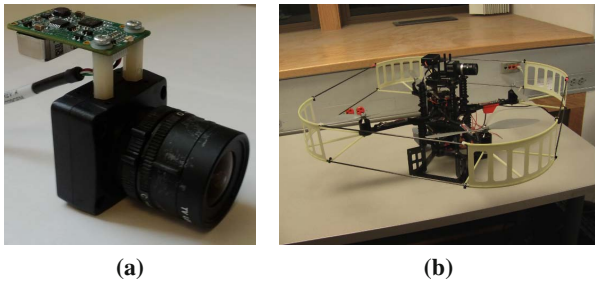
We evaluated the Root Mean Squared Error (RMSE) and Normalized Estimation Error Squared (NEES) over 30 trials (see Fig. 2) in which the camera-IMU platform traversed a circular trajectory of radius 5 m at an average speed of 60 cm/s,

<sup>3</sup> The interested reader is referred to [20] for a more complete perspective.



**Fig. 2** The average RMSE and NEES over 30 Monte-Carlo simulation trials for orientation (above) and position (below). Note that the OC-MSC-KF attains performance indistinguishable from the Ideal-MSC-KF.

and observed 50 randomly distributed features per image. The camera was modeled with a 45 deg field of view, and measurement noise with  $\sigma_{\eta} = 1$  px. The IMU was modeled with MEMS quality sensors. As evident from Fig. 2, the OC-MSC-KF outperforms the Std-MSC-KF and attains performance almost indistinguishable from the Ideal-MSC-KF in terms of RMSE and NEES. This indicates that ensuring an estimator respects the observability properties of the true system, plays a key role in improving both the accuracy and consistency of VINS.



**Fig. 3** (a) The experimental testbed comprises a light-weight InterSense NavChip IMU and a Point Grey Chameleon Camera. The dimensions of the sensing package are approximately 6 cm tall, by 5 cm wide, by 8 cm deep. (b) An AscTech Pelican on which the camera-IMU package was mounted during the experiment.

## 7 Experimental Results

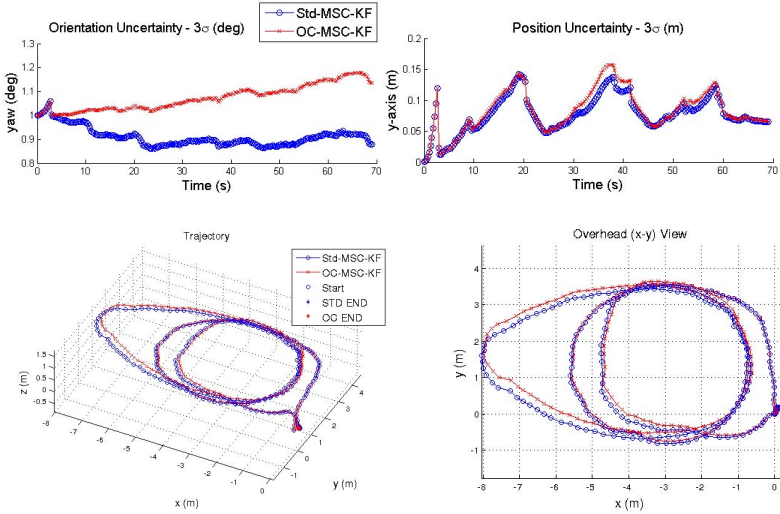
We further validated the proposed OC-MSK-KF on real-world data. Our hardware testbed consists of a Point Grey monochrome monocular camera with resolution 640x480 pixels and an InterSense NavChip IMU, both of which were rigidly mounted on an AscTec Pelican quadrotor (see Fig. 3). We begin with an overview of our image processing approach, followed by the experimental evaluation.

After acquiring image  $k$ , it is inserted into a sliding window buffer of  $m$  images,  $\{k - m + 1, k - m + 2, \dots, k\}$ . We then extract features from the first image in the window using the Shi-Tomasi corner detector [25] and track them pairwise through the window using the KLT tracking algorithm [17]. To remove outliers from the resulting tracks, we use a two-point algorithm to find the essential matrix between successive frames. Specifically, given the filter's estimated rotation (from the gyroscopes' measurements) between image  $i$  and  $j$ ,  ${}^i\hat{q}_j$ , we estimate the essential matrix from only two feature correspondences. This approach is more robust than the five-point algorithm [23] because it provides two solutions for the essential matrix rather than up to ten. Moreover, it requires only two data points, and thus it reaches a consensus with fewer hypotheses when used in a RANSAC framework.

At every time step, the robot poses corresponding to the last  $m$  images are kept in the state vector, as described in [24]. Upon completion of the image processing, all the features that first appeared at the oldest robot pose (corresponding to image  $k - m + 1$ ) are processed following the MSC-KF approach, as discussed in Sect. 5.

The sensor platform traversed three loops of total length 50 m in an indoor area and finally returned to its initial position. At the end of the trajectory, the Std-MSK-KF had a position error of 18.73 cm, while the final error for the OC-MSK-KF was 16.39 cm (approx. 0.38% and 0.33% of the distance traveled, respectively). In order to assess the impact of inconsistency on the orientation estimates of both methods, we used as ground truth the rotation between the first and last images computed independently using BLS and feature point matches. The Std-MSK-KF had final orientation error  $[0.15 \ -0.23 \ -5.13]$  deg for roll, pitch, and yaw (rpy), while the rpy errors for the OC-MSK-KF were  $[0.19 \ -0.20 \ -1.32]$  deg, respectively.

In addition to achieving higher accuracy, for yaw in particular, the OC-MSK-KF is more conservative since it strictly adheres to the unobservable directions of the system. This is evident in both the position and orientation uncertainties. We plot the y-axis position and yaw angle uncertainties in Fig. 4, as representative results. Most notably, the yaw uncertainty of the OC-MSK-KF remains approximately 1.13 deg ( $3\sigma$ ), while for the Std-MSK-KF it reduces to 0.87 deg ( $3\sigma$ ). This indicates that the Std-MSK-KF gains spurious orientation information, which leads to inconsistency. Lastly, we also show the 3D trajectory along with an overhead (x-y) view. It is evident that the Std-MSK-KF yaw error impacts the position accuracy, as the Std-MSK-KF trajectory exhibits a rotation with respect to the OC-MSK-KF.



**Fig. 4** (above) The position and orientation uncertainties ( $3\sigma$  bounds) for the yaw angle and the y-axis, which demonstrate that the Std-MSC-KF gains spurious information about its orientation. (below) The 3D trajectory and corresponding overhead (x-y) view.

### 8 Conclusion

In this paper, we analyzed a root cause of inconsistency in VINS, specifically, the gain of spurious information due to incorrect system observability properties of an EKF-based VINS estimator. We introduced an observability-constrained framework for explicitly enforcing the correct number and structure of unobservable directions by modifying the system and measurement Jacobians. We applied this methodology to the MSC-KF, and showed improved consistency both in simulations and through real-world experiments. Our future research directions include investigating other sources of VINS inconsistency, such as the existence of local minima, and extending our work to systems with multiple exteroceptive sensing modalities.

**Acknowledgement.** This work was supported by the University of Minnesota (UMN) through the Digital Technology Center (DTC) and the Air Force Office of Scientific Research (FA9550-10-1-0567). J. A. Hesch was supported by the UMN Doctoral Dissertation Fellowship.

### References

1. Bailey, T., Nieto, J., Guivant, J., Stevens, M., Nebot, E.: Consistency of the EKF-SLAM algorithm. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Beijing, China, October 9-15, pp. 3562–3568 (2006)

2. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation. John Wiley & Sons, New York (2001)
3. Bouguet, J.-Y.: Camera calibration toolbox for matlab (2006)
4. Bryson, M., Sukkarieh, S.: Observability analysis and active control for airborne SLAM. *IEEE Trans. on Aerospace and Electronic Systems* 44(1), 261–280 (2008)
5. Castellanos, J.A., Martınez-Cantin, R., Neira, J.: Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems* 55(1), 21–29 (2007)
6. Ebcin, S., Veth, M.: Tightly-coupled image-aided inertial navigation using the unscented Kalman filter. Technical report, Air Force Institute of Technology, Dayton, OH (2007)
7. Hermann, R., Krener, A.: Nonlinear controllability and observability. *IEEE Trans. on Automatic Control* 22(5), 728–740 (1977)
8. Hesch, J.A., Kottas, D.G., Bowman, S.L., Roumeliotis, S.I.: Observability-constrained vision-aided inertial navigation. Technical Report 2012-001, University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab. (February 2012)
9. Huang, G.P., Mourikis, A.I., Roumeliotis, S.I.: A first-estimates jacobian EKF for improving SLAM consistency. In: Khatib, O., Kumar, V., Pappas, G.J. (eds.) *Experimental Robotics*. STAR, vol. 54, pp. 373–382. Springer, Heidelberg (2009)
10. Huang, G.P., Mourikis, A.I., Roumeliotis, S.I.: On the complexity and consistency of UKF-based SLAM. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 12–17, pp. 4401–4408 (2009)
11. Huang, G.P., Mourikis, A.I., Roumeliotis, S.I.: Observability-based rules for designing consistent EKF SLAM estimators. *Int. Journal of Robotics Research* 29(5), 502–528 (2010)
12. Isidori, A.: *Nonlinear Control Systems*. Springer (1989)
13. Jones, E.S., Soatto, S.: Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int. Journal of Robotics Research* 30(4), 407–430 (2011)
14. Julier, S.J., Uhlmann, J.K.: A counter example to the theory of simultaneous localization and map building. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, May 21–26, pp. 4238–4243 (2001)
15. Kelly, J., Sukhatme, G.S.: Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *Int. Journal of Robotics Research* 30(1), 56–79 (2011)
16. Li, M., Mourikis, A.I.: Improving the accuracy of EKF-based visual-inertial odometry. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, St. Paul, MN, May 14–18, pp. 828–835 (2012)
17. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proc. of the Int. Joint Conf. on Artificial Intelligence*, Vancouver, B.C., Canada, August 24–28, pp. 674–679 (1981)
18. Martinelli, A.: Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Trans. on Robotics* 28(1), 44–60 (2012)
19. Mirzaei, F.M., Roumeliotis, S.I.: A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Trans. on Robotics* 24(5), 1143–1156 (2008)
20. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, April 10–14, pp. 3565–3572 (2007)
21. Mourikis, A.I., Roumeliotis, S.I.: A dual-layer estimator architecture for long-term localization. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, pp. 1–8 (June 2008)
22. Mourikis, A.I., Trawny, N., Roumeliotis, S.I., Johnson, A.E., Ansar, A., Matthies, L.: Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Trans. on Robotics* 25(2), 264–280 (2009)

23. Nistér, D.: An efficient solution to the five-point relative pose problem. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI, June 16-22, pp. 195–202 (2003)
24. Roulletiotis, S.I., Burdick, J.W.: Stochastic cloning: A generalized framework for processing relative state measurements. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, Washington D.C., May 11-15, pp. 1788–1795 (2002)
25. Shi, J., Tomasi, C.: Good features to track. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, June 27-July 2, pp. 593–600 (1994)
26. Strelow, D.W.: Motion estimation from image and inertial measurements. PhD thesis, Carnegie Mellon University, Pittsburgh, PA (November 2004)
27. Trawny, N., Roulletiotis, S.I.: Indirect Kalman filter for 3D attitude estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab. (March 2005)



# **Part V: ISER Session Summary on “Multi-Robot”**

Nathan Michael

University of Pennsylvania

## **Session Summary**

Multi-robot systems often require inter-robot sensing to enable cooperative control and localization for possibly large numbers of robots. The papers in this section address several challenges associated with the modeling of relative inter-robot sensors and the design of localization and control algorithms that are cognizant of the computational capabilities of the vehicles in the multi-robot system.

The paper by Prorok and Martinoli addresses the problem of multi-robot localization using relative range and bearing measurements resulting from multiple ultra-wideband (UWB) Time Difference of Arrival (TDoA) sensors. A measurement model is proposed to capture the variability in sensor performance for line-of-sight (LOS) and non-LOS conditions in complex environments. A multi-robot localization strategy is formulated and evaluated in experimentation via a team of ground robots to establish the characteristics and accuracy of the model and localization approach.

The paper by Charrow et al. considers the development of a cooperative strategy for a team of mobile robots to actively localize and control to a stationary target with unknown location in a complex indoor environment via range-only TDoA sensors. A sensor model and estimation approach are proposed that address the effects of LOS and non-LOS conditions on the accuracy of the target localization. An active control strategy based on the maximization of mutual information between each vehicle's target belief distribution is evaluated in experimentation on a team of ground vehicles in complex indoor environments.

The paper by Gowal and Martinoli proposes a decentralized receding-horizon control approach for multi-robot rendezvous based on noisy relative inter-robot observations. The associated cost function and optimization strategy are carefully designed to operate in real-time on a robot with limited computational capabilities and memory. Experimental evaluation of the approach and the effects of computational delays on a team of resource-constrained ground vehicles provides insight into the algorithm performance when considering multiple optimization strategies.

# Accurate Localization with Ultra-Wideband: Tessellated Spatial Models and Collaboration

Amanda Prorok and Alcherio Martinoli

**Abstract.** Ultra-wideband (UWB) localization is a recent technology that promises to outperform many indoor localization methods currently available. Despite its desirable traits, such as precision and high material penetrability, the resolution of non-line-of-sight (NLOS) signals remains a very hard problem and has a significant impact on the localization accuracy. In this work, we address the peculiarities of UWB error behavior by building models that capture the spatiality as well as the multimodal nature of the error statistics. Our framework utilizes tessellated maps that associate multimodal probabilistic error models to localities in space. In addition to our UWB localization strategy (which provides absolute position estimates), we investigate the effects of collaboration in the form of relative positioning. We test our approach experimentally on a group of ten mobile robots equipped with UWB emitters and extension modules providing inter-robot relative range and bearing measurements.

## 1 Introduction

Due to its large frequency spectrum, UWB is able to penetrate through objects in NLOS scenarios, and thus alleviates the LOS constraint imposed by other sensor types relying on media such as infrared, ultrasound, visible light or narrow-band radio. This advantage ultimately enables localization over large ranges and in dynamic environments [4], which makes UWB an attractive candidate for indoor applications such as asset management, inventory tracking and assembly control, for a variety of different industries. Nevertheless, NLOS scenarios may cause biases in the signal propagation times, which leads to significant localization errors. In order to guarantee reliable and accurate performance, these biases need to be addressed by an effective localization strategy.

---

Amanda Prorok · Alcherio Martinoli

Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture,  
Civil and Environmental Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL),  
Switzerland

e-mail: {amanda.prorok, alcherio.martinoli}@epfl.ch

In this paper, we consider the problem of absolute localization of a team of mobile robots for unknown initial conditions. We design an algorithm that works in conjunction with a tessellated spatial error model built a priori. Our solution targets miniaturized platforms equipped with low-power sensing modalities, and we ultimately envision its portability onto much smaller devices such as embedded/portable tags. Our localization strategy uses time-difference-of-arrival (TDOA) measurements from two or more UWB base station pairs and on-board dead-reckoning information. Lastly, as it is commonly known that multi-robot collaboration is able to compensate for deficiencies in the data owned by a single robot [1, 6], we extend our approach to include relative (inter-robot) range and bearing observations.

Recently, UWB has received some attention within the robotics community. The studies in [2] and [3] develop probabilistic models for biased UWB range measurements which are combined with on-board odometry data. Yet, both papers model NLOS biases within augmented-state particle filters that do not take LOS/NLOS path conditions and bias probability distributions into account explicitly, and that depend on the motion of the mobile target. Furthermore, given the novelty of UWB positioning systems in the robotics community, to the best of our knowledge, no significant studies have been performed on the fusion of UWB with on-board exteroceptive sensors, in the case of single-robot systems, nor any on-board relative positioning sensors, in the case of multi-robot systems. Lastly, this work is amongst the first to model UWB TDOA errors as a function of space.

## 2 Technical Approach

Our proposed method is a culmination of our ongoing research efforts in the domain of indoor localization, combining two main elements: an UWB system capable of absolute positioning [9, 11], and a collaborative multi-robot system capable of relative observations [7, 10].

### 2.1 UWB Localization

UWB is a radio technology which is characterized by its very large bandwidth compared to conventional narrowband systems, and in particular features high positioning accuracy (due to a time resolution in the order of nanoseconds), and high material penetrability (due to a bandwidth typically larger than 0.5 GHz). Despite these desirable traits, the resolution of multipath signals remains a hard problem—the complexity of implementing state-of-the-art direct signal path detection algorithms is exacerbated by the necessity of maintaining very high sampling rates (in the order of several GHz). Thus, our approach addresses the occurrence of both LOS and NLOS signal paths by proposing a probabilistic measurement model that captures this multimodal error behavior. But because NLOS biases are originally introduced by the clutter in a given environment, the perceived error behavior is actually a spatial phenomenon. Thus, even if a non-spatial error model is able to

represent the error distribution in mixed LOS/NLOS signal path environments, a spatial model will very likely produce better results. Yet, given the sharp (discontinuous) transitions from LOS to potentially harsh NLOS localities in space, developing a viable spatial UWB error model is a hard problem. The goal of this work is indeed to develop such a spatial error model. Given the reasons elaborated above, we resort to a mapping technique that allows us to tessellate space into areas, and where each area is associated with a unique error model. Indeed, we exploit the ability of our error model to capture and adapt to all types of UWB propagation, and spatially customize the parametrization of this model using the underlying tessellation (to ultimately produce a map, which can be fine-tuned according to user requirements).

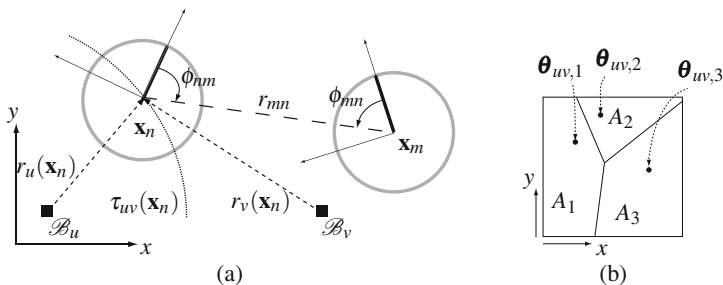
When using UWB for localization, the measured quantity is time-of-flight. However, in practice, time-of-arrival (TOA) systems are rarely implemented due to the complexity induced by the required synchronization of a mobile node with the base stations. Instead, it is a common choice to implement time-difference-of-arrival (TDOA) systems which are significantly more practical, since only the synchronization among base stations is required. This, in turn, enables a significant miniaturization of the emitter boards (to an order of a few centimeters in size), as well as a reduction of the consumption power (to an order of mW).

### 2.1.1 Framework

We consider a pair of UWB base stations  $\langle \mathcal{B}_u, \mathcal{B}_v \rangle$ , both fixed and well-localized in an absolute coordinate system, and a robot  $\mathcal{R}_n$  at position  $\mathbf{x}_n$ , equipped with an UWB emitter tag. At any given time, the robot  $\mathcal{R}_n$  may receive a measured TDOA value  $\hat{\tau}_{uv,n}$  from any pair of base stations  $\langle \mathcal{B}_u, \mathcal{B}_v \rangle$ . We denote by  $T_{n,t} = \{ \langle \hat{\tau}_{uv,n,t}, \langle \mathcal{B}_u, \mathcal{B}_v \rangle \rangle \mid \exists \langle \mathcal{B}_u, \mathcal{B}_v \rangle \in \mathbb{B} \}$  the set of TDOA measurements received by a robot  $\mathcal{R}_n$  at a given time  $t$ . The TDOA measurement error  $\Delta \tau_{uv,n}$  for robot  $\mathcal{R}_n$  and base station pair  $\langle \mathcal{B}_u, \mathcal{B}_v \rangle$  is defined as the difference between the nominal (error-free) TDOA value at the actual robot position and the measured TDOA value

$$\Delta \tau_{uv,n}(\hat{\tau}_{uv,n}, \mathbf{x}_n) = \hat{\tau}_{uv,n} - \tau_{uv,n}(\mathbf{x}_n), \quad (1)$$

where  $\tau_{uv,n}(\mathbf{x}_n) = r_u(\mathbf{x}_n) - r_v(\mathbf{x}_n)$ , and  $r_u(\mathbf{x}_n)$  is the range between base station  $\mathcal{B}_u$  and  $\mathbf{x}_n$ . In order to model the UWB error behavior, we take account of spatiality by defining a set  $\mathcal{M}_{uv}$  of  $N_A$  areas  $\mathcal{M}_{uv} = \{ \langle A_a, \boldsymbol{\theta}_{uv,a} \rangle \mid a = 1, \dots, N_A \}$  where  $\boldsymbol{\theta}_{uv,a}$  is a parameter vector, and  $A_a \subset \mathbb{R}^2$ . Note that the areas are disjoint  $\bigcap_n A_a = \emptyset$  and  $\bigcup_n A_a$  covers the whole space. In other words, each area  $A_a$  is associated with a parameter vector  $\boldsymbol{\theta}_{uv,a}$ , as illustrated in Figure 1(b). We refer to  $\mathcal{M}_{uv}$  as the *map* for base station pair  $\langle \mathcal{B}_u, \mathcal{B}_v \rangle$ , and denote the set of all maps as  $\mathcal{M} = \{ \mathcal{M}_{uv} \mid \exists \langle \mathcal{B}_u, \mathcal{B}_v \rangle \in \mathbb{B} \}$ , where  $\mathbb{B}$  is the set of all base station pairs. Furthermore, we define a function  $m_{uv} : \mathbb{R}^2 \mapsto \Theta$  that maps to any position in two-dimensional space a parameter vector in the finite set  $\Theta : m_{uv}(\mathbf{x}_n) = \boldsymbol{\theta}_{uv,a}$  such that  $\exists \langle A_a, \boldsymbol{\theta}_{uv,a} \rangle \in \mathcal{M}_{uv}$  with  $\mathbf{x}_n \in A_a$ . Finally, we model the error  $\Delta \tau_{uv,n}(\hat{\tau}_{uv,n}, \mathbf{x}_n)$  for a given base station pair  $\langle \mathcal{B}_u, \mathcal{B}_v \rangle$  with a probability density function  $p$  that covers an area  $A_a$  (such that  $\mathbf{x}_n \in A_a$ ) and that depends on the parameter vector  $\boldsymbol{\theta}_{uv,a}$ . We define our error model as



**Fig. 1** (a) System of two robots at positions  $\mathbf{x}_n, \mathbf{x}_m$  and two well-localized UWB base stations  $\mathcal{B}_U$  and  $\mathcal{B}_V$ . The figure shows the true ranges  $r_u(\mathbf{x}_n)$  and  $r_v(\mathbf{x}_n)$  of robot  $\mathcal{B}_n$  to the respective base stations, as well as a segment of the hyperbola resulting from the range-difference measurement  $\tau_{uv,n}$ . The figure also depicts the relative robot range  $r_{mn} = r_{nm}$ , and the relative bearing values  $\phi_{mn}$  and  $\phi_{nm}$ . (b) Distinct UWB error models  $p(\Delta \tau_{uv,n}; \boldsymbol{\theta}_{uv,a})$  are mapped to individual areas  $A_a$ .

$$p(\Delta \tau_{uv,n}; \boldsymbol{\theta}_{uv,a}) \triangleq p(\Delta \tau_{uv,n}(\hat{\tau}_{uv,n}, \mathbf{x}_n); m_{uv}(\mathbf{x}_n)). \quad (2)$$

Practically, we solve the problem of localizing with UWB measurements in two steps. First, we collect a data set covering the working environment to build the maps in  $\mathcal{M}$ . Second, during the actual exercise, we use  $\mathcal{M}$  to look up UWB error models in function of the current estimated positions, and then use these models within a localization filter to evaluate incoming UWB observations.

## 2.1.2 UWB Error Model

Our baseline error model [13] for the range<sup>1</sup> between a base station  $\mathcal{B}_u$  and a target node (robot) at position  $\mathbf{x}_n$  is

$$\hat{r}_u = r_u(\mathbf{x}_n) + \varepsilon + Y b_u \quad (3)$$

where  $r_u(\mathbf{x}_n)$  represents the true distance,  $b_u$  is a non-negative distance bias introduced by a NLOS signal propagation, and  $\varepsilon \sim p_{\mathcal{N}}(0, \sigma_{\mathcal{N}}^2)$  is a zero-mean Gaussian measurement noise with variance  $\sigma_{\mathcal{N}}^2$ , common to all base stations. The bias  $b_u$  is modeled as a log-normal random variable  $b_u \sim p_{ln\mathcal{N}}(\mu_u, \sigma_u)$ , supported on the semi-infinite interval  $(0, \infty)$ , and which is associated uniquely to a base station  $\mathcal{B}_u$ . The random variable  $Y$  qualifies the occurrence of a NLOS signal path and follows a Bernoulli distribution. Explicitly, it takes the value 1 with probability  $(1 - P_{L_u})$  and the value 0 with probability  $P_{L_u}$ , where  $P_{L_u}$  is the probability of measuring a LOS path, and correspondingly,  $(1 - P_{L_u})$  is the probability of measuring a NLOS path. For a range error defined as

<sup>1</sup> The terms TOA and TDOA are used interchangeably with the terms range and range difference, respectively, as they differ only by a constant factor (propagation speed).

$$\Delta r_u(\hat{r}_u, \mathbf{x}) = \hat{r}_u - r_u(\mathbf{x}) \quad (4)$$

the TOA measurement model  $p_u$  describes the likelihood of  $\Delta r_u$  occurring when a robot measures a certain range distance  $\hat{r}_u$  from a base station  $\mathcal{B}_u$  at an actual position  $\mathbf{x}$  with a nominal (actual) range  $r_u$  (we refer the reader to [11] for a detailed derivation of  $p_u$ ).

Then, in order to model TDOA instead of TOA observations, we define the difference range value (i.e. TDOA) between two base stations  $\mathcal{B}_u$  and  $\mathcal{B}_v$  to a target node  $\mathcal{R}_n$  and model the TDOA error  $\Delta \tau_{uv}$  as previously shown in Equation 1. Given this formalism, we can describe the probability density of a given TDOA measurement error  $\Delta \tau_{uv}$  as the probability density of the subtraction of two random variables drawn from the probability densities  $p_u$  and  $p_v$ , describing the TOA error models of the two respective base stations. The resulting probability density is

$$p_{uv}(\Delta \tau_{uv}) \triangleq (p_u * p_v^-)(\Delta \tau_{uv}) \quad (5)$$

which is a convolution of the probability density of the range error  $\Delta r_u$  and the mirrored probability density of  $\Delta r_v$  (i.e.,  $p_v^-(\Delta r_v) = p_v(-\Delta r_v)$ ).

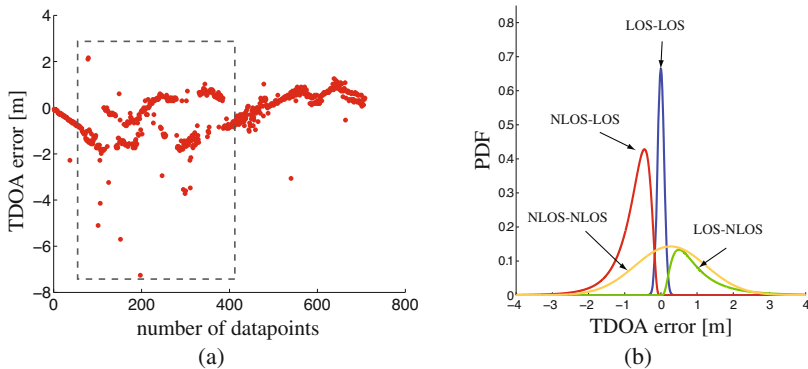
Although numerical implementations for the TDOA measurement model of Equation 5 are easily found, they imply nested integrals which may incur a substantial computational overhead when deploying the model on a real embedded platform for real-time operation. Also, the model itself is analytically non-tractable, which causes difficulties when deriving viable estimators. For these reasons, we use a closed-form approximation [11] to simplify the TDOA measurement model of Equation 5 to a sum of four terms:

$$p(\Delta \tau_{uv}; \boldsymbol{\theta}_{uv,a}) = \left( P_{L_u} P_{L_v} p_{\sqrt{2}\mathcal{N}} + P_{L_u} (1 - P_{L_v}) p_{\ln\mathcal{N},v}^- + \right. \\ \left. P_{L_v} (1 - P_{L_u}) p_{\ln\mathcal{N},u} + (1 - P_{L_u})(1 - P_{L_v}) p_{\mathcal{N}} \right) (\Delta \tau_{uv}) \quad (6)$$

Here,  $p_{\sqrt{2}\mathcal{N}} = p_{\mathcal{N}} * p_{\mathcal{N}}^-$ , and  $p_{\mathcal{N}}$  approximates the convolution  $p_{\ln\mathcal{N}} * p_{\ln\mathcal{N}}^-$  with a Gaussian that matches its moments. Hence, the parameter vector  $\boldsymbol{\theta}_{uv,a}$  is

$$\boldsymbol{\theta}_{uv,a} = [\mu_u, \sigma_u, \mu_v, \sigma_v, P_{L_u}, P_{L_v}]^T$$

and  $\mu_u, \mu_v \in \mathbb{R}$ ,  $\sigma_u, \sigma_v \in \mathbb{R}^+$ , and  $P_{L_u}, P_{L_v} \in [0, 1]$ . Finally, we note that, given some ground truth TDOA measurement errors, the parameters of the above equation can be determined via an efficient Expectation Maximization algorithm (batch mode as well as online) that maximizes their likelihood [11]. Figure 2 provides an intuition of the multimodal nature of TDOA error data, and illustrates how our error model proposes to capture this. In particular, Figure 2(b) illustrates how each of the four terms of Eq. 6 represents one of the four possible modes of operation for a base station pair: LOS-LOS, NLOS-LOS, LOS-NLOS, and NLOS-NLOS.



**Fig. 2** (a) Example of the multimodal nature of TDOA error data, in particular those points contained within the dashed box. This data was collected by a robot moving in a straight line through an indoor environment with obstacles. (b) We consider a base station pair  $(\mathcal{B}_1, \mathcal{B}_2)$ . The plot illustrates the four modes which form the complete multimodal probability density function shown in Eq. 6. The model parameters are set to:  $\mu_1 = -0.43$ ,  $\mu_2 = -0.2$ ,  $\sigma_1 = 0.6$ ,  $\sigma_2 = 0.7$ ,  $P_{L_1} = 0.3$ ,  $P_{L_2} = 0.5$ .

## 2.2 Collaborative Localization and UWB

There is abundant literature discussing various strategies toward solving the multi-robot localization problem. Our approach distinguishes itself particularly by respecting the following design goals: low cost, full decentralization and scalability, and asynchrony of relative observations [7, 11]. Given its efficiency in solving the localization problem for unknown initial conditions and its ability to accommodate arbitrary probability density functions, our method of choice is the particle filter (otherwise known as Monte Carlo Localization (MCL)). Our collaboration strategy exploits associated, inter-robot relative range and bearing observations, which are evaluated by a dedicated detection model and fused with dead-reckoning information (e.g., odometry) to form position estimates.

We subsequently introduce an augmented system by considering both relative positioning data as well as UWB, schematized in Figure 1(a). Essentially, we propose a baseline algorithm which fuses UWB TDOA measurements and relative positioning measurements with dead-reckoning information. Ultimately, the goal is to experimentally test this algorithm and provide an insight into if and how multi-robot strategies can contribute to improving the accuracy of UWB.

### 2.2.1 Framework

Our multi-robot system is composed of  $N_{\mathcal{R}}$  robots  $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_{N_{\mathcal{R}}}$ , each running an individual localization filter. The belief of a robot's pose is formulated as

$$\mathbf{Bel}(\mathbf{x}_{n,t}) \sim \{ \langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle | i = 1, \dots, M \} = X_{n,t} \quad (7)$$

where  $M$  is the number of particles,  $\mathbf{x}_{n,t}^{[i]}$  is a sample of the random variable  $\mathbf{x}_{n,t} = [x_{n,t}, y_{n,t}, \psi_{n,t}]^T$  (where  $x_{n,t}$  and  $y_{n,t}$  are Euclidian coordinates and  $\psi_{n,t}$  is the orientation), and  $w_{n,t}^{[i]}$  is its weight. The symbol  $X_{n,t}$  refers to the set of particles  $\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle$  at time  $t$  belonging to robot  $\mathcal{R}_n$ . At any given time  $t$ , a robot  $\mathcal{R}_m$  may make a range measurement  $\hat{r}_{mn,t}$  and a bearing measurement  $\hat{\phi}_{mn,t}$  of robot  $\mathcal{R}_n$ . Thus, with the knowledge of the range and bearing noise values, we define a robot detection model  $q$  which describes the probability that robot  $\mathcal{R}_m$  detects  $\mathcal{R}_n$  at position  $\mathbf{x}_{n,t}$ , given the detection data  $d_{mn,t} = \langle \hat{r}_{mn,t}, \hat{\phi}_{mn,t}, X_{m,t} \rangle$ , as

$$q(\mathbf{x}_{n,t} | d_{mn,t}) \triangleq q(\mathbf{x}_{n,t} | \hat{r}_{mn,t}, \hat{\phi}_{mn,t}, X_{m,t}). \quad (8)$$

### 2.2.2 Sensor Fusion Algorithm

The routine is shown here in Algorithm 1. Line 3 shows the application of the motion model, where  $u_{n,t}$  represents dead-reckoning information. Line 4 shows the application of the measurement model where  $T_{n,t}$  represents the TDOA data. Line 5 shows the application of the robot detection model, where  $D_{n,t} = \{d_{mn,t} | \mathcal{R}_m \in \mathcal{N}_{n,t}\}$  is the set of all communication messages received by robot  $\mathcal{R}_n$ . In other words, the detected robot will apply the detection model using data received from the robots that made the detection. A more detailed description of our robot detection model can be found in [10]. In addition to using the robot detection model for updating the belief representation **Bel** ( $\mathbf{x}_{n,t}$ ), our approach relies on a *reciprocal* sampling method [7], shown in line 13.

---

#### Algorithm 1. MultiRobot\_UWB\_MCL( $X_{n,t-1}, u_{n,t}, T_{n,t}, D_{n,t}$ )

---

```

1:  $\bar{X}_{n,t} = X_{n,t} = \emptyset$ 
2: for  $i = 1$  to  $M$  do
3:    $\mathbf{x}_{n,t}^{[i]} \leftarrow \text{Motion\_Model}(u_{n,t}, \mathbf{x}_{n,t-1}^{[i]})$ 
4:    $w \leftarrow \prod_{(\hat{r}_{uv,n,t}, \mathcal{B}_u, \mathcal{B}_v) \in T_{n,t}} P(\Delta \hat{r}_{uv,n,t}; m_{uv}(\mathbf{x}_{n,t}^{[i]}))$ 
5:    $w \leftarrow w_t^{[i]} \cdot \prod_{d_{mn} \in D_{n,t}} q(\mathbf{x}_{n,t}^{[i]} | d_{mn})$ 
6:    $\bar{X}_{n,t} \leftarrow \bar{X}_{n,t} + \langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle$ 
7: end for
8: for  $i = 1$  to  $M$  do
9:    $r \sim \text{Uniform}(0, 1)$ 
10:  if  $r \leq (1 - \alpha)$  then
11:     $\mathbf{x}_{n,t}^{[i]} \sim \bar{X}_{n,t}$ 
12:  else
13:     $\mathbf{x} \sim \prod_{d_{mn} \in D_{n,t}} P_{mn}(\mathbf{x} | d_{mn})$ 
14:  end if
15:   $X_{n,t} \leftarrow X_{n,t} + \langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle$ 
16: end for
17: return  $X_{n,t}$ 

```

---

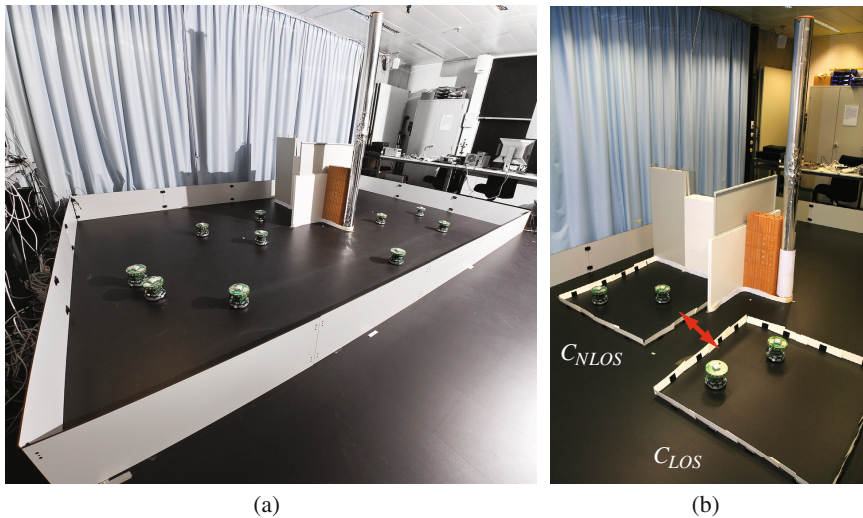


### 3 Experiments

The following section details our experimental setup. We then elaborate the practical implementation of our framework, as introduced in Section 2.1.1.

#### 3.1 Setup

Our experimental setup (Figure 3(a)) consists of three main elements (i) a group of ten mobile robots, (ii) an UWB positioning system composed of four base stations, and (iii) two overhead cameras with overlapping views of a  $5 \times 3$  m<sup>2</sup> arena.



**Fig. 3** (a) The  $5 \times 3$  m<sup>2</sup> experimental arena contains an obstacle composed of various elements made of brick, plaster, metal, wood, and a 3 meter high tube covered in aluminum. Four UWB base stations are mounted on the ceiling in the corners of the lab room. Two overhead cameras provide ground truth positioning in the experimental area. (b) In an additional setup, we test the benefits of collaboration by confining the directions of intercellular robot detections in between a strongly occluded cell (and thus, predominantly NLOS cell,  $C_{NLOS}$ ) and a predominantly LOS cell ( $C_{LOS}$ ).

To perform experiments, we use ten Khepera III robots that drive randomly in the arena at a speed of one robot size per second. The Khepera III is a differential drive robot of 12cm diameter produced by K-Team corporation<sup>2</sup>, see Figure 4. We use the robot with a KoreBot II extension board providing a standard embedded Linux operating system on an Intel XSCALE PXA-270 processor running at 624 MHz. Communication is enabled through an IEEE 802.11b wireless card which is installed in

<sup>2</sup> <http://www.k-team.com/>

**Fig. 4** The Khepera III robot is equipped with a range and bearing extension module which utilizes sixteen infrared LEDs. On top of this board, we mounted an active marker (LED) tracking module, which simultaneously carries the UWB emitter tag.



a built-in CompactFlash slot. The robot uses wheel encoders to provide odometry readings at 5 Hz. It also uses a relative range and bearing module [12], which is set to provide the measurements used by the detection model at a frequency of 1 Hz. The modules' noise characteristics were empirically determined in our actual setup ( $\sigma_r = 0.15 \cdot r_{mm}$ , and  $\sigma_\phi = 0.15$  rad). We set its maximum detection range to be 1 m. Lastly, the robot is also equipped with an LED-based active marker module for tracking that also carries the UWB emitter tag, which emits positioning pulses at a frequency of 10 Hz. The UWB localization system employed in this work is commercially available from Ubisense<sup>3</sup>, Series 7000 (sensors and compact tags). It is installed on the ceiling, in the corners of our 40 m<sup>2</sup> laboratory. The overhead camera system runs on a central processor which also enacts the synchronization of available ground truth positioning data with all incoming raw sensor data (from the UWB system as well as from the robots). In order to compute the ground truth robot positions, the camera system utilizes the open source tracking software *Swis-Track* [5]. The average error of the resulting ground truth is roughly 1cm [8]).

In order to create a NLOS setting that occludes direct paths between the UWB emitters carried by the robots and the four base stations, we install a cross-shaped obstacle in our arena (see Figure 3(a)). The obstacle is 1.5 m long, 1 m high and 20 cm thick, and is composed of several modules made of various materials (brick, plaster, metal, wood). At its extremity, in the center of the arena, we attach a 2 m high tube covered in aluminum. These shapes and materials are chosen in order to realistically emulate the various effects of a typical indoor environment on UWB propagation. Finally, Figure 3(b) shows a secondary setup, which we use to test collaborative strategies more explicitly (as discussed later in Section 4.2).

### 3.2 Mapping

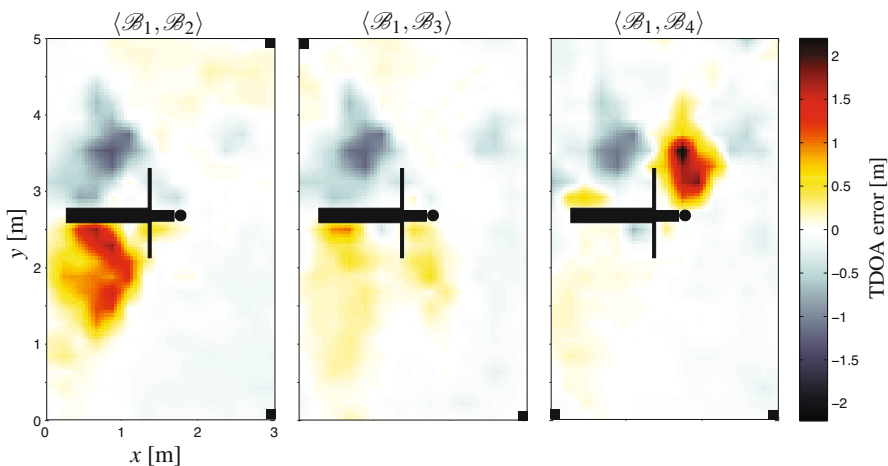
Since we consider three base station pairs ( $\langle \mathcal{B}_1, \mathcal{B}_2 \rangle$ ,  $\langle \mathcal{B}_1, \mathcal{B}_3 \rangle$ ,  $\langle \mathcal{B}_1, \mathcal{B}_4 \rangle$ ),  $\mathcal{M}$  is composed of three maps  $\mathcal{M}_{12}$ ,  $\mathcal{M}_{13}$ , and  $\mathcal{M}_{14}$ . We use the robots to collect a data set comprising over 50'000 TDOA values  $\hat{\tau}_{uv}$  for each base station pair  $\langle \mathcal{B}_u, \mathcal{B}_v \rangle$ , and record the associated ground truth positions  $\mathbf{x}$ , guaranteeing full coverage of our experimental arena. For each data point  $\hat{\tau}_{uv}$ , we then calculate the ground truth TDOA value  $\tau_{uv}(\mathbf{x})$ , which, in turn, allows us to compute the TDOA error value

<sup>3</sup> <http://www.ubisense.net>

$\Delta \tau_{uv}$  and associate it to the position  $\mathbf{x}$ . Figure 5 visualizes the resulting data set of average TDOA error values as a function of space. We observe the higher bias averages of up to  $\pm 2$  m in the vicinity of the obstacle (note that TDOA errors are positive as well as negative). Also, we note that although this figure helps us identify strongly biased areas, the plotted average error values do not accurately represent the full multimodal error behavior.

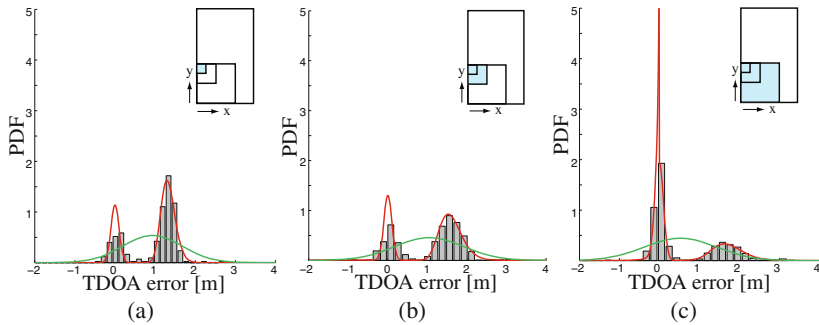
We now proceed with the creation of our set of maps  $\mathcal{M}$  to be used by the UWB localization algorithm. In order to build these maps, we discretize our space into  $N_A = 375$  equally sized grid cells of dimension  $0.2 \times 0.2$  m<sup>2</sup>, with at least 150 data points per grid cell and per base station pair. For each cell  $A_a, a \in \{1, \dots, N_A\}$  defined by our grid map, and for each of the three base station pairs, we now estimate (using Expectation Maximization, as described in Section 2.1.2) the model parameters  $\hat{\theta}_{uv,a}$  that define a unique TDOA error model per cell. This concludes the mapping step, and we denote the final set of maps  $\mathcal{M}_{MM,375}$ .

As a way of validating our multimodal error model, we calculate an additional set of maps (with the same resolution as above) composed of Gaussian distributions instead of our proposed error model (Eq. 6). Thus, in the same way as above, we associate a mean and variance value to each cell in each of the three maps. The final set of maps comprising (unimodal) Gaussian distributions is denoted  $\mathcal{M}_{UM,375}$ . Furthermore, in order to test the effect of spatiality on localization performance, we build two more sets of maps (one with our multimodal error model, and one with Gaussians), composed of one single  $5 \times 3$  m<sup>2</sup> cell. We denote these two sets by  $\mathcal{M}_{MM,1}$  and  $\mathcal{M}_{UM,1}$ .



**Fig. 5** Overhead view of the experimental arena showing the average TDOA error as a function of space, for three base station pairs (we perform 2D smoothing with a Gaussian kernel of size 1 on a grid map of  $150 \times 250$  cells, using over 50'000 UWB TDOA measurements per base station pair). The cross shape shows the placement of the obstacle and the filled squares in the panel corners schematically indicate the placements of the base stations with respect to the layout of the experimental arena.

To illustrate the concept of our mapping strategy, Figure 6 shows fits of (i) a Gaussian probability density function, and (ii) the probability density function proposed by our error model in Eq. 6, for 3 difference cell sizes, a)  $0.5 \times 0.5 \text{ m}^2$ , b)  $1 \times 1 \text{ m}^2$  and c)  $2 \times 2 \text{ m}^2$ . We note that in all cases, the multimodal approach (as proposed by our error model) suggests a better fit to the data. Also, we see that the multimodal nature of the error statistics is preserved across different scales.



**Fig. 6** Normalized histograms of TDOA error data for base station pair  $\langle \mathcal{B}_1, \mathcal{B}_2 \rangle$ . We fit a Gaussian (in green) onto the data, as well as our proposed error model of Eq. 6 (in red). The data is collected over a (a)  $0.5 \times 0.5 \text{ m}^2$  large square, (b)  $1 \times 1 \text{ m}^2$  large square, and (c)  $2 \times 2 \text{ m}^2$  large square. The areas of data collection are indicated by the shaded cells in the schematized arena, in the top right corner of each panel.

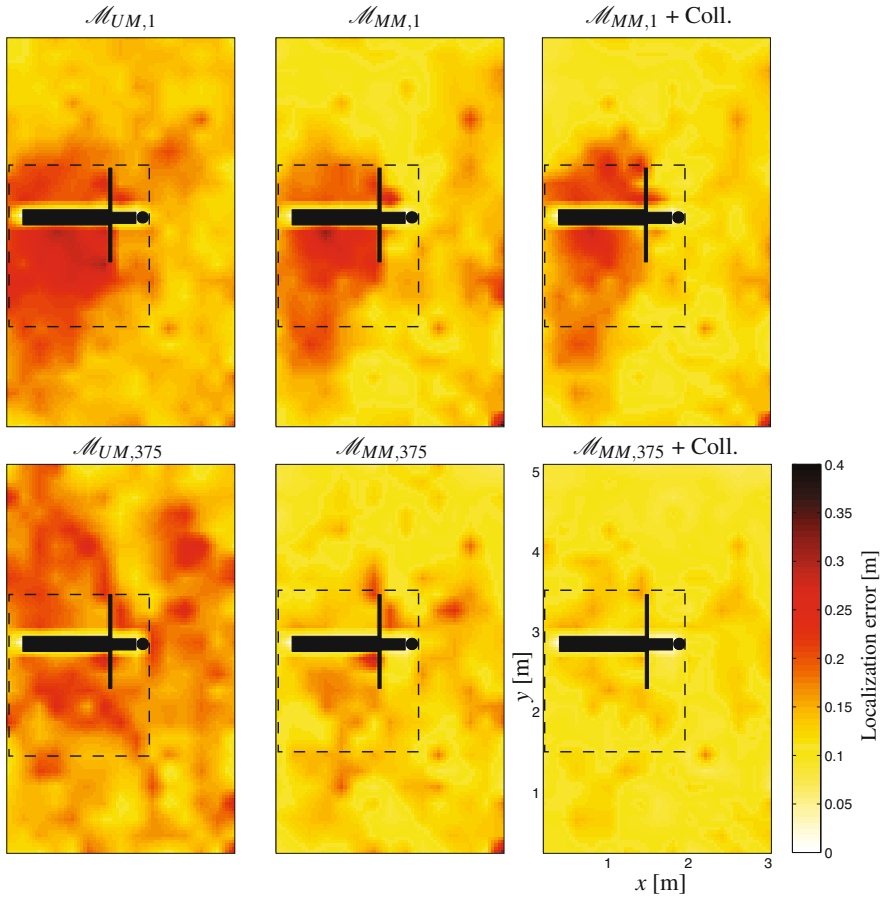
## 4 Results

The paragraphs below discuss two distinct experiments.

### 4.1 Overall localization Error

Our localization algorithm is evaluated on a data set comprising a 40 minute experiment involving ten robots, initially randomly distributed over the arena. Each robot runs Algorithm 1 with 100 particles, which are initially uniformly distributed in the arena (this problem is otherwise known as global localization). Apart from the modalities described in this paper, the robots use no other sensors to localize. We discuss the localization performance in terms of the positioning error (distance to ground truth position) of the center of mass of the particles in a robot's belief, for all ten robots used in the experiment. We test the four mapping strategies (Gaussian vs. multimodal, and 1 cell vs. 375 cells) with and without collaboration—for the non-collaborative version, we omit the robot detection model (Algorithm 1, line 5).

Figure 7 shows the localization performance as a function of space. We note the irregularity of the error distribution: higher errors tend to be in the vicinity of the obstacle. The plots indicate that our multimodal model ( $\mathcal{M}_{MM}$ ) is better than

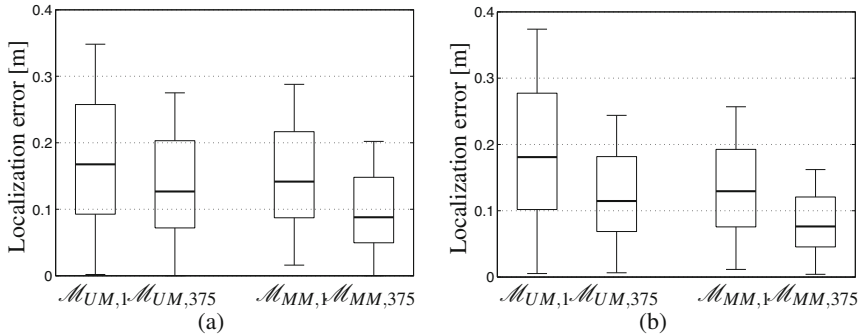


**Fig. 7** The graphs show the average localization error as a function of space, over a 40 minute experiment with ten robots. Six different models are tested. On the top row, the robots used the maps composed of one single cell, and on the bottom row, the robots used the maps composed of 375 cells. The first column shows results for the Gaussian maps, the second column for the multimodal error model maps, and the last column uses multimodal maps as well as collaboration. The dashed line delimits the critical area  $C_{obs}$  around the obstacle.

the Gaussian ( $\mathcal{M}_{UM}$ ), and that a high resolution map (375 cells) is better than a very low resolution map (1 cell). Furthermore, the results indicate that collaboration may mitigate errors, in particular in areas prone to high errors due to NLOS: the bottom right panel ( $\mathcal{M}_{MM,375} + \text{Coll.}$ ) shows a constant distribution of localization errors in the order of 10 cm.

In order to better understand the performance behavior in critical (NLOS) areas, Figure 8 discusses the localization errors measured inside the area  $C_{obs}$  as marked by a dashed line in Figure 7. Figure 8(a) summarizes the results *without* collaboration, in the form of boxplots. We can observe that for a fixed granularity, the multimodal

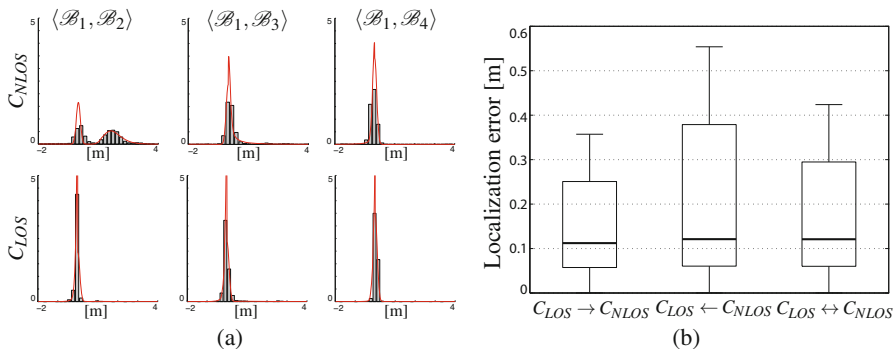
model outperforms the Gaussian model. Also, increasing the model granularity improves localization performance. Figure 8(b) summarizes the results *with* collaboration, in the form of boxplots. For three of the four cases, the performance improves (with a maximum improvement of 18%) with respect to the results of Figure 8(a). In the best case ( $\mathcal{M}_{MM,375}$ ), we have a median localization error of 7.6 cm. Finally, for comparison, we note that the median of the maximum likelihood trilateration estimates (computed with raw TDOA measurements) amounts to 56 cm.



**Fig. 8** Localization error of all ten robots for an experimental run of 40 minutes duration. (a) *Without* collaboration. (b) *With* collaboration. The results are shown in the form of boxplots (25th, 50th and 75th percentile and whiskers containing 85% of the data). Only errors in  $C_{obs}$  are considered.

## 4.2 Evaluation of Collaboration

The previous section establishes that improved performance can be obtained when collaboration is exploited. Here, we look at a more targeted experiment to better understand how collaboration contributes to this improvement. We use the setup shown in Figure 3(b), with two groups of two robots each. The robots' motion is delimited by the cell boundaries, which simultaneously defines a new set of maps (in this case, we denote our set of maps  $\mathcal{M}_{MM,2}$ , using 2 cells of size  $1 \text{ m}^2$  each). Also, as can be observed when comparing Figure 5 to Figure 3(b), the cell  $C_{NLOS}$  is located in a region where significant bias values occur (in particular for base station pair  $(\mathcal{B}_1, \mathcal{B}_2)$ ), whereas cell  $C_{LOS}$  is located in a relatively benign region. Importantly, we note that the cell boundaries are low enough to enable intercellular detections via the relative positioning modules. This setting allows us to test the following collaborative configurations (we remind that reader that if robot  $\mathcal{R}_m$  detects robot  $\mathcal{R}_n$ , it is robot  $\mathcal{R}_n$  that will execute the sensor fusion using detection data sent by robot  $\mathcal{R}_m$ , see Section 2.2.2): (i) robots in  $C_{LOS}$  detect robots in  $C_{NLOS}$ , (ii) robots in  $C_{NLOS}$  detect robots in  $C_{LOS}$ , and (iii) any detections are allowed. Figure 9(b) shows the performances of the three variant strategies. Clearly, method (i) produces the best results: LOS environments benefit from a high accuracy (due to an essentially



**Fig. 9** (a) Normalized histograms of TDOA error data for all base station pairs (columns) and both cells  $C_{LOS}$  and  $C_{NLOS}$  (rows). We fit our proposed error model of Eq. 6 onto the data, and use the resulting PDF in the set of maps  $\mathcal{M}_{MM,2}$ . (b) Localization error of four robots (as shown in Figure 3(b)), employing  $\mathcal{M}_{MM,2}$  and using three variant collaboration schemes, for an experimental run of 20 minutes duration. The results are shown in the form of boxplots.

unimodal distribution with a narrow peak, as shown in Figure 9(a), and thus, it is beneficial to allow robots in  $C_{LOS}$  to influence the beliefs of the robots in  $C_{NLOS}$ .

## 5 Experimental Insights

Although, in theory, UWB localization has the potential of providing centimeter level accuracy, in practice, sophisticated strategies are necessary to mitigate the effect of NLOS biases. Our experiments showed that the error behavior of time-of-flight based UWB measurements is dependent on the configuration of the environment, and thus, can be modeled as a function of space. In conclusion, this work has allowed us to make three main insights: **(I)** There is a clear benefit in terms of localization accuracy when using our multimodal error model instead of a unimodal Gaussian error model. We have seen that even when using the lowest map granularity, the multimodal model almost matches the performance of the unimodal, Gaussian model with the highest granularity. This conclusion is additionally strengthened by the five-fold performance improvement over the maximum likelihood estimates. **(II)** High resolution maps (i.e. with small cell sizes) result in higher localization accuracy. However, it is to be assumed that low resolution maps may be equally good, given that the cell separations faithfully separate LOS from NLOS areas, as well as separate differing NLOS cells from each other. **(III)** A performance increase can be obtained by fusing relative positioning information with UWB data, even when the relative positioning data is potentially noisy. Furthermore, by identifying LOS and NLOS cells, targeted collaboration strategies can be designed, which promise to even further improve the localization performance.

**Acknowledgement.** The work presented in this paper was supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 51NF40-111400.

## References

- [1] Fox, D., Burgard, W., Kruppa, H., Thrun, S.: A Probabilistic Approach to Collaborative Multi-Robot Localization. *Autonomous Robots* 8, 325–344 (2000)
- [2] Gonzalez, J., Blanco, J.L., Galindo, C., Ortiz-de Galisteo, A., Fernandez-Madrigal, J.A., Moreno, F.A., Martinez, J.L.: Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach. *Robotics and Autonomous Systems* 57(5), 496–507 (2009)
- [3] Jourdan, D.B., Deyst Jr., J.J., Win, M.Z., Roy, N.: Monte Carlo localization in dense multipath environments using UWB ranging. In: *IEEE International Conference on Ultra-Wideband (ICU)*, pp. 314–319 (2005)
- [4] Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man and Cybernetics* 37(6), 1067–1080 (2007)
- [5] Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J., Martinoli, A.: SwisTrack - A Flexible Open Source Tracking Software for Multi-Agent Systems. In: *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4004–4010 (2008)
- [6] Mourikis, A.I., Roumeliotis, S.I.: Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics* 22(4), 666–681 (2006)
- [7] Prorok, A., Martinoli, A.: A Reciprocal Sampling Algorithm for Lightweight Distributed Multi-Robot Localization. In: *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3241–3247 (2011)
- [8] Prorok, A., Arfire, A., Bahr, A., Farserotu, J.R., Martinoli, A.: Indoor navigation research with the Khepera III mobile robot: An experimental baseline with a case-study on ultra-wideband positioning. In: *International Conference on Indoor Positioning and Indoor Navigation, IPIN* (2010), doi:10.1109/IPIN.2010.5647880
- [9] Prorok, A., Tomé, P., Martinoli, A.: Accommodation of NLOS for Ultra-Wideband TDOA Localization in Single- and Multi-Robot Systems. In: *International Conference on Indoor Positioning and Indoor Navigation, IPIN* (2011), doi:10.1109/IPIN.2011.6071927
- [10] Prorok, A., Bahr, A., Martinoli, A.: Low-Cost Collaborative Localization for Large-Scale Multi-Robot Systems. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4236–4241 (2012)
- [11] Prorok, A., Gonon, L., Martinoli, A.: Online Model Estimation of Ultra-Wideband TDOA Measurements for Mobile Robot Localization. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 807–814 (2012)
- [12] Pugh, J., Raemy, X., Favre, C., Falconi, R., Martinoli, A.: A Fast On-Board Relative Positioning Module for Multi-Robot Systems. *IEEE Transactions on Mechatronics* 14(2), 151–162 (2009)
- [13] Sahinoglu, Z., Gezici, S., Guvenc, I.: *Ultra-wideband Positioning Systems. Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press (2008)



# Cooperative Multi-robot Estimation and Control for Radio Source Localization

Benjamin Charrow, Nathan Michael, and Vijay Kumar

**Abstract.** We develop algorithms for estimation and control that allow a team of robots equipped with range sensors to localize an unknown target in a known but complex environment. We present an experimental model for radio-based time-of-flight range sensors. Adopting a Bayesian approach for estimation, we then develop a control law which maximizes the mutual information between the robot's measurements and their current belief of the target position. We describe experimental results for a robot team localizing a stationary target in several representative indoor environments in which the unknown target is reliably localized with an error well below the typical error for individual measurements.

## 1 Introduction

Having robotic teams that are capable of quickly localizing a target in a variety of environments is beneficial in several different scenarios. Such a team can be used in search and rescue situations where a person or object must be located quickly. Cooperative localization can also facilitate localization of robots within a team towards tasks like cooperative mapping or surveillance. Given these multiple applications, it is reasonable to equip the robots with additional sensors to support localization. In this paper, we focus on the case where each robot is equipped with a range-only RF sensor. These sensors provide limited information about the state of the target, necessitating the development of an active control strategy to localize the target.

There is extensive work on algorithms for localizing stationary nodes using range sensors notably in open field environments by Kantor and Singh [8]. Spletzer and Taylor [19] also examined a multi-node stationary localization

---

Benjamin Charrow · Nathan Michael · Vijay Kumar  
GRASP Lab, University of Pennsylvania,  
3330 Walnut St., Philadelphia, PA 19104  
e-mail: {bcharrow, nmichael, kumar}@seas.upenn.edu

problem for both range and bearing sensors with bounded error. Grocholsky [4] and Stump et al. [20] examined the active control question by designing controllers which maximized the rate of change of the Fisher information matrix to localize a stationary target. However, their work assumed that the belief was Gaussian distributed, which is not the case for typical range measurements.

In our paper, we present a probabilistic approach for estimating the location of a target as well as an information-theoretic control strategy which seeks to maximize the usefulness of future measurements that the team makes in the more general non-Gaussian setting. This avenue of research was pursued by Hoffman and Tomlin [5]. They showed how particle filters can be used in obstacle-free environments with non-linear sensor models to calculate mutual information. We build on their work and extend it to work in non-convex environments. We also improve on their numerical approximation of mutual information yielding more efficient and potentially better control inputs.

Ryan and Hedrick [16] also investigated information-theoretic control and developed a receding horizon controller for a single mobile robot to track a mobile target. They approximate mutual information using a randomized algorithm which does not rely on numerical integration. Our work differs by focusing on real time control inputs for multiple robots.

These approximation methods are necessary, as naïve approaches to maximize mutual information almost immediately lead to significant computational difficulties. The fundamental complexity arises from an integration that must be performed over both the state and measurement spaces. While direct numerical calculations can be used, approximation algorithms for certain distributions exist. In this work we use the approach by Huber et al. [6] for approximating the entropy of Gaussian mixture models.

Our work is also closely related to work by Olson et al. [11] and Djughash et al. [2, 3]. In these works, a single mobile robot localizes itself and the nodes in a sensor network with many nodes using range measurements. Both of their experimental results rely on manually driving the robot throughout the environment. Our work addresses the problem of how robots should move, which is necessary as we have fewer robots generating measurements.

Both our estimation and control strategies are fully centralized and require communication throughout the team. This limitation is not significant as the team's only sensor is RF-based. If the team was in an environment where they could not communicate, they could not gather measurements either.

In the rest of this paper we detail our estimation and control algorithms that enable a team of robots to successfully localize a stationary target in non-convex environments. Our primary focus is on a series of experiments that we designed to test our approach across different indoor environments and a variety of initial conditions. Overall, we are able to repeatedly localize a target with an error between 0.8-1.9m using only two robots equipped with commercially available RF range sensors. A highlight of our work is that

despite the limited information that the team has when they are at any fixed position, they are able to effectively coordinate and use their mobility so that the estimate of the target rapidly converges.

## 2 Technical Approach

In our approach, the robotic team maintains a distribution over possible locations of the target's location using a particle filter. The team constantly seeks to maximize the mutual information between the current estimate of the target's state and expected future measurements. Both the filter and the control directions are computed in a centralized manner.

### 2.1 Measurement Model

We consider the case where the team's only way of sensing the target is through a *range*-only sensor. A standard approach for building these sensors is to use the Time Difference of Arrival (TDoA) of a signal between two nodes. Because of this, any delay in the signal's propagation results in an over-estimate of the distance. Conversely, an overestimate of the signal's speed results in an under-estimation of the distance. We use the nanoPAN 5375 – a TDoA RF sensor which operates in the 2.4GHz spectrum in our experiments and show that it exhibits both positive and negative biases, which are primarily a function of whether or not the sensors are in line of sight (LOS) or non line of sight (NLOS). Our data also show that the magnitude of the bias and variance increase with the true distance between the sensors.

There is significant empirical and theoretical support for treating LOS and NLOS measurements differently [13, 12]. TDoA methods work best when radios are not obstructed by obstacles and have clear LOS conditions. However, when radios do not have LOS they are more likely to be affected by scattering, fading, and self-interference, causing non-trivial positive biases. We model the error of the measurement conditioned on the true state as:

$$p(z | x) = \begin{cases} \mathcal{N}(z; \alpha_0 + r\alpha, r\sigma_L^2) & \text{LOS} \\ \mathcal{N}(z; \beta_0 + r\beta, r\sigma_N^2) & \text{NLOS} \end{cases} \quad (1)$$

where  $r$  is the true distance between the sensors and  $\mathcal{N}(z; \mu, \sigma^2)$  is a normal random variable with mean  $\mu$  and variance  $\sigma^2$ .  $\alpha_0$  and  $\beta_0$  are the biases, while  $\alpha$  and  $\beta$  determine how the biases change with distance. It is straightforward to calculate the maximum likelihood estimate (MLE) of these parameters for this model with labeled data. Throughout this paper, we assume that measurements are conditionally independent of each other given the true state (i.e.  $p(z_1, z_2 | x) = p(z_1 | x)p(z_2 | x)$ ).

Models for TDoA sensors typically use a biased Gaussian with constant or time-varying variance [17, 7], whereas (1) is more similar to those used in

power based range models. We have chosen this approach because, as Patwari et al. noted, the Gaussian model does not perform as well at large distances as the tails of the distribution become heavy [12]. Letting the variance increase with distance ensures that the model handles large deviations from the truth without using a Gaussian mixture model, which would increase the computational complexity of the control as we discuss later.

## 2.2 Estimation

Range measurements are a non-linear function of the target state and can easily lead to non-trivial multi-hypothesis belief distributions as well as rings and crescents. For these reasons, we use a particle filter for the estimation. Formally, let  $\mathbf{x}_t = [x, y]$  be the state of the target at time  $t$  and  $\mathbf{c}_t^i = [x, y]$  be the state of the  $i^{\text{th}}$  member of the team. The full configuration of the team is  $\mathbf{c}_t = [\mathbf{c}_t^1, \dots, \mathbf{c}_t^n]$ . Each robot makes a 1-d range measurement  $z_t^i$  as they move around the environment. Aggregating these measurements produces a vector  $\mathbf{z}_t = [z_t^1, z_t^2, \dots, z_t^n]$ . Where appropriate, we will write  $\mathbf{z}_t(\mathbf{c}_t)$  to emphasize that the measurements depend on the configuration of the team.

The belief at time  $t$  is the distribution of the state conditioned on all measurements up to time  $t$ . A typical Bayesian filter incorporates measurements over time recursively and a particle filter approximates this as a weighted sum of Dirac delta functions [21]:

$$bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \int bel(\mathbf{x}_{t-1}) p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) d\mathbf{x}_{t-1} \approx \sum_j w_j \delta(\mathbf{x}_t - \tilde{\mathbf{x}}_j) \quad (2)$$

where  $\eta$  is a normalization constant,  $\tilde{\mathbf{x}}_j$  is the location of the  $j^{\text{th}}$  particle and  $w_j$  is its weight. While this equation is standard, we wish to emphasize that the approximation is *discrete*. As we show in Sect. 2.3, this enables approximations of mutual information.

Aside from representing complex distributions, particle filters also have the advantage that they do not require each measurement to be classified as LOS or NLOS. Assuming that the robots have a map of the environment, they can use it to determine which of the two distributions in our measurement model (1) to use on a *per particle* basis by seeing if a straight line between the particle and the robot intersects any walls.

While the standard particle filter equations allow for the incorporation of non-linear control inputs,  $\mathbf{u}_t$ , in our scenario the target is stationary. Despite this, in our experiments we injected noise into the system to avoid particle degeneracy problems. Specifically, we perturbed the polar coordinates of each particle in the local frame of the robot that is making the measurement with samples from a zero-mean Gaussian distribution. We achieved the best results by rejecting samples that moved the particle more than a specified distance or caused their LOS condition to the measuring robot to change. We also used a

low variance resampler when the number of effective particles dropped below a certain threshold. All of these techniques can be found in standard books on estimation (e.g. Probabilistic Robotics [21]).

### 2.3 Control

Our control strategy is designed to drive the team so that they obtain measurements which lead to a reduction in the uncertainty of the target estimate. The mutual information between the current belief of the target's state and expected future measurements captures this intuitive notion. The advantage of this approach is that it incorporates the current belief of the target's state along with the measurement model to determine how potential future measurements will impact the state. By design the team will move in directions where their combined measurements will be useful. This is particularly important when using sensors which provide limited information about the state of the target.

Formally, we select the next configuration for the robotic team using the following objective function:

$$\mathbf{c}_{t+1} = \arg \max_{\mathbf{c} \in \mathcal{C}} \mathbf{MI}[\mathbf{x}_t, \mathbf{z}(\mathbf{c})] = \arg \max_{\mathbf{c} \in \mathcal{C}} \mathbf{H}[\mathbf{z}(\mathbf{c})] - \mathbf{H}[\mathbf{z}(\mathbf{c}) \mid \mathbf{x}_t] \quad (3)$$

where  $\mathbf{H}[\mathbf{x}]$ ,  $\mathbf{H}[\mathbf{x} \mid \mathbf{z}]$  and  $\mathbf{MI}[\mathbf{x}, \mathbf{z}]$  are the differential entropy, differential conditional entropy, and mutual information, respectively, as defined by Cover and Thomas [1]. The domain of the optimization problem is the configuration space of the team,  $\mathcal{C}$ . Different configurations of the team will affect the mutual information by affecting both entropies in (3).

#### 2.3.1 Determining Locations

Typical indoor environments are non-convex, meaning we must maximize mutual information over a non-convex set. To do this, we propose searching over a discrete set of configurations of the team. In our experiments, we do this by creating a connectivity graph along with an embedding into the environment. At each time step, robots find candidate locations by performing breadth first search and taking nodes within specified range intervals. Short ranges allow a robot to continue gathering useful measurements where it is, while long ranges enable it explore new locations.

To create the connectivity graph we perform a Delaunay triangulation of the environment and define the incenters of the triangles as nodes. This requires a polygonal representation of the obstacles, but it is straightforward to create these from an occupancy grid map. For edges, we connect nodes from adjacent triangles as well as their transitive closure. Figure 2 shows two examples of this approach, which we used in our experiments.

### 2.3.2 Calculating the Objective

Hoffmann and Tomlin [5] developed an approach for calculating mutual information with particle filters that we use here. In particular, they showed that by using the particle filter’s discrete approximation of the belief, (2), the entropies can be expressed as:

$$\mathbf{H}[\mathbf{z}] \approx - \int_{\mathbf{z}} \left( \sum_i w_i p(\mathbf{z} \mid \mathbf{x} = \tilde{\mathbf{x}}_i) \right) \log \left( \sum_i w_i p(\mathbf{z} \mid \mathbf{x} = \tilde{\mathbf{x}}_i) \right) d\mathbf{z} \quad (4)$$

$$\mathbf{H}[\mathbf{z} \mid \mathbf{x}] \approx - \int_{\mathbf{z}} \sum_i w_i p(\mathbf{z} \mid \mathbf{x} = \tilde{\mathbf{x}}_i) \log p(\mathbf{z} \mid \mathbf{x} = \tilde{\mathbf{x}}_i) d\mathbf{z} \quad (5)$$

Where we have dropped the measurement’s dependence on the configuration of the team for brevity. The approximate equalities are due to the particle filter’s approximation of the belief.

In our work, we further exploit the assumed conditional independence of the measurements given the state when calculating the conditional entropy. By exchanging summation and integration, we can rewrite (5) as  $\mathbf{H}[\mathbf{z}(\mathbf{c}) \mid \mathbf{x}] \approx \sum_i w_i \sum_j \mathbf{H}[z_j(c_j) \mid \mathbf{x} = \tilde{\mathbf{x}}_i]$ . This reduces the conditional entropy to be multiple *separate* integrals over a single measurement space – which can often be done analytically – as opposed to one integral over the joint space of all measurements the team makes.

The entropy of the measurement distribution (4) is harder to compute. However, the particle filter transforms the distribution into a finite dimensional mixture model, which enables new approximation techniques.

### 2.3.3 Approximating Mutual Information

Unfortunately, performing the mutual information calculations in real time for teams with more than 3 or 4 robots is computationally infeasible: evaluating the mutual information of a single configuration requires numerically integrating over the full measurement space to calculate the measurement entropy. Rather than performing numerical integration over subsets of the space [5], we view the measurement distribution  $p(\mathbf{z})$  as a mixture model. The  $i^{\text{th}}$  component is  $p(\mathbf{z}(\mathbf{c}) \mid \mathbf{x} = \tilde{\mathbf{x}}_i)$  with weight  $w_i$ , both of which are determined by the  $i^{\text{th}}$  particle. Because our measurement model is Gaussian, we can use a deterministic approximation algorithm for evaluating the entropy of Gaussian mixture models [6].

The algorithm is based on a Taylor series expansion of the logarithmic term in the integral. This replaces the log term by a sum. Exchanging the order of integration and summation, the entropy can be expressed as a weighted sum of the Gaussians’ central moments. The integrals can be calculated analytically, and only the weighting terms need to be computed online.

We only use the 0<sup>th</sup>-order term in the Taylor series expansion:

$$\mathbf{H}[\mathbf{z}] \approx - \int_{\mathbf{z}} \sum_k w_k \mathcal{N}(\mathbf{z}; \mu_k, \Sigma_k) \log g(\mu_k) d\mathbf{z} = - \sum_k w_k \log g(\mu_k) \quad (6)$$

$g(\mu_k)$  is the likelihood of the mixture model evaluated at the mean of the  $k^{\text{th}}$  component. The computational complexity of this approximation is  $O(n^2l)$  where  $n$  is the number of particles and  $l$  is the number of robots. The time is linear in  $l$  because the conditional independence assumption in the measurement model results in the covariance matrix of the measurements being diagonal. Table 2 summarizes the computational complexity of the entire approach.

This approach also works when the measurement model is itself a mixture of Gaussians. However, this would result in one mixture component for every separate combination of mixture components from all robots, leading to exponential growth. This is partly why we use a Gaussian measurement model with distance dependent variance rather than a Gaussian mixture model.

## 3 Experiments

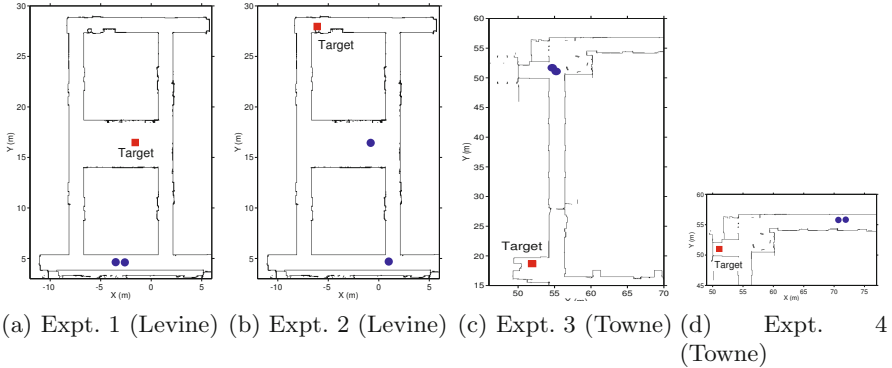
### 3.1 Experimental Design

There are four primary questions that we seek to answer with our experiments: 1) does our measurement model result in an accurate estimate of the target's location? 2) is mutual information an appropriate metric to maximize? 3) does our approximation of mutual information provide reasonable trajectories? 4) are our results repeatable across different environments and with various starting conditions?

We design three separate experiments to answer these questions. All of the experiments have two robots trying to locate a third stationary robot. To assess repeatability, we run 10 independent trials of each experiment. For each trial, we let the robots explore the environment without interacting with them, and only stop them once the filter reaches a stable estimate.

To evaluate the performance of each trial, we calculate the empirical mean of the filter's distribution as well as the volume of its covariance matrix (i.e. its determinant). These statistics are not always an accurate reflection of the filter's performance (e.g. the average of two distinct hypotheses may be far from either hypothesis), but will show whether the filter accurately converges to a single estimate over time.

We use a qualitative approach to evaluate the trajectories of the robots, and manually assess whether or not they are reasonable. As a baseline comparison, in open environments with no prior knowledge of the target's location, it is best to move two range sensors orthogonally to one another as is seen in other approaches [4, 20, 11].



**Fig. 1** Starting configurations of robots. Blue dots show the starting location of the two mobile robots and the red square shows the location of the target.

In Experiment 1, we place two robots within 0.5m of each other and put the target in NLOS conditions approximately 16m away. A good control strategy for this experiment will result in the robots moving in complementary directions rather than staying tightly clustered together.

For Experiment 2, we place the robots far away from each other to see if they still move in complementary directions. The separation also tests whether the measurement model consistently combines measurements from different modalities; if the robots follow good trajectories, they will achieve LOS to the target at different times.

In Experiment 3, we place the robots within 0.5m of each other with the target more than 30m away. Experiment 3 also takes place in a different environment. Experiments 1 and 2 take place in Levine Hall at the University of Pennsylvania, which was constructed primarily in 2003 with modern construction materials – its walls are primarily made up of wood or metal framing with drywall. Experiments 3 and 4 take place in Towne Building which was built in 1906 – its walls are typically made of brick or concrete.

For experiment 4, we again place the robots close to each other with the target 20m along a hallway that has a slight bend. Raycasts from the robots to the target barely intersect a wall, making it ambiguous whether or not the RF signal will exhibit LOS or NLOS behavior. Figure 1 shows the starting location of the target and robots for each experiment.

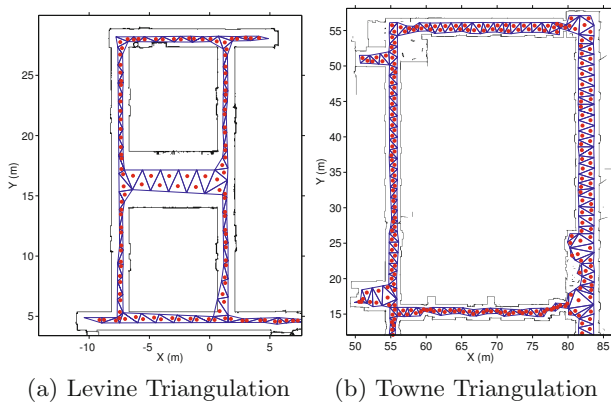
### 3.2 Equipment and Configuration

We use simple differential drive robots equipped with Hokuyo-URG04LX laser scanners and 802.11s wireless mesh cards for communication. The experimental software is developed in C++ and interfaced via Robot



Operating System [14]. The range sensor is commercially available as part of the nanoPAN 5375 Development Kit [10]. The target remains stationary for the duration of each trial while the mobile robots are limited to a maximum speed of  $0.2 \frac{m}{s}$ . The particle filter and mutual information calculations are performed on a laptop with 4GB RAM and an Intel Core Duo processor. To prevent collisions between the mobile robots, we use the Optimal Reciprocal Collision Avoidance algorithm provided in the RVO2 library [18, 15].

For localization, planning, and determining LOS conditions we use a known occupancy grid map of the environment with a resolution of 0.05m. The results of the triangulation algorithm in Sect. 2.3.1 are shown in Fig. 2. Robots consider nodes within 1.0-2.0m or 8.0-9.0m of their current location for the control update, which typically result in 7-12 locations per robot.



**Fig. 2** Graphs for candidate locations. Delaunay triangles are shown along with their incenters which form the vertices of the graph.

We uniformly sample 2500 locations throughout the environment to initialize the particle filter and run a low variance resampler when the proportion of effective particles drops below 0.3. We calculate the MLE of the measurement model's parameters, (1), using a separate dataset gathered in Levine. During the experiments, we double the MLE variances to prevent the filter from prematurely converging without affecting the location of the convergence. Specifically, if  $\sigma_{L,MLE}$  and  $\sigma_{N,MLE}$  are the MLE values of the variance for LOS and NLOS conditions, we set  $\sigma_L = 2\sigma_{L,MLE}$  for LOS and  $\sigma_N = 2\sigma_{N,MLE}$  for NLOS. Parameter values for the experiments are listed in Table 1. We emphasize that the parameters are equivalent for *both* the Towne and Levine experiments.

**Table 1** Measurement Model Parameters

	LOS NLOS	
$\alpha_0, \beta_0$	2.51	-1.34
$\alpha, \beta$	0.20	-0.24
$\sigma_L^2, \sigma_N^2$	7.00	14.01

**Table 2** Computational complexity of control law (3) with  $n$  particles,  $l$  robots, and  $d$  potential locations per robot

Task	Cost
Single Evaluation	$O(n^2l)$
Solving Objective	$O(n^2ld^l)$

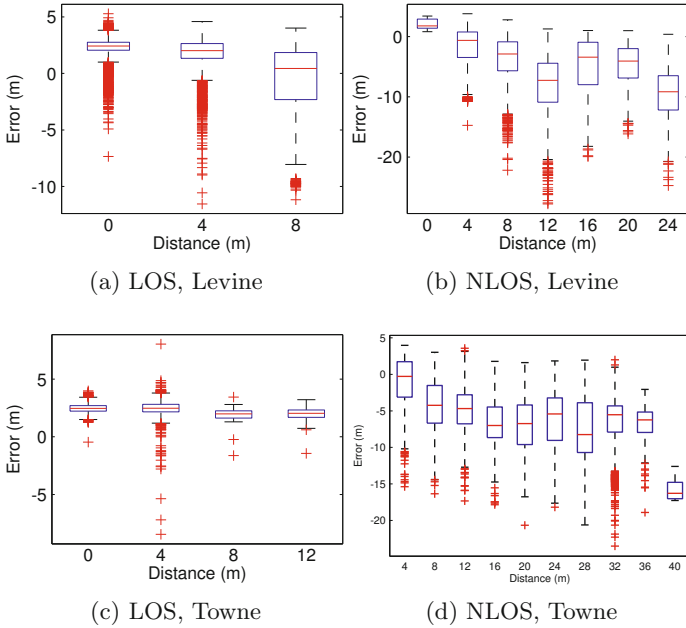
### 3.3 Results

Figure 3 shows the error of the nanoPAN 5375’s measurements as a function of distance. At short distances in LOS conditions the nanoPAN provides consistent measurements with an error larger than 2.0m. In NLOS conditions there is significant variability in both the mean and variance of the measurements as the distance between source and receiver increases. This data helps justify our choice of measurement model as it is clear that LOS and NLOS conditions have different sensor measurement biases and variances.

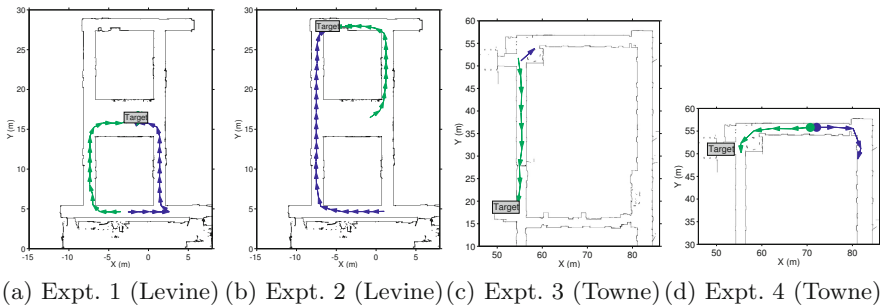
Overall, our approximation of mutual information resulted in good trajectories for the robots. Figure 4 shows that the robots moved to gain complementary measurements; they moved orthogonally to one another when possible, and once they had localized the source, they maintained LOS. Figure 5 shows this process in more detail. Initially, both robots moved away from each other. Next, the robot on the left moved up the vertical hallway, while the other robot moved laterally; an orthogonal movement pattern. Once measurements along the horizontal hallway were no longer helpful due to the symmetry of the distribution, Fig. 5b, both robots moved up the vertical hallway, which caused the filter to converge. We stress that these behaviors arose organically from our objective.

Figure 5 also serves as a good example of the types of distributions that typical parametric approaches that assume a unimodal distribution cannot reliably track as there are clearly multiple equally valid hypotheses.

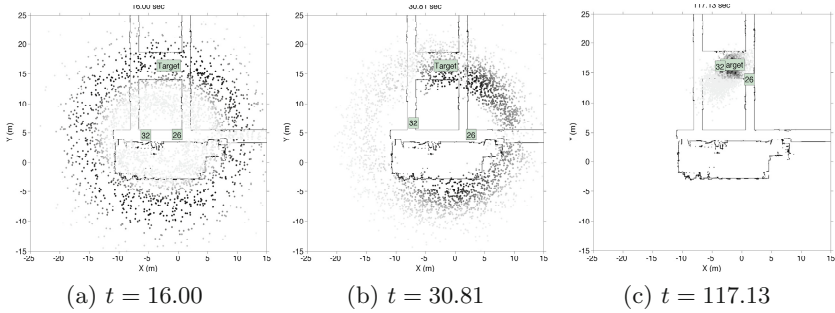
Figures 6 and 7 show the mean and covariance of the estimate over time across all trials and experiments. Except for Experiment 4, the mean converged to the true state of the target, ultimately surpassing the baseline accuracy of an individual measurement in LOS conditions. Table 3 provides the resulting root mean square error (RMSE) of the converged filter. The slightly increased error in Experiment 3 suggests that the parameters for the measurement model are not exactly the same as in Levine. However, the overall error is still low. The fact that the covariance decreases and ends in the range  $[-3, 0]$  on a natural logarithmic scale, shows that filter consistently converges to a single hypothesis. For reference, the natural logarithm of the determinant of the identity matrix (i.e. 1 meter variance in only  $x$  and  $y$ ) has



**Fig. 3** Error of nanoPAN range measurements in LOS and NLOS conditions. The subfigures show standard box-plots with outliers marked as '+'s. Each plot contains approximately 10,000 data points. NLOS conditions are significantly noisier than LOS conditions with statistics that vary with distance.



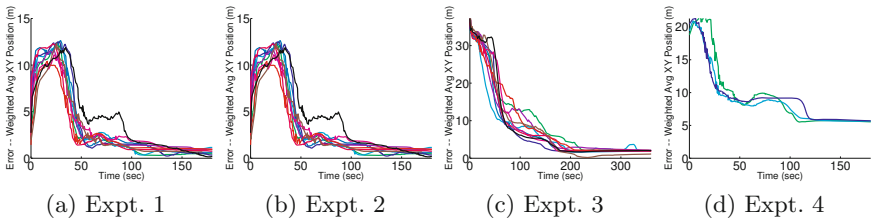
**Fig. 4** Trajectories from maximizing mutual information. Each subfigure is a typical trajectory from each experiment. The robots move to gather complementary measurements, causing the estimate of the target to converge.



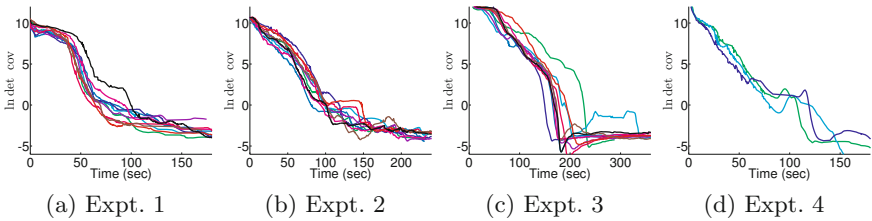
**Fig. 5** Evolution of the particle filter and the robots’ movement in Experiment 1. Darker particles have higher weight; a robot’s position is shown by a box. a) Early range measurements cause a ring. b) Both robots move laterally, generating two hypotheses. c) The robots move up, and the filter converges.

**Table 3** RMSE of converged filter across trials

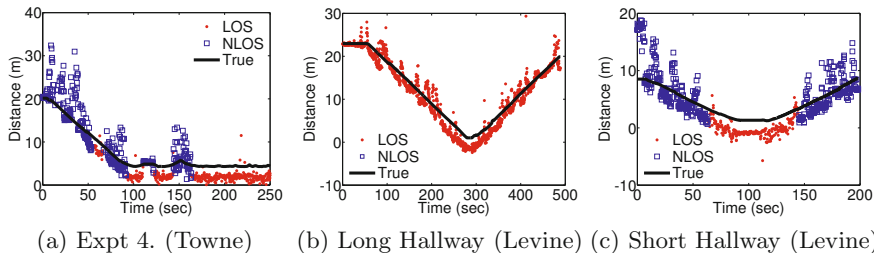
Expt. #	Time (s)	Mean RMSE (m)	Std. Dev. of RMSE (m)
1	140-160	0.83	0.23
2	230-250	0.87	0.46
3	280-300	1.94	0.41
4	130-150	5.79	0.11



**Fig. 6** Distance from weighted average of particles to target location. Each plot shows 10 trials. Except for Experiment 4, the filter converges with an error below 2m, which is below the median error of the range measurements.



**Fig. 7** Natural logarithm of the determinant of covariance. The spread of the particles decreases over time, showing the filter converges to a single estimate.



**Fig. 8** RF vs. Geometric LOS. These subfigures show measurements labeled as LOS or NLOS according to a raycast from the robot to the target location. Geometric LOS does not always accurately predict the negative bias and comparatively low variance of RF LOS measurements.

a value of 0. The consistent trends across all trials and experiments demonstrate that the fundamental approach is robust to various starting conditions and changes in the sensor across environments.

Experiment 4 is the exception to this general trend. In all 3 trials, the filter converged to a single estimate that was 5.5m away from the target’s true location. The essential problem is that at the start of the experiment, raycasts in the environment between a robot and particles near the target intersect a wall, but the RF interference is low. As Figure 8a shows, the measurements exhibit negative bias and relatively low variance which are more consistent with LOS conditions. This highlights the difference between geometric LOS and RF LOS. It is worth noting that the robots still follow a reasonable trajectory; this experiment is *not* a failure of the control.

Figure 8 shows two additional datasets from Levine which exhibit the same phenomena. Figure 8b shows a dataset gathered by having a robot drive up and down a long hallway with a target at one end. The robot and target had geometric LOS for the entire time. Figure 8c shows a dataset where a robot drove along a short hallway with a target just around the corner. The robot only had geometric LOS at one end of the hallway. In both of these experiments, geometric LOS was not always equivalent to RF LOS, highlighting the need for a different approach to classifying measurements.

## 4 Future Work

We are currently exploring various methods for classifying measurements as LOS / NLOS. Our preliminary results suggest that latent variable methods like Hidden Markov Models (HMM) will not have the same problems as our geometric method. Morelli et al. [9] previously used a similar approach to localize a mobile beacon with more than 3 static nodes.

We are also interested in determining how well our approximation of mutual information scales, both as the number of targets increases and as the

size of the robotic team increases. Longer term, we would like to extend our methods to track the state of mobile targets.

## 5 Conclusion

In this paper we described an active control strategy which leverages the current estimate of the target's state and knowledge of the sensor model to direct a team of robots towards locations where they will make informative measurements. We showed how approximations for the entropy of Gaussian mixture models can be used to calculate the necessary control inputs in real time. Most importantly, we presented extensive experimental results where a pair of robots successfully and repeatedly localize a target in non-convex indoor environments using commercially available RF-based range sensors. An implementation of our approach is available from the first author's website at <http://www.seas.upenn.edu/~bcharrow>

**Acknowledgements.** We gratefully acknowledge the support of AFOSR Grant FA9550-10-1-0567, ONR Grant N00014-07-1-0829, and ARL Grant W911NF-08-2-0004. The first author was supported by a NDSEG fellowship from the Department of Defense.

## References

- [1] Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Online Library (2004)
- [2] Djugash, J., Singh, S.: Motion-aided network slam. In: Proc. of the Intl. Sym. on Exp. Robot., New Dehli and Agra, India (December 2010)
- [3] Djugash, J., Singh, S., Kantor, G., Zhang, W.: Range-only SLAM for robots operating cooperatively with sensor networks. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Orlando, USA, pp. 2078–2084 (2006)
- [4] Grocholsky, B.: Information-theoretic control of multiple sensor platforms. PhD thesis, University of Sydney, Sydney, Australia (2002)
- [5] Hoffmann, G.M., Tomlin, C.J.: Mobile sensor network control using mutual information methods and particle filters. *IEEE Trans. Autom. Control* 55(1), 32–47 (2010)
- [6] Huber, M.F., Bailey, T., Durrant-Whyte, H., Hanebeck, U.D.: On entropy approximation for gaussian mixture random vectors. In: Multisensor Fusion and Integration for Intelligent Systems, Seoul, Korea, pp. 181–188 (2008)
- [7] Jourdan, D., Deyst, J., Win, M., Roy, N.: Monte carlo localization in dense multipath environments using UWB ranging. In: IEEE Intl. Conf. on Ultra-Wideband, Zurich, Switzerland, pp. 314–319 (September 2005)
- [8] Kantor, G., Singh, S.: Preliminary results in range-only localization and mapping. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Washington, D.C., vol. 2, pp. 1818–1823 (2002)

- [9] Morelli, C., Nicole, M., Rampa, V., Spagnolini, U.: Hidden markov models for radio localization in mixed LOS/NLOS conditions. *IEEE Trans. Signal Process.* 55(4), 1525–1542 (2007)
- [10] nanoPAN 5375 Development Kit (February 2012), [http://www.nanotron.com/EN/PR\\_tools.php#03](http://www.nanotron.com/EN/PR_tools.php#03)
- [11] Olson, E., Leonard, J.J., Teller, S.: Robust range-only beacon localization. *IEEE J. Oceanic Eng.* 31(4), 949–958 (2006)
- [12] Patwari, N., Ash, J.N., Kyperountas, S., Hero III, A.O., Moses, R.L., Correal, N.S.: Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Process. Mag.* 22(4), 54–69 (2005)
- [13] Rappaport, T.S.: *Wireless Communications: Principles and Practice*. Prentice Hall (1996)
- [14] Robot Operating System (February 2012), <http://www.ros.org/wiki/>
- [15] RVO2 Library (February 2012), <http://gamma.cs.unc.edu/RVO2/>
- [16] Ryan, A., Hedrick, J.K.: Particle filter based information-theoretic active sensing. *Robotics and Autonomous Systems* 58(5), 574–584 (2010)
- [17] Sadler, B.M., Liu, N., Xu, Z., Kozick, R.: Range-based geolocation in fading environments. In: *Allerton Conf. on Comm., Control, and Comput.*, Allerton House, USA, pp. 15–20 (2008)
- [18] Snape, J., van den Berg, J., Guy, S.J., Manocha, D.: Smooth and collision-free navigation for multiple robots under differential-drive constraints. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Anchorage, USA, pp. 4584–4589 (2010)
- [19] Spletzer, J., Taylor, C.J.: A bounded uncertainty approach to multi-robot localization. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Las Vegas, USA, vol. 2, pp. 1258–1265 (October 2003)
- [20] Stump, E., Kumar, V., Grocholsky, B., Shiroma, P.M.: Control for localization of targets using range-only sensors. *Intl. J. Robot. Research* 28(6), 743–757 (2009)
- [21] Thrun, S., Burgard, W., Fox, D.: *Probabilistic Robotics*. MIT Press (2008)

# Real-Time Optimized Rendezvous on Nonholonomic Resource-Constrained Robots

Sven Gowal and Alcherio Martinoli

**Abstract.** In this work, we consider a group of differential-wheeled robots endowed with noisy relative positioning capabilities. We develop a decentralized approach based on a receding horizon controller to generate, in real-time, trajectories that guarantee the convergence of our robots to a common location (i.e. rendezvous). Our receding horizon controller is tailored around two numerical optimization methods: the hybrid-state A\* and trust-region algorithms. To validate both methods and test their robustness to computational delays, we perform exhaustive experiments on a team of four real mobile robots equipped with relative positioning hardware.

## 1 Introduction

Since the 1960s, *consensus* problems have puzzled the minds of many researchers in various fields, ranging from computer science to information aggregation [23]. The term consensus describes the problem of reaching an agreement amongst different agents on a certain quantity or state. These agents can share information about their state either by means of communication or observations. In a network of robots, solving the consensus problem on the position of each agents refers to the task of controlling them as to reach a common *rendezvous* point. The ability to meet or to rendezvous has indeed many practical applications such as formation control [11], flocking [7], attitude alignment [25] or cooperative aerial surveillance [1]. Additionally, although this paper specifically addresses the rendezvous of differential-wheeled robots, its general concept may be applied the wider range of consensus problems.

---

Sven Gowal · Alcherio Martinoli  
Distributed Intelligent Systems and Algorithms Laboratory,  
School of Architecture, Civil and Environmental Engineering,  
École Polytechnique Fédérale de Lausanne  
e-mail: {svenadrian.gowal, alcherio.martinoli}@epfl.ch



## 1.1 Related Work

Solving the rendezvous with nonholonomic agents is complex, and proving the convergence property can be difficult. Many works employ *feedback linearization* to design relaxed control laws that recreate the holonomic properties [16, 26]; others create algorithms that are very specific to their application needs [6, 5]; but all of them rely on deterministic assumptions both in terms of actuation and sensing. Our previous work [13] incorporates insights from the *probabilistic consensus* problem [9] to guarantee that differential-wheeled robots can rendezvous under noisy measurements. However, this approach and all prior approaches to solve the rendezvous problem on nonholonomic mobile robots rely heavily on strict time-invariant controllers that yield poor trajectories without consideration to neither actuation constraints nor the energy spent.

On another front, a great body of literature starting with Meschler [19] in 1963 focuses on the optimization of the rendezvous maneuver, and although efforts to decentralize the optimization approach using communication between agents have been made [18], many works remain centralized [20, 4] and thus need global knowledge of the system. We can also observe that most work, including [18], use a pre-defined cost function and leave no design choices to the user. To tackle the problem of decentralization with an arbitrary user-defined metric, we rely on a receding horizon controller (RHC) [12]. This RHC needs to run in real-time on our platform, the Khepera III robot [24] (shown on Fig. 1(a)) equipped with an Intel XScale PXA-270 running at 624MHz without floating-point unit.

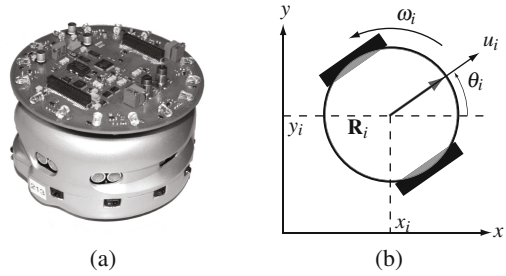
In particular, we use two distinct optimization strategies (within our RHC) that provide real-time capabilities to resource-constrained robots: (i) the hybrid-state A\* algorithm [8] – an optimization strategy based on the A\* search algorithm that generates quickly feasible trajectories for a wide range of cost functions, (ii) a subspace conjugate gradient trust-region method [2, 3] – a numerical optimization method that takes advantage of the differential flatness property of our robots. We compare both strategies in terms of performance and computational requirements. The purpose of this work is then twofold: first, to experimentally verify the convergence of our mobile robot team using our decentralized approach; second, to compare its efficiency with that of a centralized equivalent. We note that, to date, no contribution has addressed the generation of real-time, optimal rendezvous maneuvers on mobile robots performing noisy positioning observations — neither from a theoretical nor from an experimental point of view. In this work, we focus on the experimental aspect (the theory is covered in more depth in another concurrent publication [14]).

## 1.2 Problem Statement

We have a team of  $N$  differential-wheeled robots  $\mathbf{R}_1, \dots, \mathbf{R}_N$  driven by the kinematic equations:

$$\begin{cases} \dot{x}_i = u_i \cos \theta_i \\ \dot{y}_i = u_i \sin \theta_i \\ \dot{\theta}_i = \omega_i \end{cases}, \quad (1)$$

**Fig. 1** (a) A Khepera III robot with a range and bearing module attached. (b) The kinematic model of a differential-wheeled robot  $\mathbf{R}_i$ .



where  $\mathbf{u}_i = [u_i, \omega_i]^T$  is the vector of control inputs, with  $u_i$  the linear translational speed and  $\omega_i$  the rotational speed, and the vector  $\mathbf{x}_i = [x_i, y_i, \theta_i]^T$  defines the absolute pose or state of the robot  $\mathbf{R}_i$ , as shown on Fig. 1(b).

A robot  $\mathbf{R}_i$  has a set of neighbors  $\mathcal{N}_i$  containing all robots  $\mathbf{R}_j$  such that it can measure the range  $e_{ij}$  and bearing  $\alpha_{ij}$  to them. Its measurements are affected by noise such that each observation  $z_{ij}(t)$  of  $\mathbf{R}_j$  at time  $t$  is defined by

$$z_{ij}(t) = \begin{bmatrix} \tilde{e}_{ij}(t) \\ \tilde{\alpha}_{ij}(t) \end{bmatrix} = \begin{bmatrix} e_{ij}(t) \\ \alpha_{ij}(t) \end{bmatrix} + \boldsymbol{\varepsilon}_z, \quad (2)$$

where  $\boldsymbol{\varepsilon}_z$  is a random noise vector.

Our goal will be to drive all robots to the same meeting point. For each robot  $\mathbf{R}_i$ , this *rendezvous maneuver* should be performed *optimally* in *real-time* under a local user-defined metric  $\mathcal{J}_i(\mathbf{u}_i)$  which should only depend on values directly measurable (either through sensors or communication) or calculable by each individual robot  $\mathbf{R}_i$ . Without loss of generality, throughout this paper, we will use the Bolza form  $\mathcal{J}_i(\cdot) = \int L_i(\cdot) dt + V_i(\cdot)$  where  $L_i(\cdot)$  is a cost rate and  $V_i(\cdot)$  is a terminal cost (also called *salvage term*).

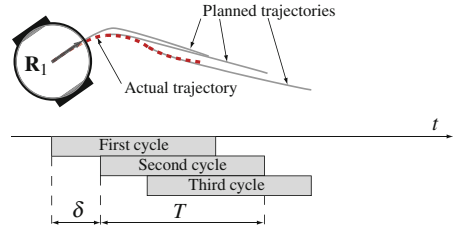
## 2 Technical Approach

In this section, we explain in brevity how RHC can guarantee rendezvous with the addition of optimization constraints, and how to perform each optimization cycle, on-board, in real-time, using, on one hand, the hybrid-state A\* algorithm and, on the other, a subspace conjugate gradient trust-region method. Additionally, we introduce a closed-loop control that follows the resulting RHC trajectories.

### 2.1 Decentralized Receding Horizon Control

To solve our optimization problem (i.e., minimizing  $\mathcal{J}_i(\cdot)$ ) whilst guaranteeing the rendezvous, as seen in Section 1.2), we will rely on RHC. RHC carries many names such as model predictive control (MPC) or real-time optimization (RTO). It is an advanced method, widely used in industry, that has the ability to use the available information on the system at hand to control it optimally under a user-defined cost.

**Fig. 2** Receding horizon trajectories:  $\mathbf{R}_1$  plans an initial trajectory for the next  $T$  seconds and executes that trajectory *blindly* for the first  $\delta$  seconds; at that point,  $\mathbf{R}_1$  plans a new trajectory (and so on)



Although its requirements in terms of computing power are high [21], it has found many successful applications, in particular when the underlying system to control has slow dynamics (i.e., in the order of minutes or seconds). The recent advances in computing power have in part alleviated this issue, but RHC reaches its limits when the underlying system is nonlinear, changing fast and has to run on a simple mobile platform.

RHC is an optimization-based control that uses online, optimal trajectory generation. The general idea is to plan a feasible and sub-optimal trajectory over a finite time  $T$  horizon and control the system (i.e. the robots) to follow this trajectory over a sampling time  $\delta$  ( $0 < \delta \leq T$ ). After  $\delta$  seconds, a new trajectory is recomputed from the current position until time  $\delta + T$  and this trajectory is again followed until time  $2\delta$ . This cycle is repeated until the goal is reached. We will denote such a receding horizon control with the symbol  $\mathcal{RH}(T, \delta)$ . This process is schematized in Fig. 2, where robot  $\mathbf{R}_1$  plans during three cycles three trajectories that are tracked sequentially.

**Theorem 1.** *Given a symmetric and connected group of  $N$  differential-wheeled robots  $\mathbf{R}_1, \dots, \mathbf{R}_N$ , the decentralized receding horizon control  $\mathcal{RH}(T_i, \delta_i)$ , with  $T_i > 0$  and  $0 < \delta_i \leq T_i$ , that solves the following optimization problem on each robot  $\mathbf{R}_i$  at time  $\tau$ :*

$$\text{minimize } \mathcal{J}_i(\mathbf{u}_i) = \int_{\tau}^{\tau+T} L_i(t, \mathbf{x}_i, \hat{\mathbf{x}}_i, \mathbf{u}_i) dt + V_i(\tau + T, \mathbf{x}_i, \hat{\mathbf{x}}_i, \mathbf{u}_i) \quad (3)$$

$$\text{subject to } \text{Eq. 1, } \mathbf{u}_i \in \mathcal{U}_i, \mathbf{x}_i \in \mathcal{X}_i \quad (4)$$

$$\text{such that } \exists k_{ij} = k_{ji} > 0 \text{ satisfying } \mathbf{u}_i = \sum_{\mathbf{R}_j \in \mathcal{N}_i} k_{ij} \hat{\mathbf{x}}_{ij} \quad (5)$$

$$\exists t \geq \tau \text{ satisfying } \omega_i(t) \neq 0, \quad (6)$$

where  $\mathcal{X}_i$  and  $\mathcal{U}_i$  are user-defined admissible sets, drives the group almost surely to a common rendezvous point if  $\hat{\mathbf{x}}_i(t) = \{\hat{\mathbf{x}}_{ij}(t) = [\hat{x}_{ij}(t), \hat{y}_{ij}(t)]^T | \mathbf{R}_j \in \mathcal{N}_i\}$  and the estimation  $\hat{x}_{ij}(t)$  of  $x_{ij}(t) = e_{ij} \cos \alpha_{ij}$  is unbiased in the time interval  $t \in [\tau, \tau + \delta_i]$ .

*Proof.* The proof is omitted for conciseness, but its complete derivation is available in [14]. □

*Remark 1.* As we will see in Section 3, the constraints (5) and (6) can in practice be ignored when an adequate salvage term  $V_i(\cdot)$  is used. In particular, it is sufficient to penalize inter-robot distances that increase.

*Remark 2.* Theorem 1 assumes for each robot  $\mathbf{R}_i$  the presence of a prediction function  $\hat{\mathbf{x}}_i(t)$  capable of estimating the position of neighboring robots. As explained in [14], this function can be implemented using an extended Kalman filter based on the observations made through the relative positioning hardware.

## 2.2 Cost Function

To ease our discussion on the algorithmic details, we describe first the cost function used in our experiments. Given a continuous trajectory for next  $T$  seconds, we discretize it by splitting it into  $N$  linear segments of  $\Delta t = T/N$  seconds each, thus generating a sequence of  $N + 1$  vertices  $\mathbf{p}_i \in \mathbb{R}^2$  with  $i \in \{0, \dots, N\}$ . Additionally, we assume that there are  $N_o$  obstacles denoted  $\mathcal{O}_j$  with  $j \in \{1, \dots, N_o\}$ . Each obstacle has a position  $\mathbf{o}_i^{(j)} \in \mathbb{R}^2$  at time  $i\Delta t$  and an associated uncertainty  $R_i^{(j)} \in \mathbb{R}^{2 \times 2}$ . We denote by  $\Delta \mathbf{p}_i = \mathbf{p}_i - \mathbf{p}_{i-1}$  the displacement vector at a vertex and by  $\mathbf{p}_f$  the final position that the trajectory aims to reach. Our cost function is then:

$$f(\mathbf{p}_{0\dots N}) = \underbrace{w_s \sum_{i=1}^{N-1} (\Delta \mathbf{p}_{i+1} - \Delta \mathbf{p}_i)^\top (\Delta \mathbf{p}_{i+1} - \Delta \mathbf{p}_i)}_{f_1} + \underbrace{w_e \sum_{i=1}^N \|\Delta \mathbf{p}_i\|_2^2}_{f_2} + \underbrace{w_o \sum_{i=0}^N \sum_{j=1}^{N_o} \Phi(\mathbf{p}_i; \mathbf{o}_i^{(j)}, R_i^{(j)})}_{f_3} + \underbrace{w_f \|\mathbf{p}_f - \mathbf{p}_N\|_2^2}_{f_4}, \quad (7)$$

where  $w_s, w_e, w_f, w_o$  are positive weights and  $\Phi(x; \mu, \Sigma)$  is the multi-variate normal probability density function with mean  $\mu$  and covariance  $\Sigma$ .

The first term  $f_1$  of the cost function forces the trajectory to be smooth: the forward acceleration and rotational speed should be small. The second term  $f_2$  penalizes fast motion and ensures that minimal energy to spend in actuation. The third term  $f_3$  guides the trajectory away from obstacles and corresponds roughly to a scaled probability of hitting any of them. The fourth term  $f_4$  steers the trajectory towards a goal position by penalizing an excessive distance to it. The first three terms correspond to the sum of all cost rates over the trajectory, whereas the last term is the salvage term. To ensure the rendezvous in practice, it is enough to set the goal position  $\mathbf{p}_f$  to the estimated center of mass of all neighboring robots at time  $T$ ,

$$\mathbf{p}_f = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{R}_j \in \mathcal{N}_i} \hat{\mathbf{x}}_{ij}(T). \quad (8)$$

## 2.3 Optimization Strategies

The sampling time  $\delta$  is related to the computational time required by the optimization of Eq. (3) [21]. Hence, it is important that this optimization takes as little time

as possible (to guarantee, in practice, the unbiasedness of the estimator of  $\hat{x}_{ij}$ ). In this section, we provide two alternatives capable of efficient real-time optimization.

### 2.3.1 Hybrid-State A\*

The first alternative, explained in [8], is a continuous optimization method derived from a discrete heuristic search method, the A\* search algorithm. The general idea is to discretize into cells the three-dimensional search space  $\langle x_i, y_i, \theta_i \rangle$  representing the robot's state. Whereas A\* explores the center of those discrete cells and generates paths that may not be feasible with respect to the kinematic constraints of Eq. (1), hybrid-state A\* associates with each cell a five-dimensional continuous state  $\langle x_i, y_i, \theta_i, u_i, \omega_i \rangle$ . Hence the transitions from a cell to the next may change according to the stored continuous state. To determine those transitions, we simply discretize the action that the robot can take during the next  $\Delta t$  seconds and perform an Euler integration of the kinematic equations. In particular, we allow the robot to either keep, increase, or decrease its forward or rotational speed by a constant increment  $\Delta u$  or  $\Delta \omega$  respectively. It is clear that hybrid-state A\* is not guaranteed to find the minimal-cost solution because of the discretization of controls and time, as well as the pruning of all but one continuous-state branches that enter a cell. Finally, to use hybrid-state A\* with RHC, we stop the search when the number of cells explored on the current branch reaches the number of points  $N + 1$  required by the trajectory. We note that although hybrid-state A\* is memory hungry, it will always generate feasible trajectories and can be easily modified to include dynamics and additional constraints with little overhead in terms of computational time.

For completeness, we show through Algorithm 1 the complete routine, where  $g(c)$  represents the real cost of the current path from the starting cell to cell  $c$ ,  $h(s, \mathbf{p})$  is the heuristic cost to reach position  $\mathbf{p}$  from a continuous state  $s$  and  $s(c)$  is the continuous state associated with cell  $c$ . Note that the cost  $g(c)$  can be computed by adding up the first three terms of our cost function ( $f_1 + f_2 + f_3$ ) until cell  $c$  on the current branch and the heuristic  $h(c, \mathbf{p})$  is the last term of this same cost function. In the context of our experimental test-bed, we observe that this algorithm can easily make use of fixed-point arithmetic as all variable ranges are known a priori. Combined with a proper implementation (i.e., efficient priority queue), we obtain a procedure on our platform, the Khepera III robot. In the experiments of Section 3, the cell discretization is done by a  $64 \times 64 \times 52$  grid on an area of  $2\text{m} \times 2\text{m} \times 360^\circ$  centered around the robot. The speed increments  $\Delta u$  and  $\Delta \omega$  are set to 0.125m/s and 1.5rad/s respectively and  $\Delta t$  is set to 0.1s. These values are selected to reach the best compromise between optimality and computational/memory requirements (the overall memory usage is 14.8 MB which can easily fit on-board the Khepera III).

### 2.3.2 Subspace Conjugate Gradient Trust-Region

This second alternative is an efficient non-convex optimization method that uses a preconditioned conjugate gradient to define a two-dimensional subspace on which a trust-region method is applied.

**Algorithm 1.** Hybrid-state-A<sup>\*</sup>( $x_i, y_i, \theta_i, u_i, \omega_i, \mathbf{p}_f$ )

---

```

1: closedSet  $\leftarrow \emptyset$ 
2: start  $\leftarrow$  getCell( $x_i, y_i, \theta_i$ )
3: s(start)  $\leftarrow \langle x_i, y_i, \theta_i, u_i, \omega_i \rangle$ 
4: g(start)  $\leftarrow 0$ 
5: openSet  $\leftarrow$  {start}
6: while openSet  $\neq \emptyset$  do
7:   cell  $\leftarrow$  argmin $_{c \in \text{openSet}} g(c) + h(s(c), \mathbf{p}_f)$ 
8:   if isGoal(cell) or depth(cell) =  $\lceil T/\Delta t \rceil$  then
9:     return generatePathTo(cell)
10:  end if
11:  openSet  $\leftarrow$  openSet  $\setminus$  cell
12:  closedSet  $\leftarrow$  closedSet  $\cup$  cell
13:   $\langle x, y, \theta, u, \omega \rangle \leftarrow s(\text{cell})$ 
14:  for all  $[u', \omega'] \in \{[u, \omega] \pm [\Delta u, \Delta \omega]\}$  do
15:     $\langle x', y', \theta' \rangle \leftarrow$  eulerIntegration( $[u', \omega'], \langle x, y, \theta \rangle, \Delta t$ )
16:    dest  $\leftarrow$  getCell( $x', y', \theta'$ )
17:    if dest = cell then
18:      continue
19:    end if
20:    newCost = g(cell) + edgeCost(cell,  $\langle x', y', \theta', u', \omega' \rangle$ )
21:    if newCost +  $h(\langle x', y', \theta', u', \omega' \rangle, \mathbf{p}_f) > g(\text{dest}) + h(s(\text{dest}), \mathbf{p}_f)$  then
22:      continue
23:    end if
24:    openSet  $\leftarrow$  openSet  $\cup$  dest
25:    closedSet  $\leftarrow$  closedSet  $\setminus$  dest
26:    s(dest)  $\leftarrow \langle x', y', \theta', u', \omega' \rangle$ 
27:    g(dest)  $\leftarrow$  newCost
28:  end for
29:  return trajectory impossible
30: end while

```

---

Let us consider a function  $f: \mathbb{R}^n \mapsto \mathbb{R}$ , which we want to minimize. We currently have an estimate  $x$  of the solution, which we wish to improve. The basic idea behind the trust-region approach is to approximate the function  $f$  with a function  $q$  reflecting the behavior of  $f$  in a neighborhood  $\Omega$  around the point  $x$ . This neighborhood is the trust-region. Hence the problem is to find a step  $s$  that minimizes  $q: \min_s \{q(s) | s \in \Omega\}$ . If the vector  $x + s$  is a better estimate of the solution (i.e.,  $f(x + s) < f(x)$ ),  $x$  is set to  $x + s$ ; otherwise, it is unchanged and the trust-region is shrunk. In practice, the approximate function  $q$  is defined by the first two terms of the Taylor expansion of  $f$  around  $x$  and the trust-region is often circular. The trust-region step then becomes

$$\min_s \left\{ \frac{1}{2} s^T H s + g^T s \quad | \quad \|s\|_2 \leq \Delta \right\}, \quad (9)$$

where  $g$  and  $H$  are the gradient and Hessian of  $f$  respectively, and  $\Delta$  is positive. Good algorithms to solve Eq. (9) based on the eigenvalues of  $H$  exist [22]. However, they become inefficient when  $H$  becomes large. Hence, a good heuristic is to reduce the original problem into a two-dimensional subspace spanned by an approximate Newton direction (given in our case by a preconditioned conjugate gradient method) and the gradient direction.

The method of conjugate gradient (CG) [15] is an effective way to iteratively solve large-scale linear equations such as  $Hv = -g$  (note that, here  $v$  is the Newton direction) without calculating the inverse of  $H$ . Using a preconditioned variant (PCG) allows for faster convergence by altering the original problem to  $M^{-1}Hv = -M^{-1}g$ , where  $M$  is called the preconditioner. Finally, the only costly operation that PCG needs to perform is the multiplication of  $H$  with a vector. Thus, PCG is very efficient when  $H$  is sparse. If the number of points is small (i.e.,  $N < 50$ ), PCG can be replaced with Newton's method for greater efficiency; but Newton's method will need more memory as the inverse of the Hessian needs to be stored.

For a fast implementation, it is important that the function  $f$  be twice differentiable and that both an analytical gradient and Hessian can be computed. In our case, for the gradient, we have

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{p}_i} &= w_s(2\Delta \mathbf{p}_{i+2} - 6\Delta \mathbf{p}_{i+1} + 6\Delta \mathbf{p}_i - 2\Delta \mathbf{p}_{i-1}) \\ &\quad - w_e(2\Delta \mathbf{p}_{i+1} - 2\Delta \mathbf{p}_i) \\ &\quad + w_o \sum_{j=1}^{N_o} (R_i^{(j)})^{-1} \Phi(\mathbf{p}_i; \mathbf{o}_i^{(j)}, R_i^{(j)}) (\mathbf{o}_i^{(j)} - \mathbf{p}_i) \\ &\quad + w_f \mathbf{1}_{i=N} (\mathbf{p}_f - \mathbf{p}_N), \end{aligned} \quad (10)$$

where  $\mathbf{1}_A$  is the indicator function of  $A$ . The Hessian is then simply the sum of two sparse matrices: a constant banded matrix  $H_1$  representing the first, second and fourth term of  $f$  and a block diagonal matrix  $H_2$  composed of  $2 \times 2$  blocks  $B_0, \dots, B_N$  (if we interleave the coordinates of every point  $\mathbf{p}_i$ ), with

$$B_i = \frac{\partial^2 f_3}{\partial \mathbf{p}_i^2} = \left( R^{-1} (\mathbf{p}_i - \mathbf{o}_i^{(j)}) \cdot (\mathbf{p}_i - \mathbf{o}_i^{(j)})^\top - I \right) R^{-1} \Phi(\mathbf{p}_i; \mathbf{o}_i^{(j)}, R), \quad (11)$$

where  $R$  means  $R_i^{(j)}$ . If no collisions are possible (i.e.,  $H_2 = \mathbf{0}$ ), it is beneficial to use Newton's method to minimize  $f$  (if memory allows). Indeed the function  $f$  becomes quadratic and Newton's method converges in one iteration. Also, as  $H_1$  is constant, its inverse only needs to be calculated once.

Although, less hungry than hybrid-state  $A^*$  in terms of memory, the subspace conjugate gradient trust-region method (which we denote from hereon as PCG-TR) may not generate a feasible trajectory. However, our specific choice of the cost function  $f$ , which penalizes non-smooth trajectories, mitigates this issue. Additionally, it can be shown that differential-wheeled robots are differentially flat and thus can

follow a *sufficiently* smooth trajectory. Algorithm 2 shows the complete routine. Lines (5-11) compute the two-dimensional subspace while lines (12) and (14-20) perform a trust-region step.

---

**Algorithm 2.** PCG-TrustRegion( $x_i, y_i, \theta_i, u_i, \omega_i, \mathbf{p}_f$ )
 

---

```

1:  $x_{\text{old}} \leftarrow \text{generateInitialTrajectory}(x_i, y_i, \theta_i, u_i, \omega_i, \mathbf{p}_f)$ 
2:  $f_{\text{old}} \leftarrow f(x_{\text{old}})$ 
3:  $\Delta \leftarrow \text{InitialTrustRegionRadius}()$ 
4: repeat
5:    $\langle g, H \rangle \leftarrow \text{computeGradientAndHessian}(x_{\text{old}})$ 
6:    $v_1 \leftarrow \text{preconditionedConjugateGradient}(g, H)$ 
7:    $v_1 \leftarrow v_1 / \|v_1\|_2$ 
8:    $v_2 \leftarrow g - v_1 (v_1^T g)$ 
9:    $v_2 \leftarrow v_2 / \|v_2\|_2$ 
10:   $g' \leftarrow [v_1 v_2]^T g$ 
11:   $H' \leftarrow [v_1 v_2]^T H [v_1 v_2]$ 
12:   $s' \leftarrow \text{argmin}_s \{ \frac{1}{2} s^T H' s + g'^T s \mid \|s\|_2 \leq \Delta \}$ 
13:   $s \leftarrow [v_1 v_2] s'$ 
14:   $x \leftarrow x_{\text{old}} + s$ 
15:   $f \leftarrow f(x)$ 
16:  if  $f < f_{\text{old}}$  then
17:     $f_{\text{old}} \leftarrow f$ 
18:     $x_{\text{old}} \leftarrow x$ 
19:  end if
20:   $\Delta \leftarrow \text{updateTrustRegionRadius}(\Delta)$ 
21: until convergence
22: return  $x_{\text{old}}$ 

```

---

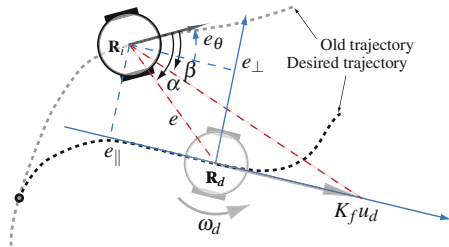
## 2.4 Computational Delays

In RHC, the optimized trajectory is followed during a time  $\delta$  during which no feedback from the environment is observed. After  $\delta$  seconds, feedback from the environment is incorporated to re-optimize the trajectory. In practice, the amount of time  $\delta$  dedicated to follow the trajectory is not fixed. Indeed, one often prefers to optimize the trajectory as fast as possible and use the result as early as possible. The sampling time  $\delta$  then directly relates to the computation time needed to optimize the new trajectory.

It is clear that while the optimization takes place, the robot continues to move according to the old trajectory which may result in a mismatch between the optimized position and the current position at the time when the optimization completes. Hence, the robot  $\mathbf{R}_i$  needs to reacquire (and track) the optimized trajectory. To do so, it needs to know its current position with respect to the desired new position, hereafter denoted by the coordinates  $(x_d, y_d)$ . This can easily be achieved by integrating the open-loop controls or, for more precision, by using odometry measurements (in our case given by wheel encoders which are deployed on most differential-wheeled robots). Fig. 3 shows a robot with its desired trajectory.



**Fig. 3** Schema of the quantities used by the control law in Eq. 12 that enable  $\mathbf{R}_i$  to reach a trajectory given by the virtual robot  $\mathbf{R}_d$  (desired trajectory) after having followed the old trajectory for *too long*



According to the desired trajectory, robot  $\mathbf{R}_i$  should be located at the position indicated by the virtual robot reference  $\mathbf{R}_d$ .  $\mathbf{R}_i$  is able to calculate the range  $e$  and the bearing  $\alpha$  to  $\mathbf{R}_d$ . It can identify the orientation  $-e_\theta$  (with respect to itself), the forward motion  $u_d$  and rotational motion  $\omega_d$  of  $\mathbf{R}_d$ . Note that  $\beta$  is the bearing to the point located at a distance  $K_f u_d$  in front of  $\mathbf{R}_d$ . We propose the following control law when  $\mathbf{R}_d$  moves forward:

$$\begin{cases} u_i = K_u e \cos \alpha + u_d \\ \omega_i = K_\omega e \sin \alpha + K_b \beta + \omega_d \end{cases} \quad (12)$$

with  $K_u$ ,  $K_\omega$ ,  $K_b$  and  $K_f$  all positive constants. An equivalent control law can be found when  $\mathbf{R}_d$  moves backward. Although omitted here for conciseness, it can be shown that this control law is stable and converges to the desired trajectory.

This strategy bears resemblance to the third strategy proposed by Milam et al. [21] to account for computation delays, with the exception that, instead of blindly applying the optimized control inputs (open-loop), we compute corrected control inputs based on the optimized trajectory using a tracking layer (closed-loop).

### 3 Experiments

Experiments are conducted using Khepera III robots in a  $3 \times 3 \text{m}^2$  arena. This robot has a diameter of 12cm, making it appropriate for multi-robot indoor experiments. As shown on Fig. 1(a), we equip each robot with a range and bearing module allowing for inter-robot positioning. A measurement campaign performed in [13] showed that the observation noise  $\varepsilon_z$  is normally distributed with a covariance  $\Sigma \approx [0.0221 \quad -0.0011; \quad -0.0011 \quad 0.0196]$ . The ground truth position and orientation of each robot is monitored using an overhead camera with SwisTrack [17], an open-source tracking software. The experiments are designed to analyze four different controllers:

**Reactive.** This controller, on top of which we add an obstacle avoidance control as explained in [10], was presented in [13]. It is a standard reactive controller, which does not optimize trajectories, nor predicts the future positions of neighboring robots or obstacles. However, it guarantees the rendezvous mathematically and was shown to perform under noisy perception particularly well.

**Hybrid-state A\*.** This controller implements Algorithm 1.

**PCG-TR.** This controller implements Algorithm 2.

**Centralized PCG-TR.** This controller implements Algorithm 2, but optimizes simultaneously the trajectories of all robots. In particular, the new cost function is the sum of the individual costs of all robots:  $\sum_{\mathbf{R}_i} f(\mathbf{p}_{0\dots N}^{(i)})$ , where  $\mathbf{p}_{0\dots N}^{(i)}$  are vertices of the trajectory of robot  $\mathbf{R}_i$ . We note that even if inter-dependencies between robots avoiding each other arise, the new Hessian matrix stays sparse and the optimization stays efficient. While all of the three above controllers use exclusively on-board resources, this centralized optimization is run off-board on a desktop computer and uses the information of the tracking system as input. It serves as an upper-bound on performance.

All controllers are tuned such that the average speed of the robots is about 15cm/s (i.e.,  $f_2$  is about the same across all controllers). Four scenarios are selected to provide a wide-range of situations upon which the different controllers can be tested:

**Scenario (a).** Four robots are randomly placed in the arena and form a complete graph (all robots are neighbors). Their task is to perform the rendezvous.

**Scenario (b).** Two robots  $\mathbf{R}_1$  and  $\mathbf{R}_2$  are placed 2 meters apart, facing each other. Each robot has to reach the initial location of the other robot. These locations are represented by 2 additional motion-less robots  $\mathbf{R}_3$  and  $\mathbf{R}_4$  (whose relative positions are artificially fed to the robots). Also,  $\mathbf{R}_1$  and  $\mathbf{R}_2$  have to avoid each other. Formally, we have  $\mathcal{O}_j = 1$  and  $\mathbf{o}_k^{(1)} = \hat{\mathbf{x}}_{12}(k\Delta t)$  for  $\mathbf{R}_1$  and  $\mathbf{o}_k^{(1)} = \hat{\mathbf{x}}_{21}(k\Delta t)$  for  $\mathbf{R}_2$ . This scenario not only tests collision avoidance, but also how each robot is able to rendezvous with a fixed goal position.

**Scenario (c).** Like the previous scenario but with four robots. This is a complex crossing and is an effective test-bed for analyzing the ability to optimize the trajectories quickly. Examples of trajectories obtained by the robots are shown in Fig. 4.

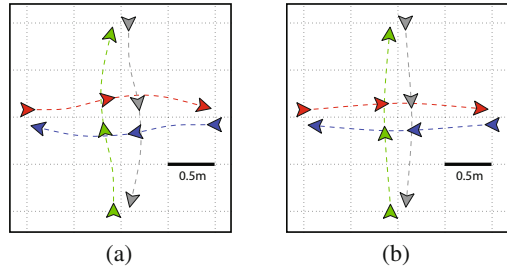
**Scenario (d).** This scenario involves two robots having to rendezvous and two other robots disturbing this rendezvous maneuver by crossing the arena.

Finally, across all scenarios and controllers, we perform two sets of experiments:

**Set I.** The first set tests the performance in terms of smoothness  $f_1$  of the resulting ground-truth trajectories of all controllers. The smoothness is a valid performance indicator, since the average forward speed was the same across controllers and scenarios, *all runs were collision-free* ( $f_3 \ll f_1$ ) and *all rendezvous maneuvers succeeded* ( $f_4 \ll f_1$ ). In this set, we did 10 runs per scenario per controller, resulting in a total of 160 experimental runs.

**Set II.** The second set aims to test the degradation of performance in **Scenario (a)** when the computational time of the controllers is increased by a fixed additional delay of 0, 200 and 400ms. Additionally, we test our approach to mitigate computational delays (Section 2.4, closed-loop) against the third strategy proposed in [21] (open-loop). The performance is measured both through the smoothness and the convergence speed towards the rendezvous point. We did 5 runs per delay per mitigation strategy per controller, resulting in a total of 90 experimental runs.

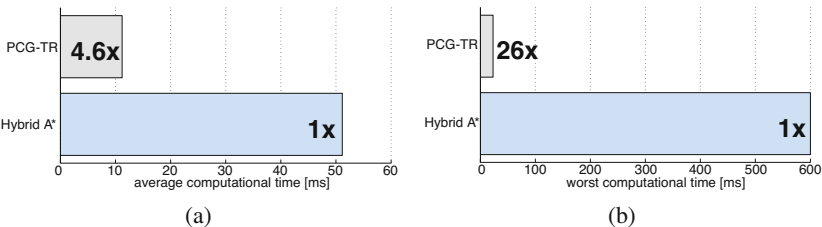
**Fig. 4** Runs performed on **Scenario (c)** with (a) the decentralized PCG-TR variant and (b) its centralized equivalent



## 4 Results

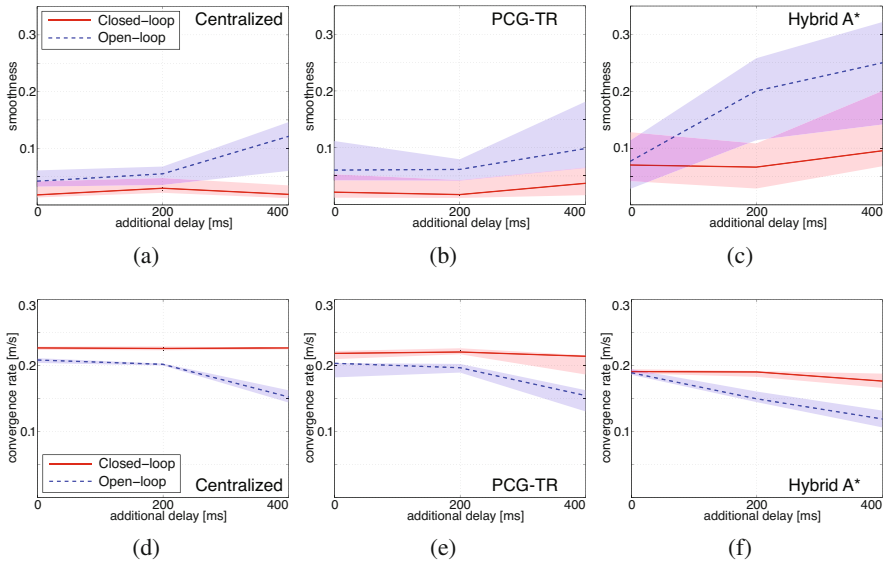
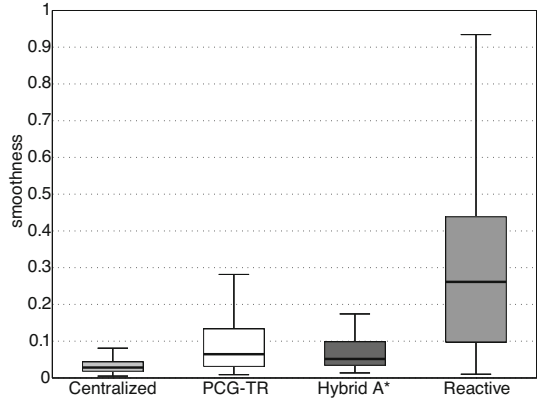
Before diving into the core results, we show that both algorithms are capable of running in real-time on board of our miniature robots. The average computational time over all robots and scenarios of **Set I** is shown in Fig. 5(a). We observe that PCG-TR is  $4.6\times$  faster than hybrid-state A\*, averaging 11.2ms per optimization cycle against 51.15ms (both algorithms run faster than 19.5Hz in average). The difference in performance is even more staggering when looking at the worst-case performance, in Fig. 5(b), yielding 23.12ms for PCG-TR and maxing out at 600ms for hybrid-state A\* (600ms is a hard computational time limit imposed on both optimization strategies to keep real-time capabilities). Indeed, hybrid-state A\* may have to explore many cells when the heuristic does not match the current situation. However, hybrid-state A\* guarantees that the optimized trajectory is feasible. Note that when run on a single core on a standard desktop computer (Intel<sup>®</sup> Core<sup>™</sup> i7 2.93GHz), PCG-TR averages 0.32ms and hybrid-state A\* 1.3ms.

Fig. 6 shows in the form of boxplots, the distribution of the smoothness of each trajectory of each robot in **Set I**. Low smoothness values indicate smooth trajectories, whereas high values indicate rough trajectories with many speed changes. Incidentally, a low value means a better minimization of the cost function. We observe that PCG-TR and hybrid-state A\* perform equally well and provide an improvement of about 300% over the standard state-of-the-art reactive controller. Both algorithms show their capability to minimize the objective function across the wide range of proposed scenarios. Their performance with respect to the centralized PCG-TR also suggests that our decentralized approach is competitive (about 74% worse). Remember that the centralized algorithm uses ground-truth positioning information



**Fig. 5** (a) Average and (b) worst computational time across all scenarios for PCG-TR and hybrid-state A\* on the real robots on **Set I**

**Fig. 6** Boxplot of the resulting smoothness of each controller across all scenarios and all robots. Smaller values indicate that trajectories are more smooth (i.e., smaller is better). We observe that the centralized controller performs best as expected, PCG-TR and hybrid-state A\* perform slightly worst but are much better than the reactive controller.



**Fig. 7** Plots of the smoothness (top row, smaller is better) and convergence rate (bottom row, higher is better) for the centralized (left column), PCG-TR (middle column) and hybrid-state A\* (right column) controllers for different additional computational delays and different tracking strategies (open versus closed-loop). The solid lines represent the median while the shaded region show the 25th and 75th percentiles. The performance worsen as the computational delay increases although the closed-loop controller (see Section 2.4) mitigates the added delays and performs better.

and requires the synchronization of our robots, whereas the decentralized variants only require local observations and no explicit communication. One can qualitatively compare trajectories obtained with PCG-TR and its centralized equivalent in Fig. 4.

The results of the second set of experiments are shown in Fig. 7. They concern only the pure rendezvous scenario, **Scenario (a)**, with four robots. On the first row, we show the smoothness degradation for both the closed-loop and open-loop control as we increase the computational delay. On one hand, the hybrid-state A\* seems to perform slightly worse and the resulting smoothness degrades more rapidly. This simple scenario may indeed exacerbate the computational time required by hybrid-state A\*. On the other hand, the benefit of our closed-loop control is to keep the smoothness almost constant even when computational delay reaches up to  $35\times$  the original computation time (for PCG-TR). The same conclusion can be made when looking at the second row of Fig. 7. This row shows the average convergence speed of the inter-robot distances (the higher, the faster the robots converge to the same rendezvous point). We observe that the performance of the open-loop control worsens as the one of the closed-loop control stays constant. Overall, the closed-loop control provides an efficient alternative when the optimization process is slow. However, it may only be implementable on robots equipped with accurate proprioceptive sensors. Hence, when one can only use the open-loop control, it is important to provide fast optimization methods such as PCG-TR or hybrid-state A\*.

## 5 Conclusion

In this work, we proposed a RHC capable of performing the rendezvous on a team of differential-wheeled robots equipped with noisy relative positioning hardware. This RHC is tested with two complementary optimization procedures and showed, in both cases, its ability to rendezvous on a wide range of experimental scenarios. The two optimization procedures are the hybrid-state A\* algorithm and a subspace trust-region method based on PCG. Both algorithms were successfully deployed on miniature robots, the Khepera III, and are able to run in real-time on-board. Finally, we developed a closed-loop control that follows the optimized trajectories and showed superior performance than its open-loop variant. This work provides an exhaustive analysis of two fast, numerical, optimal approaches to nonholonomic rendezvous for differential-wheeled robots.

## References

1. Beard, R., McLain, T., Nelson, D., Kingston, D., Johanson, D.: Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. *Proceedings of the IEEE* 94(7), 1306–1324 (2006)
2. Branch, M., Coleman, T., Li, Y.: A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing* 21, 1–23 (1999)
3. Byrd, R., Schnabel, R., Shultz, G.: Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical Programming* 40, 247–263 (1988)

4. Chung, D.H.: Time optimal rendezvous of three linear systems. *Journal of Optimization Theory and Applications* 12, 242–247 (1973)
5. Dimarogonas, D., Johansson, K.: Further results on the stability of distance-based multi-robot formations. In: *American Control Conference*, pp. 2972–2977 (2009)
6. Dimarogonas, D., Kyriakopoulos, K.: On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control* 52(5), 916–922 (2007)
7. Dimarogonas, D.V., Loizou, S.G., Kyriakopoulos, K.J., Zavlanos, M.M.: A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica* 42(2), 229–243 (2006)
8. Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J.: Path planning for autonomous driving in unknown environments. In: Khatib, O., Kumar, V., Pappas, G.J. (eds.) *Experimental Robotics. STAR*, vol. 54, pp. 55–64. Springer, Heidelberg (2009)
9. Eisenberg, E., Gale, D.: Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics* 30(1), 165–168 (1959)
10. Falconi, R., Sabattini, L., Secchi, C., Fantuzzi, C., Melchiorri, C.: A graph-based collision-free distributed formation control strategy. In: *18th IFAC World Congress* (2011), doi:10.3182/20110828-6-IT-1002.02450
11. Fax, J., Murray, R.: Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control* 49(9), 1465–1476 (2004)
12. Findeisen, F., Allgöwer, F.: An introduction to nonlinear model predictive control. In: *Benelux Meeting on Systems and Control*, pp. 119–141 (2002)
13. Goyal, S., Martinoli, A.: Bayesian rendezvous for distributed robotic systems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2765–2771 (2011)
14. Goyal, S., Martinoli, A.: Real-time optimization of trajectories that guarantee the rendezvous of mobile robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012) (to appear)
15. Hestenes, M.R., Stiefel, E.: Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards* 49, 409–436 (1952)
16. Lawton, J., Beard, R., Young, B.: A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation* 19(6), 933–941 (2003)
17. Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J., Martinoli, A.: Swistrack - a flexible open source tracking software for multi-agent systems. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4004–4010 (2008)
18. McLain, T., Chandler, P., Rasmussen, S., Pachter, M.: Cooperative control of UAV rendezvous. In: *Proceedings of the American Control Conference*, vol. 3, pp. 2309–2314 (2001)
19. Meshler, P.: Time-optimal rendezvous strategies. *IEEE Transactions on Automatic Control* 8(4), 279–283 (1963)
20. Miele, A., Weeks, M., Ciarcia, M.: Optimal trajectories for spacecraft rendezvous. *Journal of Optimization Theory and Applications* 132, 353–376 (2007)
21. Milam, M., Franz, R., Hauser, J., Murray, R.: Receding horizon control of vectored thrust flight experiment. *IEEE Proceedings on Control Theory and Applications* 152(3), 340–348 (2005)
22. Moré, J., Sorensen, D.: Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing* 3, 553–572 (1983)

23. Olfati-Saber, R., Fax, J., Murray, R.: Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE* 95(1), 215–233 (2007)
24. Prorok, A., Arfire, A., Bahr, A., Farserotu, J., Martinoli, A.: Indoor navigation research with the Khepera III mobile robot: An experimental baseline with a case-study on ultra-wideband positioning. In: *2010 International Conference on Indoor Positioning and Indoor Navigation (2010)*, doi:10.1109/IPIN.2010.5647880
25. Ren, W.: Distributed attitude consensus among multiple networked spacecraft. In: *American Control Conference (2006)*, doi:10.1109/ACC.2006.1656474
26. Ren, W., Beard, R.: *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Springer Publishing Company, Incorporated (2007)

# Part VI: ISER Session Summary on “Learning”

Matthew T. Mason

Carnegie Mellon University

## Session Summary

The session entitled "Learning" illustrates just how thoroughly machine learning permeates robotics, by gathering four papers that apply machine learning to four different robotic applications: driving, object recognition, natural language, and manipulation. In "Learning Autonomous Driving Styles and Maneuvers from Expert Demonstration", Silver et al. use machine learning to tune the driving behavior of an autonomous vehicle. Previous research has shown that learning performs well in tuning the parameters of the perception system, and that it performs well in tuning the parameters of the planning and control system. This paper shows that when both systems are tuned together, learning can perform spectacularly well. In "Unsupervised Feature Learning for RGB-D Based Object Recognition" Bo et al. adapt their previous feature learning work to take advantage of the most recent imaging technology. They test the system on several different datasets, against several existing systems. The authors' system works both on textured and untextured objects. In one case the authors' system outperforms a system tuned specifically for textured objects, when tested on textured objects. In "Learning to Parse Natural Language Commands to a Robot Control System", Matuszek et al. address a task familiar to us all: following route directions. Their system learns to take ordinary instructions like "go down the hall past a bunch of rooms and then turn left" and turns them into a robot control program. In simulations, the system correctly follows the directions around 66% of the time, which we all know exceeds a human's ability even to remember the directions. In "A Data-Driven Statistical Framework for Post-Grasp Manipulation" Paolini et al. develop stochastic models for grasping, placing, dropping, and insertion. Based on thousands of experiments, their system produces probability density functions for the pose of a grasped object, chooses optimal parameters for subsequent operations, and estimates the likelihood of success.



# Learning Autonomous Driving Styles and Maneuvers from Expert Demonstration

David Silver, J. Andrew Bagnell, and Anthony Stentz

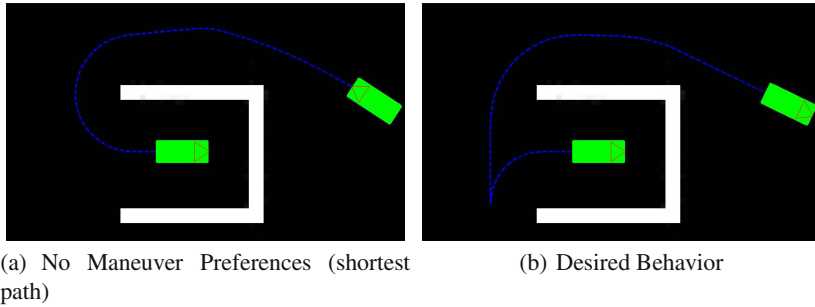
**Abstract.** One of the many challenges in building robust and reliable autonomous systems is the large number of parameters and settings such systems often entail. The traditional approach to this task is simply to have system experts hand tune various parameter settings, and then validate them through simulation, offline playback, and field testing. However, this approach is tedious and time consuming for the expert, and typically produces subpar performance that does not generalize. Machine learning offers a solution to this problem in the form of learning from demonstration. Rather than ask an expert to explicitly encode his own preferences, he must simply demonstrate them, allowing the system to autonomously configure itself accordingly. This work extends this approach to the task of learning driving styles and maneuver preferences for an autonomous vehicle. Head to head experiments in simulation and with a live autonomous system demonstrate that this approach produces better autonomous performance, and with less expert interaction, than traditional hand tuning.

## 1 Introduction

Building truly robust and reliable autonomous navigation systems remains a challenge to the robotics community. One of the many barriers to successful deployment of such systems is the large number of parameters and settings they often entail, with robust behavior dependent on correct determination of these values. While it is difficult enough to properly build and parameterize the systems themselves, it is even harder to determine the correct settings that will properly encode desired behavior in known scenarios, while also generalizing to the unknown. The traditional approach to this task involves system experts hand tuning various parameters, which then must be validated through actual system performance. However, this approach is tedious and time consuming, while typically resulting in poor system performance that does

---

David Silver · J. Andrew Bagnell · Anthony Stentz  
Carnegie Mellon University, 10 40th St Pittsburgh, PA, USA 15201  
e-mail: {dsilver, axs}@rec.ri.cmu.edu, dbagnell@ri.cmu.edu



**Fig. 1** A simulated example in a binary environment, demonstrating the necessity of preferences over maneuvers as well as chosen paths. In this example, simply following the shortest path (a) leads to a long drive in reverse. Especially for vehicles with sensors only in front, a longer turning maneuver (b) that drives in the forward direction is often preferable.

not generalize. Machine learning, specifically learning from demonstration, offers a potential solution to this problem. Rather than ask an expert to explicitly encode his own preferences, he must simply demonstrate them, allowing the system to autonomously configure itself accordingly. Recent work has demonstrated that this approach can both reduce the amount of expert interaction required, while improving the resulting system performance.

In the domain of autonomous navigation, this learning approach has mostly focused on the task of analyzing perceptual information and determining the preferability of traversing various sections of terrain. However, a robot's planning system must determine the best plan that not only considers these preferences, but also more dynamic considerations such as velocities or accelerations on a vehicle, stability, etc. This latter problem involves its own set of parameters and tuning, and the coupled problem requires proper balancing of the tradeoffs of the component problems. Failure to properly tune the overall system can result in poor performance. For example, a vehicle that is too averse to swerving may be too willing to traverse certain obstacles it should avoid, while one that is too willing to swerve will drive in a jittery fashion. Given that both of these considerations can be scalars derived from high dimensional vectors, it can be very difficult to properly tune all the necessary parameters by hand. Since a real system requires tuning for an enormous set of such problems, producing a truly robust system by hand approaches infeasibility.

This work extends the approach of [14] to the task of learning driving styles and maneuver preferences for an autonomous vehicle. That is, it not only learns *where* a robot should drive, it also learns *how* to drive. Solving these coupled problems from the same training inputs ensures the the resulting parameterization produces good system performance when applied in an online setting. The next section discusses related work in both autonomous navigation and learning from demonstration. Sections 3 and 4 discuss learning cost functions from demonstration, and extend known approaches to the coupled problem of learning maneuvers and driving styles. This approach is then validated through experimental results presented in Section 5.

## 2 Related Work

Autonomous navigation is generally framed as finding the lowest cost (i.e. optimal) feasible sequence from a start to a goal through some state space. These states could represent locations in the world, configurations of the robot, actions the robot could perform, or some combination thereof. The cost of a plan is usually defined as the sum of costs of individual states. To allow for generalization, costs are usually not explicitly assigned to specific states, but are rather produced as a function of features describing individual states. The function mapping features of states to costs essentially encodes the preferences that the robot will exhibit; therefore its configuration will have a dramatic impact on the robot's performance. Historically, the mapping of features to costs has been performed by simple manual construction of a parameterized function, and then hand tuning various parameters to create a cost function that produces desired behavior. This manual approach has been frequently used whether the cost functions in question describe locations in the world [8, 16, 17] or actions to be performed [4, 11, 19]. Unfortunately, this tedious approach typically produces subpar results, potentially leading to subpar autonomous performance.

Supervised learning is a popular solution to such tuning tasks, by automatically adjusting parameters to meet desired criteria. As opposed to just learning parameters within a specific component of an autonomous system (e.g. Terrain Classification), learning from demonstration seeks to learn parameters to directly modify end system behavior. Traditional learning from demonstration [3] seeks to learn a mapping directly from states, or features of states, to actions. The advantage of this model free formulation is that it creates a straightforward learning task. However, this comes at the cost of difficulty with both generalization to new problems, and sequentially combining decisions to achieve longer horizon planning.

Recently, model based approaches to learning from demonstration have become more popular. In contrast to model free, model based learning continues to use the optimal planning algorithms popular in navigation, and seeks to learn a feature to cost mapping such that the optimal plan in a given scenario produces desired behavior; in this way these approaches are essentially applications of inverse optimal control. Numerous applications of this approach [7, 9, 14, 20] have demonstrated its effectiveness at learning navigation cost functions over patches of terrain, improving autonomous performance and reducing the need for manual tuning.

Although learning from demonstration has been previously applied to the task of learning preferences over actions and trajectories [1, 6, 18], this work has generally been focused on more short range vehicle control and path tracking. A notable exception is [2], which investigated learning driving preferences in a parking lot from demonstration. However, the coupled problem of learning both terrain and driving preferences has not been previously addressed.

### 3 Learning Navigation Cost Functions from Demonstration

In previous work, the Maximum Margin Planning (MMP) framework [12, 14] was developed to allow learning cost functions from expert demonstration. MMP seeks to find the simplest cost function such that an example plan  $P_e$  is lower cost than all other plans, and by a margin. The margin is based on a loss function  $L_e$  that encodes the similarity of a plan to the example. Rather than enforce this constraint over all possible plans, it is only necessary to enforce it against the current minimum cost plan  $P_*$  under the current cost function  $C$ . Thus, MMP can be represented as a constrained optimization problem

$$\begin{aligned} & \text{minimize } O[C] = |C| \quad \text{subject to} & (1) \\ & \sum_{x \in P_*} (C(F_x) - L_e(x)) \geq \sum_{x \in P_e} (C(F_x)) \\ & P_* = \arg \min_P \sum_{x \in P} (C(F_x) - L_e(x)) \end{aligned}$$

$F_x$  represents the feature vector describing state  $x$  in some planning state space. Since it is generally not possible to meet this constraint, a slack term  $\zeta$  is added, and the constraint is rewritten as

$$\begin{aligned} & \text{minimize } O[C] = \lambda|C| + \zeta \quad \text{subject to} & (2) \\ & \sum_{x \in P_*} (C(F_x) - L_e(x)) - \sum_{x \in P_e} (C(F_x)) + \zeta \geq 0 \end{aligned}$$

Since  $\zeta$  is in the minimizer, it will always be tight against the constraint and equal to the difference in costs. Therefore,  $\zeta$  can be replaced by this difference to produce an unconstrained optimization

$$\text{minimize } O[C] = \lambda|C| + \sum_{x \in P_e} (C(F_x)) - \sum_{x \in P_*} (C(F_x) - L_e(x)) \quad (3)$$

This objective can be optimized by (sub)gradient descent, using the subgradient

$$\nabla O[C] = \lambda \nabla |C| + \sum_{x \in P_e} \delta_F(F_x) - \sum_{x \in P_*} \delta_F(F_x) \quad (4)$$

Simply speaking, the subgradient is positive at values of  $F$  corresponding to states in the example plan, and negative at values of  $F$  corresponding to states in the current plan. Applying gradient descent directly would involve raising or lowering the cost at specific values of  $F$ , according to the (negative) gradient. To encourage generalization and limit  $|C|$ , this gradient can instead be projected onto a limited set of directions. As in [10, 12] this projection takes the form of learning a classifier or regressor to differentiate between states whose cost needs to be raised or lowered, and then adding this learner to the current cost function.

The final algorithm, known as LEARNING to search (LEARCH) [12] iteratively computes the set of states (under the current cost function) whose cost should be raised or lowered, computes a learner to reproduce and generalize this distinction,

and adds this learner to the cost function. The result is a cost function that is the weighted sum of set of learners (of whatever form and complexity is desired).

Real world motion planning systems, when fed by a stream of onboard perceptual data, generally recompute their plan several times a second to account for the dynamics of and errors in sensing, positioning, and control. As a result, applying LEARCH to an actual vehicle requires an understanding of these dynamics with respect to expert demonstration. As a consequence, rather than treat a single demonstration as a single example, it must be considered as a large set of examples, which are chained together by actual robot motion and the passage of time. Each single example at an instant in time forms its own constraints as in (3), with the full problem the sum of these individual optimizations. Solving this problem therefore involves summing the individual gradients steps from (4) into a single learner update. For more details on the application of this approach to real world navigation problems with multiple and noisy example demonstrations, see [14].

## 4 Learning Driving Styles and Maneuver Preferences

In addition to the practical considerations mentioned above, modern planning systems are often composed of a hierarchy of individual planners, with each planner refining the results of previous plans [8, 16]. In such a scenario, LEARCH must be applied to the lowest level planner, that actually makes the final decision about the next course of action. Application to higher level planners may be necessary as well, if they use different cost functions than the lowest level.

The formulation in the previous section referred to generic states in a planner's state space. Depending on the planner, these could be locations in the world, actions to be performed, or combined state-action pairs. A straightforward application of LEARCH to state action pairs would be possible, directly learning the coupled problem of balancing terrain and action preferences. However, such a formulation would be overly complex, in that it would learn a solution in the space of terrains and actions, rather than each one individually. Depending on the specific learner used, this could result in poor generalization; for example, preferring soft turns on flat terrain would not necessarily teach the system to prefer soft turns on hilly terrain.

Instead, this work proposes to decouple the problems by defining the cost of a state action pair as the decoupled cost of the state plus the cost of the action.

$$C(P) = \sum_{x \in P} C(x) = \sum_{x \in P} C_s(F_x^s) + C_a(F_x^a) \quad x \in S \times A \quad (5)$$

That is, two separate cost functions are proposed, one over locations the vehicle will traverse, and one over actions the vehicle will perform. The overall cost of a plan is the sum of the costs over individual locations and actions. Specific couplings between locations and actions (e.g. driving through a ditch or over a bump slowly may be fine, but is very costly at high speeds) are still possible, by adding features to express such couplings specifically in  $F^s$  or  $F^a$  (the state or action feature vectors).

If this cost formulation is plugged directly into (3), taking the partial derivatives with respect to  $C_s$  and  $C_a$  yields

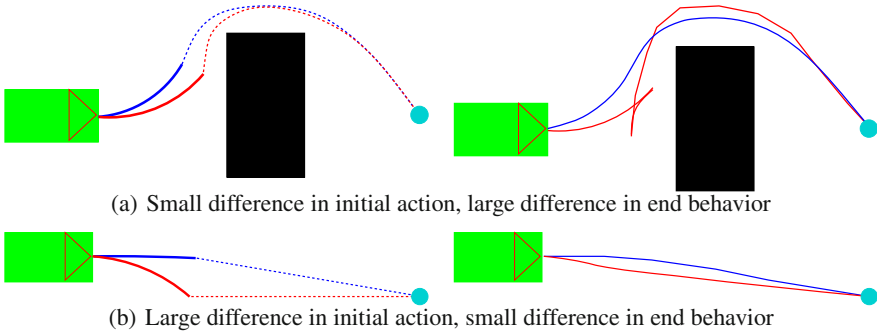
$$\begin{aligned} \frac{\partial O}{\partial C_s} O[C_s, C_a] &= \lambda \nabla |C_s| + \sum_{x \in P_e} \delta_F^s(F_x^s) - \sum_{x \in P_*} \delta_F^s(F_x^s) \\ \frac{\partial O}{\partial C_a} O[C_s, C_a] &= \lambda \nabla |C_a| + \sum_{x \in P_e} \delta_F^a(F_x^a) - \sum_{x \in P_*} \delta_F^a(F_x^a) \\ P_* &= \arg \min_P \sum_{x \in P} (C_s(F_x^s) + C_a(F_x^a) - L_e(x)) \end{aligned} \quad (6)$$

These partials describe an interleaving optimization, where  $C_s$  and  $C_a$  are each updated one at a time. However, there is still a dependence between the two, as  $P_*$  is defined with respect to both. That is, for a current pair of cost functions, the functional gradient describes a set of actions to be desired or avoided (given the current terrain preferences) and a set of terrains to be desired or avoided (given the current action preferences). Applying LEARCH in this manner leads to the construction of two separate cost functions, both of which attempt to reproduce expert behavior in concert with the other.

The LEARCH formulation up to now has assumed that the example plan  $P_e$  is something the planning system could exactly achieve and match. However, in practice this is rarely the case. Due to finite resolution and sampling of state, and small differences in perceived and actual vehicle models (and implied kinematic or dynamic motion constraints), it is rarely the case that the finite set of plans the planner could produce will exactly contain  $P_e$ . One way to avoid this dilemma is to simply consider the set of all paths  $P$  in (1) to be all possible plans the planner could produce given a specific planning problem, as opposed to all possible plans that could exist. In this way, the basic MMP constraint ensures not that the example plan is the lowest cost plan, but simply that it is preferable to anything else the planner might produce. Unfortunately, in practice this provides no actual guarantee that the planner will produce the right behavior online. That is, depending on various resolutions and discretization, it is possible that during online application, plans will become available to the planner that are undesirable and lower cost. The result is that learning in this manner may require a much larger training set than would otherwise be needed.

An alternate solution to this problem is to in some way project the example plan onto the possible planner options. Depending on the structure of the problem, the loss function may provide a natural way to perform this projection. For instance, if actions are defined simply by a vehicle curvature and velocity, than choosing the planner action that is closest in these measures to the example action would seek to ensure similar behavior to the example, while allowing the MMP constraint to be achievable. This approach has been shown to provide beneficial results to generalization, as it allows learning to terminate when it is as close to achieving the example as resolution will allow [14].

A final challenge in learning cost functions over actions is the issue of receding horizon control. Rather than compute a dynamically feasible plan all the way from a

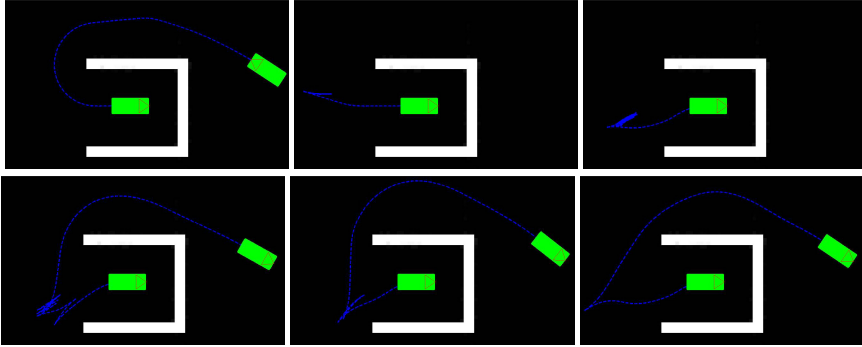


**Fig. 2** Simple examples of planning problems where, due to receding horizon planning, error between immediate actions (left) and final behavior (right) do not correspond. Desired action/behavior is in blue, actual action/behavior is in red.

vehicle’s current state to its goal, it is common for modern motion planners to only compute such a plan to a set horizon, using a heuristic or lower dimensional planner as a cost-to-go from the horizon to the goal. The result is that the planner will make a plan at a certain instant, with no expectation of running this plan to completion; rather a small section is executed before replanning. This creates a challenge for learning as it is now possible for very small errors in an immediate action to produce large errors in final behavior; this is especially problematic when the cost-to-go does not capture dynamic or kinematic constraints. It is also possible for large errors in an immediate action to end up being inconsequential, with the final behavior still quite similar. Figure 2 demonstrates both these cases for a simple planner that uses constant curvature arcs (see Section 5).

One way to identify when a small error in an immediate action choice will have a large final effect, or vice versa, is to forward simulate the repeated decision making of the planner to produce a simulated final behavior. This raises the question of whether this should actually be the current plan that is used for learning (e.g.  $P_*$ ). Unfortunately, using this plan does not produce the proper derivative as in (6) as it potentially ignores states that, while not encountered in repeated simulation, would have been encountered in a single choice at a finite point in time. With the effect of such states on end behavior hidden from the learner, the cost function can not be properly modified. Therefore, the best way to ensure the correct end behavior is to ensure the correct choice is made at each instant in time ([13]).

However, while this simulation may not be useful for providing a partial derivative, it is useful for determining how important an error in an immediate action selection is. That is, if we define a penalty function  $\mathcal{P}_e$  that defines error over end behavior (similar to how  $L_e$  defines error over an immediate action) then we can weight individual examples by how consequential they will be. The result is an optimization that, rather than bounding  $L_e$ , bounds  $\mathcal{P}_e L_e$ . In practice, this effect is achieved by weighting individual  $\frac{\partial O}{\partial c_a}$  in (6) by  $\mathcal{P}_e$



**Fig. 3** Progression of planner behavior in simulation (from top left to bottom right) as a cost function is learned. Through successive iterations, the planner learns to exit the cul-de-sac to align itself so as to drive forward to the goal, instead of in reverse.

$$\frac{\partial O}{\partial C_a} O[C_s, C_a] = \lambda \nabla |C_a| + \mathcal{P}_e \left[ \sum_{x \in P_e} \delta_F^a(F_x^a) - \sum_{x \in P_*} \delta_F^a(F_x^a) \right] \quad (7)$$

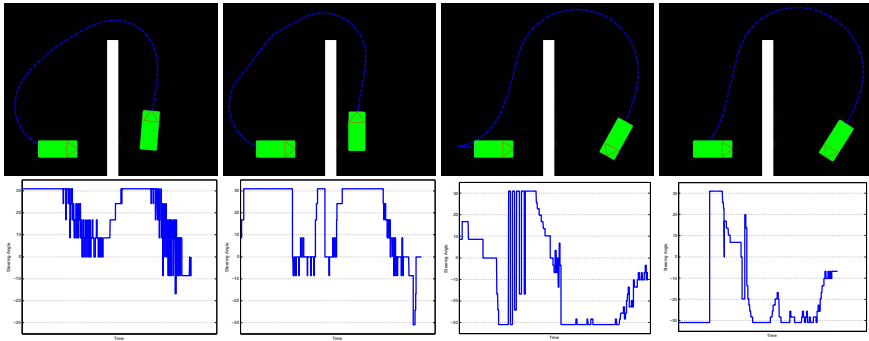
This approach is formally known as slack re-scaling; for more information (including proof of bounds and correctness) see [13]. A nice side affect of this approach is that it becomes another source of robustness to noisy demonstration (since small differences in end behavior are lessened in importance).

## 5 Experimental Results

To test the application of LEARCH and its modifications to learning driving styles and maneuvers, a local-global planning system was created as in [8, 16], with a local planner choosing between constant curvature arcs, and a global planner operating without kinematic constraints. This planner architecture was chosen because it is simple, well understood, and in practice very effective. However, its effectiveness is dependent on proper tuning. Without proper tuning it can exhibit jittery or other undesirable behavior. In addition, since it only plans a single (kinematic) action at a time, it is incapable of performing more complex maneuvers (e.g. 3 point turns) without additional tuning. This tuning often takes the form of a state machine based on the relationship between the current vehicle pose and the current global plan, as well as a state history.

In addition to any cost accrued due to the locations a plan would traverse, the planner was implemented with costs based on specific actions (e.g. the final chosen motion arc). Costs were computed as a function of the features of an action, such as direction (forward or reverse), curvature, alignment with the heading of the global plan, and changes in direction or curvature from previous actions. These features provide enough information such that a state machine is not necessary to perform more complex maneuvers; however, achieving this capability requires careful tuning.





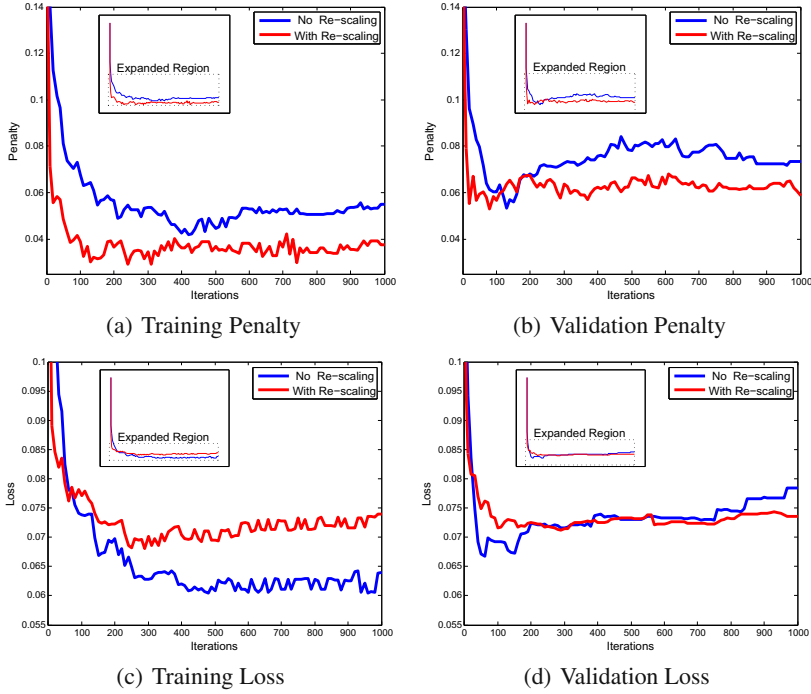
**Fig. 4** The progression of learned preference models (from left to right) in a validation scenario. The commanded curvature at each point in time is also shown for each example. The planner first learns to avoid unnecessary turning, and then to favor driving forward and being aligned with the global path.

### 5.1 Simulated Experiments

Experiments were first performed in simulation in binary environments, to decouple any effects of terrain costing. A training set was created by manually driving the simulated robot through a simple set of obstacle configurations. A cost function was learned over constant curvature actions that resulted in the planner reproducing the expert’s driving style (e.g. soft turns were preferred to hard turns when appropriate, changes in curvature were limited when possible, but obstacles were still avoided). In addition, the cost function resulted in the planner generating complex maneuvers by properly chaining together simple actions, despite never planning more than one action ahead at a time.

Figure 3 provides an example of learning a simple maneuver. Initially, the planner simply prefers the action that minimizes its cost-to-go (the global planner cost) and so tracks the global plan in reverse. Over successive iterations, a preference is developed for aligning the vehicle’s heading with the global path, allowing the robot to perform a multi-point turn maneuver to turn in reverse and then drive forward. Eventually, a preference is also learned to avoid changing directions, allowing a single reverse turn and then a forward motion. Figure 4 provides an example of how the learned cost function affects driving style and control. In early iterations, the system is heavily underdamped and very jittery. Over time, it first learns to smooth out these oscillations, but then re-allows them in order to achieve other preferences (namely turning while in reverse). Finally, the system learns to balance both desires, performing a maneuver by chaining together multiple actions, while at the same time limiting unnecessary sudden turning. The single large oscillation seen in the final iteration was in response to almost hitting the wall, demonstrating that these preferences were properly balanced with those necessary to avoid obstacles.

Experiments were also performed to demonstrate the effect of slack re-scaling (penalty weighting) as described in Section 4. The training and validation performance during these experiments is shown in Figure 5. As expected, penalty



**Fig. 5** Learning planner preference models with and without slack re-scaling (weighting examples by penalty)

weighting results in an optimization that lowers the training error between example and simulated end behavior, while increasing the error in immediate actions. The interesting result is in the validation performance, where the optimization without penalty weighting suffers from increased error and overfitting. This demonstrates the negative effects of trying too forcefully to correct small examples in actions (when they don't significantly effect end performance) and the advantage of penalty weighting.

A final simulated experiment compared the performance of learning to hand tuning. A human expert (different than the one who drove the vehicle for the training set) was tasked with manually constructing and tuning a cost function to achieve the same basic goals (a clean and reasonable driving style, while still producing complex maneuvers when necessary). During hand tuning, the expert could see the current performance of the planner in simulation. Every parameter configuration encountered during hand tuning was recorded for evaluation. The performance of both the learned and hand tuned cost function was evaluated over a large set of validation behaviors. The metrics for comparison are the average loss and average penalty; that is the average error between the example and planner immediate actions and end behavior. The hand tuned system had a final validation loss that was more than 25% higher than the learned system, and a final validation penalty that was 20% higher. Of note is that that the final configuration of the hand tuned system was not its best;



**Fig. 6** The E-Gator Robotic Platform used in these experiments

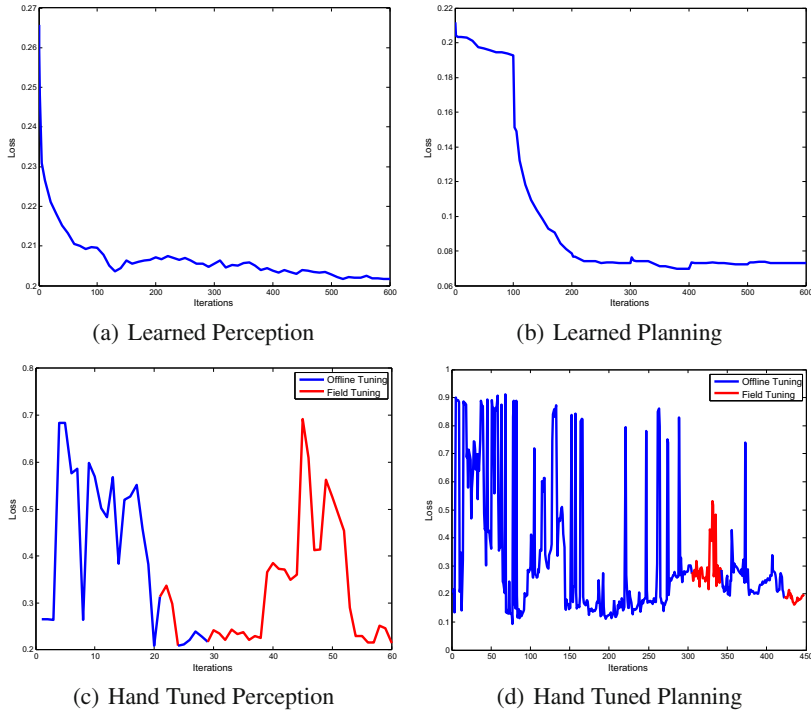
the best configuration's performance was essentially equivalent to the learned system. This exemplifies one of the major issues with hand tuning: in general a human can only evaluate a small set of examples (relative to an automated system), and so may not fully understand the quality of a specific configuration during tuning. Also of note is the amount of expert interaction that was required in each case: approximately 12 hours to hand tune the system, versus less than 2 hours to demonstrate examples in simulation.

## 5.2 *E-Gator Experiments*

Next, the planning system was applied to an autonomous E-Gator vehicle (Figure 6) for operation in outdoor, unstructured terrain. The E-Gator was equipped with a single actuated LiDAR scanner for producing 3D point clouds. Perception software described in [5] was ported to this system, allowing the computation of a supporting ground surface through vegetation, as well as geometric features relating to size and shape of potential positive obstacles. These features were used as input to a terrain cost function. The vehicle's preferred behavior was therefore determined by two cost functions: one mapping sensor data to costs over terrain (perception), and another mapping actions to costs (planning). Live expert demonstration (i.e. driving the vehicle) was then used to learn these two cost functions.

For comparison, yet another robotics expert was tasked with manually creating and tuning perception and planning cost functions to achieve the same navigation tasks for the E-Gator. The expert could test parameter configurations in offline simulation/playback, or on the live vehicle. Every tested parameter configuration was recorded for comparison. The performance of both sets of cost functions was then evaluated over a large set of validation behaviors. These results are shown in Figure 7. The metric for comparison across all experiments was validation loss.

These experiments demonstrate how the system performance changes for both the learned and hand tuned systems during tuning/learning. The learned system undergoes a relatively stable optimization procedure, quickly converging to a good solution and then iteratively refining it. In contrast, during hand tuning the expert continually changed parameters with large (and often negative) effects on system



**Fig. 7** Validation Loss for the Learned and Hand Tuned autonomy systems

performance. As seen in [14], the expert was able to tune a perception system that was close to learned quality (within 6%). However, in contrast to simulation (with a binary world), the expert performed significantly poorer when tuning the planning system. The expert’s final configuration resulted in more than 2.5 times the error in action selection, and even the expert’s best configuration (unknown during tuning) had 25% more error than the learned system. This demonstrates the inherent challenge not only in tuning perception and planning cost functions to solve independent problems, but to properly couple them to produce good autonomous performance.

The difference in performance was further explored through a series of live experiments on the actual autonomous platform. The two final system configurations (learned or hand tuned perception/planning cost functions) were tested head to head over a large set of test courses, totaling more than 4 km of autonomous driving. Several statistics were measured and compared over these runs. Table 1 shows these statistics, and the statistical significance of any differences. Of note is that the hand tuned system spent much more time in reverse (dangerous for a vehicle with sensors in front) than the learned system. It would also occasionally oscillate between forward and reverse driving very rapidly. The hand tuned system exhibited a higher preference for driving perfectly straight; when it did turn, it generally turned much faster (and changed curvature faster and more often) than the learned system.

**Table 1** Results of 4 km of autonomous E-Gator experiments comparing learned to hand-tuned performance. All statistics are on a per test average.

	Avg Dist (m)	Extra Dist (m)	Avg Roll (°)	Avg Pitch(°)	% Time in Reverse
Learned ( $\mu/\sigma^2$ )	69.27/785.5	12.02/148.5	3.71/2.61	4.45/2.29	0.075/0.012
Hand-Tuned ( $\mu/\sigma^2$ )	73.09/1791.7	17.15/1293.24	4.24/3.90	4.39/2.15	0.19/0.079
P-value	0.34/ <b>0.014</b>	0.23/ <b>0.0</b>	0.13/0.14	0.56/0.57	<b>0.016/0.0</b>

	Dir Switch Per m	Avg Steer Angle (°)	Avg $\Delta$ Angle (°)	% Time Angle $\neq 0$	Avg $\Delta$ Angle $\neq 0$ (°)	Safety E-Stops
Learned ( $\mu/\sigma^2$ )	0.029/0.0014	9.43/15.46	1.69/0.20	0.33/0.0064	5.19/0.49	0.13/0.12
Hand-Tuned ( $\mu/\sigma^2$ )	0.044/0.0079	11.74/14.43	2.08/0.76	0.19/0.0033	10.76/3.60	0.90/1.49
P-value	0.18/ <b>0.0</b>	<b>0.011/0.57</b>	<b>0.016/0.0</b>	<b>0.0/0.038</b>	<b>0.0/0.0</b>	<b>0.0/0.0</b>

These differences manifest themselves in the safety of each configuration, as the hand tuned system required nearly 7 times as many operator interventions (e.g. to prevent hitting a dangerous obstacle). One of the most common causes of these interventions was the hand tuned system trying to turn too hard around obstacles. The obstacles were clearly seen by the perception system; however the coupled tuning of perception and planning cost functions was such that the vehicle tended to clip them. It is interesting to note that this behavior was clearly observed during hand tuning, and much of the final tuning was an unsuccessful attempt to combat this problem. This demonstrates the difficulty in manually mapping desired behavior changes into appropriate parameter changes.

A final point of note is the time required to produce each system. The hand tuned system required 38 hours of expert tuning time, and an additional 18 hours from a safety operator while tuning on the live vehicle. In contrast, the learned system required less than 4 hours of combined expert and operator time to collect the necessary training data. So in addition to achieving safer and more stable autonomous performance, the learning approach required only 1/14 as much human interaction. For additional data and details of these experiments, see [13].

## 6 Conclusion

This work addressed the issue of producing properly coupled cost functions for an autonomous navigation system, to balance preferences describing both where and how a vehicle should drive. The LEARCH algorithm of the MMP framework [12] was extended to handle this coupled problem by learning cost functions separately while still considering their affect on eachother. Issues with finite planner resolution

and receding horizon control were addressed through modifications to the LEARCH algorithm, with experimental results validating these new approaches.

Previous work [14] has shown that human experts can hand tune a perception cost function that is almost equivalent in performance to one that is learned, with the primary advantage of learning being a major reduction in the amount of expert interaction required. This same result was duplicated when tuning just a planner cost function: the human expert was able to do almost as well as the learned system, but in a far less efficient manner. A closer analysis of the hand tuning process shows that human experts do not necessarily know when they have achieved their best performance, whereas a learned system has the advantage of automatic validation. In addition, it was shown that by learning the proper parameters from demonstration, a very simple planning system could be configured to result in more complex driving maneuvers.

When a human expert was tasked with tuning both a perception and planning system at the same time, this additional coupling proved a major detriment to tuning performance. As opposed to simply taking longer, the hand tuned approach also produced a system that was significantly lower quality. This decrease in quality was demonstrated in both offline (simulated) validation, and in online autonomous validation. In the latter case, the poorer quality of the cost functions manifested itself as an increase in the number of safety interventions required during autonomy, as well as a more jittery driving style. These results further demonstrate the effectiveness of the learning from demonstration approach, both for reducing the amount of time required to tune an autonomous system, and for improving the quality and robustness of the result.

Future work is currently focused on application to a variety of autonomous vehicles with a range of planning systems. In addition, the challenge of collecting and curating large amounts of training data must be addressed. While these approaches have drastically reduced the amount of time necessary to properly tune an autonomous system, further improvement is possible. This includes approaches for identifying the most useful training data [15], as well as alternate approaches to soliciting expert input and feedback [13].

**Acknowledgements.** This work was sponsored by the U.S. Army Research Laboratory under contract “Robotics Collaborative Technology Alliance” (contract DAAD19-01-2-0012). The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

## References

1. Abbeel, P., Coates, A., Ng, A.Y.: Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotics Research* 29(13), 1608–1639 (2010)
2. Abbeel, P., Dolgov, D., Ng, A.Y., Thrun, S.: Apprenticeship learning for motion planning with application to parking lot navigation. In: *IROS* (2008)

3. Argall, B., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. *Robotics and Autonomous Systems* (2008)
4. Bacha, A., Bauman, C., Faruque, R., Fleming, M., Terwelp, C., Reinholtz, C., Hong, D., Wicks, A., Alberi, T., Anderson, D., Cacciola, S., Currier, P., Dalton, A., Farmer, J., Hurdus, J., Kimmel, S., King, P., Taylor, A., Covern, D.V., Webster, M.: Odin: Team victortango's entry in the darpa urban challenge. *Journal of Field Robotics* 25(8), 467–492 (2008)
5. Bagnell, J.A., Bradley, D., Silver, D., Sofman, B., Stentz, A.: Learning for autonomous navigation: Advances in machine learning for rough terrain mobility. *IEEE Robotics & Automation Magazine* 17(2), 74–84 (2010)
6. Hamner, B., Singh, S., Scherer, S.: Learning obstacle avoidance parameters from operator behavior. *Journal of Field Robotics* 23(1), 1037–1058 (2006)
7. Kalakrishnan, M., Buchli, J., Pastor, P., Schaal, S.: Learning locomotion over rough terrain using terrain templates. In: *IROS*, pp. 167–172 (2009)
8. Kelly, A., Stentz, A., Amidi, O., Bode, M., Bradley, D., Diaz-Calderon, A., Happold, M., Herman, H., Mandelbaum, R., Pilarski, T., Rander, P., Thayer, S., Vallidis, N., Warner, R.: Toward reliable off road autonomous vehicles operating in challenging environments. *International Journal of Robotics Research* 25(5-6), 449–483 (2006)
9. Kolter, J.Z., Abbeel, P., Ng, A.Y.: Hierarchical apprenticeship learning with application to quadruped locomotion. In: *Neural Information Processing Systems* (2008)
10. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent. In: *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge (2000)
11. Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., Thrun, S.: Junior: The stanford entry in the urban challenge. *Journal of Field Robotics* 25(9), 569–597 (2008)
12. Ratliff, N., Silver, D., Bagnell, J.A.: Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots* 27, 25–53 (2009)
13. Silver, D.: Learning preference models for autonomous mobile robots in complex domains. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA (2010)
14. Silver, D., Bagnell, J.A., Stentz, A.: Learning from demonstration for autonomous navigation in complex unstructured terrain. *International Journal of Robotics Research* 29(12), 1565–1592 (2010)
15. Silver, D., Bagnell, J.A., Stentz, A.: Active learning from demonstration for robust autonomous navigation. In: *International Conference on Robotics and Automation* (2012)
16. Singh, S., Simmons, R., Smith, T., Stentz, A., Verma, V., Yahja, A., Schwehr, K.: Recent progress in local and global traversability for planetary rovers. In: *ICRA* (2000)
17. Stelzer, A., Hirschmuller, H., Gerner, M.: Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *International Journal of Robotics Research* 31(4), 381–402 (2012)
18. Sun, J., Mehta, T., Wooden, D., Powers, M., Rehg, J., Balch, T., Egerstedt, M.: Learning from examples in unstructured, outdoor environments. *Journal of Field Robotics* 23 (2007)

19. Urmson, C., Anhalt, J., Bagnell, J.A., Baker, C., Bittner, R., Clark, M.N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T.M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.W., Singh, S., Snider, J., Stentz, A., Whittaker, W.L., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., Ferguson, D.: Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics* 25(8), 425–466 (2008)
20. Zucker, M., Ratliff, N., Stolle, M., Chestnutt, J., Bagnell, J.A., Atkeson, C.G., Kuffner, J.: Optimization and learning for rough terrain legged locomotion. *International Journal of Robotics Research* 30, 175–191 (2011)



# Unsupervised Feature Learning for RGB-D Based Object Recognition

Liefeng Bo\*, Xiaofeng Ren, and Dieter Fox

**Abstract.** Recently introduced RGB-D cameras are capable of providing high quality synchronized videos of both color and depth. With its advanced sensing capabilities, this technology represents an opportunity to dramatically increase the capabilities of object recognition. It also raises the problem of developing expressive features for the color and depth channels of these sensors. In this paper we introduce hierarchical matching pursuit (HMP) for RGB-D data. HMP uses sparse coding to learn hierarchical feature representations from raw RGB-D data in an unsupervised way. Extensive experiments on various datasets indicate that the features learned with our approach enable superior object recognition results using linear support vector machines.

## 1 Introduction

Recognizing object instances and categories is a crucial capability for an autonomous robot to understand and interact with the physical world. Humans are able to recognize objects despite large variation in their appearance due to changing viewpoints, deformations, scales and lighting conditions. This ability fundamentally relies on robust visual representations of the physical world. However, most state-of-the-art object recognition systems are still based on hand-designed representations (features), such as SIFT [26], Spin Images [18], SURF [3], Fast Point

---

Liefeng Bo · Dieter Fox  
University of Washington,  
Seattle, USA  
e-mail: {lfb, fox}@cs.washington.edu

Xiaofeng Ren  
ISTC-Pervasive Computing Intel Labs,  
Seattle, USA  
e-mail: xiaofeng.ren@intel.com

\* Bo is now with the ISTC-Pervasive Computing Intel Labs.

Feature Histogram [30], LINE-MOD [15], or feature combinations [20, 7]. Such approaches suffer from at least two key limitations. Firstly, these features usually only capture a small set of recognition cues from raw data; other useful cues are ignored during feature design. For instance, the well-known SIFT features capture edge information from RGB images using a pair of horizontal and vertical gradient filters while completely ignoring color information. Secondly, the features have to be re-designed for new data types, or even new tasks, making object recognition systems heavily dependent on expert experience. It is desirable to develop efficient and effective learning algorithms to automatically learn robust representations from raw data.

Recently, several research groups have developed techniques for unsupervised feature learning from raw vision data [16, 40, 38, 24, 12, 8]. Algorithms such as deep belief nets [16], denoising autoencoders [40], deep Boltzmann machines [38], convolutional deep belief networks [24], K-Means based feature learning [12, 4], hierarchical sparse coding [43], and hierarchical matching pursuit [8] have been proposed to this end. Such approaches build feature hierarchies from scratch, and have exhibited very impressive performance on many types of recognition tasks such as handwritten digit recognition [16, 40, 38], face recognition [24], tiny image recognition [12], object recognition [24, 12, 43, 8], event recognition [8] and scene recognition [8]. However, the current applications are somewhat limited to 2D images, typically in grayscale.

Recently introduced RGB-D cameras are capable of providing high quality synchronized videos of both color and depth. With its advanced sensing capabilities, this technology represents an opportunity to dramatically increase the capabilities of object recognition. It also raises the problem of developing expressive features for the color and depth channels of these sensors. Inspired by the success of our previous work, hierarchical matching pursuit (HMP) for image classification, we propose *unsupervised feature learning for RGB-D based object recognition* by making hierarchical matching pursuit suitable for color and depth images captured by RGB-D cameras. HMP learns dictionaries over image and depth patches via K-SVD [2] in order to represent observations as sparse combinations of codewords. With the learned dictionary, feature hierarchies are built from scratch, layer by layer, using orthogonal matching pursuit and spatial pyramid pooling [8]. Two major innovations are introduced in this work: (1) Unsupervised feature learning on both color and depth channels; (2) spatial pyramid pooling over sparse codes from both layers of the HMP hierarchy. Extensive evaluations on several publicly available benchmark datasets [20, 10, 39] allowed us to gain various experimental insights: unsupervised feature learning from raw data can yield recognition accuracy that is superior to state-of-the-art object recognition algorithms, even to ones specifically designed and tuned for *textured* objects; the innovations introduced in this work significantly boost the performance of HMP applied to RGB-D data; and our approach can take full advantage of the additional information contained in color and depth channels.

## 2 Related Work

This research focuses on hierarchical feature learning and its application to RGB-D object recognition. In the past few years, a growing amount of research on object recognition has focused on learning rich features using unsupervised learning, hierarchical architectures, and their combination.

**Deep Networks:** Deep belief nets [16] learn a hierarchy of features by training multiple layers recursively using the unsupervised restricted Boltzmann machine. This pre-training phase has been shown to avoid shallow local minima. The learned weights are then further adjusted to the current task using supervised information. To make deep belief nets applicable to full-size images, Lee et al. [24] proposed convolutional deep belief nets that use a small receptive field and share the weights between the hidden and visible layers among all locations in an image. Invariant predictive sparse decomposition [17, 19] approximates sparse codes from sparse coding approaches using multi-layer feed-forward neural networks and avoid solving computationally expensive optimizations at runtime. Stacked denoising autoencoders [40] build deep networks, based on stacking layers of denoising autoencoders that train one-layer neural networks to reconstruct input data from partial random corruption. Deconvolutional networks [44] reconstruct images using a group of latent feature maps in a convolutional way under a sparsity constraint. These approaches have been shown to yield competitive performance with the SIFT based bag-of-visual-words model on object recognition benchmarks such as Caltech-101.

**Single Layer Sparse Coding:** Sparse coding [31] on top of raw images/patches has been developed for face recognition [1], digit recognition [28] and texture segmentation [27]. More recently, researchers have shown that single layer sparse coding on top of SIFT features achieves state-of-the art performance on more challenging object recognition benchmarks [23, 42, 41, 9, 12, 43]. Yang et al. [42] learn sparse codes over SIFT features instead of raw image patches using sparse coding approaches. Their comparisons suggested that such an approach significantly outperforms the standard bag-of-visual-words model. Wang et al. [41] presented a fast implementation of local coordinate coding that computes sparse codes of SIFT features by performing local linear embedding on several nearest codewords in a codebook learned by K-Means. Boureau et al. [9] compared many types of recognition algorithms, and found that the SIFT based sparse coding approaches followed by spatial pyramid max pooling work very well, and macrofeatures can boost recognition performance further. Coates and Ng [12] evaluated many feature learning approaches by decomposing them into training and encoding phases, and suggested that the choice of architecture and encoder is the key to a feature learning system. Yu et al. [43] showed that hierarchical sparse coding at pixel level achieves similar performance with SIFT based sparse coding.

**Feature Learning for RGB-D:** Kernel descriptors [6] learn patch level feature descriptors based on kernels comparing manually designed pixel descriptors such as gradients, local binary patterns or colors. Adapting this view to depth maps and 3D points, RGB-D kernel descriptors are proposed in [5, 7], and the experiments

showed that they obtain higher recognition accuracy than hand-designed feature sets on the RGB-D object dataset [20]. By adapting K-Means based feature learning proposed by Coates and Ng [12] to the RGB-D setting, Blum and colleagues showed that it is possible to learn RGB-D descriptors from raw data that are competitive with RGB-D kernel descriptors on the RGB-D object dataset [4].

### 3 Unsupervised Feature Learning

This section provides an overview of our feature learning architecture. We review the key ideas behind dictionary learning and discuss our two-layer architecture to generate features over complete RGB-D images.

Building on our previous work on feature learning for object recognition [8], we propose two innovations to make the approach suitable for RGB-D based object recognition. Firstly, we perform feature learning on both color and depth images. The original HMP work [8] uses grayscale images only, insufficient in many cases: color is distinctively useful for object instance recognition where appearance details matter, and the depth channel in RGB-D can greatly improve object category recognition and its robustness. We learn dictionaries and encode features using full RGB-D data: gray, RGB, depth and surface normal channels. Secondly, as described in Section 3.2, we extract features not only from the top of the feature hierarchy, but also from lower layers.

#### 3.1 Dictionary Learning via K-SVD

The key idea of sparse coding is to learn a dictionary, which is a set of vectors, or codes, such that the data can be represented by a sparse, linear combination of dictionary entries. In our case, the data are patches of pixel values in RGB-D frames. For instance, a dictionary for  $5 \times 5$  RGB-D patches would contain vectors of size  $5 \times 5 \times 8$ , where the last component is due to grayscale intensity, RGB, depth and surface normal values. Grayscale intensity values are computed from the associate RGB values and normal values are computed from the associated depth values and their coordinates.

K-SVD [2] is a popular dictionary learning approach that generalizes K-Means. It learns dictionaries  $D = [d_1, \dots, d_m, \dots, d_M]$  and the associated sparse codes  $X = [x_1, \dots, x_n, \dots, x_N]$  from a matrix  $Y$  of observed data by minimizing the reconstruction error

$$\min_{D, X} \|Y - DX\|_F^2 \quad s.t. \quad \forall m, \|d_m\|_2 = 1 \quad \text{and} \quad \forall n, \|x_n\|_0 \leq K \quad (1)$$

Here, the notation  $\|A\|_F$  denotes the Frobenius norm, the zero-norm  $\|\cdot\|_0$  counts the non-zero entries in the sparse codes  $x_n$ , and  $K$  is the sparsity level controlling the number of the non-zero entries. When the sparsity level is set to be 1 and the sparse code matrix is forced to be a binary(0/1) matrix, K-SVD exactly reproduces the K-Means algorithm.

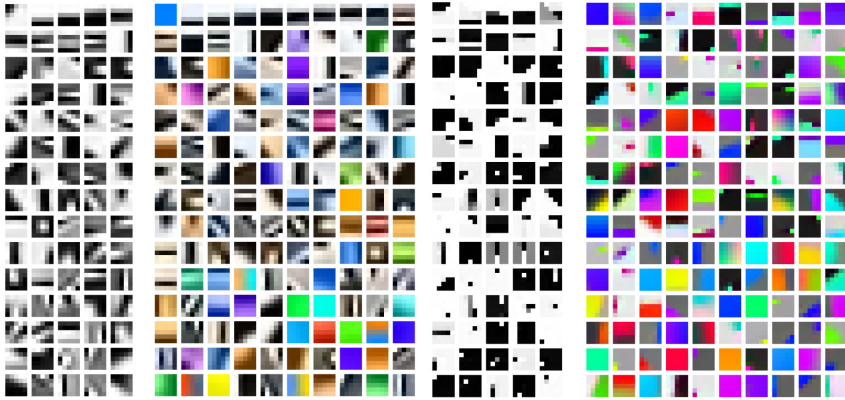
K-SVD solves the optimization problem (1) in an alternating manner. During each iteration, the current dictionary  $D$  is used to encode the data  $Y$  by computing the sparse code matrix  $X$ . Then, the codewords of the dictionary are updated one at a time, resulting in a new dictionary. This new dictionary is then used in the next iteration to recompute the sparse code matrix followed by another round of dictionary update. We now briefly outline these steps, see [2, 8] for details.

**Computing the Sparse Code Matrix via Orthogonal Matching Pursuit:** Given the current dictionary  $D$ , optimizing the sparse code matrix  $X$  can be decoupled into  $N$  sub-problems; one for each data item  $y_n$ . The sparse code  $x_n$  for each item  $y_n$  is computed efficiently using orthogonal matching pursuit (OMP) [34], a greedy algorithm. In each iteration, OMP selects the codeword  $d_m$  that best matches the current residual, which is the reconstruction error remaining after the codewords chosen thus far. In the first iteration, this residual is exactly the observation  $y_n$ . Once the new codeword is selected, the observation is orthogonally projected onto the span of all the previously selected codewords and the residual is recomputed. The procedure is repeated until the desired  $K$  codewords are selected. In our unsupervised feature learning setting, a large number of image patches share the same dictionary and the total cost of OMP can be reduced by its batch version that keeps some quantities in memory to save computational cost [13, 36, 8].

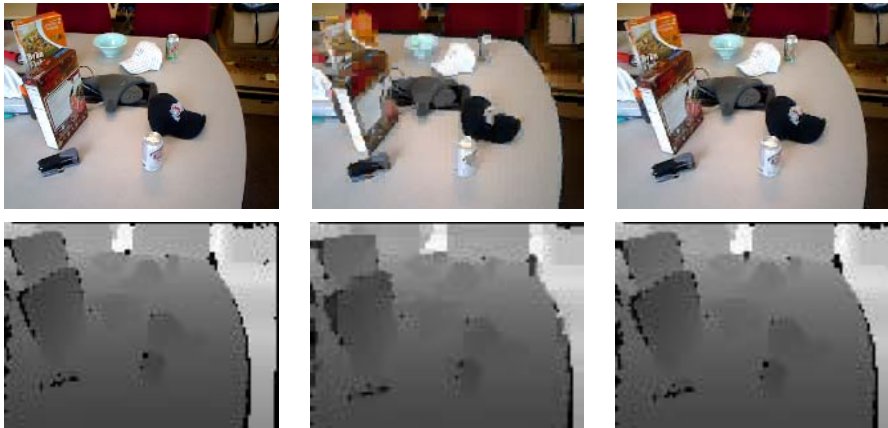
**Updating the Dictionary via Singular Value Decomposition:** Given the sparse code matrix  $X$ , the dictionary  $D$  is optimized sequentially via Singular Value Decomposition (SVD). In the  $m$ -th step, the  $m$ -th codeword and its sparse codes can be computed by performing SVD of the residual matrix corresponding to that codeword. This matrix contains the differences between the observations and their approximations using all other codewords and their sparse codes. To avoid introducing new non-zero entries in the sparse code matrix  $X$ , the update process only considers observations that use the  $m$ -th codeword. It can be shown that each iteration of sparse coding followed by dictionary updating decreases the reconstruction error (1). In practice, K-SVD converges to good dictionaries for a wide range of initializations [2].

In our hierarchical matching pursuit, K-SVD is used to learn dictionaries at two layers, where the data matrix  $Y$  in the first layer consists of patches sampled from RGB-D images, and  $Y$  in the second layer are sparse codes pooled from the first layer (details below). Fig. 1 visualizes the learned dictionaries in the first layer for four channels: grayscale and RGB for RGB images, and depth and surface normal for depth images. As can be seen, the learned dictionaries have very rich appearances and include separated red, green, blue colors, transition filters between different colors, gray and color edges, gray and color texture filters, depth and normal edges, depth center-surround (dot) filters, flat normals, and so on, suggesting many recognition cues of raw data are well captured.

Once dictionaries are learned via K-SVD, sparse codes can be computed for new images using orthogonal matching pursuit or the more efficient batch tree orthogonal matching pursuit [8]. Fig. 2 shows an example of an RGB / depth image pair along with reconstructions achieved for different levels of sparsity. The shown results are



**Fig. 1** Dictionaries learned for different channels. From left to right: Grayscale intensity, RGB, depth, 3D surface normal (3 normal dimensions color coded as RGB). The codeword sizes are  $5 \times 5 \times 1$  for grayscale intensity and depth values, and  $5 \times 5 \times 3$  for RGB and surface normal values. Dictionary sizes are 75 for grayscale intensity and depth values, and 150 for RGB and surface normal values.

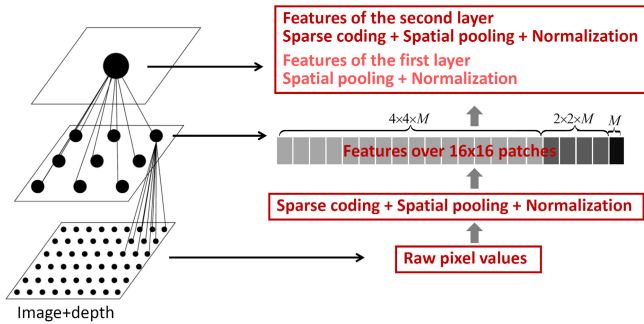


**Fig. 2** Reconstructed images using the learned dictionaries. **Left:** Original RGB and depth images. **Middle:** Reconstructed RGB and depth images using only two codewords per  $5 \times 5$  patch. **Right:** Reconstructions using five codewords.

achieved by non-overlapping  $5 \times 5$  reconstructed patches. As can be seen, a sparsity level of  $K = 5$  achieves results that are virtually indistinguishable from the input data, indicating that this technique could also be used for RGB-D *compression*, alternative to [37]. For object recognition, the sparse codes become the features representing the image or segment, as we describe next.

### 3.2 Hierarchical Matching Pursuit

With the learned dictionaries  $D$ , hierarchical matching pursuit builds a feature hierarchy by applying the orthogonal matching pursuit encoder recursively (Fig. 3). This encoder consists of three modules: batch orthogonal matching pursuit, pyramid max pooling, and contrast normalization (see also [8]).



**Fig. 3** Hierarchical matching pursuit for RGB-D object recognition. In the first layer, sparse codes are computed on small patches around each pixel. These sparse codes are pooled into feature vectors representing  $16 \times 16$  patches, by spatial pyramid max pooling. The second layer encodes these feature vectors using a dictionary learned from sampled patch level feature vectors. Whole image features are generated from sparse codes of both first and second layers.

**First Layer:** The goal of the first layer is to generate features for image patches whose size is typically  $16 \times 16$  pixels or larger. Each pixel in such a patch is represented by the sparse codes computed for the pixel and a small neighborhood (for instance,  $5 \times 5$  pixel region). Spatial pyramid max pooling is then applied to these sparse codes to generate patch level features. Spatial pyramid max pooling partitions an image patch  $P$  into multiple level spatial cells. The features of each spatial cell  $C$  are the max pooled sparse codes, which are simply the component-wise maxima over all sparse codes within a cell:

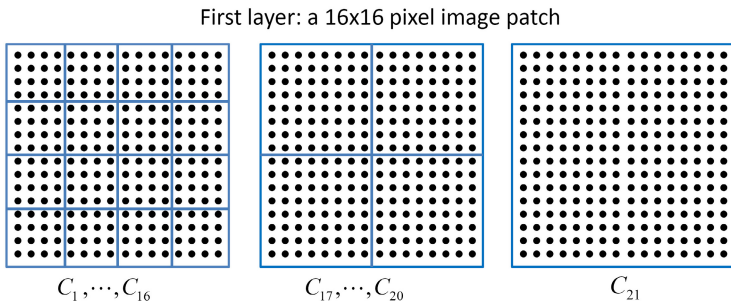
$$F(C) = \left[ \max_{j \in C} |x_{j1}|, \dots, \max_{j \in C} |x_{jm}|, \dots, \max_{j \in C} |x_{jM}| \right] \quad (2)$$

Here,  $j$  ranges over all entries in the cell, and  $x_{jm}$  is the  $m$ -th component of the sparse code vector  $x_j$  of entry  $j$ . Note that  $F(C)$  has the same dimensionality as the original sparse codes. The feature  $F_P$  describing a  $16 \times 16$  image patch  $P$  are the concatenation of aggregated sparse codes in each spatial cell

$$F_P = [F(C_1^P), \dots, F(C_s^P), \dots, F(C_S^P)] \quad (3)$$

where  $C_s^P \subseteq P$  is a spatial cell generated by spatial pyramid partitions, and  $S$  is the total number of spatial cells. As an example, we visualize spatial cells generated by a





**Fig. 4** Spatial pyramid partitions. Each black dot denotes sparse codes of a pixel that are computed on a  $5 \times 5$  small patch around this pixel using batch orthogonal matching pursuit. **Left:** Level 2. The  $16 \times 16$  image patch is partitioned into  $4 \times 4 = 16$  spatial cells. Each cell is represented by the component-wise maximum of the sparse codes of pixels within the cell. **Middle:** Level 1. The  $16 \times 16$  image patch is partitioned into  $2 \times 2 = 4$  spatial cells. Here, maxima are computed over the level 2 cells. **Right:** Level 0. The whole  $16 \times 16$  image patch is one spatial cell. The concatenation of  $C_1$  through  $C_{21}$  gives the max pooled feature for the whole patch.

3 level spatial pyramid pooling on a  $16 \times 16$  image patch in Fig. 4. In this example, the dimensionality of the pooled feature vector  $F_P$  is  $(16 + 4 + 1)M$ , where  $M$  is the size of the dictionary (see also Fig. 3). The main idea behind spatial pyramid pooling is to allow the features  $F_P$  to encode different levels of invariance to local deformations [24, 42, 8], thereby increasing the discrimination of the features.

We additionally normalize the feature vectors  $F_P$  by their  $L_2$  norm  $\sqrt{\|F_P\|^2 + \varepsilon}$ , where  $\varepsilon$  is a small positive number. Since the magnitude of sparse codes varies over a wide range due to local variations in illumination and occlusion, this operation makes the appearance features robust to such variations, as commonly done in the hand-designed SIFT features. We found that  $\varepsilon = 0.1$  works well for the recognition problems we considered.

**Second Layer:** The goal of the second layer in HMP is to generate features for a whole image/object. To do so, HMP applies the sparse coding and max pooling steps to image patch features  $F_P$  generated in the first layer. The dictionary for this level is learned by sampling patch features  $F_P$  over RGB-D images. To extract the feature describing a whole image, HMP first computes patch features via the first layer (usually, these patches cover  $16 \times 16$  pixels and are sampled with a step size of  $4 \times 4$  pixels). Then, just as in the first layer, sparse codes of each image patch are computed using batch orthogonal matching pursuit, followed by spatial max pooling ( $3 \times 3$ ,  $2 \times 2$ , and  $1 \times 1$  cell sizes). However, in this layer, we perform max pooling over the sparse codes *and* the patch level features computed in the first layer:

$$G(C) = \left[ \max_{j \in C} |z_{j1}|, \dots, \max_{j \in C} |z_{jU}|, \max_{j \in C} |F_{j1}|, \dots, \max_{j \in C} |F_{jV}| \right] \quad (4)$$



Here,  $C$  is a cell and  $F_j$  and  $z_j$  are the patch features and their sparse codes, respectively.  $U$  and  $V$  are the dimensionality of  $z_j$  and  $F_j$ , where  $U$  is given by the size of the layer two dictionary, and  $V$  is given by the size of the patch level feature (3). Jointly pooling  $z_j$  and  $F_j$  integrates both fine-grained cues captured by the codewords in the first layer and coarse-grained cues by those in the second layer, increasing the discrimination of the features (joint pooling improves results over those reported in our original HMP work [8]).

The features of the whole image/object are the concatenation of the aggregated sparse codes within each spatial cell. The image feature vector  $G_I$  is then normalized by dividing with its  $L_2$  norm  $\sqrt{\|G_I\|^2 + 0.0001}$ .

It should be noted that hierarchical matching pursuit is a fully unsupervised feature learning approach: no supervision (e.g. object class) is required for dictionary learning and feature coding. The feature vectors  $G_I$  of images/objects and their corresponding labels are then fed to classifiers to learn recognition models.

## 4 Experiments

We evaluate our RGB-D hierarchical matching pursuit framework on three publicly available RGB-D object recognition datasets and two RGB object recognition datasets. We compare HMP to results achieved by state-of-the-art algorithms published with these datasets. For all five datasets, we follow the same training and test procedures used by the corresponding authors on their respective data.

In the first layer, we learn the dictionaries of size 75 with sparsity level 5 on 1,000,000 sampled  $5 \times 5$  raw patches for grayscale and depth channels, and dictionaries of size 150 on 1,000,000 sampled  $5 \times 5 \times 3$  raw patches for RGB and normal channels using K-SVD (see their visualizations in Fig. 1). We remove the zero frequency component from raw patches by subtracting their means. With these learned dictionaries, we compute sparse codes of each pixel ( $5 \times 5$  patch around it) using batch OMP with sparsity level 5, and generate patch level features by max pooling over  $16 \times 16$  image patches with  $4 \times 4$ ,  $2 \times 2$ , and  $1 \times 1$  partitions. Note that this architecture leads to fast computation of patch level features.

In the second layer, we learn the dictionaries of size 1,000 with sparsity level 10 on 1,000,000 sampled  $16 \times 16$  patch level features for all four channels using K-SVD. With these learned dictionaries, we compute sparse codes of image patches that are densely sampled from the whole image with a step size of  $4 \times 4$  pixels. We then pool both patch level features and their sparse codes on the whole image with  $3 \times 3$ ,  $2 \times 2$ , and  $1 \times 1$  partitions to generate the image level features. The final image feature vectors are the concatenation of those from all four channels, resulting in a feature size of 188,300 dimensions.

The above hyperparameters are optimized on a subset of the RGB-D object recognition dataset we collected. We empirically found that they work well on different datasets. In the following experiments, we will keep these values fixed, even though the performance might improve via tuning these parameters for each dataset using cross validation on the associated training data. With the learned HMP

features, linear support vector machines (SVMs) are trained for recognition. Linear SVMs are able to match the performance of nonlinear SVMs with the popular histogram intersection kernel [29] while being scalable to large datasets [8].

#### 4.1 RGB-D Object Dataset

The first dataset, called RGBD, contains 41,877 RGB-D images of 300 physically distinct everyday objects taken from different viewpoints [20]. The objects are organized into 51 categories arranged using WordNet hypernym-hyponym relationships. The objects in the dataset are segmented from the background by combining color and depth segmentation cues. The RGBD dataset is challenging since it not only contains textured objects such as food bags, soda cans, and cereal boxes, but also texture-less objects such as bowls, coffee mugs, fruits, or vegetables. In addition, the data frames in RGBD additionally exhibit large changes in lighting conditions.

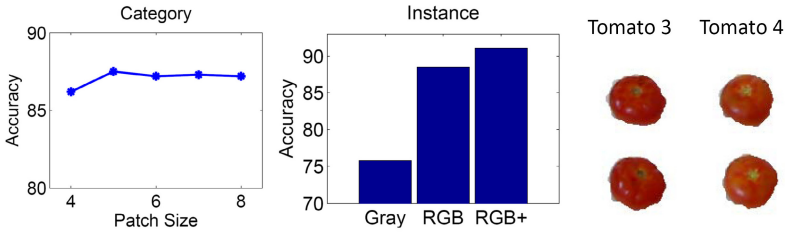
**Object Recognition.** We distinguish between two types of object recognition tasks: instance recognition and category recognition. Instance recognition is to recognize known object instances. Category recognition is to determine the category name of a previously unseen object. Each object category consists of a number of different object instances. Following the experimental setting in [20], we randomly leave one object instance out from each category for testing, and train models on the remaining  $300 - 51 = 249$  objects at each trial for category recognition. We report the accuracy averaged over 10 random train/test splits. For instance recognition, we train models on images captured from  $30^\circ$  and  $60^\circ$  elevation angles, and test them on the images of the  $45^\circ$  angle (leave-sequence-out).

We compare HMP with the baseline [20], kernel descriptors [7], convolutional k-means descriptors (CKM Desc) [4], and the original HMP [8] (features from the second layer only; grayscale and depth, but no color and normal) in Table 1. The recognition systems developed in [20, 7, 22] use a rich set of manually designed features. As can be seen, HMP outperforms all previous approaches for both category and instance recognition. For instance recognition, features learned on color images substantially improve the performance relative to those on grayscale images.

We performed additional experiments to shed light on different aspects of our approach. Fig. 5 (*left*) shows category recognition accuracy as a function of the patch size in the first layer. A larger patch size helps to improve the accuracy, but becomes saturated around  $5 \times 5$ . In Fig. 5 (*middle*), we show instance recognition accuracy using features on grayscale images and features on color images from the second layer only, and features on color images from both layers. As can be seen, features on color images work much better than those on grayscale images. This is expected since color information plays an important role for instance recognition. Object instances that are distinctive in the color space may have very similar appearance in grayscale space. We investigated the object instances misclassified by HMP, and found that most of the mistakes are from fruits and vegetables. Two misclassified tomatoes are shown in Fig. 5. As one can see, these two tomato instances are so

**Table 1** Comparisons with the baseline [20], kernel descriptors [7], convolutional k-means descriptor [4] and the original HMP [8]

RGBD Methods	Category			Instance		
	RGB	Depth	RGB-D	RGB	Depth	RGB-D
ICRA11 [20]	74.3 ± 3.3	53.1 ± 1.7	81.9 ± 2.8	59.3	32.3	73.9
Kernel descriptors [7, 22]	80.7 ± 2.1	80.3 ± 2.9	86.5 ± 2.1	90.8	54.7	91.2
CKM Desc [4]	N/A	N/A	86.4 ± 2.3	82.9	N/A	90.4
HMP [8]	74.7 ± 2.5	70.3 ± 2.2	82.1 ± 3.3	75.8	39.8	78.9
This work	82.4 ± 3.1	81.2 ± 2.3	87.5 ± 2.9	92.1	51.7	92.8



**Fig. 5** **Left:** category recognition accuracy as a function of the filter size. **Middle:** instance recognition accuracy by using features on grayscale images (Gray), features on color images (RGB) from the second layer only, features on color images from both layers (RGB+). **Right:** two tomato instances confused by our approach.

similar that even humans struggle to tell them apart. If such objects are excluded from the dataset, our approach has more than 95% accuracy for instance recognition on the RGBD dataset.

We also investigate recognition accuracies using features from the first layer only, from the second layer only, and from both layers. We observe that integrating features from both layers improves performance by about 2 percents. Features from the second layer are better than those from the first layer for category recognition while features from the first layer are better than the second layer for instance recognition. This makes sense intuitively, since coarse-grained information (second layer) is more important for category recognition whereas fine-grained information (first layer) is more important for instance recognition.

**Pose Estimation.** We further evaluated the HMP features for pose estimation, where the pose of every view of every object is annotated as the angle about the vertical axis. Each object category has a canonical pose that is labeled as 0°, and every image in the dataset is labeled with a pose in [0,360°]. Similar to instance recognition, we use the 30° and 60° viewing angle sequences as training set and the 45° sequence as test set. For efficiency, we follow an independent tree approach to estimate pose, where each level is trained as an independent classifier [21]: Firstly, one-versus-all category classifiers are trained in the category level; secondly,

**Table 2** Pose estimation error (in degrees) and running time (in seconds) comparison of several approaches. Indep Tree is a tree of classifiers where each level is trained as independent linear SVMs, NN is nearest neighbor regressor, and OPTree is the Object-Pose Tree proposed in [21]. Median pose accuracies for MedPose, MedPose(C) and MedPose(I) are 88.9%, 89.6% and 90.0%, respectively. Mean pose accuracies for AvePose, AvePose(C) and AvePose(I) are 70.2%, 73.6% and 75.1%, respectively.

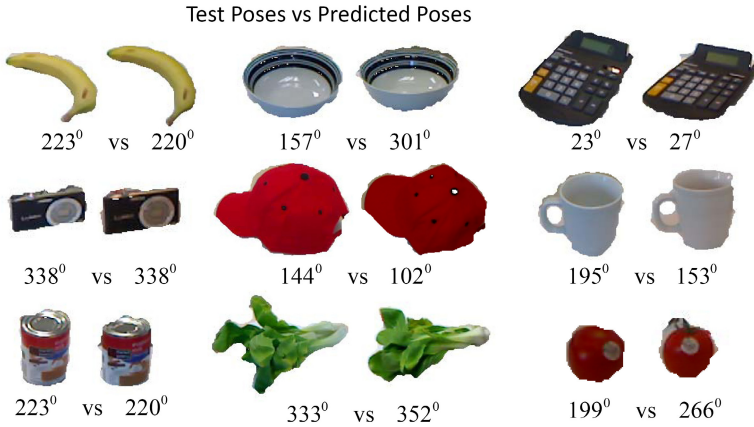
Technique	MedPose	MedPose(C)	MedPose(I)	AvePose	AvePose(C)	AvePose(I)	Test Time(s)
NN	144.0°	105.1°	33.5°	109.6°	98.8°	62.6°	54.8
Indep Tree	73.3°	62.1°	44.6°	89.3°	81.4°	63.0°	0.31
OPTree	62.6°	51.5°	30.2°	83.7°	77.7°	57.1°	0.33
This work	20.0°	18.7°	18.0°	53.6°	47.5°	44.8°	0.51

one-versus-all instance classifiers are trained in the instance level within each category; and finally one-versus-all pose classifiers in the pose level are trained within each instance. At test time, category, instance and pose classifiers are run in turn to estimate the pose of a query object.

Table 2 shows pose estimation errors under three different scenarios. We report both median pose (MedPose) and mean pose (AvePose) errors because the distribution across objects is skewed [21]. For MedPose and AvePose, pose errors are computed on the entire test set, where test images that were assigned an incorrect category or instance label have a pose error of 180.0°. MedPose(C) and AvePose(C) are computed only on test images that were assigned the correct category by the system, and, MedPose(I) and AvePose(I) are computed only on test images that were assigned the correct instance by the system. We compare HMP to our previous results [21]. As can be seen from Table 2, with our new HMP features, pose estimation errors are significantly reduced under all scenarios, resulting in only 20° median error even when classification errors are measured as 180.0° offset. We visualize test images and the best matched images in Fig. 6. The results are very intuitive: estimations are quite accurate for non-symmetric objects and sometimes inaccurate for symmetric objects for which different poses could share very similar or exactly same appearances.

## 4.2 Willow and 2D3D Datasets

We evaluated HMP on two other publicly available RGB-D recognition datasets. The first dataset, 2D3D, consists of 156 object instances organized into 14 categories [10]. The authors of this dataset also use a large set of 2D and 3D manually designed shape and color features. SVMs are trained for each feature and object class, followed by multilayer perceptron learning to combine the different features. The second dataset, Willow, contains objects from the Willow and Challenge datasets for training and testing, respectively [39]. Both training and test data contain 35 rigid, textured, household objects captured from different views by



**Fig. 6** Test images and the best matched images using HMP features



**Fig. 7** Ten of the thirty-five textured household objects from the Willow dataset

**Table 3** Comparisons with the previous results on the two public datasets: Willow and 2D3D

2D3D	Category Recognition			Willow	Instance Recognition
Methods	RGB	Depth	RGB-D	Methods	Precision/Recall
ICCVWorkshop [10]	66.6	74.6	82.8	ICRA12 [39]	96.7/97.4
This work	<b>86.3</b>	<b>87.6</b>	<b>91.0</b>	This work	<b>97.4/100.0</b>

Willow Garage. The authors present a processing pipeline that uses a combination of SIFT feature matching and geometric verification to perform recognition of highly textured object instances [39]. Note that 2D3D and Willow only contain highly textured objects.

We report the results of HMP in Table 3. Following the experimental setting in [10], HMP yields 91.0% accuracy for category recognition, much higher than the 82.8% reported in [10]. Learning models on training data from the Willow dataset and testing them on the training data from the Challenge dataset [39], HMP achieves higher precision/recall than the system proposed in [39], which won the 2011 Perception Challenge organized by Willow Garage. Note that that system is specifically designed for textured objects and thus could not, in contrast to our learned features, be applied to untextured objects such as those found in the RGBD dataset.

### 4.3 Learning and Vision Datasets

We also tested our model on the feature learning dataset *STL-10* [11] and on the vision dataset *MITScenes-67* [35]. We used the same architecture for these datasets as for RGB-D datasets. The dictionaries are learned on both RGB and grayscale channels and the final features are the concatenation of HMP features from these two channels. Following the standard setting in [11], we train linear SVMs on 1000 images and test on 8000 images using our HMP features and report the averaged accuracy over 10 pre-defined folds by the authors. As can be seen in Table 4, HMP achieves much higher accuracy than the receptive field learning algorithm [12] that beat many types of deep feature learning approaches as well as single layer sparse coding on top of SIFT (SC) [12]. Training linear SVMs on 80 images and testing on 20 images per category on the pre-defined training/test split by the authors, HMP achieves higher accuracy than many state-of-the-art algorithms: spatial pyramid matching (SPM) [32], deformable parts models (DPM) [14], object bank (OB) [25], Reconfigurable Models (RBoW) [33], and even the combination of SPM, DPM, and color GIST [32].

**Table 4** Comparisons with the previous results on the *STL-10* and *MITScenes-67*

STL-10		MITScenes-67			
VQ [11]	54.9 ± 0.4	GIST-color [32]	29.7	OB [25]	37.6
SC [12]	59.0 ± 0.8	DPM [14]	30.4	RBoW [33]	37.9
Learned RF [12]	60.1 ± 1.0	SPM [32]	34.4	DPM+Gist-color+SPM [32]	43.1
This work	64.5 ± 1.0	SC [8]	36.9	This work	47.6

## 5 Conclusions

We demonstrated that recent advances in unsupervised feature learning make it possible to learn very rich features from raw RGB-D data. Our approach, HMP, consistently outperforms state-of-the-art techniques on five benchmark datasets. Importantly, even though HMP is designed for very general object recognition, it even outperforms techniques specifically designed for highly textured objects, when applied to such data. These results are extremely encouraging, indicating that current recognition systems can be significantly improved without resorting to careful, manual feature design. We believe this work opens up many possibilities for learning rich, expressive features from raw RGB-D data. In the current implementation, we manually designed the architecture of HMP. Automatically learning such structure is interesting but also very challenging and left for future work. Our current experience is that learning dictionaries separately for each channel works better than learning them jointly. We plan to explore other possibilities of joint dictionary learning in the future.

**Acknowledgements.** This work was funded in part by the Intel Science and Technology Center for Pervasive Computing and by ONR MURI grant N00014-07-1-0749.

## References

1. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust Face Recognition via Sparse Representation. *IEEE PAMI* 31(2), 210–227 (2009)
2. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing* 54(11), 4311–4322 (2006)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110(3), 346–359 (2008)
4. Blum, M., Springenberg, J., Wülfing, J., Riedmiller, M.: A Learned Feature Descriptor for Object Recognition in RGB-D Data. In: *ICRA* (2012)
5. Bo, L., Lai, K., Ren, X., Fox, D.: Object Recognition with Hierarchical Kernel Descriptors. In: *CVPR* (2011)
6. Bo, L., Ren, X., Fox, D.: Kernel Descriptors for Visual Recognition. In: *NIPS* (2010)
7. Bo, L., Ren, X., Fox, D.: Depth Kernel Descriptors for Object Recognition. In: *IROS* (2011)
8. Bo, L., Ren, X., Fox, D.: Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms. In: *NIPS* (2011)
9. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning Mid-level Features for Recognition. In: *CVPR* (2010)
10. Browatzki, B., Fischer, J., Graf, B., Bühlhoff, H., Wallraven, C.: Going into Depth: Evaluating 2D and 3D Cues for Object Classification on a New, Large-scale Object Dataset. In: *1st ICCV Workshop on Consumer Depth Cameras in Computer Vision* (2011)
11. Coates, A., Lee, H., Ng, A.: An Analysis of Single-layer Networks in Unsupervised Feature Learning. In: *International Conference on AI and Statistics* (2011)
12. Coates, A., Ng, A.: The Importance of Encoding versus Training with Sparse Coding and Vector Quantization. In: *ICML* (2011)
13. Davis, G., Mallat, S., Avellaneda, M.: Adaptive Greedy Approximations. *Constructive Approximation* 13(1), 57–98 (1997)
14. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models. *IEEE PAMI* 32, 1627–1645 (2010)
15. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal Templates for Real-Time Detection of Texture-less Objects in Heavily Cluttered Scenes. In: *ICCV* (2011)
16. Hinton, G., Osindero, S., Teh, Y.: A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation* 18(7), 1527–1554 (2006)
17. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the Best Multi-Stage Architecture for Object Recognition? In: *ICCV* (2009)
18. Johnson, A., Hebert, M.: Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE PAMI* 21(5) (1999)
19. Kavukcuoglu, K., Sermanet, P., Boureau, Y., Gregor, K., Mathieu, M., LeCun, Y.: Learning Convolutional Feature Hierarchies for Visual Recognition. In: *NIPS* (2010)
20. Lai, K., Bo, L., Ren, X., Fox, D.: A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In: *ICRA* (2011)
21. Lai, K., Bo, L., Ren, X., Fox, D.: A Scalable Tree-based Approach for Joint Object and Pose Recognition. In: *AAAI* (2011)



22. Lai, K., Bo, L., Ren, X., Fox, D.: RGB-D Object Recognition: Features, Algorithms, and a Large Scale Benchmark. In: Fossati, A., Gall, J., Grabner, H., Ren, X., Konolige, K. (eds.) *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*. Springer (2012)
23. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient Sparse Coding Algorithms. In: *NIPS* (2007)
24. Lee, H., Grosse, R., Ranganath, R., Ng, A.: Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In: *ICML* (2009)
25. Li, L., Su, H., Xing, E., Fei-Fei, L.: Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification. In: *NIPS* (2010)
26. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 91–110 (2004)
27. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. In: *CVPR*, pp. 1–8 (2008)
28. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised Dictionary Learning. In: *NIPS*, pp. 1033–1040 (2008)
29. Maji, S., Berg, A., Malik, J.: Classification Using Intersection Kernel Support Vector Machines is Efficient. In: *CVPR* (2008)
30. Morisset, B., Bogdan Rusu, R., Sundaresan, A., Hauser, K., Agrawal, M., Latombe, J., Beetz, M.: Leaving Flatland: Toward Real-Time 3D Navigation. In: *ICRA* (2009)
31. Olshausen, B., Field, D.: Emergence of Simple-cell Receptive Field Properties by Learning a Sparse Code for Natural Images. *Nature* 381, 607–609 (1996)
32. Pandey, M., Lazebnik, S.: Scene Recognition and Weakly Supervised Object Localization with Deformable Part-Based Models. In: *ICCV* (2011)
33. Naderi Parizi, S., Oberlin, J., Felzenszwalb, P.: Reconfigurable Models for Scene Recognition. In: *CVPR* (2012)
34. Pati, Y., Rezaifar, R., Krishnaprasad, P.: Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In: *The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 (1993)
35. Quattoni, A., Torralba, A.: Recognizing Indoor Scenes. In: *CVPR* (2009)
36. Rubinstein, R., Zibulevsky, M., Elad, M.: Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit. Technical report (2008)
37. Ruhnke, M., Steder, B., Grisetti, G., Burgard, W.: Unsupervised Learning of Compact 3D Models Based on the Detection of Recurrent Structures. In: *IROS* (2010)
38. Salakhutdinov, R., Hinton, G.: Deep Boltzmann Machines. In: *International Conference on AI and Statistics* (2009)
39. Tang, J., Miller, S., Singh, A., Abbeel, P.: A Textured Object Recognition Pipeline for Color and Depth Image Data. In: *ICRA* (2012)
40. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and Composing Robust Features with Denoising Autoencoders. In: *ICML* (2008)
41. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Guo, Y.: Locality-constrained Linear Coding for Image Classification. In: *CVPR* (2010)
42. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. In: *CVPR* (2009)
43. Yu, K., Lin, Y., Lafferty, J.: Learning Image Representations from the Pixel Level via Hierarchical Sparse Coding. In: *CVPR* (2011)
44. Zeiler, M., Krishnan, D., Taylor, G., Fergus, R.: Deconvolutional Networks. In: *CVPR* (2010)



# Learning to Parse Natural Language Commands to a Robot Control System

Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox

**Abstract.** As robots become more ubiquitous and capable of performing complex tasks, the importance of enabling untrained users to interact with them has increased. In response, unconstrained natural-language interaction with robots has emerged as a significant research area. We discuss the problem of parsing natural language commands to actions and control structures that can be readily implemented in a robot execution system. Our approach learns a parser based on example pairs of English commands and corresponding control language expressions. We evaluate this approach in the context of following route instructions through an indoor environment, and demonstrate that our system can learn to translate English commands into sequences of desired actions, while correctly capturing the semantic intent of statements involving complex control structures. The procedural nature of our formal representation allows a robot to interpret route instructions online while moving through a previously unknown environment.

## 1 Motivation and Problem Statement

In this paper, we discuss our work on *grounding* natural language—interpreting human language into semantically informed structures in the context of robotic perception and actuation. To this end, we explore the question of interpreting natural language commands so they can be executed by a robot, specifically in the context of following route instructions through a map.

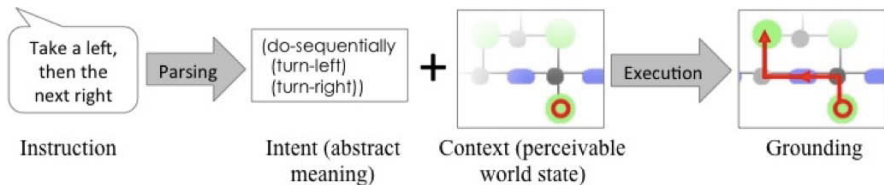
Natural language (NL) is a rich, intuitive mechanism by which humans can interact with systems around them, offering sufficient signal to support robot task planning. Human route instructions include complex language constructs, which robots must be able to execute without being given a fully specified world model such as a map. Our goal is to investigate whether it is possible to learn a *parser* that produces

---

Cynthia Matuszek · Evan Herbst · Luke Zettlemoyer · Dieter Fox

University of Washington, Seattle, USA

e-mail: {cynthia, eherbst, lsz, fox}@cs.washington.edu



**Fig. 1** The task: Going from NL to robot control. First, the natural language command is parsed into a formal, procedural description representing the intent of the person. The robot control commands are then used by the executor, along with the local state of the world, to control the robot, thereby grounding the NL commands into actions while exploring the environment.

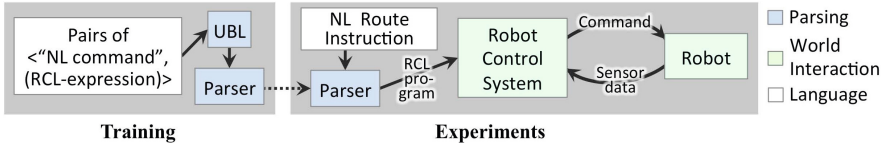
correct, robot-executable commands for such instructions. We treat grounding as a problem of parsing from a natural language to a formal control language capable of representing a robot’s operation in an environment. Specifically, we train a *semantic parsing* model that defines, for any natural language sentence, a distribution over possible robot control sequences in a LISP-like control language called Robot Control Language, or RCL.

The key contributions of this work are to learn grounding relations from data (rather than predefining a mapping of natural language to actions), and to execute them against a previously unseen world model (in this case, a map of an environment), as illustrated in Fig. 1. Training is performed on English commands annotated with the corresponding robot commands. This parser can then be used to transform new route instructions to execution system inputs in an unfamiliar map. The resulting system can represent control structures and higher-order concepts. We test our approach using a simulator executing the commands produced.

The remainder of this paper is organized as follows. In the next section, we discuss related work in human-robot interaction, natural language understanding, and robot navigation and instruction-following. Sec. 3 describes the technical approach, the formal execution language we define for this work, and our parser learning system. Sec. 4 and Sec. 5 describe the experimental evaluation performed and the results obtained, and we close with a discussion of insights gained from this work.

## 2 Related Work

Robot navigation is a critical and widely-studied task in mobile robotics, and following natural-language instructions is a key component of natural, multi-modal human/robot interaction. Previous efforts have treated the language grounding task as a problem of parsing commands into formal meaning representations. Several efforts [14, 23] parse natural route instructions to sequences of atomic actions that must be grounded into fully specified world models. Other systems learn to parse navigation instructions, but limit their formal language to a set of predefined parses [25].



**Fig. 2** The high-level architecture of the end-to-end system. Training is performed by learning a parser from English instructions to RCL. In the experimental phase, the learned parser maps NL instructions to an RCL program that is executed by the robot.

Our work falls also into the broader class of grounded language acquisition [24], in which language is learned from situated context, usually by learning over a corpus of parallel language and context data. Other work shows how parsed natural language can be grounded in a robot’s world and action models, taking perceptual and grounding uncertainty into account, thereby enabling instruction for robot navigation, GUI interaction, and forklift operation [30, 4, 28].

Parsing natural language to expressive formal representations such as  $\lambda$ -calculus has been demonstrated [31, 1, 18].  $\lambda$ -calculus is able to represent complex robot control systems [8]; however, to the best of our knowledge, such parser learning approaches have not yet been applied in the context of robotics. Logic-based control systems have been used successfully in robotics [11, 3, 5, 9, 16], providing a powerful framework that can be readily mapped to robot actions, and combinatory categorial grammars have been used for semantic mapping [21]; in contrast to our framework, however, most approaches rely on a manually constructed parser to map from NL commands to  $\lambda$ -calculus, rather than learning grounding relations from data.

Our work is most similar to that of Chen & Mooney [7, 6], who perform parser learning over a body of route instructions through a complex indoor environment containing objects and landmarks with no prior linguistic knowledge. However, their work assumes initial knowledge of a map, and does not represent complex control structures. Compared to our previous approach to parser learning for route instruction following, the system presented here can represent control structures such as ‘while,’ higher-order concepts such as ‘n<sup>th</sup>,’ and set operations, and is able to follow such directions through unknown maps.

### 3 Technical Approach

As noted in Sec. 2, much of the work on learning to parse natural language route instructions assumes previous knowledge of the map to be traversed. Intuitively, our approach is to parse natural language into a high level *specification* of robot behavior, which can then be executed in the *context* of an arbitrary environment. Differentiating context and meaning in this way lends itself well to tasks in robotics, where full knowledge of the environment is rarely available. Our approach also separates the parsing problem from execution, providing a more appropriate layer of abstraction for robot control. This architecture is shown in Fig. 2.

### 3.1 Robot Control Language

We model control structures in a logic-based Robot Control Language (RCL), inspired by task execution systems such as PRS [12], RPL [2], and GOLEX [11]. For a given set of route instructions, RCL represents the high-level execution intended by the person. Fig. 3 illustrates RCL programs for paths through an automatically generated, semantically-labeled map [10], along with the instructions corresponding to that path. (RCL is a subset of the typed lambda calculus, and therefore can be represented in a LISP-like format.)

The language is intended to be high-level enough that different execution systems can be used for various tasks, rather than specifying low-level actuation and perception tasks. As such, RCL expressions can encode actions with nontrivial sub-tasks such as combined perception and action. For example, `(exists left-loc)` is a high-level command to the execution system to determine whether there is an opening to the left, while `(move-to forward-loc)` tells the robot to move one node forward in the semantic map. This also allows the language to be quite compact; a full list of RCL commands is given in Fig. 3(a), including terms for logical expressions, queries about local state, and actions to be performed.

### 3.2 Parsing

RCL is a formal language defined by a grammar; *parsing* is the process of producing an expression in that grammar from some input. In our case, the input is natural language route instructions, and the parsing target is a statement in RCL that can be passed to a robot controller for planning or further disambiguation.

For this work, parsing is performed using an extended version of the Unification-Based Learner, UBL [17]. The grammatical formalism used by UBL is a probabilistic version of *combinatory categorial grammars*, or CCGs [27], a type of phrase structure grammar. CCGs model both the syntax (language constructs such as NP for noun phrase) and the semantics (expressions in  $\lambda$ -calculus) of a sentence. UBL creates a parser by inducing a probabilistic CCG (PCCG) from a set of training examples.

PCCG-based algorithms have several characteristics that make them a good choice for parsing NL instructions. They are robust against noise found in language, and they are able to efficiently generate  $n$ -best parses using CKY-style parsing [29, 18], allowing for jointly considering a parse model and a world model derived from sensor data when interpreting instructions into grounded action; next-best parse search can also be used for “fallback” exploration, e.g., when performing local error correction. Additionally, the probabilistic nature of PCCGs offers a clear objective for learning, that is, maximizing the conditional likelihood of training data.

Importantly, UBL can learn a parser solely from training data of the form  $\{(x_i, z_i)\}$ , where  $x_i$  is a natural-language sentence and  $z_i$  is a corresponding semantic-language sentence. In brief, UBL learns a model for  $p(z_i, y_i | x_i; \theta)$ , where  $\theta$  parameterizes the learned grammar  $G$  and  $y_i$  is a derivation in  $G$  ( $z_i$  is completely specified by  $y_i$ ). UBL uses a log-linear model:

locations	
current-loc:loc	robot's current position
forward-loc:loc	location ahead of robot
left-loc:loc	to robot's left
right-loc:loc	to robot's right
exists:t [loc]	does [loc] exist?
movement	
move-to:t [loc]	move to [loc]
turn-left:t	take next available left
turn-right:t	take next available right
logic	
and:t [t] [t]	boolean 'and'
or:t [t] [t]	boolean 'or'
not:t [t]	boolean 'not'
loops	
do-until:t [t] [e]	do [e] until [t] is true
do-n-times:t [n] [e]	do [e] [n] times
querying the type of a node	
room:t [loc]	is [loc] a room?
junction:t [loc]	is [loc] a junction?
junction3:t [loc]	is [loc] a 3-way junction?
junction4:t [loc]	is [loc] a 4-way junction?
hall:t [loc]	is [loc] of type hallway?
mid-level perception	
turn-unique-corner:t	take available turn
take-unique-exit:t	leave a room with one exit
other	
<#>:n	integers
do-sequentially:t e+	do each thing in turn
verify:t t	error if arg is false

<p><b>“Go left to the end of the hall.”</b></p> <pre>(do-sequentially   (turn-left     (do-until       (or         (not           (exists forward-loc))         (room forward-loc))       (move-to forward-loc)))</pre>
<p><b>“Go to the third junction and take a right.”</b></p> <pre>(do-sequentially   (do-n-times 3     (do-sequentially       (move-to forward-loc         (do-until           (junction current-loc             (move-to forward-loc))))))   (turn-right))</pre>
<p><b>“Go straight down the hallway past a bunch of rooms until you reach an intersection with a hallway on your left.”</b></p> <pre>(do-sequentially   (do-until     (and       (exists left-loc)       (hall left-loc))     (move-to forward-loc))   (turn-left))</pre>

(a)

(b)

**Fig. 3** (a) The complete list of terms in Robot Control Language. Hallways, rooms and intersections are treated as nodes of a map. The return type of each term is given after its name, followed by the types of any parameters: e (entity), t (boolean), n (number), loc (map location). (b) gives examples of English sentences from the test corpus and their RCL interpretations.

$$p(z_i, y_i | x_i; \theta) \propto e^{\theta \cdot \phi(x_i, y_i, z_i)}$$

UBL first generates a set of possibly useful *lexical items*, made up of natural language words, a  $\lambda$ -calculus expression, and a syntactic category. (An example lexical item might be <“left”, turn-left, S>.) The algorithm then alternates between increasing the size of this *lexicon* and estimating the parameters of  $G$  via stochastic gradient descent [20].

<u>go to</u>	<u>the</u>	<u>second</u>	<u>junction</u>	<u>and</u>	<u>go left</u>
$S/NP$	$NP/NP$	$NP/N$	$N$	$S \setminus S/S$	$S$
<i>(move-to forward)</i>	<i>[null]</i>	<i>(do-n-times 2 x)</i>	<i>(until (junction current-loc) y)</i>	<i>(do-seq g f)</i>	<i>(turn-left)</i>
		$NP$	$NP$	$S \setminus S$	
		<i>(do-n-times 2 (until (junction current-loc) y))</i>	<i>(do-n-times 2 (until (junction current-loc) y))</i>	<i>(do-seq g turn-left)</i>	
		$NP$	$NP$		
		<i>(do-n-times 2 (until (junction current-loc) y))</i>	<i>(do-n-times 2 (until (junction current-loc) y))</i>		
		$S$	$S$		
		<i>(do-n-times 2 (until (junction current-loc) (move-to forward)))</i>	<i>(do-n-times 2 (until (junction current-loc) (move-to forward)))</i>		
		$S$	$S$		
		<i>(do-seq (do-n-times 2 (until (junction current-loc) (move-to forward))) (turn-left))</i>	<i>(do-seq (do-n-times 2 (until (junction current-loc) (move-to forward))) (turn-left))</i>		

**Fig. 4** CCG parse of a test sentence performed by the learned parser. Here the natural language input is first, followed by alternating CCG syntactic categorization and  $\lambda$ -calculus logical forms. The bottom row shows the RCL program to be executed by the robot. (Some syntax has been changed for conciseness.)

Two types of features  $\theta$  are used. *Lexical* features fire when the associated lexical items are found in a training example (an example lexical item might be  $\langle$ “left”,  $turn-left$ ,  $S$  $\rangle$ ). *Semantic* features are functions of the logical RCL expression  $z_i$ . These are binary features that indicate the co-occurrence of different types of terms in  $z_i$ : predicates and their arguments, argument types, predicate co-occurrences, and shared variables. Once a parser is trained, parses are produced via *derivations*, using the learned lexical items and a small set of fixed production rules. Fig. 4 gives an example derivation of an RCL program (last line) from an input sentence (first line).

### 3.2.1 Parsing Extensions: Initialization of Lexicon and Parameters

In [17], the lexicon—the set of lexical entries and their weights—was initialized with entries covering the entirety of each training example: for each pair of terms found in  $(x_i, z_i)$ , one initial lexical entry was created. The model defined by  $\theta$  contained a parameter corresponding to each lexical item, and these weights were set using cooccurrence statistics of single words in the natural language with constants in the semantic language. For example, given the training example:

<u>exit the room and go left</u>
<i>(do-sequentially (take-unique-exit) (turn-left))</i>

the algorithm would count one cooccurrence for each of ( $\text{‘exit’}$ ,  $do-sequentially$ ), ( $\text{‘exit’}$ ,  $take-unique-exit$ ), ( $\text{‘exit’}$ ,  $turn-left$ ), and each other (NL-word, logical-term) pair. The more often such a pair cooccurred, the more important it was considered for parsing and so the higher its weight in the initial model.

In this work we use a new initialization that is better able to handle our semantic language. Intuitively, rather than generating lexical items from each word, we allow arbitrary subsequences of natural-language words, along with semantic subexpressions as defined by the splitting procedure of [17]. As before, this favors (NL, semantics) pairs that are very predictive of each other, while remaining tractable.

More specifically: for each training example  $i$ , we let  $W(x_i)$  be all subsequences of words in the NL sentence  $x_i$  (up to a fixed length,  $N = 4$  in our experiments).

We define  $L_i$  to be the initial *sentential CCG category* of the sentence  $x_i$ , having syntactic category  $S$  (sentence) and meaning  $z_i$ . Splits of  $L_i$  into CCG categories are recursively explored to depth two, yielding a set of possible syntactic sub-categorizations  $R$ . We then define three count functions:  $C(w \in W)$ , instances of phrase  $w$  in the training data;  $C(r \in R)$ , occurrences of each syntactic category in the training data; and  $C(w, r)$  as the cooccurrence count. The score for lexical entry  $(w, r)$  is then defined to be  $p(w|r) + p(r|w)$ , where probabilities are computed using counts.

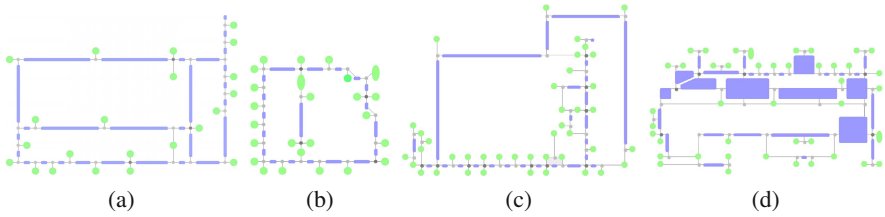
Two other improvements made to UBL involve synonyms and ordinals. During both training and evaluation, if UBL is unable to find a parse for a sentence, it tries to substitute known synonym lists from a standard dictionary for individual words. In addition, numbers are special-cased: when an ordinal (numbers such as ‘three’, or counting terms such as ‘second’) is encountered in a training sentence, it is replaced by a standard symbol, turning that training sentence into a *template* into which other ordinals can be substituted freely. As a result, not each ordinal term has to be encountered in all contexts in the training data.

## 4 Dataset and Maps

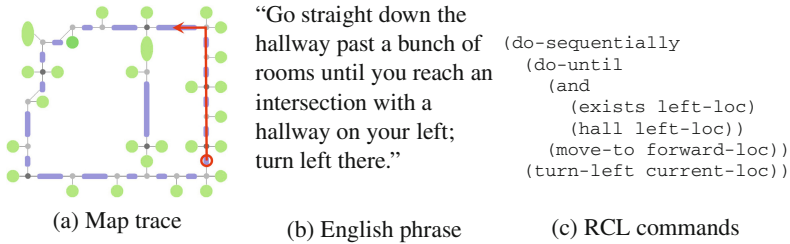
Our primary goal is to evaluate the ability of the system to generate RCL command sequences that allow a robot to navigate through a labeled map according to NL instructions. Maps are labeled according to area type (room, hallway, etc.), but are not known to the robot in advance. Instead, the robot explores the map simultaneously with following an RCL program. These experiments are performed in simulation.

We use four maps for our experiments, two each for training and testing (see Fig. 5). Each map has an associated set of routes through the map that have been described in English; for training and test purposes, each English route description is also paired with an associated RCL annotation. Maps A and B—the training and testing maps from earlier work—were automatically generated using Voronoi Random Fields [10] on data gathered by a SICK laser range-finder mounted on a Pioneer robot. Because these original maps were fairly simple, two additional manually constructed maps, C and D, were introduced for training and testing, in order to increase experimental complexity. All navigation experiments were performed in simulated execution of RCL commands in these maps.

Language datasets were generated as follows. First, English training and testing data from earlier work [23] was re-annotated in RCL; as in [23] and [7], all English instructions are segmented into individual movement phrases. This training set,  $S_{\text{base}}$ , contained 189 unique sentences generated by non-experts. We then supplemented this data with additional sentences taken from descriptions of four more complex routes in map C. Twelve non-experts supplied NL directions for those routes. The resulting *enriched* dataset,  $S_{\text{enr}}$ , contains 418 NL route instructions along with corresponding RCL command sequences, including structures requiring loops and counting—for example, “Go until you reach a four-way intersection”, or “Go into the ninth door on the right”. Fig. 6 shows an example of an NL route instruction set with RCL annotation.



**Fig. 5** Four maps were used: A, B, C, and D are respectively a computer science building, an industry research lab, an apartment building, and an academic support building at Cuesta College. Buildings C and D were selected for their relatively complex structure. Areas are color-coded according to type: green areas are rooms, blue areas are hallways, and dark and light gray circles are 4-way and 3-way intersections.



**Fig. 6** An example training/testing triplet. (a) The red line shows the path through map B; (b) a description of the path, written by a non-expert person; (c) the language’s RCL annotation in  $S_{\text{enr}}$ . In testing, the annotation is compared to the annotation produced by the learned parser.

## 5 Experiments

Experimentally, we are interested in whether the robot reaches the desired destination by exactly following the described path. Since all maps contain loops, any destination can be reached via an incorrect path; we consider this a failed trial. In addition to the generated RCL program, the robot begins at a known starting location and orientation; local knowledge of the map (adjacent and line-of-sight areas) is updated continuously as the robot progresses. A trial is considered successful only if the robot reached the correct destination along the route intended by the instructor, as our goal is to test instruction-following rather than navigation.

### 5.1 Baseline, Parser, and Route-Following Experiments

We report on three different experiments used to test the capabilities of our system: two which evaluate its ability to successfully guide a robot through a map, and one which evaluates the parsing component directly. Because the complete number of routes is small and does not necessarily capture map complexity, a large number of



test routes of various lengths were generated in the test map, and NL descriptions were generated using route instructions drawn from  $\mathbf{S}_{\text{base}}$  and  $\mathbf{S}_{\text{enr}}$  (see Fig. 7 for an example test route and description).

### 5.1.1 Baseline

As a baseline, we test our robot control system on the training, test, and map data used in our previous work, which only contains the 189 comparatively simple route instructions in  $\mathbf{S}_{\text{base}}$ . For this experiment, data was collected on Map A, and tested against Map B, using the evaluation paradigm from previous work [23]. Performance was virtually identical to that work, with the simulated robot successfully following 71.4% of route instructions. It is not surprising that our new technique did not significantly improve over the performance of the prior approach, as that data set was rather small and did not contain complex control structures (while loops, counting) requiring the additional capabilities provided by this work. Additionally, this accuracy was achieved using only collected training data, without amplifying the training set by adding a large number of hypothetical map routes to each natural language instruction, as was done in [23].

### 5.1.2 Parser Cross-Validation

In order to evaluate the parser learning system independent of route-following, we perform ten-fold cross-validation on the enriched dataset  $\mathbf{S}_{\text{enr}}$ , which contains more complex instructions. This test compares the learned parser output against the expert-created RCL annotation, rather than testing the full system against a map, and evaluates performance on individual sentences rather than sets of route instructions. Table 1 shows precision, recall, and F1 score. A parse is reported to be correct if it matches the human-supplied RCL program for that phrase, so this test shows the frequency with which the parsing system recovers the exact RCL program of unseen commands.

**Table 1** Precision, recall, and F1 on cross-validation tests of the extended UBL parser learner

data set	precision	recall	F1 score
enriched	71.0%	72.6%	71.8%

### 5.1.3 Route Following with Complex Language

The primary experiment of this work is to test following more difficult route instructions through the complex map D. To determine directly whether this framework can be used to produce executable Robot Control Language, we performed end-to-end tests on a large number of paths through map D in Fig. 5. For each test,  $\mathbf{S}_{\text{enr}}$  is split into training and test data by *participant*: all English instructions from some number of non-expert instruction givers ( $N = 2$  in our experiments) is withheld from training and used to create a test set  $\mathbf{S}_{\text{test}}$ . The training set is then  $\mathbf{S}_{\text{train}} = \mathbf{S}_{\text{enr}} - \mathbf{S}_{\text{test}}$ , or the set of all sentences collected from the remaining participants.

For each of 10 different sets  $\mathbf{S}_{\text{test}1-10}$  (every combination of five different participants), 1,200 paths through map D were randomly generated: 1,000 short paths described by single NL sentences, and 200 paths that required multiple sentences to express (5, on average). An example test path through the map with NL instructions is shown in Fig. 7. The learned parser is asked to interpret these novel route instructions into RCL commands, which were executed by the simulated robot. Table 2 gives a summary of the results.

**Table 2** Testing the end-to-end system on 1,000 short and 200 longer sets of route instruction. Longer routes average five sentences/route; examples of long and short routes and their associated RCL programs can be seen in Fig. 7 and Fig. 3(b).

data set	success (short)	success (long)
enriched	$66.3 \pm 10.7\%$	48.9%

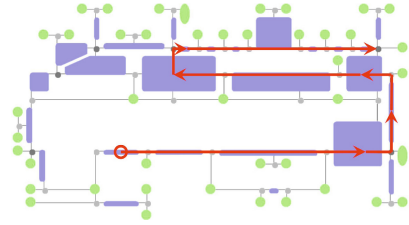
## 5.2 Discussion

Execution of complex language instructions is successful in approximately 66% of short paths, and 49% of complex paths. In general, longer paths are more likely to fail than shorter ones: our simulated control system does not attempt any local error recovery if it encounters an unexpected situation, such as discovering that the current map location is not of the expected type or failing to find an opening to the left after a (`turn-left`) instruction. Intuitively, this metric does not give any credit for partly-correct parses of route instructions, e.g., cases where the robot *almost* reaches the destination.

The high variability of results is likely a product of the experimental setup, in which  $\mathbf{S}_{\text{test}}$  is generated from statements by people who did not contribute to training data, making the learned parser vulnerable to forms of instruction that are idiosyncratic to an individual. Given that each person contributed route instructions for only four routes (five sentences each, on average), it is also possible that only one or two unparseable instructions occur in a particular data set  $\mathbf{S}_{\text{test}}$ , such that only randomly-generated paths containing that instruction fail. Both failure modes could be improved by either gathering training data from a wider range of people (for example, using Amazon Mechanical Turk), or by splitting  $\mathbf{S}_{\text{test}}$  from  $\mathbf{S}_{\text{enr}}$  by *paths*, allowing an individual to contribute route instructions to both training and testing data, but describing different paths or different maps.

Fig. 4 demonstrates the successful derivation of an RCL command from a short test route instruction using our learned parser. As shown, the parser is able to correctly express concepts such as counting ('second') and can generate differing, contextually correct interpretations of the word 'go' (move vs. turn). Fig. 7 shows an example of a long path description that has been successfully parsed by our system.

“Go down the long hallway past three intersections, turn left, take the hallway to your left, go through two intersections, turn right, and go forward until you reach the end of the hall.”



**Fig. 7** (left) Example route instruction and (right) corresponding path through map D. Our approach is able to correctly parse 62.5% of such complex instructions into RCL programs that control a simulated robot all the way from the start to the end of such a path.

## 6 Experimental Insights

Several conclusions can be drawn from these experiments. First, it is possible to learn a parser able to handle complex, procedural natural language commands for robot instruction. This makes it possible to target a rich robot control language capable of handling abstract notions such as counting, loops, and conditionals. We experimentally demonstrate that such a system is able to transform complex natural-language instructions into a language suitable for use by a robot control system.

In this work we demonstrate that it is possible to combine advanced natural language processing with robotic perception and control systems. As an example consider the different, contextually appropriate interpretations of the word ‘go’ in Fig. 4, where the system learned to interpret ‘go to’ and ‘go left’ as having quite different meanings. We established that parsing into a procedural language does require extensions to the parser learning process. We also find that local error recovery would improve results notably, as would a larger, more diverse natural language corpus to train over. In future work, the latter might be obtained from Amazon Mechanical Turk, a system which allows web-based distribution of small tasks to a group of workers in exchange for small payments [13], which it has been used successfully for crowdsourcing natural language data-gathering [26].

This work raises a number of possible directions for future research. It will be worthwhile to see how this system behaves with more complex data that is still drawn from a real robot; for example, using landmarks retrieved from developing object recognition systems [19]. While the effort involved in expert annotation of natural language is far less than that of writing control systems for the tasks, it still requires experts to be involved in the teaching process. Learning solely from execution traces, as in [6], has the potential to increase the efficiency of learning.

We find these results extremely encouraging toward grounding complex NL instructions. The focus of this work is on parsing complex natural language into a formal Robot Control Language. As such, it is complementary to recent efforts in mapping such representations to actions and perceptions in the real world [28, 15, 22]. Since our parser is probabilistic in nature, it can also generate a ranked list of

possible RCL programs, each of which could generate a joint model for grounding. We believe that such an approach would enable robots to execute very complex natural language commands.

**Acknowledgements.** This work was funded in part by the Intel Science and Technology Center for Pervasive Computing, through collaborative participation in the Robotics Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program (Cooperative Agreement W911NF-10-2-0016), and by NSF grant IIS-1115966.

## References

1. Artzi, Y., Zettlemoyer, L.S.: Bootstrapping semantic parsers from conversations. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing (2011)
2. Beetz, M., Arbuckle, T., Belker, T., Bennewitz, M., Burgard, W., Cremers, A.B., Fox, D., Grosskreutz, H., Hähnel, D., Schulz, D.: Integrated plan-based control of autonomous service robots in human environments. *IEEE Intelligent Systems* 16(5) (2001)
3. Boutilier, C., Reiter, R., Soutchanski, M., Thrun, S.: Decision-theoretic, high-level agent programming in the situation calculus. In: Proc. of the National Conference on Artificial Intelligence, AAAI (2000)
4. Branavan, S.R.K., Zettlemoyer, L., Barzilay, R.: Reading between the lines: Learning to map high-level instructions to commands. In: *ACL*, pp. 1268–1277 (2010)
5. Burgard, W., Cremers, A.B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S.: Experiences with an interactive museum tour-guide robot. *Artificial Intelligence* 114(1-2), 3–55 (1999)
6. Chen, D.L.: Fast online lexicon learning for grounded language acquisition. In: Proc. of the Annual Meetings of the Association for Computational Linguistics, *ACL* (2012)
7. Chen, D.L., Mooney, R.: Learning to interpret natural language navigation instructions from observations. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011), pp. 859–865 (2011)
8. Dzifcak, J., Scheutz, M., Baral, C., Schermerhorn, P.: What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In: Proc. of the 2009 IEEE Int'l Conf. on Robotics and Automation, *ICRA 2009* (2009)
9. Ferrein, A., Lakemeyer, G.: Logic-based robot control in highly dynamic domains. *Robotics and Autonomous Systems* 56(11) (2008)
10. Friedman, S., Pasula, H., Fox, D.: Voronoi random fields: Extracting topological structure of indoor environments via place labeling. In: *IJCAI*, pp. 2109–2114 (2007)
11. Hähnel, D., Burgard, W., Lakemeyer, G.: GOLEX - bridging the gap between logic (GOLOG) and a real robot. In: Proc. of the German Conference on Artificial Intelligence (KI), Germany (1998)
12. Ingrand, F., Chatila, R., Alami, R., Robert, F.: PRS: A high level supervision and control language for autonomous mobile robots. In: Proc. of the IEEE International Conference on Robotics & Automation, *ICRA* (1996)
13. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: Proc. of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems, *CHI 2008*. ACM (2008)

14. Kollar, T., Tellex, S., Roy, D., Roy, N.: Toward understanding natural language directions. In: HRI 2010: Proc. of the 5th Int'l Conf. on Human-Robot Interaction. ACM Press (2010)
15. Kress-Gazit, H., Fainekos, G., Pappas, G.: Translating structured english to robot controllers. *Advanced Robotics* 22(12) (2008)
16. Kress-Gazit, H., Wongpiromsarn, T., Topcu, U.: Correct, reactive robot control from abstraction and temporal logic specifications. *IEEE Robotics and Automation Magazine*, special issue on Formal Methods for Robotics and Automation 18(3), 65–74 (2011)
17. Kwiatkowski, T., Zettlemoyer, L.S., Goldwater, S., Steedman, M.: Inducing probabilistic CCG grammars from logical form with higher-order unification. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing (2010)
18. Kwiatkowski, T., Zettlemoyer, L.S., Goldwater, S., Steedman, M.: Lexical generalization in CCG grammar induction for semantic parsing. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing (2011)
19. Lai, K., Bo, L., Ren, X., Fox, D.: A scalable tree-based approach for joint object and pose recognition. In: The AAAI Conference on Artificial Intelligence, AAAI (2011)
20. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278–2324 (1998)
21. Lison, P., Kruijff, G.-J.M.: An integrated approach to robust processing of situated spoken dialogue. In: Proc. of SRSL 2009, the 2nd Workshop on Semantic Representation of Spoken Language, Athens, Greece, pp. 58–65. Association for Computational Linguistics (March 2009)
22. Matuszek, C., FitzGerald, N., Zettlemoyer, L., Bo, L., Fox, D.: A Joint Model of Language and Perception for Grounded Attribute Learning. In: Proc. of the 2012 International Conference on Machine Learning, Edinburgh, Scotland (June 2012)
23. Matuszek, C., Fox, D., Koscher, K.: Following directions using statistical machine translation. In: ACM/IEEE International Conference on Human-Robot Interaction, HRI (2010)
24. Mooney, R.: Learning to connect language and perception. In: Fox, D., Gomes, C.P. (eds.) Proc. of the Twenty-Third AAAI Conf. on Artificial Intelligence, AAAI 2008, Chicago, Illinois, pp. 1598–1601. AAAI Press (2008)
25. Shimizu, N., Haas, A.: Learning to Follow Navigational Route Instructions. In: Int'l Joint Conf. on Artificial Intelligence, IJCAI (2009)
26. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In: EMNLP, pp. 254–263 (2008)
27. Steedman, M.: *The Syntactic Process*. MIT Press (2000)
28. Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., Roy, N.: Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine* 32(4) (2012)
29. Tsuruoka, Y., Tsujii, J.: Iterative CKY parsing for probabilistic context-free grammars. In: Su, K.-Y., Tsujii, J., Lee, J.-H., Kwong, O.Y. (eds.) *IJCNLP 2004*. LNCS (LNAI), vol. 3248, pp. 52–60. Springer, Heidelberg (2005)
30. Wei, Y., Brunskill, E., Kollar, T., Roy, N.: Where to go: Interpreting natural directions using global inference. In: Proc. of the IEEE International Conference on Robotics & Automation, ICRA (2009)
31. Zettlemoyer, L.S., Collins, M.: Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence (2005)

# A Data-Driven Statistical Framework for Post-Grasp Manipulation

Robert Paolini, Alberto Rodriguez,  
Siddhartha S. Srinivasa, and Matthew T. Mason

**Abstract.** Grasping an object is usually only an intermediate goal for a robotic manipulator. To finish the task, the robot needs to know where the object is in its hand and what action to execute. This paper presents a general statistical framework to address these problems. Given a novel object, the robot learns a statistical model of grasp state conditioned on sensor values. The robot also builds a statistical model of the requirements of the task in terms of grasp state accuracy. Both of these models are constructed by offline experiments. The online process then grasps objects and chooses actions to maximize likelihood of success. This paper describes the framework in detail, and demonstrates its effectiveness experimentally in placing, dropping, and insertion tasks. To construct statistical models, the robot performed over 8000 grasp trials, and over 1000 trials each of placing, dropping and insertion.

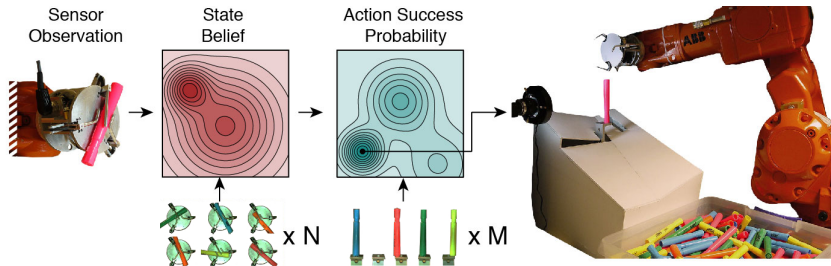
## 1 Introduction

Knowledge of the grasp state is often critical to any subsequent manipulation task. Intuitively, harder tasks demand a more accurate estimation of the state of a grasp than simpler ones. For example, balancing a cylinder on a table requires more accuracy than dropping it into a hole. More generally, consider a manipulator, an object to manipulate, a task, and a set of actions designed to accomplish the task. In this paper we build a data-driven framework to automate the process of deciding whether the task is solvable with the available hardware and set of actions, and find the action most likely to succeed.

The statistical framework proposed in this paper is best suited to model the execution of tasks that require grasping an object prior to execution, i.e., post-grasp

---

Robert Paolini · Alberto Rodriguez · Siddhartha S. Srinivasa · Matthew T. Mason  
The Robotics Institute,  
Carnegie Mellon University, Pittsburgh, USA  
e-mail: {rpaolini, albertor}@cmu.edu,  
          {siddh, matt.mason}@cs.cmu.edu



**Fig. 1** Procedure to choose the optimal action to accomplish a manipulation task. First, we learn the belief of the state of the system from sensor readings. Based on that belief, we then estimate the probability of success of available actions and choose on the best action to take. Both the state estimation and task requirements are learned using real data.

manipulation tasks. We address the problem by separating it into two independent steps. First, estimate the state of the grasp with in-hand sensors, and second, model the accuracy requirements that the particular task imposes on our state estimation. This separation yields the benefit that we can use the same model of state estimation for different tasks, and the same model of task requirements for different manipulators. Using this framework, each sensor reading generates a probability distribution in task action space, enabling us to find not only the optimal action, but to understand just how likely that action is to succeed.

Figure 1 illustrates the process for placing an object. Sensors in the hand provide information of the grasp state. First, we estimate the probability distribution of the pose of the object in the hand. Second, we predict the probability of success of each available action. Both of these are computed based on data-driven models. Finally, we choose the action most likely to succeed.

In this paper, we test the framework with three different manipulation tasks: placing an object, dropping it into a hole, and inserting it. The experimental setup in Figure 1 consists of a simple gripper [17, 14] mounted on a robotic arm that iteratively grasps an object from a bin, estimates the distribution of the pose of the object, computes the probability of success for all available actions, chooses the optimal one, and executes it.

For the experiments in this paper, the manipulated objects are highlighter markers. The chosen state representation is that of a symmetrical cylinder in the plane as parametrized by the polar coordinates  $x = (r, \theta)$  of its axis. That is  $x \in X = \mathbb{R} \times \mathbb{SO}(2)$ .

## 2 Related Work

In this paper, we study how uncertainty affects post-grasp manipulation. While post-grasp manipulation has not been extensively explored, grasp planning with

uncertainty is a well studied problem. Brost [3] found planar grasps that succeed even with large uncertainty in object position by exploiting friction. Goldfeder and Allen [9] approached the problem of grasp planning from a data-driven perspective. Goldberg and Mason [8] proposed a Bayesian framework to model uncertainty propagation in manipulation plans and apply it to grasping. Stulp et al. [19] learned motion primitives to optimize the chance of grasping an object with Gaussian uncertainty on its location.

The closest work to this paper is probably by Brost and Christiansen [4], who provided a framework for probabilistic analysis of manipulation tasks to overcome the shortcomings of the worst-case-configuration-space approach to manipulation. They applied the framework to plan grasps of planar objects with a parallel jaw gripper. Dogar and Srinivasa [6] applied a similar idea to clutter and uncertainty in the context of push-grasping.

Some work has been done on analyzing the grasp outcome as well. Morales et al. [15] used real grasps on a collection of objects to predict the reliability of the grasp process. Kang and Goldberg [12] used a random sequence of parallel-jaw grasps to classify grasped objects using a Bayesian process.

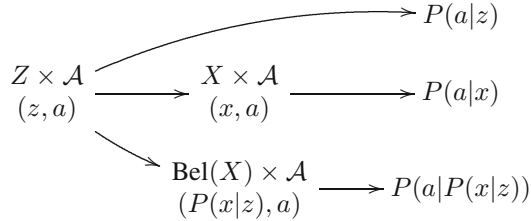
There is plenty of work on statistical frameworks to model uncertainty. POMDPs [5] (Partially Observable Markov Decision Processes) are a general framework to optimally plan under uncertainty by tracking the distribution of a system under some provided state representation. Hsiao et al. [10] used a POMDP framework to track the belief of the pose of an object and tactile exploration to localize it by planning among grasping and information-gathering trajectories. PSRs [20, 1] (Predictive State Representation) are also introduced as a general framework to learn compact models directly from sequences of action-observation pairs without the need for a hand-selected state representation. Lavalle [13] introduced information-spaces to formalize the process of propagating uncertainty along motion strategies.

In the context of post-grasp manipulation, Jiang et al. [11] looked at scenes to determine good locations to place objects. However, they did not study how robust the final process of actually placing an object is, which is the subject of our work. Fu et al. [7] addressed the problem of batting an object to a goal in the presence of uncertainty. They first maximized information gain in an observation step, and then chose the action most likely to succeed.

### 3 Statistical Framework

Our goal in this paper is to find the action  $a$  from a set of available actions  $\mathcal{A}$  that, given sensor inputs  $z \in Z$ , maximizes the expected performance of accomplishing a task. The following diagram illustrates three different paths to approach the problem:





The first path proposes to model the performance of an action directly as a function of sensor observations. How likely a specific action is to succeed and the action to execute are decided based upon the history of sensor readings. It makes the least assumptions about the system but is also the most difficult to implement, since the complexity of the model depends strongly on the dimension of the sensor space, which might be large.

In the second path, sensor inputs  $z$  are first projected into a more compact representation of state, noted here by  $x$ . In this work, we chose  $x$  to be the pose of the grasped object. The probability of success of an action is then modeled as a function of the most likely pose of the object rather than the sensor observations  $z$  directly. The intermediate representation  $x$  potentially reduces the model complexity, since the dimension of state space is generally smaller than that of sensor space. On the other hand, it introduces the possibility of information loss or lack of observability. Note that it also fails to address uncertainty in the system induced by noisy sensors.

In this paper, we implement the third path, which encapsulates uncertainty by representing the system by its state belief  $P(x|z)$  rather than just by its most likely value  $x$ . By maintaining the distribution of all possible poses of the object, we can later make a more informed prediction on the probability of success of a given action.

The dimension of the space of belief distributions  $\text{Bel}(X)$  is too large to model the probability of success of an action  $P(a|z)$  directly from the belief  $P(x|z)$ . We simplify this problem by marginalizing the probability of success of an action  $P(a|z)$  with respect to the true state of the system  $x$ :

$$P(a|z) = \int_X P(a|z, x) \cdot P(x|z) dx = \int_X P(a|x) \cdot P(x|z) dx \quad (1)$$

where in the last step, we make the assumption that the state representation  $x$  is informative enough such that the output of an action is conditionally independent of  $z$ , given the true state  $x$ .

We will show later that this assumption is key to enable the computation of the probability of success  $P(a|z)$ . Note, however, that for some tasks the pose of an object is not always fully representative of the grasp state. For example, in a compliantly actuated gripper, the state of the actuators also contains information on how tight the grasp is, which might be relevant to determine the outcome of an action.

Equation (1) models the performance of an action as a function of both  $P(x|z)$  and  $P(a|x)$ . These characterize the *sensing capabilities* of the gripper and the

precision *task requirements* for a successful task execution, respectively. The following subsections detail the approach to model them, as well as the process to combine them to give an accurate estimation of  $P(a|z)$ .

### 3.1 Learning Sensing Capabilities

The shape of the posterior distribution  $P(x|z)$  of the grasp state depends on several factors, including the geometries of the manipulator and object, the location and type of sensors, and the type of grasp. Assuming fixed geometries for the manipulator, object, and sensors, we will see that in general, different grasps yield different shaped belief distributions. We will pay special attention to the sharpness of those distributions, as an indicator of the confidence we get on the pose of the object.

In this section, we describe the process to model  $P(x|z)$  from data. Learning  $P(x|z)$  directly is usually expensive in terms of the amount of data required, since it can be arbitrarily shaped and the complexity of the model depends on the dimension of sensor space. To simplify the process, we use Bayes rule to flip the conditioning in  $P(x|z)$  to  $P(z|x)$ , the likelihood of the system. The likelihood is the distribution of sensor readings given the true state of the system, which is usually unimodal and we assume here to follow a Gaussian distribution  $P(z|x) \sim \mathcal{N}(z; \mu(x), \sigma^2(x))$ . This leads us to the following equation for our posterior distribution:

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)} \simeq \mathcal{N}(z; \mu(x), \sigma^2(x)) \cdot \frac{P(x)}{P(z)} \quad (2)$$

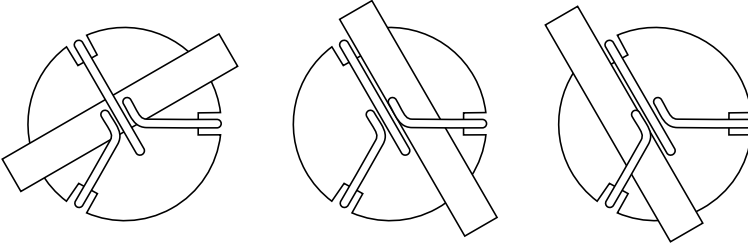
where  $P(x)$  is the prior state distribution, and both  $\mu$  and  $\sigma$  are functions of the true state of the system  $x$ . Given that  $P(z)$  is independent of  $x$ , we can obviate it and normalize  $P(x|z)$  a posteriori. We now detail the process to estimate the prior  $P(x)$  and posterior  $P(x|z)$  distributions from a collected dataset  $C_1 = \{(z^i, x^i)\}_i$  of pose/sensor readings pairs. Figure 3 shows the data collected for 2000 grasps .

#### 3.1.1 Prior Distribution

The prior distribution  $P(x)$  is the distribution of the state of the system before considering any information in the sensor readings. We regress  $P(x)$  by estimating the density of the pose of the object in state space. We use Kernel Density Estimation that models  $P(x)$  as a sum of kernels:

$$P(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x^i}{h}\right) \quad (3)$$

where  $K$  is a Gaussian kernel,  $h$  is the bandwidth parameter and  $x^i$  are the state points in the dataset  $C_1$ . The bandwidth parameter is chosen automatically to minimize the mean integrated squared error following the algorithm in Botev et. al [2].



**Fig. 2** The three most stable configurations of the object/gripper pair used our experiments. We label them from left to right as clusters I, II and III.

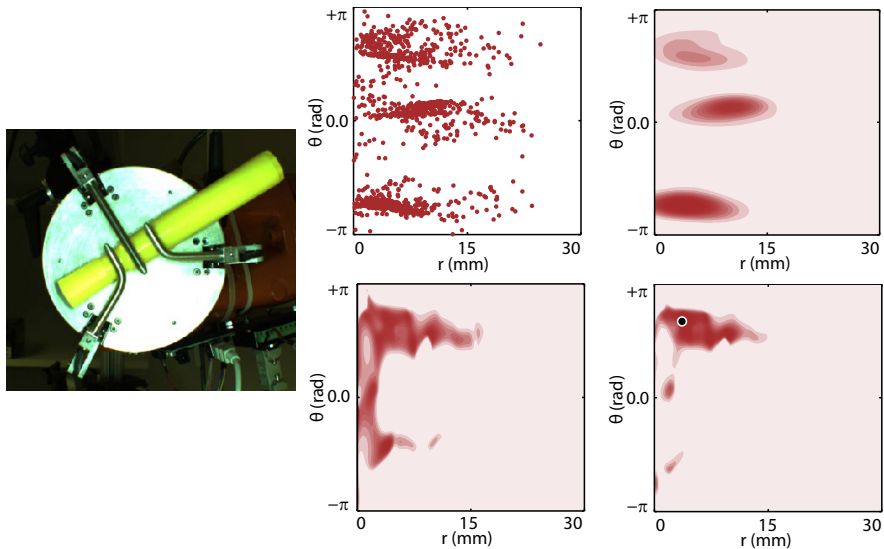
In this paper, the state of the system is represented by the coordinates  $(r, \theta)$  of the axis of a grasped cylindrical object. The prior grasp distribution in Figure 3 shows three main clusters. These correspond to the three most stable grasps yielded by the combined geometries of object and gripper, shown in Figure 2. The expectation is for grasps in cluster II and III to be the most stable and informative, since one of the fingers acts as an alignment feature.

### 3.1.2 Posterior Distribution

Equation (2) expresses the posterior distribution as a function of  $\mu$  and  $\sigma$ . Here we use Gaussian Processes (GP) [16] to regress them as functions of  $x$ . For that we again use the dataset  $C_1$ . The process is detailed in the following steps:

1. Use a GP on the first half of the data points in  $C_1$  to estimate the mean of the likelihood or observation model  $P(z|x), \mu : X \rightarrow \mathcal{Z}$ . This provides the most likely set of observed sensor readings for every possible state of the system  $x$ . This implies training one independent GP for every sensory input, as a function of  $r$  and  $\theta$ . Note that we will get a better estimation of the observation model for the regions of the state space that are most often observed, since those areas will be more populated with the collected data.
2. Complement the second half of the dataset  $C_1$  with the sensor readings  $z_0^i = \mu(x^i)$  predicted by the learned observation model, and the squared error yielded by that prediction  $\Delta^2 z^i = (z^i - z_0^i)^2$ ,  $C_1^+ = \{(z^i, x^i, z_0^i, \Delta^2 z^i)\}_i$
3. Use GPR on  $C_1^+$  to regress the variance of the observation model  $\sigma^2 : X \rightarrow \Delta^2 \mathcal{Z}$ . Again, this implies training one independent GP for every sensor in the system.

By following these steps, we can now estimate  $P(z|x)$  as  $\mathcal{N}(z; \mu(x), \sigma^2(x))$ . Figure 3 shows the likelihood  $P(z|x)$  for an example grasp and the corresponding posterior distribution  $P(z|x)$ .



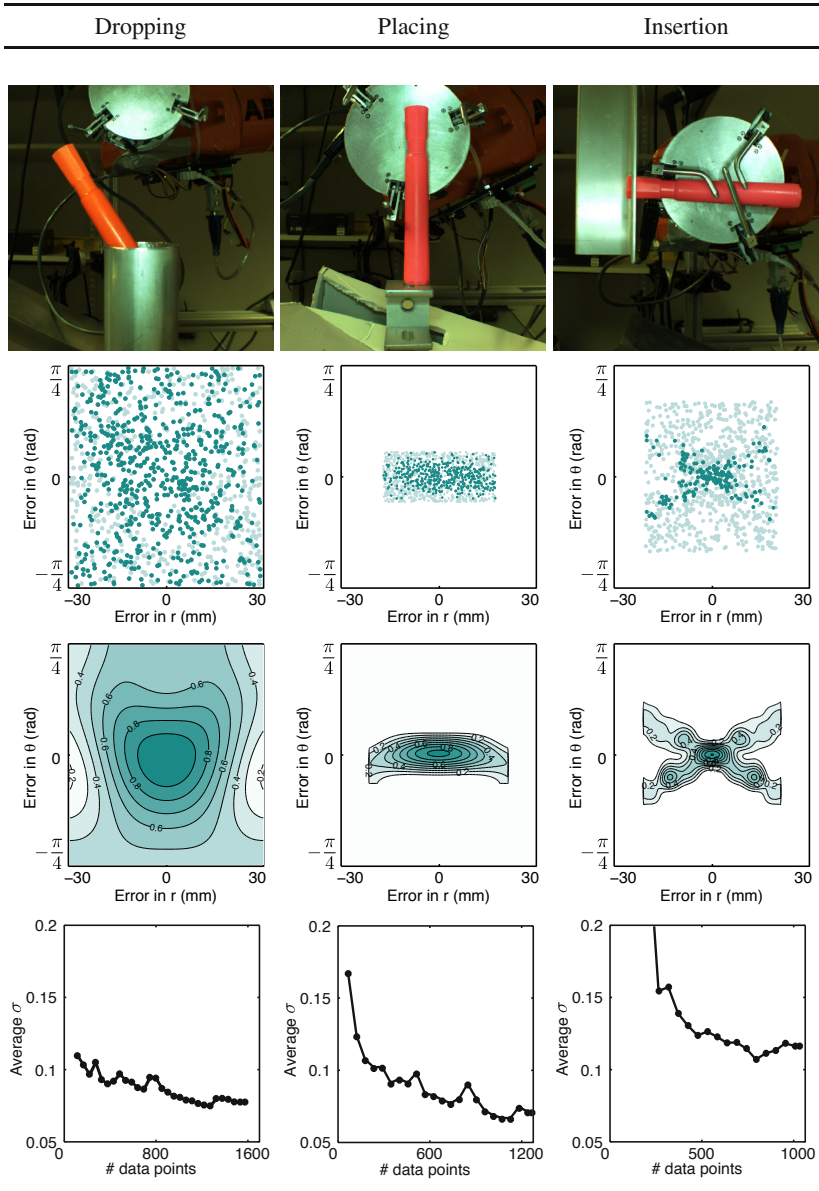
**Fig. 3** [Top-center] State from 2000 grasps collected to model the *sensing capabilities* of the manipulator. [Top-right] Prior state distribution  $P(x)$ . [Bottom-center] Likelihood of pose of the object for an example grasp in cluster I. [Bottom-right] Corresponding posterior distribution  $P(x|z)$  for the example grasp, and the most likely pose of the object.

### 3.2 Learning Task Requirements

We now model the probability of success of an action,  $P(a|x)$ . This will tell us how accurate our estimation of the state of the grasp must be for an action to successfully execute a task. While not required for our framework, we choose to state parameterize the set of actions. For example, for the task of placing a cylindrical object, if assumed to be at pose  $p$ , we design an action  $a_p$  that turns the cylinder so it is upright with respect to the ground, and then set it down. Note that  $a_p$  is an action parameterized by a chosen state  $p$ .

In general, the success of an action depends both on the action  $a_p$  itself and the true state of the system  $x$ . Since we assume state parametrized actions, we assume that the probability of success only depends on the difference  $(x - p)$ . For example, when placing a cylinder whose estimated axis is 1 degree off from its true state, we are more likely to succeed than if we try to place an object several degrees off. We model the outcome of an action  $a_p$  as a Bernoulli random variable of parameter  $\phi_{a_p}$ , so that:

$$P(a_p = 1|x) = \phi_{a_p}(x) = \phi(x - p) \quad (4)$$



**Fig. 4** [1st Row] Examples of successful executions of dropping, placing, and insertion. [2nd Row] Datapoints from perturbed task experiments. Dark points are successes and light ones are failures. Notice that the magnitude of the perturbation noise is different for each task. [3rd Row] Distribution of task requirements  $P(a_p = 1|x)$  for dropping, placing and insertion, as a function of the error in state estimation. [4th Row] Average standard deviation of the regression of the Bernoulli parameter of  $P(a_p = 1|x)$  obtained with a GP. This is used as a rough estimate of the convergence of the algorithm and stop criteria.

The use of state parameterized actions allows us to randomly sample the space of mismatches ( $x - p$ ) by choosing to execute the action  $a_p$  with  $p = x + \epsilon$ , where  $\epsilon$  is a uniformly distributed error in the space of system states instead of the optimal one  $a_x$ .

We next detail the process of estimating the function  $\phi(\epsilon)$  from a dataset  $C_2 = \{(z^i, x^i, \epsilon^i, y^i)\}_i$ , where  $\epsilon^i$  is the error in system state and  $y^i$  is the success/failure output of the trial. For each task, we uniformly sample  $\epsilon^i$  from  $\mathcal{E} = [-\Delta r_{\max}, \Delta r_{\max}] \times [-\Delta \theta_{\max}, \Delta \theta_{\max}]$ , where  $\Delta r_{\max}$  and  $\Delta \theta_{\max}$  are chosen to be large enough where we expect failure for anything outside of that range. We use a GP on  $C_2$  to regress the Bernoulli parameter  $\phi$  with the outcome of over 1000 executions of each task.

Figure 4 shows the requirements  $\phi(x - p)$  for three different manipulation tasks: dropping, placing and insertion of a highlighter marker with a simple hand. As  $|x - p|$  increases, our likelihood of task success decreases, which is as expected. Note that for the dropping task the probability decreases much slower than placing or insertion. This indicates that dropping a marker into a hole is easier than balancing it on a platform or inserting into a small hole. We can be more inaccurate and still succeed at dropping. Insertion is also interesting, as it resembles the shape of an X. This can be explained by noticing that if we incorrectly try to insert the marker too high, but also tilted downward, the end of the marker still manages to fit in the hole. Generating these task requirement distributions allows us to gain key insight into how robust our post-grasp manipulation tasks are.

In general, the more data, the more accurate the regressed distributions of task requirements are. The magnitude of the variance returned by the Gaussian Process Regression can be used to define a stopping criteria. In our case, we use the average standard deviation to assess how certain we are about the learned distribution. The bottom graphs in Figure 4 shows how the average standard deviation changes with the number of experiments for each task.

### 3.3 Matching Task Requirements with Sensing Capabilities

Here we combine the models of  $P(x|z)$  and  $P(a|x)$  to estimate the probability of success of an action  $a_p$ . For that, we extend (1) as:

$$\begin{aligned} P(a_p = 1|z) &= \int_X P(a_p = 1|x)P(x|z)dx \\ &= \int_X \phi(x - p)\mathcal{N}(z; \mu(x), \sigma(x)^2) \frac{P(x)}{P(z)} dx \\ &\simeq \int_{\mathcal{E}} \phi(\epsilon)\mathcal{N}(z; \mu(p + \epsilon), \sigma(p + \epsilon)^2) \frac{P(p + \epsilon)}{P(z)} d\epsilon \end{aligned} \quad (5)$$

where we apply the change of variables  $\epsilon = x - p$ .

In the experiments we approximate the integral numerically. If we grid the space of mismatches between real state and estimated state into  $N_r \times N_\theta$ , and call the deviation from the center  $\epsilon_{ij}$ , then we can approximate the integral in (5) by the following convolution:

$$P(a_p = 1|z) \simeq \sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} \phi(\epsilon_{ij}) \mathcal{N}(z; \mu(p + \epsilon_{ij}), \sigma(p + \epsilon_{ij})) \frac{P(p + \epsilon_{ij})}{P(z)} \Delta A \quad (6)$$

where

$$\Delta A = \frac{4\Delta r_{\max} \Delta \theta_{\max}}{(N_\theta - 1)(N_r - 1)}$$

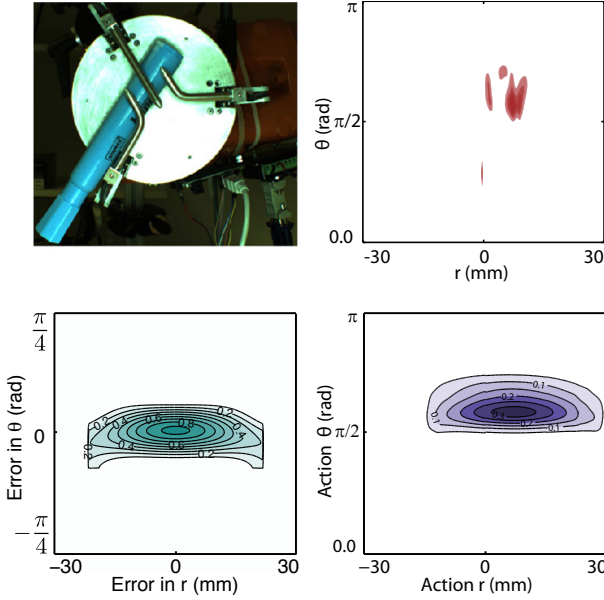
Depending on the maximum value of  $P(a|z)$ , we can decide either to execute the task with the optimal action or to abort the execution. Figure 5 shows the predicted task success distribution for an example grasp.

Note that our framework allows for the complete decoupling of sensing capabilities and task requirements. For a given hand and object, once we have determined its sensing capabilities, whenever we have a new task, we only have to compute the task requirements, and then we can follow the above analysis to compute our overall probability. Imagine another scenario where we have an industrial robot learning in a room for days at a time. If a mobile manipulator robot learns its sensing capabilities, it can directly use the task requirements learned by the industrial robot to immediately predict its likelihood of success. If we had chosen to go directly from sensors to actions, each time we wanted to learn a new task, we would have to start all over again, without being able to reuse any learned models.

## 4 Experimental Validation

To validate our framework, we use the same hand and object to complete three different tasks. This requires one training set for the *sensing capabilities* of the hand,  $P(x|z)$ , and estimating the *task requirements*,  $P(a|x)$ , for each one of the tasks. After learning these functions, for any new grasp, we can predict the action most likely to successfully execute a task and its expected probability of success.

Figure 6 compares experimental results to model predictions for each of our tasks. We group grasps by their predicted task success probability and compare it with their correspondent experimental success rate. For example, if we take all grasps that were predicted to succeed at an action around 40% of the time, the average experimental success rate for those grasps should ideally be 40%. Each task is executed 500 times for validation, and the most likely action is estimated from information provided by the three finger encoders. For all three tasks, the experimental probability follows the predicted probability, supporting the validity of the framework.

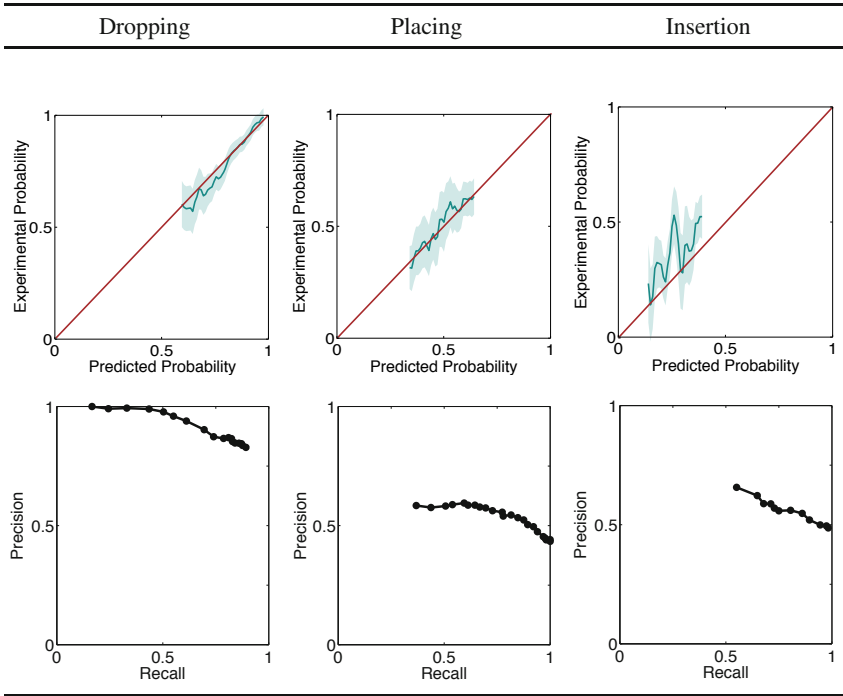


**Fig. 5** [Top-left] Example grasp. [Top-right] Posterior distribution  $P(x|z)$  of the pose of the object for the example grasp. Note that, for visualization purposes, we have rearranged the set of possible values for the pose of the marker from  $\theta \in [-\pi, \pi]$  to  $\theta \in [0, \pi]$ , and allowed the distance  $r$  to take on negative values. [Bottom-left] Estimated task requirements,  $P(a_p = 1|x)$ , for the task of placing. [Bottom-right] Estimated probability of success for all placing actions,  $P(a_p|z) \forall p$ .

Depending on the adequacy of the in-hand sensors to capture the real state of the system and the difficulty of the specific task, each task has a region in which most predicted probabilities fall. Dropping is the easiest task, followed by placing, followed by insertion. This could have been predicted by looking at the task requirements and noting that dropping has the widest distribution of success in the presence of pose error. It is clear from these experiments that the proposed framework successfully predicts the probability of success of action independently of the complexity of the task.

Predicting the probability of success of an action allows us to make an informed decision on what action to execute and improve the overall system performance. The bottom row of Figure 6 shows the precision-recall curves of the three tasks. As expected, when we choose to execute a task only when our expected probability is above a certain threshold, our average success rate of the task increases. By choosing where to set the threshold, we can move along the precision-recall curve, and achieve a desired performance. Dropping increases from 80% to near 100% success, placing increases from 40% to 60%, and insertion increases from 50% to 60%.





**Fig. 6** [Top] Comparison between the experimental probability of success and the predicted one. The shaded region is a 95% confidence interval of the estimation of the Bernoulli parameter, according to a binomial distribution. The plots show that the predictions follow the experimental observations quite well. [Bottom] Precision-recall curves of the success in task execution when conditioning the task execution to the predicted probability go above a certain thresholding. The plot shows that we can increase our success rate by rejecting low probability grasps in the three tasks.

## 5 Conclusion

In this paper, we introduce a general statistical framework to model the likelihood of success of a post-grasp manipulation task. We contribute the following three steps:

**Sensing Capabilities:** Using state and sensor pairs, we regress the posterior distribution of the state of an object. Generating a probability distribution of object location allows us to understand which grasps give good object localization and which do not, possibly informing both hand design and grasping strategies. We could attempt to find strategies to reduce uncertainty along the most troublesome directions and design hands that don't have grasps where object pose uncertainty is high.

**Task Requirements:** By perturbing our state, we learn a function for how our object state accuracy affects our task success probability. Generating a probability function of task success for a given object state error would be an excellent tool

for identifying weaknesses in task execution. We could discover, for example, that placing fails when our marker is in a certain location, so adding a move to fix this could greatly improve our overall task success rate.

**Task Success Distribution:** By combining the two functions above, we can generate a function of how likely an action is to succeed at a task for any new sensor input. Note that if we have the same hand but different tasks, we only have to learn a new task requirement function. If we are executing the same task with different hands, we only need to learn its sensing capabilities. Knowing the probability of succeeding at a task before we execute it has many benefits. We can choose to abort in order to increase our overall success rate and not take unnecessary risks when success is vital. We should even be able to find an optimal policy to minimize mean time to success, similar to [18]. Also, because we calculate the entire probability distribution in action space, this would allow us to choose the best action even in the presence of constraints or other cost functions.

We performed over 8000 grasps, and 1000 trials each for placing, dropping, and insertion.

## 6 Discussion

**Assumptions:** While the framework discussed in Section 3 is quite general, we make several simplifying assumptions to actually implement it on a real system. Although already mentioned in the paper, we summarize them here. First, we assume that the probability of success of an action is conditionally independent of our sensors given the true state of the system. This is reasonable when we have a good representation; however, if it is incomplete, our estimate of  $P(a|x)$  may be incorrect. For example, in the dropping task, the location of the center of mass of the marker relative to the center of the hand seemed to greatly affect the probability of success. Given that our state representation is only the axis that the marker lies on, the center of mass is not observable. This discrepancy was simply treated as extra noise in our system, and we were still able to accurately predict our probability of dropping success. Greater accuracy could have been obtained by adding this extra dimension in our state, although this would have also required us to collect much more data.

Our second assumption is that given an object pose, each of our sensors was unimodal and normally distributed. This is very dependent on the hand, object, and sensors in question. If the sensors being used cannot fully identify the object pose, one might consider adding more sensors. Note that even without this Gaussian assumption, one could still find the distribution  $P(x|z)$ , but comparatively more data would be required to accurately learn this distribution.

Finally, we assume that the set of actions for executing a task are state parameterized. While this is true in most cases, for the instances where it is not, in order to find  $P(a|x)$ , for every object state, we would need to sample the space of all actions *and* pose errors, again requiring much more data.

**Robustness:** In collecting all of the data for the robot to learn, it was clear that having a robust system is important but also extremely challenging. For our post-grasp manipulation tasks we had to focus on three different aspects: object acquisition, task execution, and post-task reset. With thousands of experiments needed, a human could not sit there handing the robot objects, as this would not only be time-consuming but also introduce bias into our system, since the robot might always grasp the object in the same way. We solved the object acquisition problem by having a large bin of objects and training an open-loop grasping strategy that singulates a marker out of the bin approximately one third of the time.

For task execution, it was important to make sure that the hand did not collide with the environment, regardless of the pose of the marker or action chosen by the robot. Finally, for resetting the system after task execution, different strategies were used depending on the task. For placing, the object was placed at the top of a ramp and knocked back into the bin. For dropping, after some constraining moves, the object was grasped out of the hole and dropped back into the bin. For insertion, the object was held the entire time and then dropped back in. It is important to note that a fair amount of time was spent designing robust experiments, and this should not be overlooked when attempting to use our framework in a real setting.

**Statistical Analysis:** Another area which we would like to focus more effort on is that of understanding the statistical significance of our distributions. We suggested the use of the covariance learned using a Gaussian Process as a stopping criterion, which worked well in practice, but a more thorough analysis is needed.

Carrying statistical significance through all of the distributions would enable us to compute a confidence bound on our final probability distribution in action space. This would expand the usefulness of our framework and help us understand how collecting more data affects our estimates. Another direction that we are interested in exploring is using active learning to selectively sample so as to reduce overall data requirements.

**Acknowledgements.** This work was supported by National Science Foundation [NSF-IIS-0916557], the Defense Advanced Research Projects Agency [DARPA-BAA-10-28] and Army Research Laboratory [W911NF-10-2-0016]. This work does not necessarily reflect the position or the policy of the U.S. Government, DARPA or ARL. No official endorsement should be inferred.

## References

1. Boots, B., Siddiqi, S.M., Gordon, G.J.: Closing the Learning-Planning Loop with Predictive State Representations. *The International Journal of Robotics Research* 30(7), 954–966 (2011)
2. Botev, Z.I., Grotowski, J.F., Kroese, D.P.: Kernel density estimation via diffusion. *The Annals of Statistics* 38(5), 2916–2957 (2010)
3. Brost, R.: Automatic Grasp Planning in the Presence of Uncertainty 3, 1575–1581 (1986)
4. Brost, R.C., Christiansen, A.D.: Probabilistic Analysis of Manipulation Tasks: A Conceptual Framework. *The International Journal of Robotics Research* 15(1), 1–23 (1996)

5. Cassandra, A.R., Kaelbling, L.P., Littman, M.L.: Acting Optimally in Partial Observable Stochastic Domains. In: AAAI (1994)
6. Dogar, M., Srinivasa, S.S.: A Framework for Push-Grasping in Clutter. In: Robotics: Science and Systems (RSS) (2011)
7. Fu, J., Srinivasa, S., Pollard, N., Nabbe, B.: Planar batting under shape, pose, and impact uncertainty. In: IEEE International Conference on Robotics and Automation, ICRA (2007)
8. Goldberg, K., Mason, M.T.: Bayesian grasping. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1264–1269 (1990)
9. Goldfeder, C., Allen, P.K.: Data-driven grasping. *Autonomous Robots* 31(1), 1–20 (2011)
10. Hsiao, K., Kaelbling, L., Lozano-Pérez, T.: Robust grasping under object pose uncertainty. *Autonomous Robots* 31(2), 253–268 (2011)
11. Jiang, Y., Lim, M., Zheng, C., Saxena, A.: Learning to place new objects in a scene. *The International Journal of Robotics Research* (2012)
12. Kang, D., Goldberg, K.: Sorting parts by random grasping. *IEEE Transactions on Robotics and Automation* 11(1), 146–152 (1995)
13. Lavalle, S.M., Hutchinson, S.A.: Evaluating Motion Strategies under Nondeterministic or Probabilistic Uncertainties in Sensing and Control. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3034–3039 (April 1996)
14. Mason, M.T., Rodriguez, A., Srinivasa, S.S., Vazquez, A.S.: Autonomous Manipulation with a General-Purpose Simple Hand. *The International Journal of Robotics Research* 31(5), 688–703 (2012)
15. Morales, A., Chinellato, E., Fagg, A.H., del Pobil, A.P.: Using Experience for Assessing Grasp Reliability. *International Journal of Humanoid Robotics* 1(4), 671–691 (2004)
16. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press (2006)
17. Rodriguez, A., Mason, M.T., Srinivasa, S.S.: Manipulation Capabilities with Simple Hands. In: International Symposium on Experimental Robotics, ISER (2010)
18. Rodriguez, A., Mason, M.T., Srinivasa, S.S., Bernstein, M., Zirbel, A.: Abort and Retry in Grasping. In: IEEE International Conference on Intelligent Robots and Systems, IROS (2011)
19. Stulp, F., Theodorou, E., Buchli, J., Schaal, S.: Learning to Grasp under Uncertainty. In: IEEE International Conference on Robotics and Automation, ICRA (2011)
20. Wingate, D.: Exponential family predictive representations of state. Ph.D. thesis, University of Michigan (2008)

# Part VII: ISER Session Summary on “Social Robotics”

Maja J. Matarić

University of Southern California

## Session Summary

As robots get closer to people, different forms of interaction are spawning new research topics and subfields. Human-robot interaction (HRI) can be broadly classified into two categories: physical and social/emotional. Physical interaction involves the research areas of manipulation and haptics, among others, and is used in medical and rehabilitation robotics. In contrast, social/emotional interaction involves verbal and non-verbal expression and communication, and thus the research areas of assistive robotics, social robotics, and socially assistive robotics. The field of assistive robotics spans both physical and social/emotional types of human-robot interaction. Socially assistive robotics focuses on the challenges of providing motivation, coaching, training, and rehabilitation through non-physical interaction; such systems have been validated in hands-off stroke rehabilitation, social skill training of children with autism, and eldercare, among others. In contrast, most of rehabilitation robotics focuses on physical interaction with the patient; such systems have been validated in hands-on stroke rehabilitation. Social robotics focuses on endowing robots with the ability to behave in socially-aware and engaging ways; such systems have been validated in museums, movies, classrooms, and informal settings. Service robotics can be seen as the overarching field that encompasses much of the work above.

In spite of the session name, the four papers are best described as being in the area of **human-robot interaction for service robotics**, rather than the more specific area of social robotics. Specifically, Bolini et al. is in the area of service robotics, Fasola & Matarić is in socially assistive robotics, and Vasquez et al. and Weisz et al. are in the area of assistive robotics.

“Interpreting and Executing Recipes with a Cooking Robot” by Bollini et al. presents a system that enables a robot to look up a recipe on line, prepare, and bake cookies in a realistic kitchen environment. As a first step, the robot parses a recipe written in natural language into a state/action sequence, which is then interpreted into a robot action plan of baking primitives (e.g., pour, mix, scrape). The primitives are then executed sequentially, unless replanning is necessary due to failure. Each primitive involves the use of recognition system, grasping, and local navigation. The system was successfully validated with a PR2 robot; the cookies were sampled by the authors.

"Socially Assistive Robot Exercise Coach: Motivating Older Adults to Engage in Physical Exercise" by Fasola and Matarić describes the design, implementation and user study evaluation of a socially assistive robot coach designed to engage

elderly users in seated exercises (i.e., chair aerobics). The robot coach provides three different games to the user and interact verbally and through gestures, utilizing social science theories to ensure it is motivating, fluid and highly interactive, personable, intelligent, and task-driven. The system was successfully validated with 33 elderly users in a multi-session user study that compared a robot coach vs. a computer version of the same coach, demonstrating a strong preference for the robot.

“Human Aware Navigation for Assistive Robotics” by Vasquez et al. describes a robotic wheelchair navigation system that takes into account safety, usability, comfort and respect of social conventions. These abilities are achieved through an approach that integrates motion planning with long-term motion prediction that allows for safe and smooth movement in dynamic environments. For socially-aware spatial behavior, the approach involves the use of circular interaction zones around people to ensure appropriate social distances are observed in interactions. The system was successfully validated both in simulation and in physical robot trials with an autonomous robotic wheelchair.

“Grasping With Your Face” by Weisz et al. describes a brain computer interface (BCI) system that enables users with physical disabilities to command a robot to grasp objects in the environment. A vision system identifies the grasping targets using a database of models, and then a real-time grasp planner generates a grasp for reaching the target object. The user wears a light-weight headset which collects EMG signals that are used both to indicate the target object and to correct the generated trajectory as needed. The end-to-end system was successfully validated in several trials involving common household objects.

# Grasping with Your Face

Jonathan Weisz, Benjamin Shababo, Lixing Dong, and Peter K. Allen

**Abstract.** BCI (Brain Computer Interface) technology shows great promise in the field of assistive robotics. In particular, severely impaired individuals lacking the use of their hands and arms would benefit greatly from a robotic grasping system that can be controlled by a simple and intuitive BCI. In this paper we describe an end-to-end robotic grasping system that is controlled by only four classified facial EMG signals resulting in robust and stable grasps. A front end vision system is used to identify and register objects to be grasped against a database of models. Once the model is aligned, it can be used in a real-time grasp planning simulator that is controlled through a non-invasive and inexpensive BCI interface in both discrete and continuous modes. The user can control the approach direction through the BCI interface, and can also assist the planner in choosing the best grasp. Once the grasp is planned, a robotic hand/arm system can execute the grasp. We show results in using this system to pick up a variety of objects in real-time, from a number of different approach directions, using facial BCI signals exclusively. We believe this system is a working prototype for a fully automated assistive grasping system.

## 1 Motivation

With recent advances in robotics and computer vision, it is possible to imagine a robotic system to assist people with severely limiting disabilities in activities of daily living (ADL), improving their quality of life. ADL frequently require the user to grasp an object stably in a context aware way. Complex hands and manipulators increase the flexibility and grasping capabilities of a robotic assistant, but at the cost of requiring more complex control of many DOFs. Our goal is to create a robust system that can control a dexterous grasping system in real-time using only a small number of signals from an inexpensive and non-invasive BCI device with minimal training.

---

Jonathan Weisz · Benjamin Shababo · Lixing Dong · Peter K. Allen

Department of Computer Science, Columbia University,  
500 W. 120th Street, M.C. 0401 New York, New York 10027

e-mail: {jweisz, allen}@cs.columbia.edu,  
{bms2156, ld2505}@columbia.edu

Online, interactive control of robotic arms and hands for grasping in natural environments is a difficult problem. Typically, most systems use simple, parallel jaw grippers which simplifies the grasping process. Using more complex and higher DOF robotic hands increases the versatility of the system but at the cost of higher complexity for control. This generally requires more input from the user. More input, however, requires more channels, more processing, higher latencies, and generally higher cost. Our solution to this trade-off is to use a higher DOF hand with an accompanying high level interface that offloads the complexity of input requirements for the user to a simple and intelligent user interface. To avoid the issue of increasing the complexity of the input to the hand but maintain the flexibility of the more complex hand, the user needs a high level and intuitive interface.

## 2 Prior Work

A significant proportion of BCI-robotics research has focused on manipulating robotic arms and hands and different strategies for implementing solutions via electrophysiological signals have been investigated. Vogel et al. [26] showed that using the BrainGate cortically implanted electrode, a subject was able to exercise Cartesian velocity control over an end effector and control opening and closing of the hand. However, this approach requires an invasive device capable of recording a large number of high quality signals.

It has also been shown that non-invasive devices can exercise effective control over robotic arms. For example, distal limb surface EMG signals have been used to control robotic arms in several applications [1, 4]. However, thus far accurate real time control has only been demonstrated for simple trajectory tracking tasks while using a large number of signals.

In order to reduce the number and quality of human signals needed, some intermediate level abstractions have been used. In [23], Shenoy et al. demonstrated basic cartesian control of a robot arm and gripper from 9 forearm electrodes to perform basic pick and place tasks. Another approach to abstraction is the use of forearm EMG signals to quickly switch between preset discrete states of various robotic hands [29, 28, 10, 6, 13]. In many situations in which assistive BCI-robotics would be applicable, limb EMG signals may not always be available or convenient. Therefore, various authors have proposed control schemes using face and head EMG signals to control robotic arms and grippers [20, 8, 18].

EEG has also been developed for BCIs to control robotic arms and hands in simple tasks. In [24, 9] BCI signals are used to control functional electrical stimulation to close and open a subject's wrist. In [17], surface electrode signals related to eye gaze direction are used to control 2D arm position and EEG signals are used to detect eye blinks to control gripper closing. In [11] hand opening/closing and elbow flexion/extension are controlled by EEG signals.

The majority of this previous work concentrates on trajectory control. However, it has been shown that users find BCI control easier using even higher level, goal oriented paradigms [19]. We have begun to see work that attempts to exploit higher



level abstractions to allow users to perform more complex tasks with robotic arms. In [2], EEG signals were used to select targets for pick and place operations for a small humanoid robot. Waytowich et al. [27] used EEG signals to control pick and place operations of a 4-DOF Stäubli robot. Bryan et al. [12] presented preliminary work extending this approach to a grasping pipeline on the PR2 robot. In that work, a 3D perception pipeline is used to find and identify target objects for grasping and EEG signals are used to choose between them. In [15], grasping is decomposed to a 4 stage pipeline where EEG signals are used to control transitions between stages. And in [22], the authors demonstrate an interface to navigate in two dimensions and select goals in a complex virtual environment and propose a hierarchical control scheme for learning high level tasks dynamically.

While previous work has shown that complex interactions can be mediated by BCI signals, *thus far a fully developed, end-to-end, real-time, BCI-based grasping system for complex hands has not been demonstrated*. A full grasping pipeline, which our system addresses, must integrate target selection and localization, multi-DOF hand configuration planning, and approach trajectory planning with user intent decoded from a noisy low dimensional BCI signal.

### 3 Technical Approach

Figure 1 is an overview of our system. A 3D range camera is used to image an object to be grasped. This range image is used to both identify and align the object from a database of existing models [7]. Once the model is chosen and aligned, we show that four simple signals are sufficient to supervise an online grasp planning system resulting in robust grasps using real sensor data for object localization. By carefully reducing the configuration space of our planning and control architecture, our system enables subjects to select grasps appropriate to the desired task. The user interacts in real-time with the robot through a kinematic simulator which allows the user to visualize and supervise all of the elements necessary to plan a task specific grasp. The goal of this approach is to make simple, inexpensive BCI devices powerful enough to allow users to grasp many objects that are important for everyday living in a context aware way.

The grasping pipeline is divided up into four stages: object identification and alignment, grasp planning, grasp review, and grasp execution, which are described below. This pipeline is controlled using only four facial gestures. The use of these gestures in each stage of the pipeline is explained in Table 1. In general, gesture 1 serves as a “click” and moves the user through the pipeline. The exception to this is at points of decision for the user. In these cases, gesture 2 serves as the “YES” option and gesture 1 becomes “NO” and returns the user to an earlier point in the pipeline. Because false positive readings of gesture 1 and 2 have strong consequences, we found that both are best associated with a concise and strong gesture such as closing one eye or clenching the jaw. Gestures 3 and 4 control the approach direction of the hand relative to the object during the grasp planning stage of the pipeline.

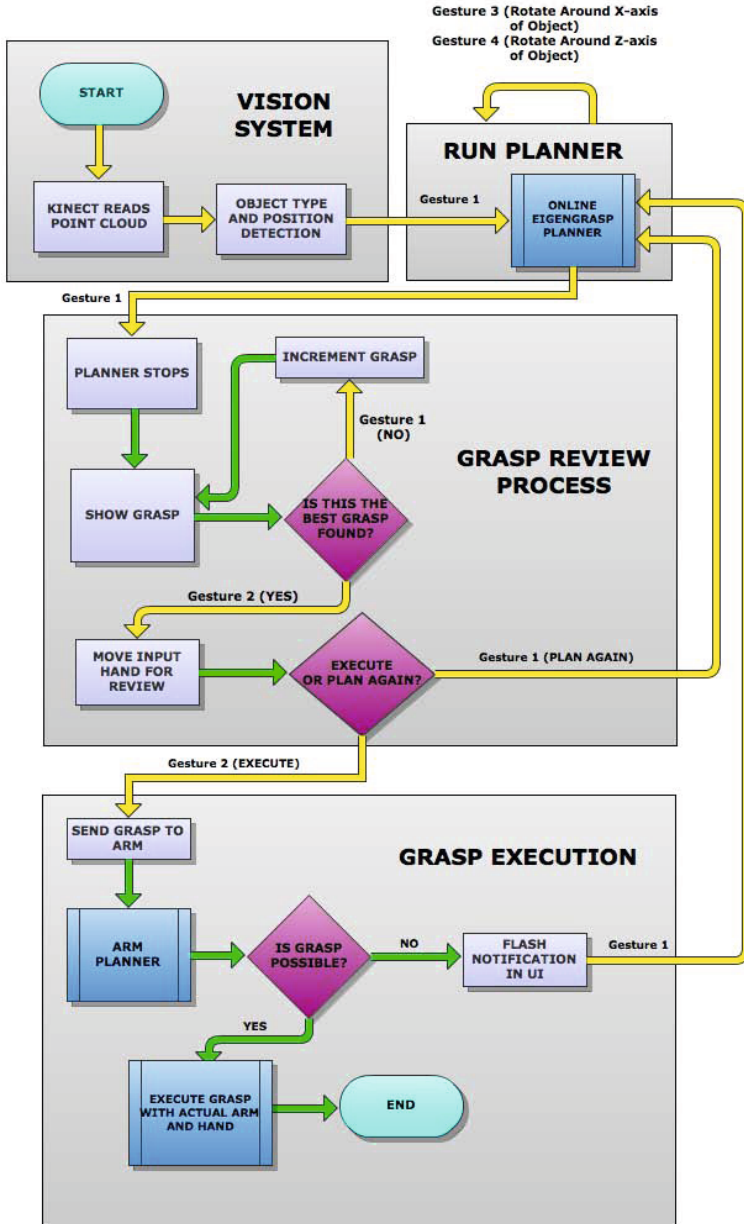
**Table 1** A description of the user interface as the user progresses through phases of the pipeline

Gesture	Run Planner	Review Grasps	Execution
1	start/stop planner	cycle through grasps	restart
2	n/a	select grasp	confirm grasp
3	rotate around $x$ -axis	n/a	n/a
4	rotate around $z$ -axis	n/a	n/a

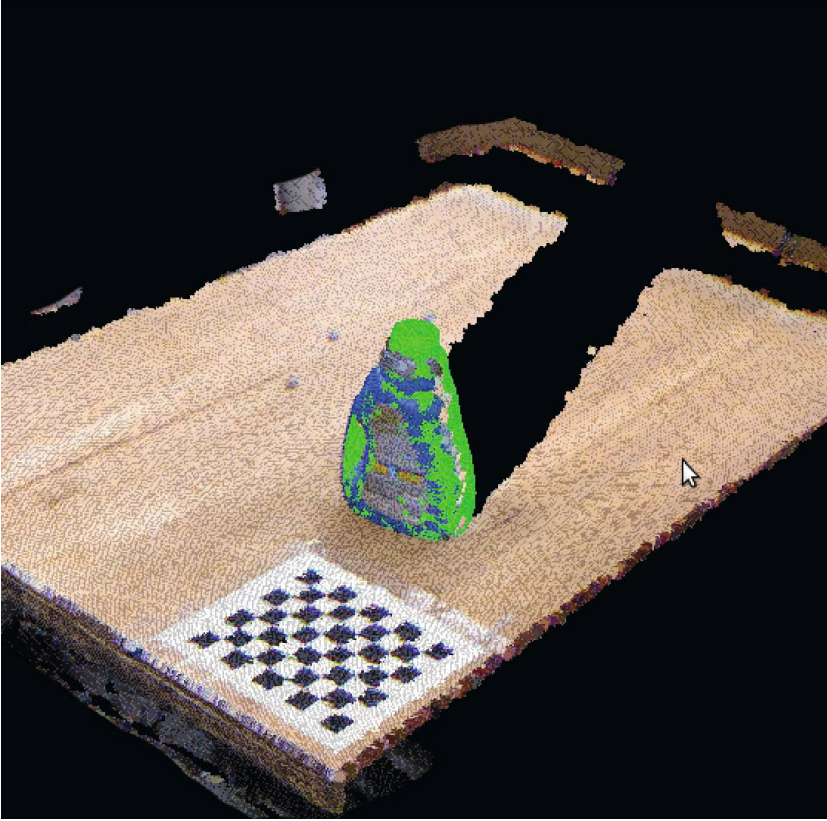
These gestures can be maintained to generate *continuous* motion of the hand over two degrees of freedom and therefore are best associated with gestures that can be contracted for several seconds without too much twitching or fatigue.

**Object Identification and Localization:** Our grasp planner requires a complete description of the geometry and location of the target object. The raw point cloud data is gathered by a Microsoft Kinect. In this work, we assume the target is a member of a known set of objects. We use the method described in [16] to identify and localize the target object in the scene. Briefly, this method generates features from pairs of oriented points on the surface of the object. Prospective models are processed offline and put in to a hash table. Features are sampled from the sensor data and tested for collision in the hash table. If a sufficient number of collisions occurs with points on the same model, a variant of RANSAC is used to test the hypothesis that a set of points in the sensor data corresponds to a particular model at a particular location. Fig. 2 shows a correctly chosen model aligned with the range scan. This method is robust and fast enough to demonstrate the efficacy of our BCI-grasping pipeline.

**Grasp Planning Phase:** In this work we use the Eigengrasp Grasp Planner developed by Ciocarlie and Allen, the details of which can be found in [5]. In this system, grasps are planned through stochastic optimization using simulated annealing. Recent advances in neuroscience research have shown that control of the human hand during grasping is dominated by movement in a configuration space of highly reduced dimensionality[25, 21]. The Eigengrasp Grasp Planner uses these insights to reduce the the dimensionality of the hand's joint postures to a subspace with a lower dimensionality. This allows the planner to function in real time with a human user in the loop by guiding the planner by partial demonstration. The planner allows the operator to specify how much the stochastic optimization process can vary each parameter that describes the demonstration pose against the gradient of the grasp energy function in searching for a valid, complete grasp. The planner optimizes a grasp energy function as the sum of two parts, a measure of how nearly a hand conforms to the object, and a continuous approximation of the canonical Ferrari-Canny grasp quality measure. Hand configurations with a high enough quality function are marked as potentially good grasps. These configurations will put the hand in a position which is near the object but where the finger configuration is restricted to the postural subspace described above. The quality of these configurations as starting



**Fig. 1** An overview of the grasping pipeline. In the first phase, the system uses the kinect to identify and localize the object in the scene. In the second phase, the user guides the planner to an appropriate approach direction. In the third phase, the user stops the planner and reviews the available grasps. In the fourth phase, the user sends the grasp to the robot for execution.

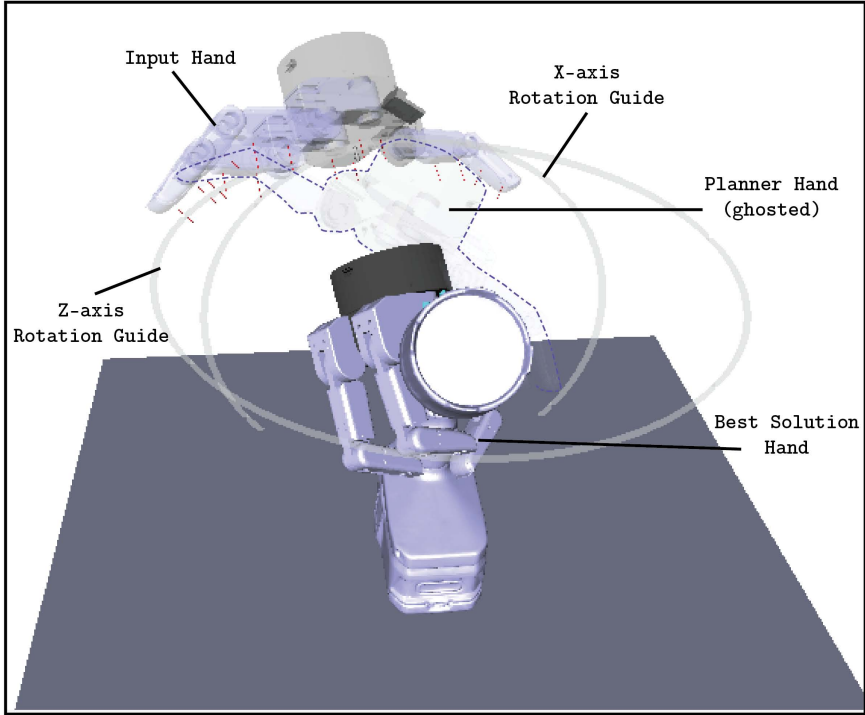


**Fig. 2** The point cloud with RGB texture from the vision system. The blue bottle is the object point cloud, while the green represents the detected model overlain on the scene.

grasp poses is then reassessed after a kinematic simulation of grasping. The hand approaches the object and closes all of its joints, leaving the postural subspace as necessary to conform to the object.

The user is presented with a world in our grasping simulator, *GraspIt!* [14], that contains the hand, the object detected by the vision system, and the surface on which the object sits. Elements of the user interface for the planning phase are shown in Fig. 3. When planning begins, two clones of the hand are placed into the world, and the transparency of each of the three hands identifies it to the user. The most transparent hand represents the planning process and shows samples of the grasp space region the planner is exploring. This online feedback helps the user choose an approach direction that best communicates their intent to the planner.

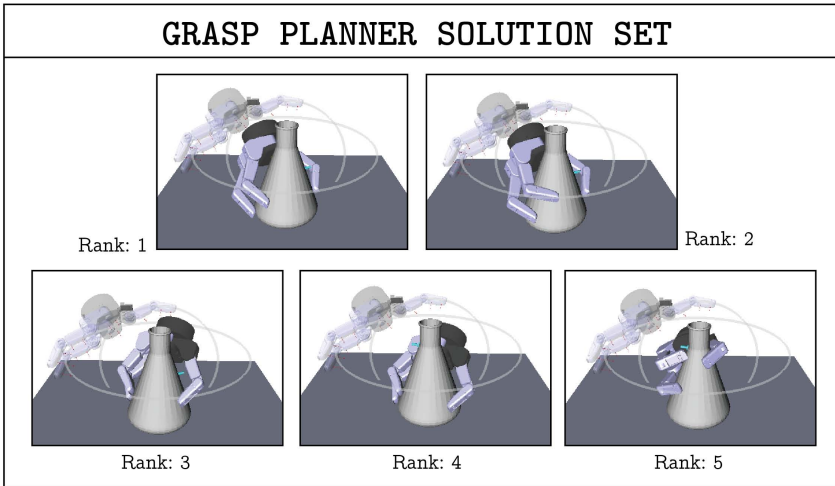
The input hand has intermediate transparency, and the user can control the rotation of the input hand around the  $x$  and  $z$  axis of the object. This effectively moves the hand around the surface of a sphere while maintaining that the approach



**Fig. 3** An illustration of the grasp planning user interface in GraspIt!. Here we demonstrate the three hands of the system. The ‘Planner Hand,’ which is most transparent hand, demonstrates the current state of the planner. The ‘Input Hand,’ which is of intermediate transparency, is the hand through which the user directs the planning system. Here you can see the rotational guides which allow the user to visualize their available control directions. The ‘Solution Hand’, which is fully opaque, demonstrates the best grasp currently available. This is the grasp which is closest to the approach direction that the Input Hand is demonstrating and which also has the minimal grasp energy.

direction of the hand faces the center of the object. The third hand during the planning phase is the solution hand which is fully opaque. As the planner runs, it stores the ten best grasps in a list. The solution hand demonstrates the current best grasp. The list of best grasps is sorted such that preference is given to solutions that reflect the user’s desired approach. When the user is satisfied, they can stop the planner and progress into the review process using gesture 1.

**Review Phase:** Once the planner is stopped the user has an opportunity to review the list of best grasps and choose one for execution. As in the previous stage, the solution hand is used to display the grasps to the user. At any point, the user can select a grasp which removes the solution hand from the world and closes the input hand into the chosen grasp. Now the user can evaluate the grasp more closely and



**Fig. 4** An illustration of the review phase. Using one of the facial gestures, the user can cycle through the available grasps and visualize them in the simulator. The grasps are sorted by their closeness to the last demonstrated approach vector. If none of the grasps are suitable, the user can go back to the planning phase with a second facial gesture. Otherwise, the user selects a grasp to be executed using a third gesture.

examine the quality metrics for the grasp. If the user is satisfied, they can confirm selection of the grasp and send it to the next stage of the pipeline. If the user does not want to execute any of the found grasps, they can select the grasp that is closest to what they have in mind and restart the planning process. Importantly, the next iteration of planning will not only take the input hand's position as a constraint but also the eigengrasp values of this selected grasp.

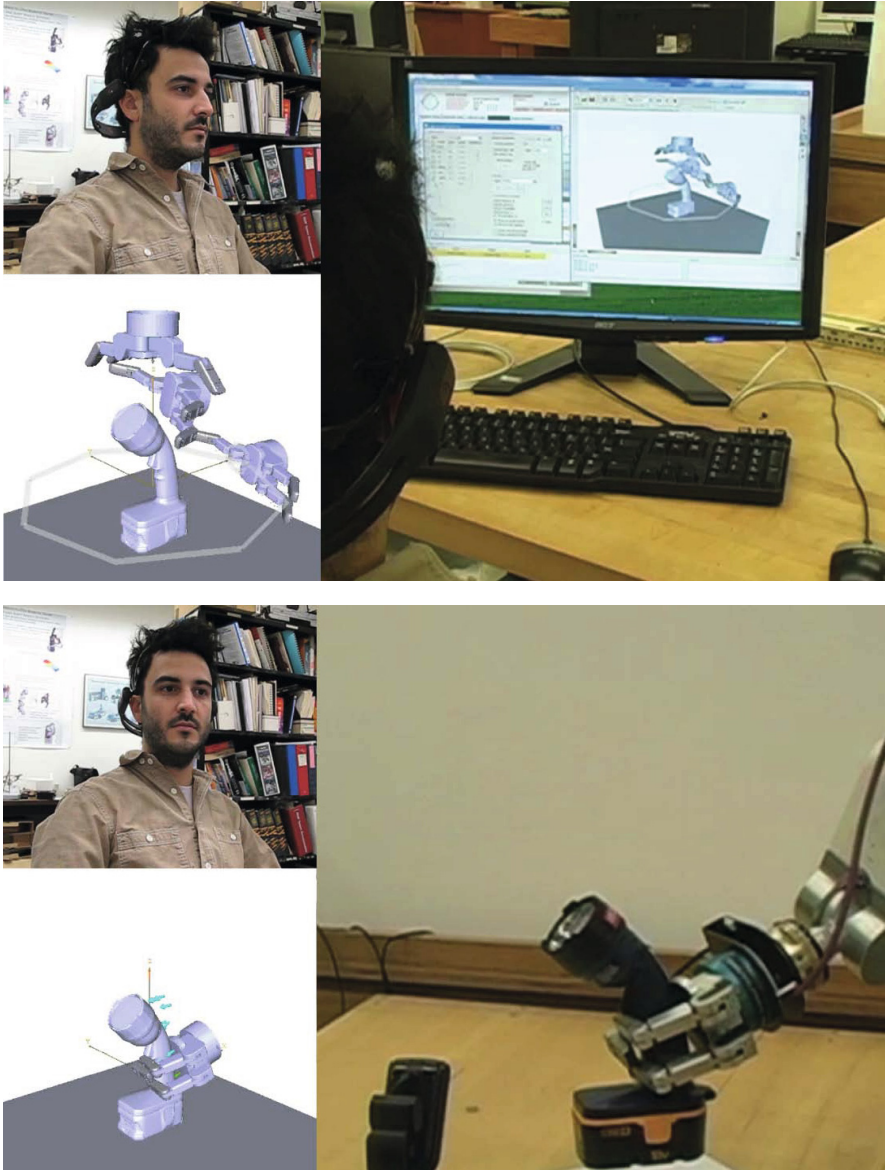
**Execution Phase.** Once the robotic control software receives the selected grasp which is planned relative to the object's coordinate system, the arm planner decides if that grasp is achievable given the kinematic constraints of the arm and the vision system's estimate of the object's position in the world. If the planner cannot find a solution to execute the grasp, it will send a message back to GraspIt! which then notifies the user. If this occurs, the user can restart the planning phase to find a new grasp. If the planner can find a solution, then the grasp is executed with the actual arm and hand.

## 4 Experimental Results

### 4.1 Brain-Computer Interface

*Hardware:* The BCI hardware used for this experiment was the Emotiv EPOC which is a low-cost 16 electrode headset. Of the 16 electrodes, 14 are positioned at the





**Fig. 5** Two sets of video stills from the actual execution of the system. **Top:** Clockwise from the top-left are an image of the user wearing the Emotiv headset and operating the system, the computer monitor with user interface, and a screen capture of the simulation. **Bottom:** User watching the robotic arm implementing the selecting grasp as shown in the simulator window on the lower left.

International 10-20 locations corresponding to AF3, AF4, F3, F4, F7, F8, FC5, FC6, P7, P8, T7, T8, O1, O2. The remaining 2 channels are references and are positioned at P3 and P4. Each electrode has a sampling rate of roughly 128 hz and a resolution of 16 bits (14 bits effective). The headset communicates with the computer via a bluetooth connection.

*Software:* We used the Cognitiv Suite and Expressiv Suite software that come bundled with the Emotiv Epoc to process the electrophysiological signals. The Cognitiv Suite is designed to be trained on different cognitive states, usually associated with mental imagery or imagined movement. However, because the EMG signals from facial gestures are much stronger and more easily produced in a consistent manner, we chose to train the Cognitiv Suite on these types of signals instead of cognitive states. First, the system is trained on a neutral signal to define a baseline signal, then a set of facial gesture states are learned. In the examples presented here, gesture 1 is mapped to clenching the jaw muscles and is detected through the Expressiv Suite. The remaining gestures are detected through the Cognitiv Suite. Gesture 2 is mapped to closing the right eye, gesture 3 is mapped eyebrow movement, and gesture 4 is mapped to tensing the muscles around the ear. This combination of classifiers and gestures was found by trial and error to be reliable for the particular user discussed in this paper, but the system as a whole is in no way dependent on which particular gestures are used.

## 4.2 Robot Setup

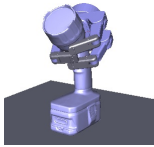

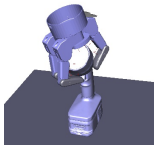

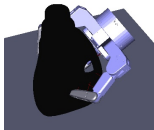

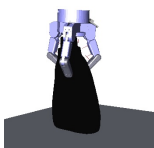

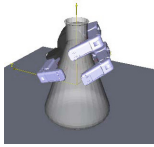
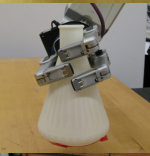
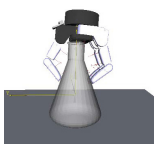

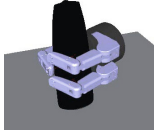

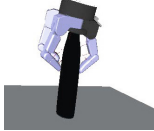


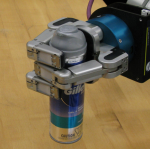


*Hardware:* Our grasping platform is comprised of a 280 model BarrettHand mounted on a Stäubli TX60L 6-DOF robotic arm. A Microsoft Kinect sensor is used to generate point clouds of the object.

*Software:* The Barrett hand is commanded in position control mode using the OpenWAM driver. Planning for the motion of the arm is done in OpenRave using a bidirectional random tree planner by Berenson et al. [3], and small linear motions near the object are planned using the built-in inverse kinematics planner on the Stäubli TX60L arm.

## 4.3 Training

One subject completed 10 successful training periods on the neutral signal and each of the three facial gestures that would be processed by the Cognitiv Suite. Each training period lasts 8 seconds, and after each period the subject is asked whether or not they were able to maintain the appropriate gesture for the entire time. Only sessions in which the subject answered yes are considered successful and included in the training data. During training, the subject is given visual feedback via the motion of



Results				
Object	Approach 1		Approach 2	
	Simulation Grasp	Physical Grasp	Simulation Grasp	Physical Grasp
Flashlight				
Detergent Bottle				
Flask				
Shampoo Bottle				
Shaving Cream				

**Fig. 6** The results of ten different attempts to grasp five objects using two different user-selected approach directions per object. In each case, the user was able to generate an appropriate grasp using the simulator which was then transferred to the robot.

a cube and a power meter, both of which represent the strength of current classification. The Expressiv Suite does not need any training, however, the sensitivity of the classifier for jaw-clenching was adjusted for best performance for the subject.

### 4.4 Grasping Experiments

The subject used the system to grasp and pick up five common objects: a flashlight, a flask, a bottle of laundry detergent, a bottle of shampoo, and a canister of shaving cream. For each setup, the Stäubli arm starts such that it is completely vertical with the BarrettHand at the top. A simulation world consisting of the hand, the object, and a surface is presented to the subject who can move the input hand and start the

planner at any time. For each object, the subject executed grasps from two different approach directions.

In Fig. 6, we show the results of ten different attempts to grasp five objects using two different approach directions per object. In each case, we demonstrate the grasp planned in simulation and the final grasp achieved by the physical robot. We found that using this grasp planning environment, we were easily and reliably able to grasp objects. In each case, the physical grasp was successful on the first attempt. Planning the grasp takes on the order of 10-30 seconds. The review process takes an additional 10-15 seconds. We refer the reader to the video at <http://robotics.cs.columbia.edu/jweisz/bciGraspingISER2012> for an example of the entire procedure.

## 4.5 Discussion

In this work we have described an end-to-end system for grasping objects using a low-cost, non-invasive BCI device. Although further refinements and user studies still need to be done, we have shown three important ideas in this paper. First, we have shown that a user interface using only two user controlled dimensions is expressive enough to demonstrate the user's intent by controlling the approach direction of the hand and that the Eigengrasp planner is able to produce reliable and appropriate grasps using this information alone, vastly simplifying the grasp planning process. Second, we have shown that a low cost, non-invasive, noisy, low-bandwidth BCI interface is sufficient to guide the grasp planner. By reducing the complexity of the grasp planning process, we can accommodate such a BCI device. Third, a central tenet of this work is that by keeping a human in the loop, we are able to support context-aware grasping, spanning different grasps for different tasks. Although the grasp quality measure used by the planner is sufficient to produce a reasonable list of plausible grasps, it is not sufficient to choose the most reliable and appropriate grasp among them, which can lead to failures. With a human adding their own intuition to select a grasp from the candidate list, we have never seen a failure in grasping the object.

We believe our user interface works well, but it is yet to be tested with multiple subjects, which is the subject of ongoing research in our lab. As part of this work, we are also analyzing how much training of the BCI control is needed to create a competent user. Making the system easy to learn and use is an important goal of this work. Finally, we are also experimenting with using EEG signals from the Emotiv EPOC instead of using facial gestures which generate EMG signals.

## References

1. Artemiadis, P.K., Kyriakopoulos, K.J.: A switching regime model for the EMG-based control of a robot arm. *IEEE Transactions on Systems, Man, and Cybernetics* 41(1), 53–63 (2011)

2. Bell, C.J., Shenoy, P., Chalodhorn, R., Rao, R.P.N.: Control of a humanoid robot by a noninvasive brain-computer interface in humans. *Journal of Neural Engineering* 5(2), 214–220 (2008)
3. Berenson, D., Srinivasa, S.S., Kuffner, J.: Task Space Regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research* (2011)
4. Castellini, C., van der Smagt, P.: Surface EMG in advanced hand prosthetics. *Biological Cybernetics* 100(1), 35–47 (2009)
5. Ciocarlie, M.T., Allen, P.K.: Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research* 28(7), 851–867 (2009)
6. Cipriani, C., Zaccone, F., Micera, S., Carrozza, M.: On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User Prosthesis Interaction. *IEEE Transactions on Robotics* 24(1), 170–184 (2008)
7. Goldfeder, C., Ciocarlie, M., Peretzman, J., Dang, H., Allen, P.K.: Data-driven grasping with partial sensor data. In: *IROS 2009: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1278–1283. IEEE Press, Piscataway (2009)
8. Gomez-Gil, J., San-Jose-Gonzalez, I., Nicolas-Alonso, L.F., Alonso-Garcia, S.: Steering a Tractor by Means of an EMG-Based Human-Machine Interface. *Sensors* 11(7), 7110–7126 (2011)
9. Hazrati, M.K., Erfanian, A.: An on-line BCI for control of hand grasp sequence and holding using adaptive probabilistic neural network. In: *Conference Proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1009–1012 (2008)
10. Ho, N.S.K., Tong, K.Y., Hu, X.L., Fung, K.L., Wei, X.J., Rong, W., Susanto, E.A.: An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation. In: *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–5. IEEE (2011)
11. Horki, P., Solis-Escalante, T., Neuper, C., Müller-Putz, G.: Combined motor imagery and SSVEP based BCI control of a 2 DoF artificial upper limb. *Medical & Biological Engineering & Computing* 49(5), 567–577 (2011)
12. Bryan, M., Thomas, V., Nicoll, G., Chang, L., Smith, J., Rao, R.: What You Think is What You Get: Brain-Controlled Interfacing for the PR2. Tech. rep., Iros 2011: The PR2 Workshop, San Francisco (2011)
13. Matrone, G., Cipriani, C., Carrozza, M.C., Magenes, G.: Two-channel real-time EMG control of a dexterous hand prosthesis. In: *2011 5th International IEEE/EMBS Conference on Neural Engineering*, pp. 554–557 (2011)
14. Miller, A.T., Allen, P.K.: Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine* 11, 110–122 (2004)
15. Müller-Putz, G.R., Scherer, R., Pfurtscheller, G., Rupp, R.: EEG-based neuroprosthesis control: a step towards clinical practice. *Neuroscience Letters* 382(1-2), 169–174 (2005)
16. Papazov, C., Burschka, D.: An efficient ransac for 3d object recognition in noisy and occluded scenes. In: *Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492*, pp. 135–148. Springer, Heidelberg (2011)
17. Postelnicu, C.C., Talaba, D., Toma, M.I.: Controlling a Robotic Arm by Brainwaves and Eye. *IFIP International Federation For Information Processing*, pp. 157–164 (2011)
18. Ranky, G.N., Adamovich, S.: Analysis of a commercial EEG device for the control of a robot arm. In: *Proc. IEEE Northeast Bioengineering Conference (NEBEC), New York, NY*, pp. 1–2 (2010)

19. Royer, A.S., Rose, M.L., He, B.: Goal selection versus process control while learning to use a brain-computer interface. *Journal of Neural Engineering* 8(3), 036,012 (2011)
20. Sagawa, K., Kimura, O.: Control of robot manipulator using EMG generated from face. In: *ICMIT 2005: Control Systems and Robotics*, vol. 6042, pp. 604,233-604,233–6 (2005)
21. Santello, M., Flanders, M., Soechting, J.F.: Patterns of hand motion during grasping and the influence of sensory guidance. *The Journal of Neuroscience* 22(4), 1426–1435 (2002)
22. Scherer, R., Friedrich, E.C.V., Allison, B., Pröll, M., Chung, M., Cheung, W., Rao, R.P.N., Neuper, C.: Non-invasive brain-computer interfaces: Enhanced gaming and robotic control. In: Cabestany, J., Rojas, I., Joya, G. (eds.) *IWANN 2011, Part I. LNCS*, vol. 6691, pp. 362–369. Springer, Heidelberg (2011)
23. Shenoy, P., Miller, K.J., Crawford, B., Rao, R.N.: Online electromyographic control of a robotic prosthesis. *IEEE Transactions on Bio-medical Engineering* 55(3), 1128–1135 (2008)
24. Tavella, M., Leeb, R., Rupp, R., Millan, J.D.R.: Towards natural non-invasive hand neuroprostheses for daily living. In: *Conference Proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 126–129 (2010)
25. Tsoli, A., Jenkins, O.C.: 2d subspaces for user-driven robot grasping. In: *RSS Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, Atlanta, GA (2007)
26. Vogel, J., Haddadin, S., Simeral, J.D., Stavisky, S.D., Bacher, D., Hochberg, L.R., Donoghue, J.P., van der Smagt, P.: Continuous Control of the DLR Light-weight Robot III by a human with tetraplegia using the BrainGate2 Neural Interface System. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) *Experimental Robotics. STAR*, vol. 79, pp. 125–136. Springer, Heidelberg (2012)
27. Waytowich, N., Henderson, A., Krusienski, D., Cox, D.: Robot application of a brain computer interface to Staubli TX40 robots - early stages. In: *World Automation Congress (WAC)*, pp. 1–6 (2010)
28. Wolczowski, A., Kurzyński, M.: Human-machine interface in bioprosthesis control using EMG signal classification. *Expert Systems* 27(1), 53–70 (2010)
29. Yang, D., Zhao, J., Gu, Y., Jiang, L., Liu, H.: EMG pattern recognition and grasping force estimation: Improvement to the myocontrol of multi-DOF prosthetic hands. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 516–521. IEEE (2009)

# Human Aware Navigation for Assistive Robotics

Dizan Vasquez, Procópio Stein, Jorge Rios-Martinez, Arturo Escobedo, Anne Spalanzani, and Christian Laugier\*

**Abstract.** Ensuring proper living conditions for an ever growing number of elderly people is a significant challenge for many countries. The difficulty and cost of hiring and training specialized personnel has fostered research in assistive robotics as a viable alternative. In this context, an ideally suited and very relevant application is to transport people with reduced mobility. This may involve either autonomous or semi-autonomous transportation devices such as cars and wheelchairs.

For a working solution, a number of problems including safety, usability and economic feasibility have to be solved. This paper presents PAL's robotic wheelchair, an experimental platform to study and provide solutions to many of the aforementioned problems.

## 1 Motivation, Problem Statement and Related Work

Ensuring proper living conditions for an ever growing number of elderly people is a significant challenge for many countries. The difficulty and cost of hiring and training specialized personnel has fostered research in assistive robotics as a

---

Dizan Vasquez · Procópio Stein · Jorge Rios-Martinez · Arturo Escobedo ·  
Anne Spalanzani · Christian Laugier  
INRIA Rhône-Alpes, Montbonnot, 38334 Saint-Ismier cedex,  
France and Laboratoire d'Informatique de Grenoble, 38041 Grenoble, France

Dizan Vasquez · Anne Spalanzani  
Université Pierre Mendès France, 38040 Grenoble, France

Procópio Stein  
Departamento de Engenharia Mecânica, Universidade de Aveiro, 3800 Aveiro, Portugal

\* Procópio Stein and Arturo Escobedo are funded by the INRIA large initiative scale called PAL (Personal Assisted Living) e-mail: pal.inria.fr. Procópio Stein is also funded by FCT SFRH/BD/46604/2008 and personally thanks his Thesis supervisor Prof. Vítor Santos. Jorge Rios-Martinez is also funded by CONACYT 250140/308006. Dizan Vasquez was funded by UPMF as invited Professor.

viable alternative. In this context, an ideally suited and very relevant application is to transport people with reduced mobility.

In particular, this paper studies the case of a robotic wheelchair. For such a system, it is crucial to take into account the actual needs and characteristics of both its users and the people around them. The platform discussed in this paper has been designed around the following requirements:

- *Safety*: The system should avoid collisions with both static and dynamic entities.
- *Usability*: People with motor disabilities often have problems using joysticks and other standard control devices. The system should account for this, for example by favoring the most “reasonable” actions when presented with an ambiguous command.
- *Comfort*: Strong accelerations can be intolerable and even dangerous for a wheelchair user, this imposes an additional constraint on how the robot may move.
- *Respect of social conventions*: When moving, a robot may considerably disturb people around it, especially when its behavior is perceived as unsocial. Even worse, the wheelchair’s passenger may be held responsible for that behavior. It is thus important to produce socially acceptable motion.

From the technical standpoint these requirements imply that, in addition to the conventional robot tasks (e.g. localization, path execution) the following points should be specifically addressed:

- *Integrated motion-planning and long-term motion prediction*: Most human-populated environments are highly dynamic, requiring considerable look-ahead about how other objects will move in order to ensure collision-free robot motion under “comfortable” accelerations. This motivates the proposed integration of a long-term motion prediction algorithm based on the idea of typical behavior with a risk-based motion planning algorithm.
- *Interaction detection for socially acceptable robot-motion*: Our approach is based on the simple idea that, when people interact, they often adopt spatial formations implicitly forming “interaction zones”. Thus, socially acceptable motion can be enforced by first detecting interaction zones and then computing the risk to invade them.

One of our main ambitions with this platform is to provide an open benchmark that could be used to compare and evaluate different approaches. This is an important task given the large diversity of existing wheelchairs [12], including autonomous [6], semi-autonomous [9] and social aware systems [5], [11].

## 2 Technical Approach

Figure 1 presents an overview of our system’s architecture. It is divided into several subsystems:

1. *Tracking subsystem*: mobile objects are tracked both off-board and on-board the robotic wheelchair.
2. *Prediction subsystem*: the prediction subsystem processes data from the trackers and transforms it into probabilistic predictions about the configuration of the free space in the environment. It also features a “social filter”, which detects present and future interactions and creates virtual obstacles corresponding to interaction zones.
3. *Navigation subsystem*: the navigation subsystem includes a laser-based localization module and a motion-planner which integrate predictions to compute safe trajectories that are fed to the execution module.

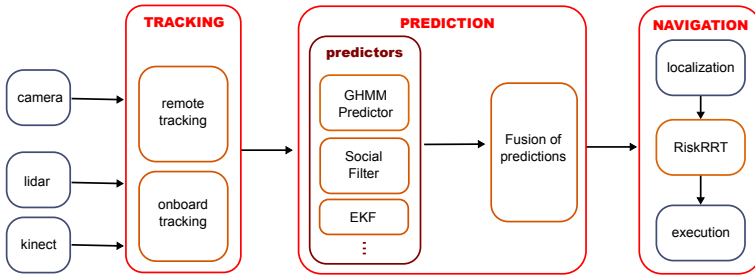


Fig. 1 Achitecture overview

### 2.1 Tracking Systems

The off-board tracker provides global information about moving obstacles and provides learning input for our motion prediction module.

At this point, we are still developing and testing our tracking systems. Meanwhile, we have performed several tests using augmented reality markers that people wear as hats. This has allowed us to validate the overall architecture, even if it is not a viable solution in the long run.

For the definitive version of the platform, we are working on a basic detect-then-track system, where moving objects are first detected using a Self-organizing network [13], after this, objects are encoded as a color histogram, and then detected in later frames using the mean-shift algorithm [1]. Finally, the different detections are used as input for a tracker based on the Joint Probabilistic data Association Filter.

On the other hand, the on-board system will provide detailed information about the objects that appear in the robot’s perceptual field. Its main use is to identify interactions between people (e.g. two persons shaking hands). The on-board tracking performs leg detection using a LIDAR sensor and people detection using the kinect sensor, according to the technique described in [7].

## 2.2 Motion Prediction

The motion prediction subsystem takes tracking data (i.e. position, orientation and velocity) and outputs  $K$  grids, representing the posterior probability of the space being occupied at times  $\{t_1, \dots, t_K\}$  in the future. Prediction itself is accomplished with a Growing Hidden Markov Model [14] and an Extended Kalman Filter but the grid representation makes it easy to experiment with other prediction algorithms. The prediction grids are then processed by a fusion module, which currently performs bayesian sensor fusion as described in [8].

In order to produce socially acceptable motion, we have proposed the ‘‘Social Filter’’, which integrates constraints inspired by social conventions in order to evaluate the risk of disturbance represented by navigation decisions. We focus on detecting and predicting conversations in the environment surrounding the wheelchair [11].

## 2.3 Navigation

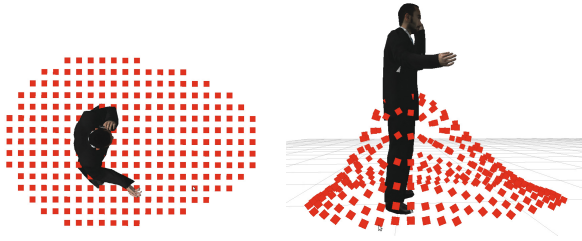
Our navigation system is based on Risk-RRT [2], a partial motion planner which integrates motion predictions to provide safe trajectories. We have also extended the approach by including a mechanism to obtain socially acceptable behavior.

When the wheelchair is transporting a human, it will have to move in a populated environment where an ‘‘optimal’’ behavior may be perceived as unsocial. People will become uncomfortable if they are approached at a distance that is deemed to be too close, where the level of discomfort experienced by the person is related to the importance of his or her space. This simple idea was formalized introducing the concept of *personal space*, first proposed by Hall [3], which characterizes the space around a human being in terms of comfort to social activity.

Another interesting social situation arises when two or more of the persons in the environment are interacting. We model interactions using the concept of *o-space* which has been developed by sociologists [4]. This space can be observed in casual conversations among people where participants’ position and orientation are used to establish boundaries of the space. This space is respected by other people and only participants are allowed to access to it, therefore the intrusion of a stranger causes discomfort. In our path planner, human friendly paths are generated by including a ‘‘Social Filter’’ which transforms those spaces into corresponding cost functions which lead the robot to avoid them. As a result, the choice of a best path done by RiskRRT is based on the ‘‘probability of success’’ calculated for every partial path considering the probability of not encountering a collision along the path and not entering in a personal space or an o-space [11].

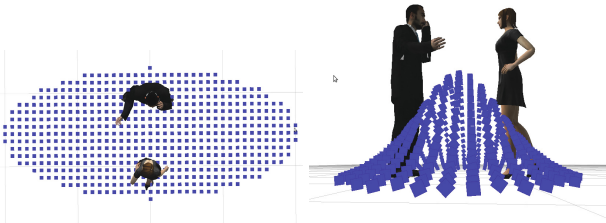
**Modeling Personal Space.** We have modeled personal space as a mixture of two gaussians human centered, one for the front and one for the back of the space, the front is larger as people is more sensitive to this space. Fig. 2 shows an example of personal space as provided by the Social Filter.





**Fig. 2** Personal space calculated by Social Filter Module. Height of the gaussian means Risk of disturbance then maximum disturbance is located at human position.

**Modeling o-Space.** When more than two people are in conversation, they tend to make a formation with circular shape. The o-space could be taken as a circle whose center coincides with that of the inner space. For the specific case of two people, some formations, called F-formations, have been identified as being particularly frequent [4]. The social filter identifies individual F-formations (Vis-a-vis, L-Shape, C-Shape or V-Shape) and builds the corresponding o-space. in Fig. 3, the calculated o-space for a Vis-a-Vis interaction is shown.



**Fig. 3** O-space calculated by Social Filter Module for a Vis-a-Vis F-formation. Maximum risk of disturbance is located at o-space center, in the picture the disturbance is represented by height of Gaussian.

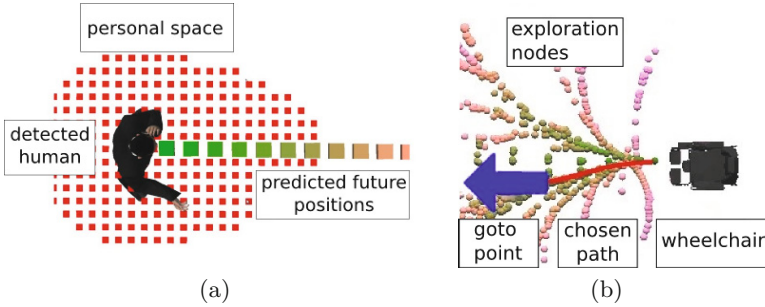
### 3 Experimental Results

It is important to highlight that the proposed experimental platform is an ongoing effort. Thus, the results described below should be considered preliminary. We have conducted experiments both in simulation and with the real platform as described in § 3.1 and 3.2, respectively.

Before going into the details of our results, it is convenient to discuss the graphical elements we will use in our figures. In our tests, humans are represented by a 3D model of a man or woman (4a), red points are used to represent the personal space that should be avoided by the robot. Finally, colored squares in front of the human

represent a simple estimation of future positions, with each color representing a different moment in time.

The wheelchair (Fig. 4(b)) is represented by a 3D model of a wheelchair surrounded by rounded points that represent explored RiskRRT nodes. As in the case of people, different colors are associated with different moments in time. The size of the points represents the computed risk of navigation to that position, where larger points mean bigger risks. Finally, a red solid line is used to represent the path to be followed, with a blue arrow indicating the robot’s goal.



**Fig. 4** (a) Human visualization and symbols meaning; (b) Robot visualization and symbols meaning

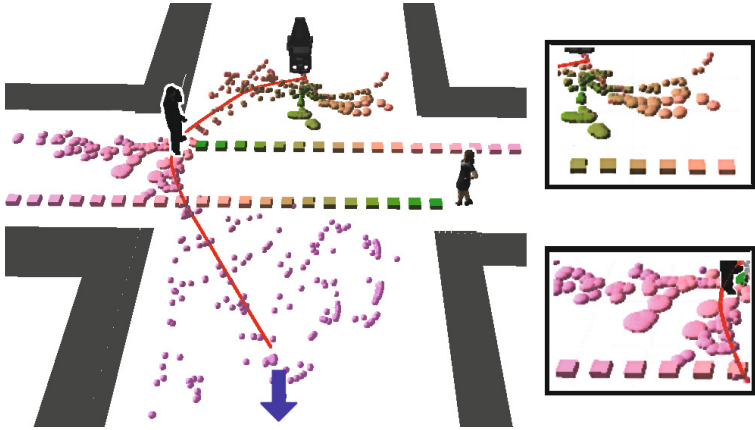
### 3.1 Simulation

**Test Scenarios.** The tests focused on two main functions: predictive navigation and socially acceptable navigation. In the first case, people interfered with the robot’s plans by either following the same path than the robot in the opposite direction or intersecting it at some point. In both cases the robot had to anticipate the human trajectories and generate an alternative collision-free plan.

In the second case, we aimed to assess the capability of the robot to avoid disturbing or causing discomfort to persons that were not moving but were interacting with each other. People were arranged in a manner that the direct path to the robot’s goal would be inside a social interaction zone, so a straight movement to the goal would violate the interaction zone and therefore, the robot had to find alternative paths.

**Prediction and Navigation.** We have conducted extensive tests of the RiskRRT algorithm in simulation. Fig. 5 shows one iteration of the navigation main loop. As it can be seen, the resulting trajectory differs from optimal trajectories obtained by traditional planning algorithms, the robot actually opts for a larger trajectory that avoids obstructing the moving pedestrians.

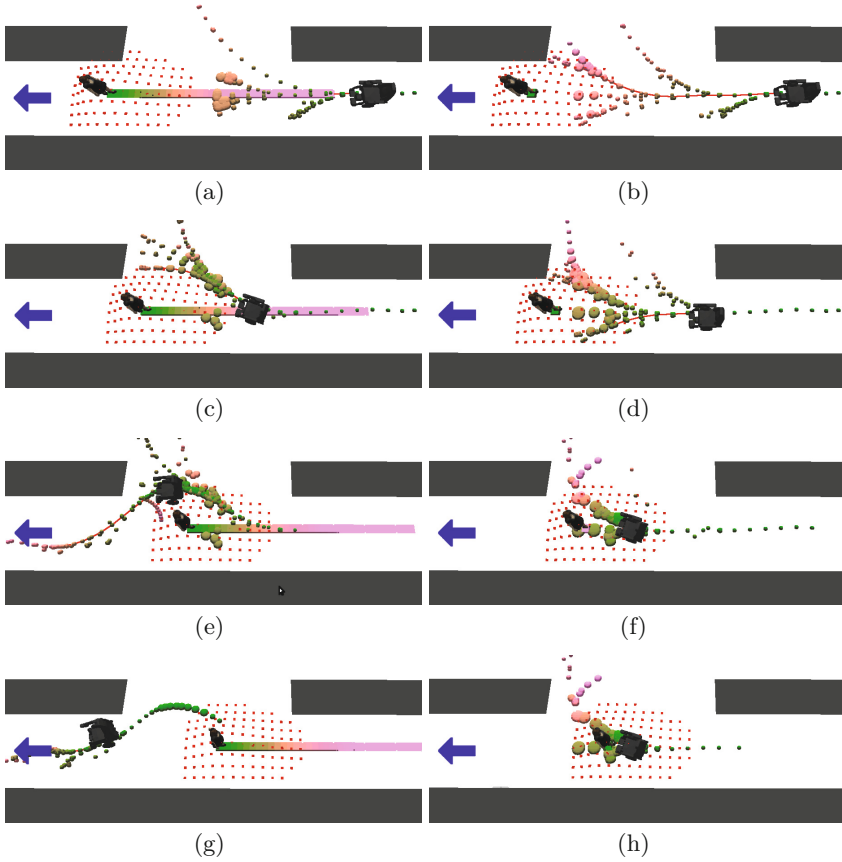
In all our simulations the speed of pedestrians has been fixed to one m/s and maximum speed of our wheelchair is also one m/s.



**Fig. 5** Predictive navigation example. RiskRRT selected a plan (red line) to the goal (blue arrow). The chosen path leads the robot to pass by the back of the first person, and then reduces the speed to let the second person to pass as well. With this strategy, the robot minimizes the risk of collision and the discomfort caused for the two pedestrians. Once second person has passed, the algorithm chooses a straighter path to the goal. Frames at the right of the figure show that estimated risk is bigger at future positions of the wheelchair (circles) which are close to predicted positions of pedestrians (squares).

We have performed a number of tests to assess the effect of including prediction in our motion planning algorithm. Fig. 6 compares the paths that were obtained using predictions of pedestrian movements (left column) with those obtained without predictions (right column). The robot's initial position is on the left end of the corridor while the goal is at right end. Since the corridor is narrow, the only way to avoid colliding or disturbing the pedestrian is by moving aside in the corridor opening before continuing to the goal. In the figure, it is possible to see how, by using predictions, the wheelchair is capable to detect a possible collision in the middle of the corridor (6 a) and to choose an alternative path to reach the goal. Without prediction it takes a straight path to reach the goal which, at first does not seem to pose any risk (6 b) later, when the wheelchair detects the collision (6 d)) and tries to avoid the person, it is already too late.

**Socially Acceptable Navigation.** In order to test socially acceptable behavior, we conducted several simulation tests. Our first test scenario consisted of two interacting people, together with the wheelchair. We realized thirty executions of the planner in very similar conditions, as it can be seen in Fig. 7, when the social filter is off, the plans do avoid people but do not respect social space. When the social filter is turned on again, all the plans managed to respect interaction space without disturbing the involved people.

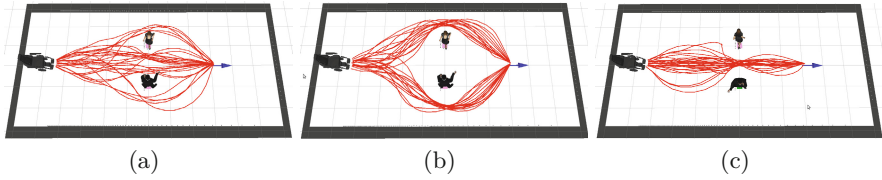


**Fig. 6** Qualitative comparison of predictive navigation (first column) vs non predictive navigation (second column). Prediction helps to discover future high-risk states (a) and anticipate avoidance paths to finally reach the goal (g). Without prediction avoidance begins too late (f) and a collision is unavoidable (h).

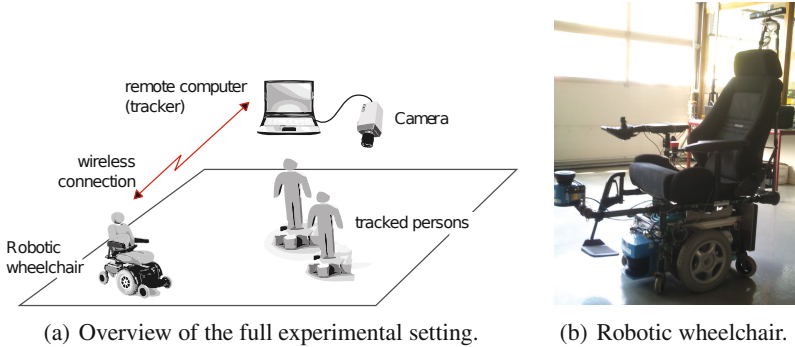
### 3.2 Real Platform

**Experimental Platform.** Our mobile platform (Fig. 8(b)) is a robotic wheelchair manufactured by BlueBotics for the European project MOVEMENT. It is built on a mobile base equipped with a SICK LMS-200 LIDAR, and a Microsoft Kinect RGBD camera. The wheelchair is also equipped with an on-board computer to take care of the low-level hardware control tasks, on top of that it also carries a notebook computer with the navigation, prediction and tracking software.

In addition to the mobile platform, there is also an external camera (Fig. 8(a)) overlooking the test environment. It is connected to an external computer that communicates with the wheelchair via wireless network.



**Fig. 7** Socially acceptable navigation. Each figure shows a sample of generated plans (in red) for thirty executions of RiskRRT: a) without social filter social spaces are not respected, b) and c) with social filter, navigation is socially acceptable. In c) people are looking towards walls, therefore there is no social interacting zone, then navigation respects only their personal spaces.



**Fig. 8** Experimental platform

From the software perspective, the system has been implemented as a number of independent modules using the Robot Operating System (ROS) [10].

**Motion Prediction.** The proposed prediction algorithms has been extensively validated and compared about other state of the art techniques [14]. Our approach consistently yields comparable predictions with much smaller models and is able to update its knowledge as new motion patterns are observed.

To validate the results obtained with our predictor, the scenario chosen to conduct the experiments is the main hall of INRIA Rhône Alpes (Fig. 9(a)). It is an interesting choice as it has a large flow of people during different times of the day, entering and leaving the building during lunch hours and at the beginning and the end of a working day. These conditions provide a realistic and challenging place to conduct experiments on dynamic environments.

The GHMM has been trained using a set of 190 real trajectories. Volunteers were asked to move naturally among interest points in the environment, as the entrance of the hall, the two corridors and the two doors. Fig. 9(b) shows a sample of these trajectories, where the tree interest points located at the stairs illustrates the three separate paths that can be taken when climbing it.

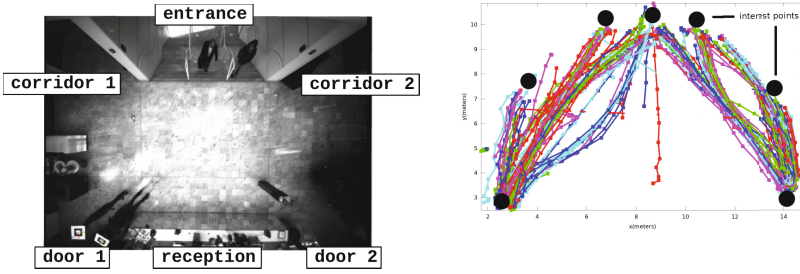


Fig. 9 (a) INRIA Rhône Alpes entrance hall. (b) Real trajectories used in the GHMM training.

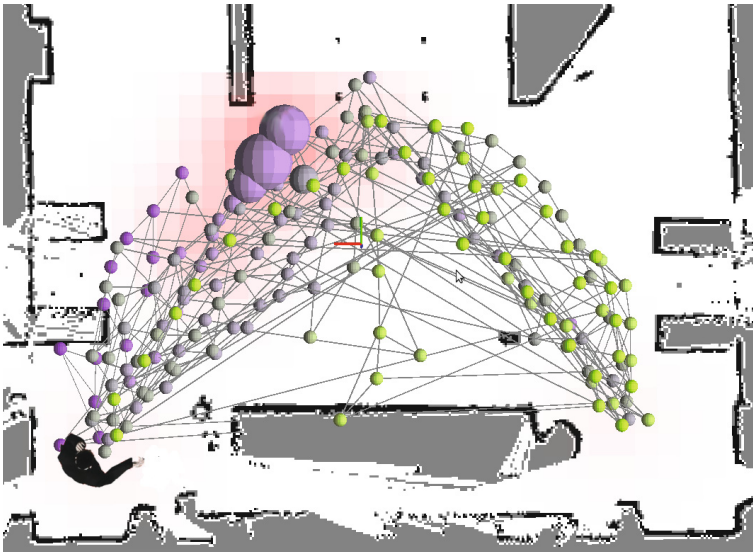
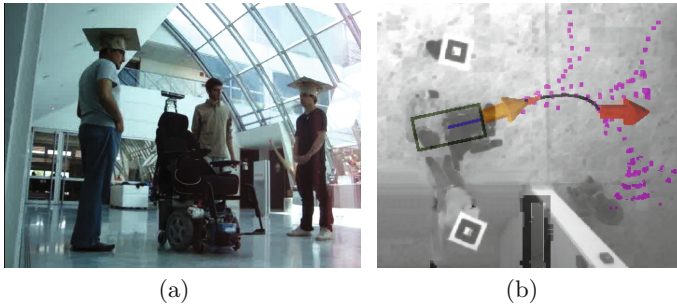


Fig. 10 GHMM learned states (represented by connected nodes) and the prediction of a goal for a person beginning to move from the left door (represented by larger nodes at the left portion of the stairs)

A great advantage of the GHMM is its capability to automatically create, remove and merge redundant states while learning, which result in a more efficient training compared to classical HMM. Fig. 10 illustrates the learned states (represented by spheres) along the INRIA’s hall.

**Socially Acceptable Navigation.** Tests were conducted in the INRIA hall, linking together the tracking, social filters and navigation modules, previously presented. The tracking module fed information to the social filter module which computed social interaction zones, according to the orientation and position of humans in the scene.

Fig. 11(a) shows one image of two persons interacting while the robot passes by, with a researcher closely following. Fig. 11(b) shows the same situation but taken from the overhanging camera linked to the tracker computer, where the robot position, its plan and intended destination can be seen.



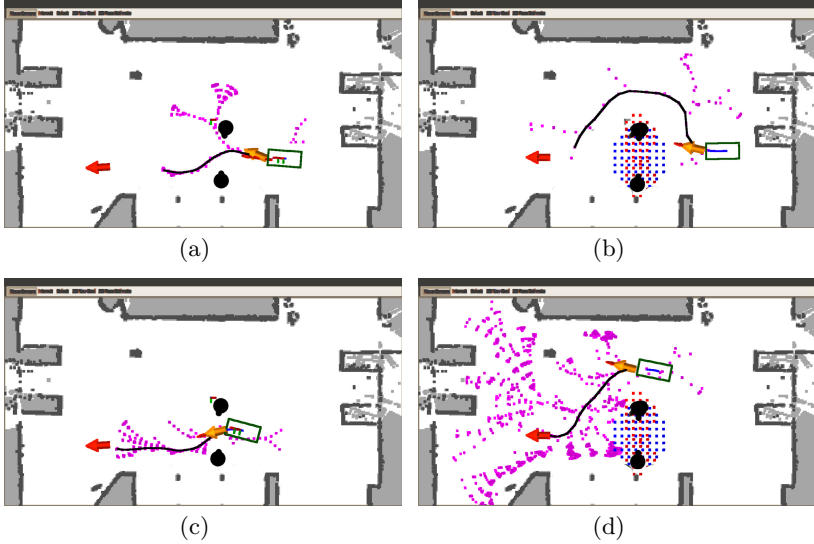
**Fig. 11** (a) Experimental test with two interacting humans and a robotic chair navigating among them. (b) Overview camera image of the test scenario with the robot plan overlaid.

Several tests were conducted to evaluate the capability of the robot to avoid zones that would cause discomfort to the people interacting with each other. We also compared results with and without the social filter module, to demonstrate that not taking into account the zones of social interaction would result in paths that are shorter but “rude” or even frightening.

Fig. 12 shows the two experiments that were performed. The image shows roughly the same initial configuration for the robot and the interacting persons, as well as the same goal. The only difference is that, in the left column the social filter has been disabled while in the right one it is active, which is illustrated by the point cloud between persons.

Due to the absence of a social space, in the left column images, the planning algorithm treats the humans as simple obstacles, and the chosen path is the one that moves straight to the goal. However, when the social filter is active, nodes that are generated inside the interaction zone are penalized with a high risk, and then are excluded during the path search.

This example clearly shows that although a straight path to the goal can be considered to be more efficient in terms of energy and total distance that was traveled, it moves in such a way that it causes discomfort to interacting groups of people in the environment. On the other hand, the example shown in the right column, manages to avoid the zone of interaction, at the cost of traveling a longer distance.



**Fig. 12** The robot is represented by a rectangle, the goal by the leftmost arrow, and interacting people by black circles. Images (a,c,e) show the social filter module deactivated, the resulting trajectory is shorter but socially unacceptable; (b,d,f) images shows a trajectory that is longer but respects the social interaction zones displayed as clouds of blue points.

## 4 Conclusions and Future Work

As we have mentioned above, the platform presented in this paper should be considered work in progress. Nevertheless, we consider that the results we have obtained until now are both relevant and promising and had been instructive in relation to several aspects of the problem at the application and the technical level:

- *Socially acceptable behavior is very important.* Even in our scripted tests, both interacting people and the wheelchair’s user reported that they felt very uncomfortable when the robot passed right through the middle of a talking group.
- *Predictive behavior leads to socially acceptable behavior.* For example, when pedestrians were passing through the robot’s path, it often happened that it stopped (knowing that the path was going to be free) to let the person pass. This seems to indicate that in many cases, knowing how people will move, the most reasonable thing to do is to be polite. It also suggests game theory as a possible way to analyze these interactions.

On the other hand, there are several open fundamental issues that need to be addressed, in particular, the problem of defining proper ways of evaluating comfort and social compliance has not been tackled here. The reason lies in the difficulty to put together experiments which really factor out all those variables that are not being studied. For example, during our experiments, we were applying questionnaires



to the wheelchair passenger with very inconclusive results because the environment contained a flight of stairs going down. The result was that people were too frightened about the wheelchair falling there to be able to consider social discomfort.

As future work we plan to have the help of sociologists to aid in the formulation of questionnaires that can better capture the variables we want to study, as the comfort, for example. We also noticed that the reduced size of the useful space of our test environment (approximately  $70m^2$ ) posed limitations to the variety of tests that we could perform. So future tests will be conducted in a larger and safer (without stairs) environment, with a larger number of humans and more free space for the robot to maneuver, so we can better explore the limitations and advantages of our techniques.

## References

1. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), 603–619 (2002)
2. Fulgenzi, C., Spalanzani, A., Laugier, C.: Probabilistic motion planning among moving obstacles following typical motion patterns. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), <http://hal.inria.fr/inria-00398059/en/>
3. Hall, E.T.: *The hidden Dimension: Man's Use of Space in Public and Private*. The Bodley Head Ltd., London (1966)
4. Kendon, A.: Spacing and orientation in co-present interaction. In: Esposito, A., Campbell, N., Vogel, C., Hussain, A., Nijholt, A. (eds.) *Second COST 2102. LNCS*, vol. 5967, pp. 1–15. Springer, Heidelberg (2010)
5. Kirby, R., Simmons, R., Forlizzi, J.: Companion: A constraint-optimizing method for person acceptable navigation. In: *The 18th IEEE International Symposium on Robot and Human Interactive Communication* (2009)
6. Kollar, T., Tellex, S., Roy, D., Roy, N.: Grounding Verbs of Motion in Natural Language Commands to Robots. In: *International Symposium on Experimental Robotics* (2010), [http://people.csail.mit.edu/stefiel0/publications/tkollar\\_iser2010.pdf](http://people.csail.mit.edu/stefiel0/publications/tkollar_iser2010.pdf)
7. Loong, S.D.: People detection using the Microsoft Kinect (December 2011), [http://www.ros.org/wiki/ppl\\_detection](http://www.ros.org/wiki/ppl_detection) (retrieved February 2012)
8. Moravec, H.: Sensor fusion in certainty grids for mobile robots. *AI Magazine* 9(2), 61 (1988)
9. Parikh, S.P., Grassi Jr., V., Kumar, V., Okamoto Jr., J.: Usability study of a control framework for an intelligent wheelchair. In: *IEEE International Conference on Robotics and Automation*, pp. 4745–4750 (2005)
10. Quigley, M., Gerkey, B., Conley, K., Fausty, J., Footey, T., Leibs, J., Bergery, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: *ICRA Workshop on Open Source Software* (2009)
11. Rios-Martinez, J., Spalanzani, A., Laugier, C.: Understanding human interaction for probabilistic autonomous navigation using Risk-RRT approach. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2014–2019 (September 2011)
12. Simpson, R.C.: Smart wheelchairs: A literature review. *The Journal of Rehabilitation Research and Development* 42(4), 423 (2005), <http://www.rehab.research.va.gov/jour/05/42/4/simpson.html>

13. Vasquez, D., Fraichard, T.: A novel self organizing network to perform fast moving object extraction from video streams. In: IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, Beijing (CN), pp. 4857–4862 (October 2006), <http://emotion.inrialpes.fr/bibemotion/2004/VF04>
14. Vasquez, D., Fraichard, T., Laugier, C.: Growing hidden markov models: a tool for incremental learning and prediction of motion. *International Journal of Robotics Research* 28(11-12), 1486–1506 (2009)

# Socially Assistive Robot Exercise Coach: Motivating Older Adults to Engage in Physical Exercise

Juan Fasola and Maja J. Matarić

**Abstract.** We present the design, implementation, and user study evaluation of a socially assistive robot (SAR) system designed to engage elderly users in physical exercise aimed at achieving health benefits and improving quality of life. We discuss our design methodology, which incorporates insights from psychology research in the area of intrinsic motivation, and focuses on maintaining engagement through personalized social interaction. We describe two user studies conducted to test our design principles in practice with our system. The first study investigated the role of praise and relational discourse in the exercise system by comparing a relational robot coach to a non-relational robot coach. The second study compared physical vs. virtual embodiment in the task scenario. The results of both studies demonstrate the feasibility and overall effectiveness of the robot exercise system.

**Keywords:** socially assistive robotics, human-robot interaction, exercise therapy, intrinsic motivation, embodiment, older adults.

## 1 Introduction

The growing aging population is increasing the demand for healthcare services worldwide. By the year 2050, the number of people over the age of 85 will increase five-fold [1], while the shortfall of nurses and caregivers is already an issue [2]. Regular physical exercise has been shown to be effective at maintaining and improving the overall health of elderly individuals [3]. Social interaction, and specifically high perceived interpersonal social support, has also been shown to have a positive impact on general mental and physical

---

Juan Fasola · Maja J. Matarić  
Interaction Lab, Computer Science Department,  
Viterbi School of Engineering, University of Southern California,  
Los Angeles, CA, USA  
e-mail: {fasola,mataric}@usc.edu

wellbeing [4], in addition to reducing the likelihood of depression [5]. Thus, the availability of physical exercise therapy, social interaction, and companionship will be critical for the growing elderly population; socially assistive robotics (SAR) has the potential to help to address this need.

In this paper, we present the approach, design methodology, implementation details, and user study evaluation of a novel socially assistive robot system that aims to motivate and engage elderly users in simple physical exercise. Our SAR system approach incorporates insights from psychology research into intrinsic motivation and contributes clear design principles for SAR-based therapeutic interventions. For system evaluation, we conducted two user studies with older adults with the following aims: 1) to validate the system approach and its effectiveness in gaining user acceptance and motivating physical exercise in older adults; 2) to study the effect of praise and relational discourse in the system towards increasing user motivation; and 3) to investigate the effect of embodiment in the system by comparing user evaluations of similar physically and virtually embodied SAR exercise coaches.

## 2 Related Work

The literature that addresses assistive robotics intended for and evaluated by the elderly is limited but growing. Representative work includes robots that focus on providing assistance for functional needs, such as mobility and health monitoring [6], navigation and schedule reminders [7], as well as social and emotional needs, such as reducing depression [8] and increasing social interaction [9]. Matsusaka et al. developed an exercise demonstrator robot to aid lead human demonstrators during simple arm exercises to a training group [10]. While similar to our work, this robot was not autonomous and did not have any sensors for which to perceive the users; hence, it did not provide any real-time feedback, active guidance, or personalized training, all of which are employed by our system.

Social agent coaches have previously been developed to autonomously assist individuals in tasks such as physical exercise [11, 12], but have largely been conversational and minimally interactive, without actively monitoring the activity itself. Our system provides a clear distinction in that the agent, a robot in our case, not only provides active feedback and task monitoring, but is also directly responsible for instructing and steering the task as well. Hence, our agent is both an administrator and active participant in the health-related activity.

While previous studies have investigated the positive effect of physical embodiment within the context of human-agent interaction (e.g., [12, 13, 14]), most have recruited a participant pool consisting primarily of young adults. Embodiment studies that have targeted the elderly population include the work of Heerink et al. [15], which investigated the acceptance of assistive

social agents by older adults. While similar to our work, the robot used in their evaluation was a table-top robot (the iCat), and was either controlled via a human operator during interaction with elderly users (Wizard of Oz study), or, like their screen agent, interacted with users through a touch-screen interface. Furthermore, the interaction consisted primarily of short, informational or utility interactions (e.g., medication/agenda reminders, weather forecast, companionship), lasting about 5 minutes and often for a single session. In contrast, our SAR system was designed to engage elderly users in fluid, highly interactive exercise sessions, completely autonomously, while providing active feedback, motivation, and guidance on the task, across multiple sessions of interaction, each lasting 10-20 minutes in duration.

### 3 Robot Exercise Coach

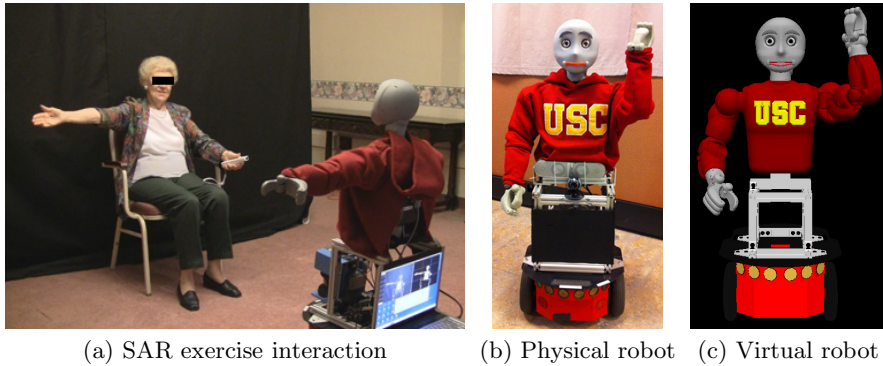
Our approach to designing our SAR system to help address the physical exercise needs of the elderly population was motivated by two basic axioms regarding what a SAR agent must possess, namely: 1) the ability to influence the user's intrinsic motivation to perform the task, and 2) the ability to personalize the social interaction to maintain user engagement and build trust in the task-based human-robot relationship

#### 3.1 Design Principles

In following the above basic axioms, we developed five design principles for the SAR system; all are general and can be applied to any SAR-based therapeutic intervention. The design principles stated that the robot coach should be:

1) *Motivating*. The coaching style and interaction methodology of our SAR exercise system was guided by psychology research in the area of intrinsic motivation, which has been shown to be more effective than extrinsic motivation in achieving long-term user task compliance and behavior change [16]. The motivational techniques utilized by our system were primarily derived from Csikszentmihalyi's theory of flow [17], which asserts that people are intrinsically motivated under conditions of optimal challenge. Towards this end, we focused on providing a variety of challenging exercise games, of varying degrees of difficulty, and alternating the games at a regular pace to prevent user boredom and/or frustrations. We also incorporated indirect competition and user autonomy into the system design, which have also been shown to increase intrinsic motivation [18], by having the robot periodically report the user's high score and by giving the user control over the exercise routine in one of the exercise games.

2) *Fluid and Highly Interactive*. For any task to achieve a state of flow, or maximal enjoyment, in the user, it must establish a clear set of goals, combined with immediate and appropriate feedback [17]. A primary goal of our



**Fig. 1** Exercise scenario and robot coach embodiments

coaching approach was to provide a fluid interaction, which required the robot to both perceive the user’s activity and provide active feedback and guidance in real-time, all with the aim of maintaining user engagement in the task.

3) *Personable*. The SAR coach employed relationship building characteristics such as praise, empathy, humor, references to mutual knowledge, continuity behaviors, politeness, and trust, among others [11]. Praise, which has been shown to increase intrinsic motivation [19], was given upon successful user completion of exercise gestures. The user’s name was also often used to personalize the interaction and to promote the user-robot relationship.

4) *Intelligent*. Trust is a key component to the success of any careprovider-user relationship, and one that is closely linked to the intelligence/helpfulness of the careprovider as perceived by the user [20]. Toward this end, we placed special attention on adding variety to the robot’s utterances to minimize any perceived repetitiveness, in addition to accurate monitoring of user activity.

5) *Task-Driven*. In gaining user trust in the system, it is also important that the tasks employed not only be healthcare-driven, but also be successful in achieving the desired therapeutic behavior. In the case of our SAR exercise coach, this means the tasks must elicit consistent physical exercise among the users (measurable through objective quantitative metrics).

### 3.2 Interaction Scenario

The robot exercise system consists of a socially assistive robot whose purpose is to monitor, instruct, evaluate, and encourage users to perform simple seated physical exercise (“chair aerobics”). The one-on-one interaction scenario consists of the user sitting in a chair across from the robot, with each facing the other. The system’s physical robot platform is a 19-DOF humanoid torso with expressive face mounted on a MobileRobots Pioneer base; the virtual robot platform, used in our embodiment comparison user study, is a

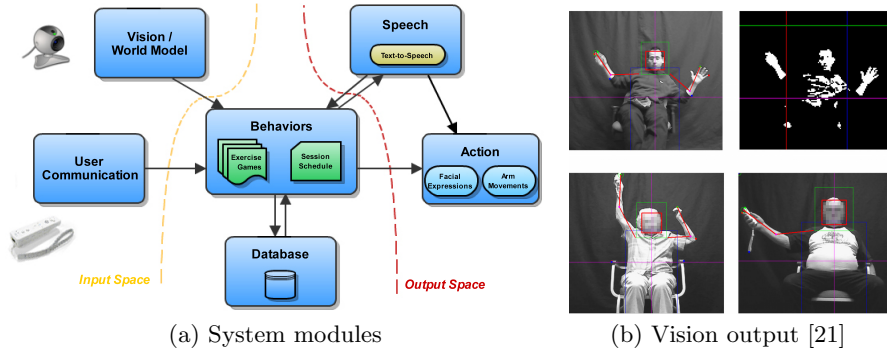


Fig. 2 SAR system architecture and user activity recognition output

computer simulation of the same robot. The exercise interaction setup and robot platforms are shown in Fig. 1.

Four exercise games are available in our system: the *Workout game* (traditional exercise coach), the *Sequence game* (increased repetitions), the *Imitation game* (user autonomy), and the *Memory game* (cognitive game). In the *Workout game*, the robot demonstrates the arm exercises with its own arms, and asks the user to imitate. The robot shows only one exercise gesture at a time, and upon successful completion by the user, generates a different gesture, and the process repeats. The robot gives the user feedback in real-time, providing corrections when appropriate (e.g., “Raise your left arm and lower your right arm” or “Bend your left forearm inward a little”), and praise in response to each successful imitation (e.g., “Great job!” or “Now you’ve got the hang of it.”). This game has the fastest pace of all the four exercise games, as the users generally complete the requested gestures quickly. Difficulty increases in the *Sequence game*, where the robot demonstrates a gesture pair for the user to repeat three times in sequence (i.e., six gestures per sequence). Contrastly in the *Imitation game*, the robot instead imitates user movements in real-time. Lastly, the goal of the *Memory game* is for the user to try and memorize an ever-longer sequence of arm gesture poses, and thus compete against his/her own high score.

### 3.3 System Architecture

The system architecture is comprised of six independent software modules, representing: vision and world model, speech, user communication, behaviors, robot action, and database management. The vision and world model module is responsible for providing information regarding the state of the user to the behavior module for the robot to make task-based decisions during interaction. We developed a novel visual user activity recognition algorithm [21] to recognize user arm poses in real-time, without markers on the user, using only a monocular camera, which can be performed using either color

or motion-based segmentation, and is generalizable to other domains. The speech module communicates to the user through a text-to-speech engine, while the user communicates through a Wiimote wireless button interface. The behavior module interfaces with the robot action and database modules, and is responsible for producing all of the coaching behaviors of the system; hence, it is the module that implements most of the motivational techniques outlined in our SAR design principles. A diagram of the system architecture, along with example vision output, is provided in Fig. 2.

## 4 Study I: Praise and Relational Discourse

We designed and conducted an intrinsic motivation study to investigate the role of praise and relational discourse (politeness, humor, empathy, etc.) in the robot exercise system. Toward that end, the study compared the effectiveness and participant evaluations of two different coaching styles used by our system to motivate elderly users to engage in physical exercise. This section discusses the study methods employed, the dependent measures that were evaluated, and the outcomes of the study with elderly participants.

### 4.1 Study Design

The study consisted of two conditions, Relational and Non-Relational, to explore the effects of praise and communicative relationship-building techniques on a user's intrinsic motivation to engage in the exercise task with the SAR coach. The study design was within-subject; participants saw both conditions, one after the other, and the order of appearance of the conditions was counter-balanced among the participants. Each condition lasted 10 minutes, totaling 20 minutes of interaction, with surveys being administered after both sessions to capture participant perceptions of each study condition independently. The following describes the two conditions in greater detail:

1) *Relational Condition*: In this condition the SAR exercise coach employed all of the social interaction and personalization approaches described in Sect. 3. Specifically, the robot always gave the user praise upon correct completion of a given exercise gesture (an example of positive feedback) and provided reassurance in the case of failure (an example of empathy). The robot also displayed continuity behaviors (e.g., by referencing past experiences with the user), humor, and referred to the user by name, all with the purpose of encouraging an increase in the user's intrinsic motivation to engage in the exercise session.

2) *Non-relational Condition*: In this condition the SAR exercise coach guided the exercise session by providing instructional feedback as needed (e.g., user score, demonstration of gestures, verbal feedback during gesture attempts, etc.), but did not employ explicit relationship building discourse of any kind.



Specifically, the robot did not provide positive feedback (e.g., praise) in the case of successful user completion of an exercise gesture, nor did it demonstrate empathy (e.g., reassurance) in the case of user failure. Furthermore, the SAR coach did not display continuity behaviors, humor, or refer to the user by name.

We recruited elderly individuals to participate in the study through a partnership with be.group, an organization of senior living communities in Southern California, using flyers and word-of-mouth. Thirteen participants (12 female, 1 male) responded and successfully completed both conditions of the study; their ages ranged from 77-92 ( $M = 83$ ,  $S.D. = 5.28$ ). Half of the participants ( $n = 7$ ) engaged in the Relational condition in the first session, whereas the other half ( $n = 6$ ) engaged first in the Non-Relational condition. The following describes the specific evaluation measures captured in the post-session surveys:

1) *Evaluation of Interaction*: Two dependent measures were used to evaluate the interaction with the robot exercise system: the *enjoyableness of the interaction* (6 items; Cronbach's  $\alpha = .93$ ), and the *perceived value or usefulness of the interaction* (4 items; Cronbach's  $\alpha = .95$ ). For both measures, participants were asked to rate how well each item described the interaction on a 10-point scale, anchored by "Describes Very Poorly" (1) and "Describes Very Well" (10).

2) *Evaluation of Robot*: Three dependent measures were used to evaluate the robot coach: as a *companion* (9 items; Cronbach's  $\alpha = .86$ ), as an *exercise coach* (5 items; Cronbach's  $\alpha = .88$ ), and the *social presence* of the robot (7 items; Cronbach's  $\alpha = .82$ ). All items were rated on a 10-point scale.

3) *Direct Comparison of Conditions*: In addition to the above evaluation measures, at the end of the last exercise session we administered one final survey asking the participants to directly compare the two study conditions (labeled "first" and "second") according to ten evaluation categories.

## 4.2 Results

*Evaluation of Interaction Results.* 85% of the participants (11 of 13) rated the Relational condition higher than the Non-Relational condition in terms of enjoyment, and 77% of the participants (10 of 13) rated the Relational condition higher in terms of usefulness than the Non-Relational condition. A Wilcoxon signed-rank test was performed on the data to analyze matched pairs from the sample population's evaluations of both study conditions according to the dependent measures; the results show that the participants evaluated the interaction with the relational robot as significantly more enjoyable/entertaining than the interaction with the non-relational robot ( $W[12] = 4, p < .005$ ), and as somewhat more valuable/useful than the interaction with the non-relational robot, although not to a significant degree ( $W[12] = 15.5, p < 0.10$ ).

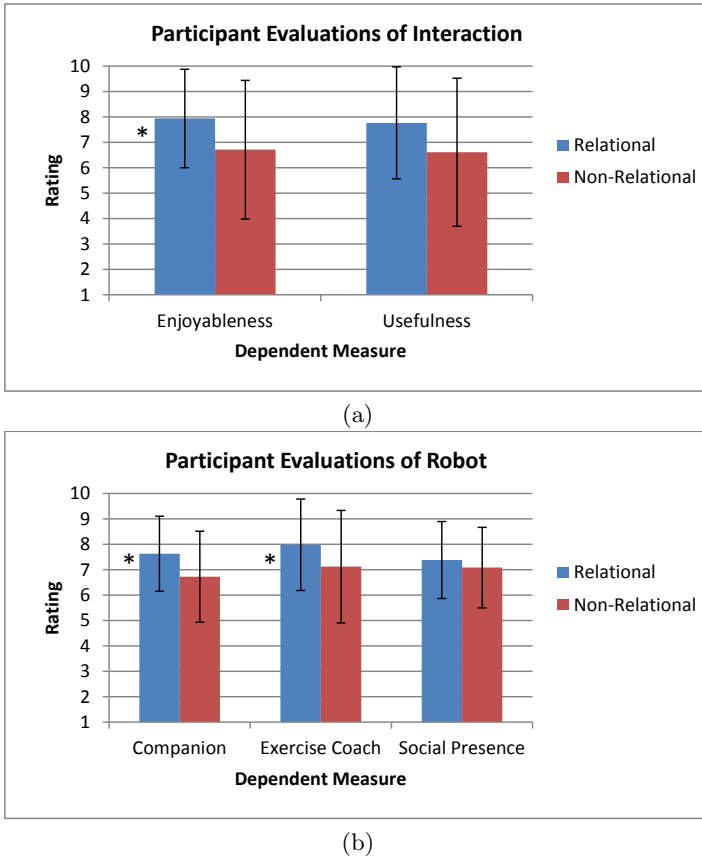
For illustration purposes, Fig. 3(a) shows the average participant ratings of the enjoyableness and usefulness of the interaction for both study conditions.

*Evaluation of Robot Results.* 77% of the participants (10 of 13) rated the relational robot higher than the non-relational robot in terms of companionship, 77% of the participants (10 of 13) rated the relational robot more positively as an exercise coach, and the comparative ratings of social presence between the robot conditions were approximately equal, as 54% of participants (7 of 13) reported higher social presence for the relational robot. A Wilcoxon signed-rank test was again performed on the data; the results show that the participants rated the relational robot as a significantly better companion than the non-relational robot ( $W[13] = 14, p < .05$ ), and as a significantly better exercise coach than the non-relational robot ( $W[11] = 7, p < .02$ ). There was no significant difference in the participant evaluations of social presence between both robot conditions ( $W[12] = 28.5, p > 0.2$ ), with both robots receiving equally high ratings. The average participant ratings of both robot conditions for all three dependent measures are shown in Fig. 3(b).

*Direct Comparison Results.* The direct comparison results demonstrate that, regardless of the order of condition presentation, the participants expressed a strong preference for the relational robot over the non-relational robot. Specifically, the relational robot received 82% of the positive trait votes vs. 16% for the non-relational robot, with the remaining 2% shared equally between them. Notable results include the high number of participants who rated the relational robot as more enjoyable (10 votes, 77%), better at motivating exercise (11 votes, 85%), more useful (11 votes, 85%), and the robot they would choose to exercise with in the future (11 votes, 85%). In contrast, the non-relational robot received a high number of votes for being more frustrating (10 votes, 77%) and more boring (10 votes, 77%) than the relational robot.

### 4.3 Discussion

The results of the study show a strong user preference for the relational robot over the non-relational robot, demonstrating the positive effects of praise and relational discourse in a healthcare task-oriented human-robot interaction scenario. Comments made by participants after the study further illustrate the positive response to the relational robot, including “It’s nice to hear your name, it’s personal. I felt more positive reinforcement,” and from another participant “The robot encourages you, compliments you; that goes a long way.” These results provide significant insight into how people respond to socially assistive robots, and confirm the positive influence that praise and relational discourse have on intrinsic motivation. Furthermore, the results validate our SAR design principles, with particular emphasis on the personable nature of the robot coach, which in turn influences the motivational



**Fig. 3** Participant evaluations of (a) the interaction and (b) the robot coach, for both study conditions [21]. Note: significant differences are marked by asterisks (\*).

capabilities of the system (e.g., by increasing enjoyment) and its perceived usefulness (task-driven), as evidenced by the participant evaluation results. For further analysis and discussion of the study results, the reader is referred to [21].

## 5 Study II: SAR Evaluation and Embodiment

We designed and conducted a second, larger, user study with older adult participants in order to: 1) evaluate the effectiveness of our SAR approach and system design, and 2) investigate the role of physical embodiment in the robot exercise system. Specifically, the study compared the effectiveness and participant evaluation of our physical humanoid robot to that of a computer simulation of the same robot shown on a flat-panel display.

## 5.1 Study Design

To analyze the differences between the physical and virtual embodiments in the exercise system, we implemented both a between-subjects and within-subjects study design. Study participants were divided into two groups, physical robot embodiment vs. virtual robot embodiment, and the study consisted of a total of five 20-minute sessions of exercise interaction with the system, conducted over a two-week period. The between-subjects portion of the study constituted the analysis across both conditions of the data pertinent to the first four exercise sessions, where participants in both groups interacted solely with their designated robot embodiment. Distinct from all previous sessions, in the fifth exercise session participants in both groups interacted with the alternative robot embodiment (physical robot group with the virtual robot, virtual robot group with the physical robot). The within-subjects portion of the study consisted of the data analysis between the fourth and fifth exercise sessions within each group.

Surveys were administered at the end of the fourth and fifth sessions for participant evaluation of the SAR system. The same five measures from the first user study were again used to evaluate both the interaction and robot coach, with the addition of three measures: *helpfulness* of the robot (4 items; Cronbach's  $\alpha = .96$ ), *intelligence* of the robot (4 items; Cronbach's  $\alpha = .93$ ), and *social attraction* towards the robot (4 items; 7-point scale; Cronbach's  $\alpha = .88$ ).

To help assess the effectiveness of the SAR exercise system in motivating exercise among the participants, we also collected fifteen different objective measures during the exercise sessions regarding user performance and compliance in the exercise task. Five performance measures were captured during user interaction in the Workout game, including: the *average time to gesture completion* (from the moment the robot demonstrates the gesture, to successful user completion of the gesture), *number of seconds per exercise completed*, *number of failed exercises*, *number of movement prompts* by the robot to the user due to lack of arm movement, and *feedback percentage*. The feedback percentage measure refers to the fraction of gestures, out of the total given, where the robot needed to provide verbal feedback to the user regarding their arm positions in order to help guide them to correct gesture completion. Two measures concerned user activity during the entire exercise session; they were: the *average total number of exercises completed*, and *number of breaks* taken by the user. The eight remaining measures were captured in the other three exercise games, and were similar in nature to those captured in the Workout game.

We recruited thirty-three older adults to participate in the study, again through a partnership with be.group. Half of the participants were placed in the physical robot group ( $n = 16$ ), and the other half in the virtual robot group ( $n = 17$ ). The sample population consisted of 27 female (82%) and 6 male (18%) participants, with ages ranging from 68 to 88 ( $M = 76$ ,  $S.D. = 6.32$ ).

**Table 1** Results of between-subjects and within-subjects embodiment comparison

Dependent Measure	Physical Robot	Virtual Robot	P.V.	V.P.
<i>Interaction Evaluation</i>				
<i>Between-Subjects Analysis</i>				
<i>Within-Subjects Analysis</i>				
Enjoyable	7.51 (1.77)*	6.00 (2.01)	6.94 (2.21)	7.11 (2.35) <sup>†</sup>
Valuable/Useful	8.14 (1.66)*	6.19 (2.39)	7.70 (2.13) <sup>††</sup>	7.51 (2.26)*
<i>Robot Evaluation</i>				
Helpful	8.11 (1.98)*	6.26 (1.98)	8.28 (1.61)	7.44 (2.48)*
Social Attraction	4.70 (1.40)*	3.61 (1.54)	4.31 (1.43) <sup>†</sup>	4.36 (1.58) <sup>††</sup>
Social Presence	7.88 (0.94)*	6.47 (2.01)	6.98 (0.97)**	7.22 (1.66) <sup>†</sup>
Companion	7.48 (2.07) <sup>††</sup>	6.23 (1.84)	7.12 (1.94)	7.42 (1.87)*
Intelligence	8.17 (2.02) <sup>†</sup>	6.76 (2.09)	7.61 (1.54)	7.79 (2.66)*
Exercise Partner	7.18 (2.17) <sup>††</sup>	5.76 (2.18)	6.95 (1.60)	7.01 (2.16)*

P.V. = Physical robot group evaluating the virtual robot (5th session).

V.P. = Virtual robot group evaluating the physical robot (5th session).

<sup>††</sup> $p < .10$ , <sup>†</sup> $p < .06$ , \* $p < .05$ , \*\* $p < .01$ , \*\*\* $p < .001$ .

## 5.2 Coach Embodiment Comparison Results

A two-tailed independent T-test was performed on the survey data following the fourth exercise session, to compare participant evaluations of the robot embodiments as well as their interactions with them across the two study groups. Survey results from the fourth session were used to perform the comparison analysis as they were less likely to contain scores influenced by the effect of novelty. Table 1 provides the complete set of embodiment comparison results.

The participants evaluated the interaction with the physical robot embodiment as more enjoyable ( $t[31] = 2.29, p < .03$ ) and as more valuable/useful ( $t[29] = 2.72, p = .01$ ) than the interaction with the virtual robot embodiment. In addition, the participants rated the physical robot as significantly more helpful than the virtual robot ( $t[31] = 2.66, p = .01$ ), more socially attractive ( $t[30] = 2.09, p < .05$ ), and as having significantly stronger social presence ( $t[23] = 2.59, p < .02$ ). Evaluations of the remaining measures were also favorable to the physical robot, though not to a significant degree, as the participants rated the physical robot as somewhat more of a companion ( $t[30] = 1.81, p < .08$ ), more intelligent ( $t[31] = 1.96, p < .06$ ), and a moderately better exercise partner ( $t[31] = 1.87, p = .07$ ) than the virtual robot.

To test for significant differences between embodiments within each study group, we used a two-tailed paired T-test to analyze the data gathered from the fourth and fifth session post-session surveys. The within-subjects results largely agree with the results of the between-subjects comparison (see Table 1), with the exception of the physical robot group's ratings of

**Table 2** Results of direct comparison survey for all  $n = 33$  participants

	Physical	Virtual	Both Equal	Binomial Test
Enjoy More	25 (76%)**	6 (18%)	2 (6%)	$p < .01$
More Intelligent	13 (40%)	6 (18%)	14 (42%)	$p = 0.3$
More Useful	21 (64%)*	7 (21%)	5 (15%)	$p < .05$
Prefer to Exercise with	27 (82%)**	4 (12%)	2 (6%)	$p < .0001$
Better at Motivating	22 (67%)**	4 (12%)	7 (21%)	$p < .01$
More Frustrating	10 (30%)	14 (43%)	9 (27%)	$p = 0.5$
More Boring	4 (12%)	17 (52%)*	12 (36%)	$p < .05$
More Interesting	23 (70%)**	5 (15%)	5 (15%)	$p < .01$
More Entertaining	25 (76%)**	4 (12%)	4 (12%)	$p < .001$
Choice from now on	25 (76%)**	7 (21%)	1 (3%)	$p < .01$

\* $p < .05$ , \*\* $p < .01$ , \*\*\* $p < .001$ .

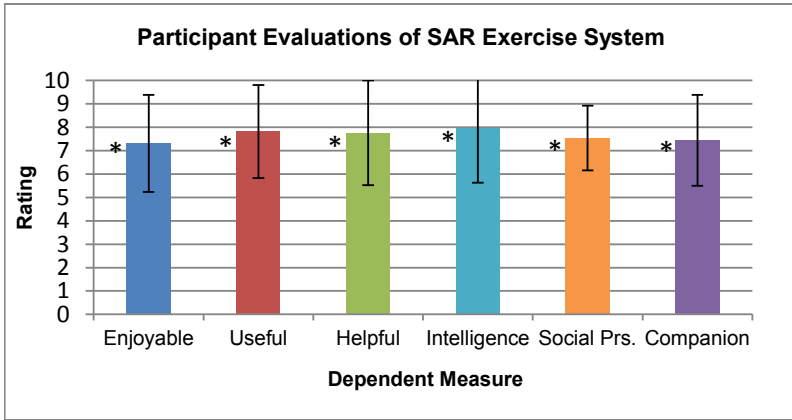
the virtual embodiment, which although lower on average than the physical embodiment were not significantly different, indicating possible carryover effects in the evaluation.

As in the first user study, participants were asked to directly compare both embodiments with respect to ten evaluation categories. The results, provided in Table 2, show clear preference for the physical robot, which received 81% of the positive trait votes vs. 19% for the virtual robot among participants who chose one embodiment over the other (85% of the sample); a significant margin as indicated by a two-sided exact binomial test (201 votes vs. 63,  $p < .0001$ ).

### 5.3 SAR System Evaluation Results

In order to evaluate the effectiveness of the SAR exercise system, we analyzed the data of the physical robot group's fourth session of interaction, together with the data of the virtual robot group's fifth session of interaction. Therefore, the SAR system evaluation results, regarding user perceptions and user exercise performance, were gathered from all 33 older adult participants and consist only of participant interactions with the physically embodied robot, as this condition represented the ideal interaction scenario for users of the system.

**User Evaluations of SAR System.** To analyze the user evaluations of the SAR exercise system, we performed a two-tailed independent T-test to test for significant differences between the participant ratings of the subjective measures and a neutral evaluation rating. The neutral evaluation rating distribution was obtained from a uniform sampling of the rating scale (integers from 1 to 10) for the approximate number of participants, and has a



**Fig. 4** Participant evaluations of SAR system interaction and robot coach. Note: Significant differences in comparison to neutral rating are marked by asterisks (\*).

mean rating of 5.5 (*S.D.* = 2.90). This uniform sampling assumes no prior information regarding user perceptions of the system, and thus is deemed neutral.

The results of the user evaluation of the SAR exercise system were very encouraging, as they showed a notable level of user acceptance of the system, as evidenced by the high ratings across each of the subjective measures, and highlighted the effectiveness of our SAR system design principles. The participants evaluated the interaction with the SAR exercise system as enjoyable ( $M = 7.3, S.D. = 2.07$ ) and valuable/useful ( $M = 7.8, S.D. = 1.99$ ). The ratings for both measures were found to be significantly more positive than a neutral evaluation (enjoyableness:  $t[52] = 2.81, p < .01$ ; usefulness:  $t[50] = 3.64, p < .001$ ). These results illustrate the effectiveness of the system in promoting intrinsic motivation within the users to engage in the healthcare task (intrinsic motivation is characterized by enjoyment [17]), and in guiding the task-driven interaction towards achieving beneficial health outcomes for the user.

Regarding user perceptions of the robot coach of the SAR exercise system, the participants rated the robot highly and significantly more positive than neutral in terms of helpfulness ( $M = 7.8, S.D. = 2.25; t[54] = 3.42, p < .01$ ), intelligence ( $M = 8.0, S.D. = 2.35; t[56] = 3.68, p < .001$ ), social presence ( $M = 7.5, S.D. = 1.38; t[41] = 3.48, p < .01$ ), and as a companion ( $M = 7.4, S.D. = 1.94; t[50] = 3.08, p < .01$ ). The participants also rated the robot coach favorably in terms of social attraction ( $M = 4.5$  (on a 7-point scale),  $S.D. = 1.48; t[62] = 1.22, n.s.$ ), and as an exercise partner ( $M = 7.1, S.D. = 2.14; t[53] = 2.45, p < .02$ ). These results illustrate the personable nature and intelligence of the robot coach, as perceived by the participants, both of which aid in the development of trust within the human-robot relationship, and

**Table 3** User exercise performance statistics for all  $n = 33$  older adult participants

Performance Measure	Mean (std.)
<i>Workout game:</i>	
Time to Gesture Completion (seconds)	2.54 (0.80)
Seconds per Exercise	5.37 (0.88)
Feedback Percentage	7.4% (4.8%)
Number of Failed Gestures	0
Number of Movement Prompts <sub>W</sub>	0
<i>Sequence game:</i>	
Time to Gesture Completion (seconds)	5.73 (1.37)
Number of Sequences Completed	4.97 (1.16)
Number of Gesture Pairs Completed	14.9 (3.41)
Feedback Percentage	19.6% (11.2%)
<i>Memory game:</i>	
Maximum Score	6
Average Maximum Score	3.52 (1.25)
Time per Gesture Attempt (seconds)	7.62 (3.98)
<i>Imitation game:</i>	
Number of Movement Prompts <sub>I</sub>	0.37 (0.63)
<i>Entire Session:</i>	
Total Number of Exercises Completed	107.75 (18.1)
Number of Breaks Taken	1 (1.26)

were design goals of our SAR system approach towards providing successful therapeutic interventions. A plot showing participant evaluations of the SAR system interaction and robot coach is shown in Fig. 4.

**User Exercise Performance Statistics.** The collected statistics regarding participant performance in the exercise task were also very encouraging, as they demonstrated a consistently high level of user exercise performance and compliance within the exercise task.

User compliance and performance in the Workout game were high. The average gesture completion time was 2.54 seconds ( $S.D. = 0.80$ ), and the overall exercise performance averaged 5.37 seconds per exercise ( $S.D. = 0.88$ ), which also includes time taken for verbal praise, feedback, and score reporting from the robot. The low percentage of necessary corrective feedback, averaging 7.4%, combined with zero failures, and zero movement prompts during the interaction sessions, are all very encouraging results, as they suggest that the participants were motivated to do well on the exercises consistently throughout the interaction.

A summary of all statistics regarding user performance, including those from the Sequence, Memory, and Imitation games, can be found in Table 3.



The consistently high level of exercise performance achieved by the study participants, as evidenced by the results, validates the effectiveness of the SAR exercise system approach and design methodology in motivating elderly users to engage in physical exercise, and demonstrates the potential of the technology to provide guided care and to help elderly users achieve beneficial health outcomes.

#### ***5.4 Study Expansion with Young Adults***

To analyze and compare user evaluations and embodiment effects across age groups, we expanded the study to include 33 young adult participants (6 female, 27 male), yielding a combined sample of 66 participants, all of whom engaged in five sessions of interaction with our SAR exercise system (330 sessions total). The results of the study with young adults were largely consistent with those observed with the older adult participants. Among the combined results, a two-sided exact binomial test showed the physical robot coach received significantly more positive votes than the virtual robot coach upon direct comparison (425 votes vs. 103,  $p < .0001$ ). For further discussion of the results of both the young adult and combined populations, we refer the reader to [22].

## **6 Conclusions**

We have presented a set of design principles for socially assistive robots in therapeutic contexts, and a novel robot system that embodies and validates those principles, designed to motivate and engage elderly users in physical exercise. Our SAR system approach, design methodology, and implementation details were discussed, including five SAR design principles which can be applied to a variety of human-robot interaction-based healthcare interventions.

Results of the user evaluation of the SAR exercise system in two user studies with older adults were presented, which together showed strong participant preferences for relational and physically embodied coaches. The successful acceptance of the SAR exercise system from elderly users, as evidenced by the high participant evaluations of the system and consistent exercise performance in both user studies, validates our SAR system approach, design, algorithms, and effectiveness, and illustrates the potential of SAR technology to help older adults achieve beneficial health outcomes and improve quality of life.

**Acknowledgements.** We are grateful to be.group, their staff and participating residents, and to Aras Akbari and Farrokh Langroodi for the study data collection. This research was supported by a grant from the Robert Wood Johnson Foundation's Pioneer Portfolio through its national program, Health Games Research, and National Science Foundation grants IIS-0713697, CNS-0709296, and IIS-1117279.

## References

1. United States Census Bureau: U.S. Census Bureau Report (Issue Brief No. CB09-97) (2009), <http://www.census.gov/population/international>
2. Buerhaus, P.: Current and future state of the US nursing workforce. *J. Amer. Med. Assoc.* 300(20), 2422–2424 (2008)
3. Dawe, D., Moore-Orr, R.: Low-intensity, range-of-motion exercise: invaluable nursing care for elderly patients. *J. Adv. Nursing* 21, 675–681 (1995)
4. Moak, Z.B., Agrawal, A.: The association between perceived interpersonal social support and physical and mental health: results from the national epidemiological survey on alcohol and related conditions. *J. Public Health* 32, 191–201 (2010)
5. George, L., Blazer, D., Hughes, D., Fowler, N.: Social support and the outcome of major depression. *British J. Psychiatry* 154, 478–485 (1989)
6. Dubowsky, S., Genot, F., Godding, S., Kozono, H., Skwersky, A., Yu, H., Shen Yu, L.: PAMM - a robotic aid to the elderly for mobility assistance and monitoring: a helping-hand for the elderly. In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 1, pp. 570–576 (2000)
7. Montemerlo, M., Pineau, J., Roy, N., Thrun, S., Verma, V.: Experiences with a mobile robotic guide for the elderly. In: *Proc. AAAI Nat. Conf. on Artificial Intelligence*, pp. 587–592 (2002)
8. Wada, K., Shibata, T., Saito, T., Tanie, K.: Analysis of factors that bring mental effects to elderly people in robot assisted activity. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1152–1157 (2002)
9. Kidd, C., Taggart, W., Turkle, S.: A sociable robot to encourage social interaction among the elderly. In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3972–3976 (2006)
10. Matsusaka, Y., Fujii, H., Okano, T., Hara, I.: Health exercise demonstration robot TAIZO and effects of using voice command in robot-human collaborative demonstration. In: *Proc. IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, pp. 472–477 (2009)
11. Bickmore, T.W., Picard, R.W.: Establishing and maintaining long-term human-computer relationships. *Trans. Comput.-Hum. Interact.* 12(2), 293–327 (2005)
12. Kidd, C.D., Breazeal, C.: Robots at home: understanding long-term human-robot interaction. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 3230–3235 (2008)
13. Powers, A., Kiesler, S., Fussell, S., Torrey, C.: Comparing a computer agent with a humanoid robot. In: *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pp. 145–152 (2007)
14. Jung, Y., Lee, K.M.: Effects of physical embodiment on social presence of social robots. In: *Proceedings of Presence*, pp. 80–87 (2004)
15. Heerink, M., Kröse, B., Evers, V., Wielinga, B.: Assessing acceptance of assistive social agent technology by older adults: the almere model. *Int. J. Social Robotics* 2(4), 361–375 (2010)
16. Dienstbier, R.A., Leak, G.K.: Effects of monetary reward on maintenance of weight loss: an extension of the overjustification effect. *Am. Psychol. Assoc. Conv.* (1976)
17. Csikszentmihalyi, M.: *Beyond boredom and anxiety*. Jossey-Bass, San Francisco (1975)

18. Fisher, C.D.: The effects of personal control, competence, and extrinsic reward systems on intrinsic motivation. *Organizational Behavior and Human Performance* 21, 273–288 (1978)
19. Vallerand, R.J., Reid, G.: On the causal effects of perceived competence on intrinsic motivation: a test of cognitive evaluation theory. *J. Sport Psychol.* 6, 94–102 (1984)
20. Tway, D.C.: A construct of trust. Dissertation, University of Texas at Austin (1994)
21. Fasola, J., Matarić, M.J.: Using socially assistive human-robot interaction to motivate physical exercise for older adults. In: Kanade, T. (ed.) *Proc. of the IEEE, Special Issue on Quality of Life Technology* (2012)
22. Fasola, J., Matarić, M.J.: Comparing physical and virtual embodiment in a socially assistive robot exercise coach for the elderly. University of Southern California CRES Technical Report, CRES-11-003 (2011)

# Interpreting and Executing Recipes with a Cooking Robot

Mario Bollini, Stefanie Tellex, Tyler Thompson,  
Nicholas Roy, and Daniela Rus

## 1 Motivation

The creation of a robot chef represents a grand challenge for the field of robotics. Cooking is one of the most important activities that takes place in the home, and a robotic chef capable of following arbitrary recipes would have many applications in both household and industrial environments. The kitchen environment is a semi-structured proving ground for algorithms in robotics. It provides many computational challenges, such as accurately perceiving ingredients in cluttered environments, manipulating objects, and engaging in complex activities such as mixing and chopping. Yet it also allows for reasonable simplifying assumptions due to the inherent organization of a kitchen around a human-centric workspace, the consistency of kitchen tools and tasks, and the ordered nature of recipes. We envision a robotic chef, the BakeBot, which can collect recipes online, parse them into a sequence of low-level actions, and execute them for the benefit of its human partners. We present first steps towards this vision, by combining techniques for object perception, manipulation, and language understanding to develop a novel end-to-end robot system able to follow simple recipes and by experimentally assessing the performance of these approaches in the kitchen domain.<sup>1</sup>

## 2 Problem Statement

This paper describes progress towards a robotic system which is able to read and execute simple recipes. The robot is initialized with a set of ingredients laid out on the

---

Mario Bollini · Stefanie Tellex · Tyler Thompson · Nicholas Roy · Daniela Rus

Computer Science and Artificial Intelligence Laboratory,

Massachusetts Institute of Technology

e-mail: {mbollini, stefie10, tcthompson, nickroy, rus}@csail.mit.edu

<sup>1</sup> Video of recipe execution:

[http://people.csail.mit.edu/mbollini/bakebot\\_planning.mp4](http://people.csail.mit.edu/mbollini/bakebot_planning.mp4)

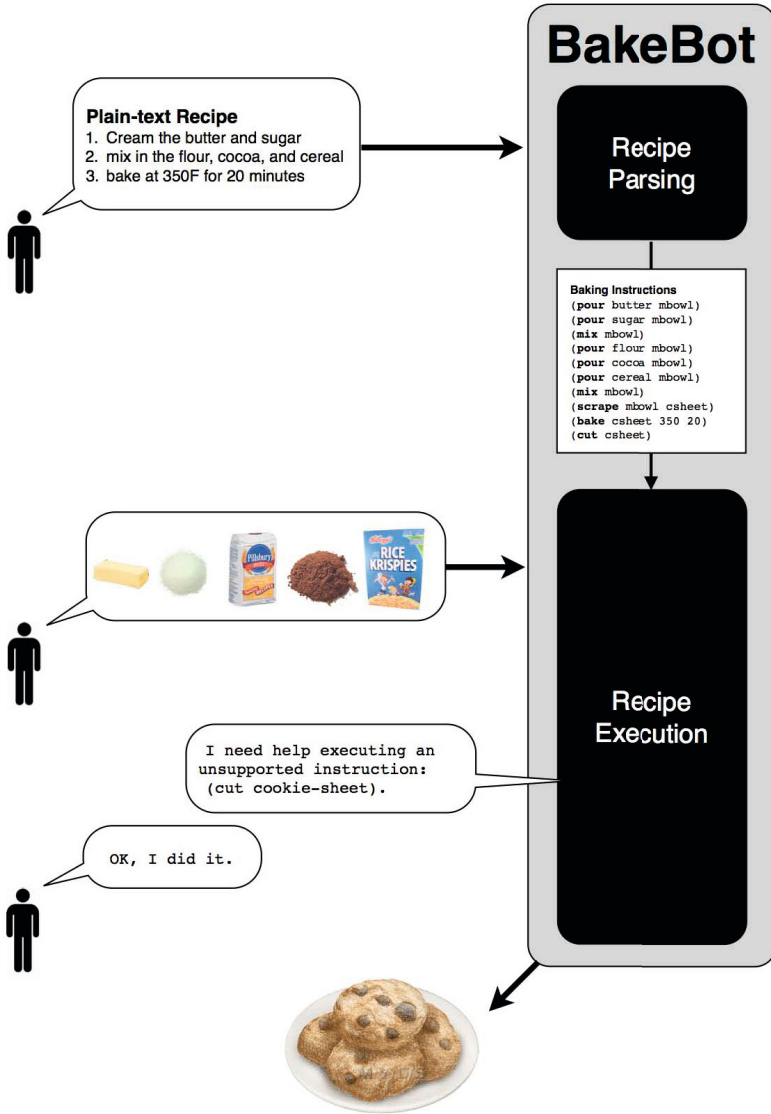
table and a set of natural language instructions describing how to use those ingredients to cook a dish such as cookies, salad, or meatloaf. For example, ingredients might include flour, sugar, butter, and eggs arrayed in labeled bowls on the table, and instructions might include statements like “Add sugar and beat to a cream.” We assume the robot has access to a model of the environment including the set of ingredients and their labels, and the location of tools such as the oven. In addition, it has a repertoire of primitive actions, such as pouring ingredients into a bowl and mixing them. The robot must parse the text of recipe and infer an action sequence in the external world corresponding to the instructions. It then executes the inferred action sequence on the PR2 robotic manipulation platform, performing the motion and task planning necessary to follow the recipe to create the appropriate dish. For example, for the instructions above, the robot might empty the bowl of sugar into the mixing bowl, then stir the ingredients in the bowl. Figure 2 shows the BakeBot system inputs and outputs.



**Fig. 1** BakeBot, a PR2 robot system that executes simple baking recipes. The PR2 is modified with a protective covering to prevent damage from spills and a spatula which is bolted to the end effector to enable mixing.

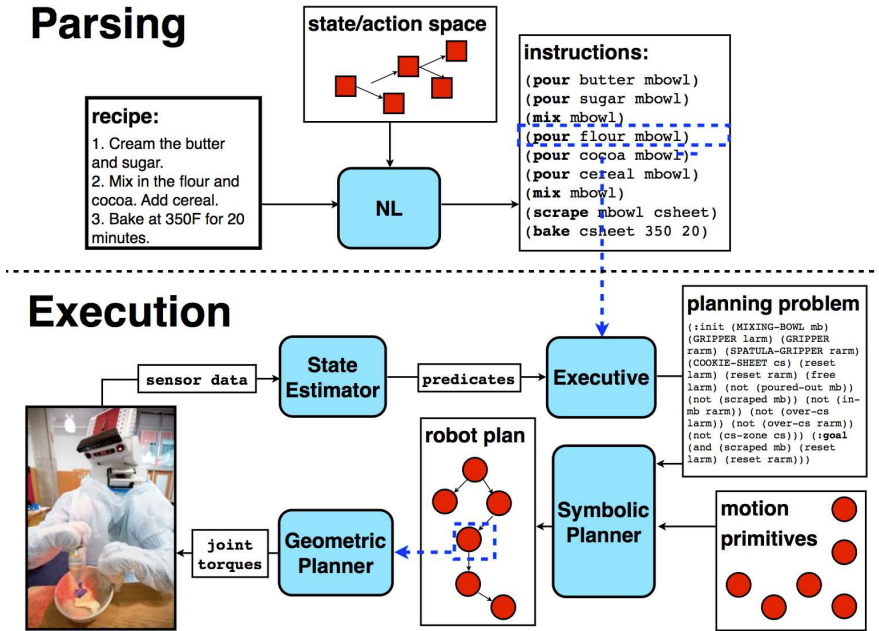
### 3 Related Work

Many previous authors have described robotic systems for following natural language instructions. Previous work focused on understanding natural language route directions [9, 10, 6] and mobile-manipulation commands [12]. Similar to existing approaches, our system learns a model that maps between the text of the instructions and a reward function over actions in the predefined state-action space, specifically focusing on the domain of following recipes. Recipes present a unique challenge to these approaches because of the richer space of actions inherent to the cooking domain. BakeBot uses the low-level manipulation and perception system described



**Fig. 2** The human interaction with the BakeBot system for recipe execution. First the person provides the plain-text recipe and the measured ingredients. Then BakeBot infers a sequence of baking primitives to execute that correspond to following the recipe. If BakeBot encounters an unsupported baking primitive, it asks its human partner for help executing the instruction. The end result is baked cookies.

in Rusu et al. [11]. Beetz et al. [2] have demonstrated dispensing pancake batter from a premixed container and flipping the pancakes on a skillet. In this paper we demonstrate an end-to-end robot cooking system capable of implementing any baking recipe that requires pouring, mixing, and oven operations on premeasured ingredients provided to the system. Our system is able to follow recipes downloaded from the internet; we demonstrate it by following two different recipes in the real world and by further evaluating its performance on a larger test set in simulation.



**Fig. 3** Architecture of the BakeBot system. The NL system processes the plain text recipe, producing a high-level plan which is sent to the robot. For each instruction in the high-level plan, the motion planner assembles a motion plan and executes it on the PR2 robot.

### 4 Technical Approach

The robot’s goal is to read the text of a natural language recipe, and use it to infer an action sequence in the environment that corresponds to preparing the dish described in the recipe. The robot first segments the recipe into sentences based on punctuation. Then for each sentence, it infers an action sequence in the environment corresponding to the words in the sentence. After executing the action sequence successfully, the robot will have produced the appropriate dish, for example cookies, brownies, or meatloaf. The system architecture appears in Figure 3.

We formally define a state-action space for the kitchen domain using primitive actions such as *mix*, *pour*, and *bake*. Within this state space, many specific action

trajectories can be followed, yielding a variety of different dishes. In order to follow a specific recipe, we define a reward function based on the text of the recipe. The robot uses forward search to find the sequence of states and actions that maximizes reward, then executes those actions in the external world. Because the text of the recipe generates the reward function, this optimization corresponds to finding an action sequence that follows the recipe.

### 4.1 State/Action Space for the Kitchen

We define a state  $S_k$  as the collection of unused ingredients  $S_k^i$  in the workspace, the mixing bowl  $S_k^b$  and its contents, the cookie sheet  $S_k^s$  and its contents, and a toaster oven  $S_k^o$  and its temperature and contents. Given any state  $S_k$ , we define  $actions(S_k)$  to be the set of available actions in that state:

- For each unpoured ingredient  $S_k^i$ ,  $pour(S_k^i, S_k^b) \in actions(S_k)$ .
- If  $nonempty(S_k^b)$  then  $mix(S_k^b) \in actions(S_k)$ .
- If  $nonempty(S_k^b)$  then  $scrape(S_k^b, S_k^s) \in actions(S_k)$ .
- If  $empty(S_k^o)$  then  $preheat(S_k^o) \in actions(S_k)$ .
- If  $empty(S_k^o) \wedge nonempty(S_k^s)$  then  $bake(S_k^s) \in actions(S_k)$ .

After executing an action such as *pour* in state  $S_k$ , the next state  $S_{k+1}$  contains one less unused ingredient, and the mixing bowl  $S_{k+1}^b$  contains the corresponding poured ingredient. Although the currently available actions do not always execute robustly on the real robot, they support a surprising array of recipes, ranging from brownies to meatloaf to salads. Over time we plan to increase the robustness of the existing actions and add additional ones such as chopping and whisking in order to increase the range of recipes that the robot supports.

If no supported action exists for an instruction, the robot prompts the user, visually via the terminal and audibly via a text-to-speech synthesizer, to execute the action. This enables a wider array of actions than the manipulation platform and motion primitive set currently support by leveraging support from a human partner.

### 4.2 Reward Function

In order to follow a recipe, the system takes as input the natural language text of a recipe and induces a reward function over the state/action space. The reward function is learned from a labeled dataset of recipes paired with the correct action sequence. Formally, a recipe consists of a list of sentences  $d_1 \dots d_N$ . For each sentence, the system infers a sequence of states  $S_1 \dots S_K$  that maximizes the reward function  $R$ :

$$\operatorname{argmax}_{S_1 \dots S_K} \sum_j R(d_j, S_m \dots S_n) \quad (1)$$

We define the reward function as a probability distribution parametrized using a log-linear model [3]. The model is trained from a corpus of recipes annotated with



the correct action. The feature functions are bag-of-words features crossed with the presence of specific actions and arguments in the state sequence. For example, a feature would be “Mix”  $\in d_j \wedge pour \in S_1 \dots S_k$ . The system takes positive examples from the annotations. To form negative examples, it finds action trajectories which result in an incorrect final state. The system then finds model parameters which maximize the likelihood of this training set using a gradient descent algorithm.

We use a beam search to jointly optimize the action sequence across multiple sentences. The sequence of states and actions is then passed to the execution model to run on the physical robot.

### 4.3 Plan Assembly

The inferred state/action sequence is interpreted into a robot action plan of *baking primitives*. Each baking primitive corresponds to a single cooking action and consists of a collection of motion primitives with STRIPS-style preconditions and effects and a STRIPS-style description of the goal[8]. The action plan is represented as a sequential finite state machine of ordered baking primitives corresponding to the inferred state/action sequence.

If no supported baking primitive exists for an action in the state/action sequence a *user interaction* state is created and inserted into the state machine, enabling the system to follow recipes that might involve instructions for which no baking primitives exist. For example, if the system encounters a primitive for chopping walnuts, which it cannot execute itself, it asks its human partner for assistance (via text output and audibly via a text to speech synthesizer).

### 4.4 Plan Execution

Once a sequence of baking primitives has been inferred, the robot executes them sequentially. When executing a baking primitive, the system (1) infers a set of Boolean planning predicates that correspond to the world state, (2) symbolically plans a sequence of motion primitives to execute the baking primitive from the current state, (3) parametrizes the motion primitive sequence into an executable sequential state machine, and (4) executes the state machine. If execution fails the system estimates the current state and replans a new sequence of motion primitives to accomplish the baking primitive. This enables the system to recover from many errors experienced during recipe execution.

#### Detecting and Pouring Ingredients

We utilize the ROS tabletop manipulation pipeline [11] to find the ingredients on the preparation table. First the system finds objects on the table using an algorithm [5] that combines multiple detections across the tabletop into a single state representation of the objects in front of the robot. The system is given a mapping between segmented objects and ingredient names; no object recognition is used to match

the clusters to an object database. This approach provides more robust performance to bowl size variation, and more importantly, is more robust to varying ingredient volumes (and shapes) within the ingredient and mixing bowls.

Once the robot has detected ingredients on the table, it manipulates them using the *pour* baking primitive. The *pour* baking primitive contains the following motion primitives: ingredient bowl grasps, lifting, moving an object over the table with orientation constraints, pouring, shaking, and discarding an empty ingredient bowl. The task space for bowl grasps and pours are discretized into four cardinal bowl orientations at the 12, 3, 6, and 9 o'clock positions around the bowl circumference. This simplified task and motion-plan specification limits bowl positions to a small, intuitive set while providing versatility. If planning or pouring from a bowl grasp pose fails another pose is attempted. After the ingredient is poured, the robot discards the ingredient bowl.

### Mixing and Scraping

The *mix* and *scrape* baking primitives use motion primitives for bowl grasps, moving an object over the table with orientation constraints, pouring, moving the spatula across the table, mixing, and scraping. These primitives use both end effector position and compliance control techniques.

For mixing, the robot performs a grasp of the mixing bowl to hold it in place. It moves the other end effector, with an attached spatula, via end effector position control over the mixing bowl. It then activates the end effector Cartesian force/compliant controller [1], plunging the spatula down into the batter. Next the robot executes a series of linear and circular mixing trajectories to ensure the batter is well-combined. Mixing is performed open-loop: the system does not know how well-mixed the batter is. We attempted to use joint-torques to estimate batter consistency but this was unsuccessful. Other approaches include using computer vision techniques to visually sense the batter consistency. The mixing strategy aims for well-combined batter, at the expense of time and the risk of over-mixing.<sup>2</sup> When mixing is completed the system uses a compliant trajectory to wipe the spatula on the rim of the mixing bowl before switching back to position control to reset the end effector.

To initiate scraping, the robot grasps the mixing bowl and lifts it over the cookie sheet. It pours the contents of the bowl into the cookie sheet, then moves spatula inside of the bowl. Next it executes a compliant scrape trajectory to dislodge any batter that remains stuck inside of the mixing bowl. This trajectory also presses the batter down into the cookie sheet, ensuring one large flat cookie.

### Oven Operation

The *bake* baking primitive uses motion primitives for base navigation, handle recognition, grasping, following endpoint trajectories, and compliant oven opening.

---

<sup>2</sup> Overmixing a flour-based batter can cause baked goods to become tough.

The robot first drives an open-loop path to the oven and servos to its final position based on the detection of the oven handle. It locates the handle using point cloud segmentation in front of the combined surface of the front of the oven and the hanging tablecloth under the oven. Localization on the oven handle eliminates the need for mapping as the odometry of the PR2 base is sufficient to drive from the preparation table to within range to perform oven handle detection. The robot opens the oven by grasping the handle and executing a hybrid force/compliant trajectory. The trajectory follows position waypoints pulling away from the oven while maintaining a constant downward force, opening the oven door with limited *a priori* information about the oven [4].

The robot drives back to the table, grasps the cookie sheet, and returns to the oven. It inserts the cookie sheet into the oven by following a path of end-effector waypoints in the workspace. Finally it closes the oven with the same manipulator by following a joint trajectory that brings the end effector into contact with the open door and pushes the door upwards. After baking is complete, the robot follows a similar process to remove the finished product from the oven.

## 5 Experiments

We evaluate our approach in three ways. First, we performed two experiments using our platform: we passed recipes through our language processing system to create robot instruction sets, and we executed the instruction sets on the physical robot system. The robot's workspace is initialized with a set of objects containing the ingredients and implements necessary to follow the recipe; the system does not parse the ingredients list. Finally, we compared our recipes to human performance by comparing the number of baking primitives supported by our framework to those needed for a set of ten recipes sampled from Joy of Cooking.

We collected a dataset of 60 recipes from the internet, describing how to make simple dishes such as brownies, meat loaf, and peach cobbler. For each recipe, we formally specified the specific ingredients and implements necessary to follow the recipe. This initialization is given to the robot. Next, for each instruction in the recipe text, we annotated the sequence of primitive actions the robot should take in order to follow that instruction. We used 45 recipes from this corpus to train the model, and 15 recipes to test it. A sample recipe from the test set appears in Figure 4, together with the automatically inferred action sequence.

### 5.1 Real-World Demonstration

The robot system operates in a kitchen environment consisting of two work surfaces, one for preparation and another to support a standard toaster oven (preheated to the temperature specified in the instruction set).

We assume that the kitchen is *mise en place*; ingredients are pre-measured and distributed on bowls on the table. Equipment includes four plastic ingredient bowls of various sizes and colors containing premeasured ingredients, a large plastic

Recipe Text	Inferred Action Sequence
Afghan Biscuits	
200g (7 oz) butter	
75g (3 oz) sugar	
175g (6 oz) flour	
25g (1 oz) cocoa powder	
50g cornflakes (or crushed weetbix)	
Soften butter.	<i>pour(butter, bowl); mix(bowl)</i>
Add sugar and beat to a cream.	<i>pour(sugar, bowl); mix(bowl)</i>
Add flour and cocoa.	<i>pour(flour, bowl); pour(cocoa, bowl)</i>
Add cornflakes last.	<i>pour(cornflakes, bowl); mix(bowl)</i>
Put spoonfuls on a greased oven tray.	<i>scrape()</i>
Bake about 15 minutes at 180°C (350°F).	<i>preheat(350); bake(pan, 20)</i>

Fig. 4 Text from a recipe in our dataset, paired with the inferred action sequence for the robot

mixing bowl, and a metallic pie pan. The items are arranged in a grid on the table, with the relative position of every item noted in our ingredient-resolution program.

First, we assessed the robustness of the physical capabilities of the robot by performing extensive tests on a single recipe: “Afghan Biscuits” (which appears in Figure 4). Minor failures, such as the fingers slipping off of the oven door halfway through the opening procedures or the inverse kinematic planner requiring a restart, were corrected during runtime and the tests were allowed to continue. More serious failures, such as spilling an ingredient or scraping the contents of the mixing bowl onto the floor or table, that required the system to be fully restarted or a new piece of code to be written caused the termination of the test. We tested the full end-to-end system, from robot instructions through finished dish, 27 times on this recipe, of which 16 ran to completion, with an average runtime of 142 minutes from start to finish. On average, mixing ingredients in the bowl takes 27 minutes to execute. Adding an ingredient takes on average 8 minutes; scraping from the bowl onto a cookie sheet takes 16 minutes, and putting an item into the oven takes roughly 18 minutes. The robot is able to successfully execute actions in a cooking domain, although much more slowly and with many more failures than a human. A pictorial representation of the baking process is shown in Figure 5.

Next, we selected two recipes from our test set and used our recipe following system to execute them from plain-text recipes through the finished dish. We executed two runs each of “Afghan Biscuits” and “Quick’N Easy Sugar Cookies,” demonstrating the end-to-end system integration and the robot’s ability to physically execute several recipes.

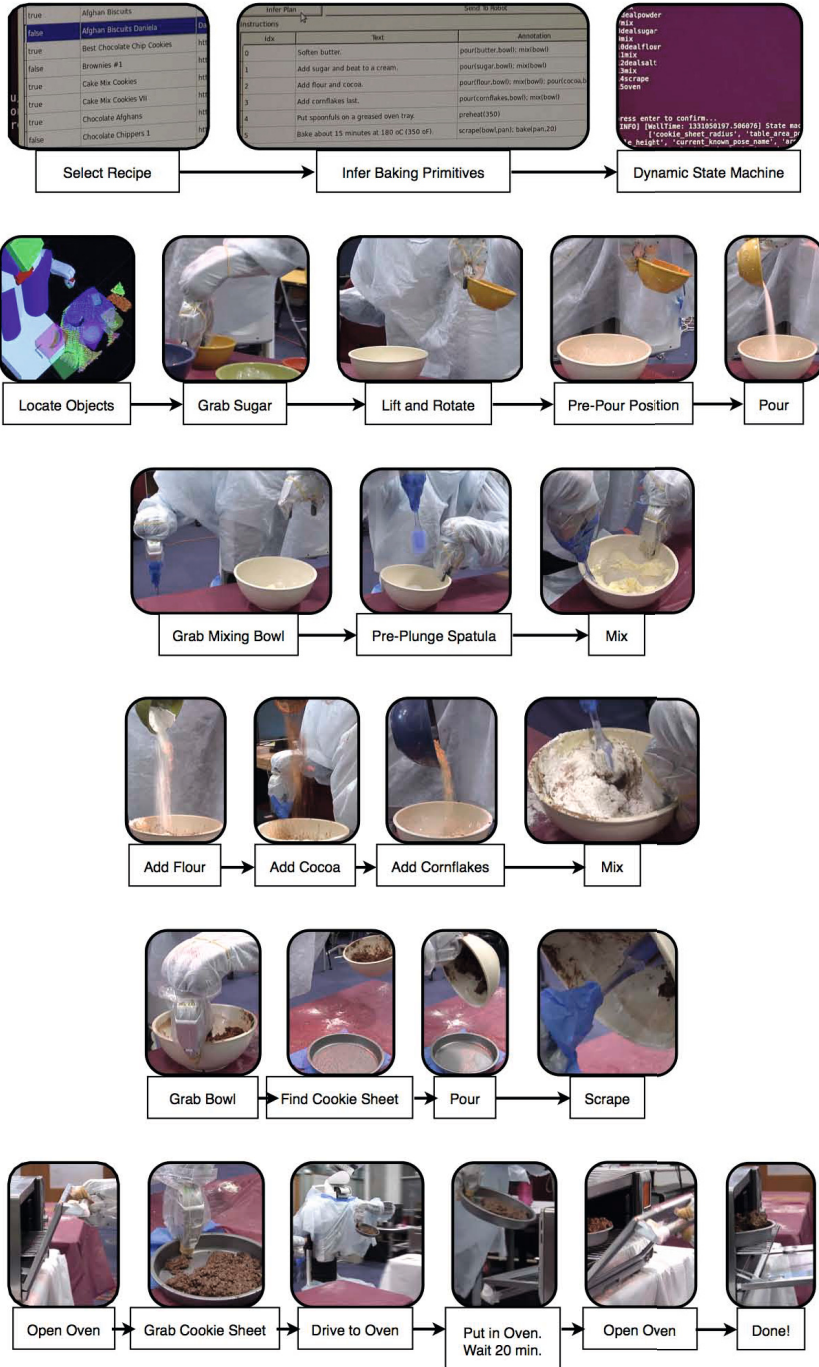


Fig. 5 A pictorial timeline of the baking process

### 5.2 Quantitative Evaluation of Recipe Processing

We performed a quantitative evaluation in simulation to assess the system’s ability at finding a plan to understand a variety of recipes. First, we assessed the system’s performance at inferring an action sequence for specific sentences in the recipes. The 15 recipes in the test set contained a total of 92 individual sentences. Of these sentences, the system correctly inferred exactly correct plans for 49% of them. Many of the errors consist of omitting an ingredient, as in “Mix with hands – form small balls and flatten on ungreased pan.” which referred to all ingredients in the recipe; for this command the system inferred only *pour(sugar, bowl); mix(bowl)* but excluded other ingredients such as butter and flour.

Next, we evaluated the system’s performance when inferring the complete action sequence for the recipes. The system correctly inferred action sequences for 26.67% of the test set (4/15). As in the instruction-level tests, most of the errors consist of omitting an ingredient for a single instruction, resulting in the rest of the action sequence becoming incorrect.

Finally, we assessed the performance of instruction-level and end-to-end inference with respect to changing training corpus size in Figure 6 and Figure 7. We kept the recipe test set constant through these tests. We started with a training set of one random recipe, trained the system with that training set, and evaluated the performance of the system using the test set. At each step, we added 5 random recipes to the training set. The results for the instruction-level evaluations are shown in Figure 6 and the results of the end-to-end evaluations are shown in Figure 7. These graphs reflect the asymptotic nature of the training set size; we see a large initial improvement in inference performance when we add more recipes to the training set, but the improvement diminishes with each addition step.

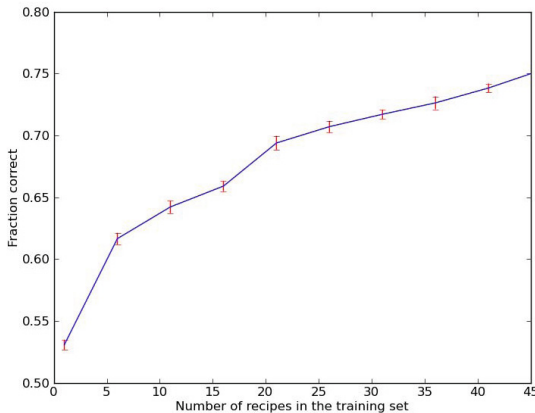


Fig. 6 Instruction-level recipe inference performance with varying training set size

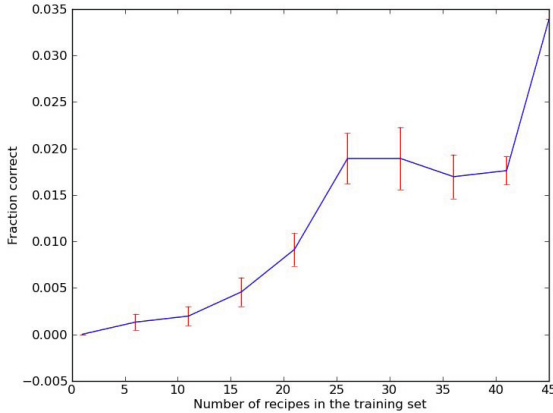


Fig. 7 End-to-end recipe inference performance with varying training set size

### 5.3 Comparison to Human Performance

Finally, we assess the types of actions supported by the system compared to a human chef. First, we compared our supported primitives to a brownie recipe performed by a human, annotated within the CMU Multimodal Activity Database [7]. We found that our *mise en place* assumption eliminated 67% of the annotated actions. Of the remaining actions performed by the human, our instruction set supports 79%. The two unsupported types of actions were greasing a cookie sheet and cracking eggs. Our compliant controller could be used to implement a greasing technique (using Crisco and a paper-towel, for example). However, cracking eggs is unsupported because of limitations in the PR2’s manipulation and perception capabilities.

Next, we compared our instruction set to a set of ten recipes sampled from Joy of Cooking [13], ranging from corn dogs to salsa fresca.<sup>3</sup> The system is able to execute 67% of the 120 instructions from these recipes. Adding motion primitives to support waiting for specific events, such as boiling or tenderness, as well as stove operations would increase the fraction of supported instructions considerably. It is worth noting that while the BakeBot system supports a large subset of the recipe instructions, it lacks the dexterity and perception to perform more complicated actions. Any aspiring chef can attest that simply going through the motions of a recipe is insufficient to guarantee tasty food. For example, when making scones, an instruction like “Cut in [cold butter]... until the size of small peas” is challenging even for human chefs to execute so that the scones have their distinctive flakiness.

We compare these figures to our corpus of simple baking recipes, of which 76% were supported by the BakeBot system. Of the 101 unsupported sentences, 24 were

<sup>3</sup> Recipes from Joy of Cooking: corn dogs, tart green salad, eggs in a basket, fettuccini with butter and cheese, Israeli couscous pilaf, carrot cake, classic scones, salsa fresca, Belgian beef stew, and roasted whole salmon.



waiting for the cookies/brownies to cool before removing from the pan, 17 were associated with greasing the cookie sheet, and 8 were simply descriptive statements or friendly notes (i.e. “Do not overbake” and “Enjoy!”).

While performance can be increased by adding more manipulation primitives, we believe that improved system exteroception would yield the most significant recipe execution benefits. Many recipes in the Joy of Cooking set and within our baking recipe corpus require feedback to determine whether or not a step of the recipe is completed. For example, a recipe may require onions to be sauted until translucent. First order implementations to sense this onion state may use vision systems to identify color transformations. More capable systems may use thermal imaging to sense the temperature of the onions and correlate this temperature to the desired activation temperature of the chemical reaction associated with onion translucence. We see the long-term potential for robotic cooking systems to leapfrog human chefs through the integration of advanced sensing and feedback systems.

## 6 Main Experimental Insights

In this paper we presented BakeBot, a robotic chef that is capable of following textual recipes in a kitchen environment. We demonstrated that the robot is able to successfully follow two different recipes in the real world. In addition, we performed a quantitative evaluation of the system’s performance at following instructions from a larger set of recipes in simulation. Our approach represents first steps towards a robotic chef capable of executing arbitrary recipes in the home environment.

Executing the recipes on the robot shows a functioning end-to-end system that successfully integrates language processing, object recognition, task planning, and manipulation. Our set of motion primitives enable a variety of simple baking recipes to be executed on the robot, and we demonstrated two successfully experimentally. The compartmentalization of motion planning and execution into primitives enables the system to support a wide array of recipes while minimizing software development and debugging effort.

The natural language component of the system shows promising performance at understanding individual instructions; it inferred exactly correct actions for many instructions, and many of the mistakes were due to missing one or two ingredients from aggregate phrases such as “Sift together the dry ingredients.” However, the robot’s physical and perceptual capabilities limited the scope of the recipe processing. For example, the system ignores a warning such as “Do not brown; do not over bake” because detecting the color of overcooked cookies through an oven door is an unsolved perceptual problem. BakeBot is a demonstration of the challenges faced when integrating large robotic control systems. The overall robustness of the BakeBot is limited by the inherent complexity of a system executing such a difficult task; the top-level cooking system is vulnerable to faults in its software substrate. The difficulty of maintaining a system built upon a constantly updated software base reflects a need for improved software management for robotic systems.



The key limitation of the system is its lack of robustness. Failure in any of the robot's systems leads to a failure to successfully follow the recipe: in the natural language system, the perceptual system, or the control system. Actions in the real-world demonstration often failed to execute successfully because of inability to adapt to unexpected situations in the environment. For example, flour being piled too high in a bowl caused the grasp planner to aim for the flour rather than the edge of the bowl. For another example, the tabletop manipulation package occasionally chose to grasp the ball of batter in the center of the mixing bowl, rather than the rim of the bowl, because the batter presented a more salient grasp feature than the rim of the bowl. The large number of environmental interactions and sequential baking primitives to complete the baking task make BakeBot particularly sensitive to even low probabilities of subsystem failure.

Our experimental performance was demonstrative of the issues faced in the construction of large robotic systems. The performance of complex tasks such as cooking require the interaction of many systems and subsystems. This iteration can often be error-prone and difficult to fully characterize or debug, oftentimes leaving the top-level system vulnerable to faults in its software substrate.

We have focused thus far on creating a functioning end-to-end system for semi-structured home environments. System robustness will improve as its underlying manipulation software matures and as better perception of the environment increases the system's ability to deal with uncertainty and correct its own mistakes. We plan to expand the repertoire of the system by designing additional motion primitives to enable actions such as chopping, blending, and frying, introducing new challenges for manipulation, perception, and natural language understanding.

## References

- [1] Barry, J., Bollini, M., Liu, H.: End effector Cartesian controller (2012), [http://www.ros.org/wiki/ee\\_cart\\_imped](http://www.ros.org/wiki/ee_cart_imped)
- [2] Beetz, M., Klank, U., Kresse, I., Maldonado, A., Mösenlechner, L., Pangercic, D., Rühr, T., Tenorth, M.: Robotic roommates making pancakes. In: IEEE-RAS International Conference on Humanoid Robots (2011)
- [3] Berger, A.L., Della Pietra, S.A., Della Pietra, V.J.: A maximum entropy approach to natural language processing. *Computational Linguistics* 22, 39–71 (1996)
- [4] Bollini, M.: Following recipes with a cooking robot. Master's thesis, MIT (2012)
- [5] Bollini, M., Barry, J., Rus, D.: Bakebot: Baking cookies with the PR2. In: International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal (October 2012) (in submission)
- [6] Chen, D.L., Mooney, R.J.: Learning to interpret natural language navigation instructions from observations. In: Proc. AAAI (2011)
- [7] De la Torre Frade, F., Hodgins, J.K., Bargteil, A.W., Artal, X.M., Macey, J.C., Castells, A.C.I., Beltran, J.: Guide to the Carnegie Mellon University multimodal activity (CMU-MMAC) database. Technical Report CMU-RI-TR-08-22, Robotics Institute, Pittsburgh, PA (April 2008)
- [8] Hoffmann, J., Nebel, B.: The FF planning system: Fast plan generation through heuristic search 14, 253–302 (2001)

- [9] Kollar, T., Tellex, S., Roy, D., Roy, N.: Toward understanding natural language directions. In: Proc. ACM/IEEE Int'l. Conf. on Human-Robot Interaction (HRI), pp. 259–266 (2010)
- [10] Matuszek, C., Fox, D., Koscher, K.: Following directions using statistical machine translation. In: Proc. ACM/IEEE Int'l. Conf. on Human-Robot Interaction (HRI), pp. 251–258 (2010)
- [11] Rusu, R.B., Sukan, I.A., Gerkey, B., Chitta, S., Beetz, M., Kavraki, L.E.: Real-time perception-guided motion planning for a personal robot. In: Intelligent Robots and Systems, pp. 4245–4252. IEEE (2009)
- [12] Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S., Roy, N.: Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine* 32(4), 64–76 (2011)
- [13] von Starkloff Rombauer, I., Becker, M.R., Becker, E., Guarnaschelli, M.: *Joy of Cooking*. Scribner Book Company (1997)

# Part VIII: ISER Session Summary on “Manipulation”

Evan Drumwright

George Washington University

## Session Summary

The session on manipulation concurrently illustrates how far the hardware for robotic manipulation has come and—through the sophisticated strategies employed to achieve robustness—how further improvements would be welcome. Each paper in this session addresses different aspects of manipulation: precision and power grasping, manipulating bulky objects and non-rigid bodies, and non-prehensile manipulation. The research described in these works share techniques (e.g., manipulation primitives, motion planning), hardware (the Willow Garage PR2 robot), or both. ‘*Load Equalization on a Two-Armed Robot via Proprioceptive Sensing*’ by Leontie et al. introduces the problem of equalizing loads from bulky objects, for which force and form closure may not be possible, on the two arms of a robot. The paper demonstrates robust application of the strategy to multiple bulky objects, including non-rigid objects. ‘*Mapping Grasps from the Human Hand to the DEXMART Hand by Means of Postural Synergies and Vision*’ uses a mapping from human hand motions to the DEXMART robotic hand toward improving grasp planning and control. The authors, Ficuciello et al., apply the Microsoft Kinect vision system to demonstrations of human subjects grasping, and then map these performances to produce power and precision grasps with the DEXMART hand. ‘*Manipulation with Multiple Action Types*’, by Barry et al., introduces a randomized sampling algorithm, *DARRT*, for planning with multiple types of possibly non-prehensile manipulation actions. They apply their method toward manipulating plates from an initial configuration to a goal configuration using manipulation primitives. Finally Lakshmanan et al.’s paper, ‘*A Constraint-Aware Motion Planning Algorithm for Robotic Folding of Clothes*’, presents a motion planning algorithm for the problem of manipulating clothing via manipulation primitives. The use of primitives helps avoid the problem of infeasible plans that afflicted the same research group’s initial foray into robotic folding.

# Load Equalization on a Two-Armed Robot via Proprioceptive Sensing

Roxana Leontie, Evan Drumwright, Dylan A. Shell, and Rahul Simha

**Abstract.** As humans we use our arms and bodies in addition to our hands to grasp objects. We (and robots) often cannot use caging or closure strategies when manipulating bulky objects. This paper studies manipulating such objects in the context of a particular task: equalizing a load across the arms of a two-armed robot. Our PR2 robot performs this task using only proprioceptive force sensing and a simple, reactive equalization strategy. We demonstrate the robot robustly performing this task using numerous and various objects (*e.g.*, boxes, pipes, broomsticks, backpacks).

## 1 Introduction

Recent research in robotic manipulation has delivered mobile robots with advanced autonomous capabilities (*e.g.*, [1, 2, 3]). The robot typically manipulates via a gripper or hand, though manipulation using the whole arm and even the body—thereby providing enhanced manipulation capabilities (*e.g.*, grasping when the hands are already loaded)—has been studied [4, 5, 6, 7, 8].

We have been investigating applications of arm and body manipulation for which *caging* and *closure* are difficult or impossible (as in the figure above), such as manipulating bulky objects, manipulating multiple objects simultaneously (*e.g.*, a stack of boxes, a bundle of ski equipment), or manipulating articulated objects (*e.g.*, a squirming child). In applications such as these, the manipulated object or



---

Roxana Leontie · Evan Drumwright · Rahul Simha  
The George Washington University, Washington, DC  
e-mail: {roxana, drum, simha}@gwu.edu

Dylan A. Shell  
Texas A&M University, College Station, TX  
e-mail: dshell@cs.tamu.edu

objects are restrained from moving—sometimes only unilaterally—in some directions while being free to move in others.

## 2 Background and Related Work

We introduce the following subproblem of caging-free and closure-free arm/body manipulation: *given a two-armed robot manipulating an uneven load with its arms, equalize the load such that the arms maintain the load with nearly identical torques.* Here we define a *load* as one or more objects.

Research has established that a load is statically balanced if the projection of the load’s center of mass (COM) lies within the polygon of support (defined for our problem as the contact manifold between the load and the robot’s arms). It should be clear to the reader that load equalization trivially yields static balancing. We also note that load equalization bestows several benefits apart from stabilizing the load dynamically, including reducing energy consumption by the actuators (if such consumption is approximated using the integral of squared torque, as customary), avoiding lateral postural shifts in humanoids [9], and increasing dynamic stability of the robot [10].

Similar in spirit to our problem of bimanual load equalization is the “devil sticking” task investigated by Schaal and Atkeson [11, 12, 13]. That research also sought to achieve dynamic stability of the object (the “baton”), though by using dynamic movements of the robot rather than precise manipulation. We view both approaches as effective strategies for manipulation without caging or closure; our approach (the latter) is more applicable to larger loads and more powerful on slower robots.

Our work is the first (of which we are aware) to use proprioceptive force sensing in a Robotics application, though our approach—which uses PID control state to sense load on our robot’s arms—is identical to that described by Wolpert and Kawato [14] (they also use the error between a motor controller’s prediction and the robot’s state).

## 3 Technical Approach

Our experimental platform is the Willow Garage PR2. The dynamics equations for this multibody system would nominally take the form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}(\mathbf{q})^T \mathbf{f}_{load} \quad (1)$$

where  $\mathbf{q}$  are the generalized coordinates of the system,  $\mathbf{M}(\mathbf{q})$  is the generalized inertia matrix,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  is the matrix of fictitious forces,  $\mathbf{G}(\mathbf{q})$  is the vector of gravitational forces,  $\boldsymbol{\tau}$  is torque applied by the arms to counteract the load,  $\mathbf{J}(\mathbf{q})$  is a Jacobian matrix and  $\mathbf{f}_{load}$  is a vector of spatial forces (corresponding to the load) acting on one or more of the robot’s links. The PR2 employs counter-balancing

springs (as a safety mechanism) that nearly eliminate gravitational acceleration of the arms; thus,  $\mathbf{G}(\mathbf{q})$  is approximately zero. Assuming that the robot moves sufficiently slowly (*i.e.*,  $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \approx \mathbf{0}$ ) the torques applied by the actuators effectively equal the generalized forces from the load; thus, we can study load equalization with only the PR2's PID control state, albeit at some loss of modeling accuracy (a truly accurate dynamics model of the PR2 does not currently exist, to our knowledge). Thus, we use only proprioceptive sensing (*i.e.*, no vision, LIDAR, *etc.*) to equalize the load.

We formulated our fully reactive (*i.e.*, no planning or intensive computation is employed) equalization strategies from anecdotal experience. For example, humans may move their arms outward to redistribute a load; we tried this simple strategy first. Unfortunately, that strategy failed to account for frictional forces acting between the object and the robot's arms: Coulomb's friction model predicts greater frictional forces will be applied to the arm under greater load. Accordingly, the object (undesirably) moved along with that arm in our experiments, thus failing to shift the COM as desired. We developed two new strategies to address this difficulty: the *Tilting* strategy (Section 3.1) and the *Lock-Move-Release* strategy (Section 3.2).

### 3.1 Tilting Strategy

The *Tilting* strategy, depicted in Figure 1, tilts the object by raising one arm and lowering the other until sufficient load has been shifted such that the object will slide along the arm to be moved outward. This strategy is predicated on the object possessing highly similar frictional properties at the points of contact with the robot's arms. Additionally, frictional forces must be sufficiently large such that the object does not slide off the robot's arms when tilted.

The strategy begins by tilting the arms until the imbalance is swapped: if we initially detect  $x$  units of force on arm A and  $y$  units of force on arm B, where  $x - y = \Delta > 0$ , we move arm A upward and B downward until the force on B becomes  $\epsilon > 0$  units greater than the force on A. Arm A is then moved outward until  $\Delta$  load has been transferred, though the object may need to be tilted again during this process to keep the load on B greater than on A. We note that this strategy is sensitive to the frictional properties of manipulated objects: the strategy is not only predicated on nearly equal frictional characteristics at the contact points, but also depends on sufficient frictional effects to keep the manipulated object from sliding off the robot's arms when tilted.

We found this strategy worked poorly. Manipulated objects would slide off of the arms before sufficient weight had shifted; alternatively, joint limits would frequently prevent the object from being tilted enough for sufficient load redistribution. An successful series of steps showing the robot performing this strategy can be seen in Figure 2.

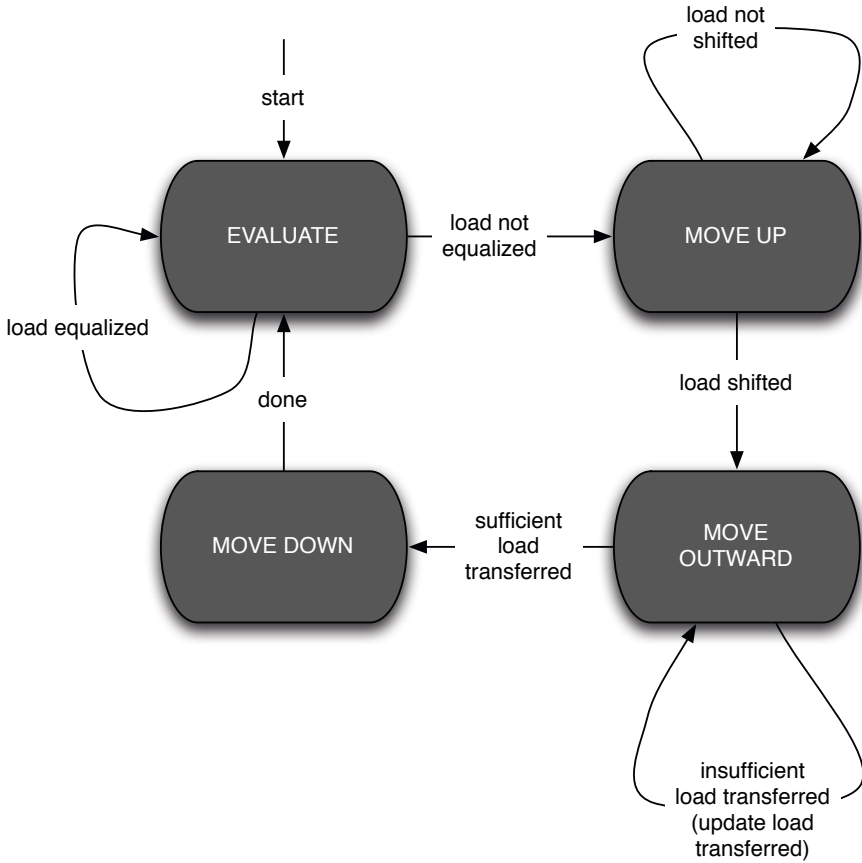


Fig. 1 State transition diagram for the *Tilting* strategy

Our failure to get this strategy working reliably led to the creation of the second strategy, described below.

### 3.2 Lock-Move-Release Strategy

In the *Lock-Move-Release* strategy, the robot first grasps the object between its upper arm and forearm (*Lock*). The other arm then moves outward (*Move*), and finally the object is released from the grasp (*Release*). This process repeats until the load is equalized.

In the **EVALUATE** state, the robot analyzes the proprioceptive feedback from the arms (obtained from their PID controllers) to evaluate whether a load (if any) is equalized. To account for settling time (*i.e.*, the time it takes for the system to converge to a steady state), we apply a simple filtering strategy: the controller cannot

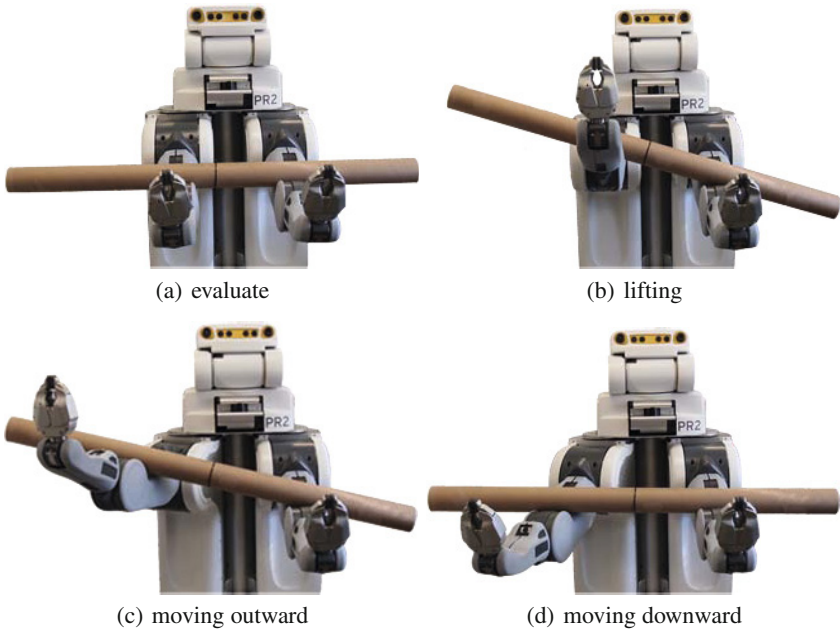


Fig. 2 A series of photos depicting the robot performing the *Tilting* strategy

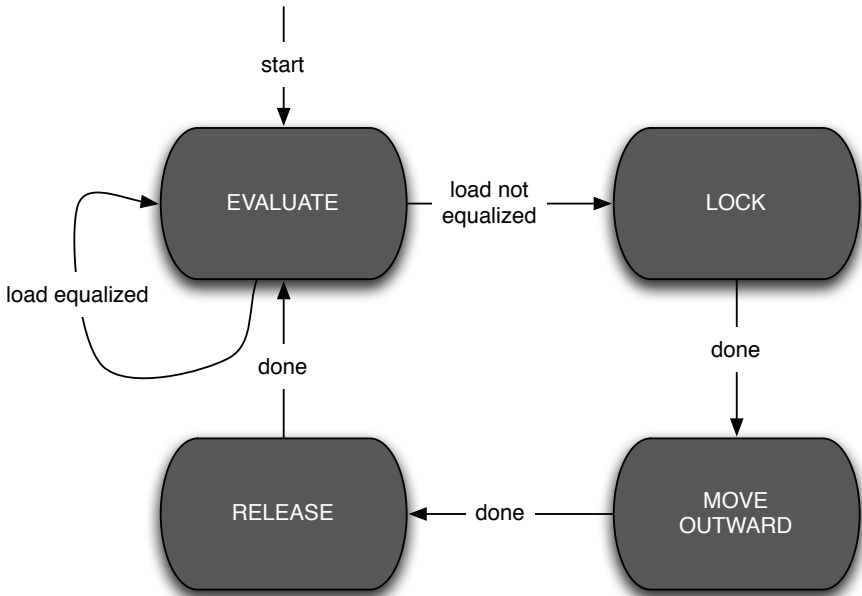


Fig. 3 State transition diagram for the *Lock-Move-Release* strategy



leave the **EVALUATE** state until the sensed change in load on the two arms is smaller than 0.1 units<sup>1</sup> for at least 100ms. If—after the robot is determined to be in a steady state—the sensed difference in load on the two arms is 1.0 units or greater, the controller transitions to the **LOCK** state.

In the **LOCK** state, the robot immobilizes the object on the arm with lesser load by grasping it between the upper arm and the forearm: we assume that the object is placed approximately on the elbow joints. By restraining the object in this manner, we can move the other arm outward without the object also moving. The grasping movement takes place over two seconds; although the movement can generally be performed more quickly, we find that faster movements often reduce the accuracy of the proprioceptive feedback (due to the need for the system to settle).

The **LOCK** state transitions immediately to the **MOVE OUTWARD** state. This state predetermines the distance to move the arm outward toward minimizing repeated adjustments (*i.e.*, repetitions of holding, moving, and releasing). We compute this distance using several steps. First we employ forward kinematics to estimate the location of points of contact between the arms and object. Next we compute the object's initial COM (*i.e.*, its COM before immobilization) as depicted in Figure 4. Given the contact points  $s$  and  $p$  and the loads  $x$  and  $y$  on the arms in the **EVALUATE** state, we determine the position of the COM as follows. Noting that the COM is at  $p$  if  $x = 0$  and the COM is at  $s$  if  $y = 0$ , then the COM is at  $\frac{s+p}{2}$  if  $x = y$ . By extension, for arbitrary  $x$  and  $y$ , the COM is located at:

$$\bar{x} = \frac{xs + yp}{x + y} \quad (2)$$

Using the current location of the object's COM, our final step employs inverse kinematics to estimate the distance to move the arm outward to equalize the load (this calculation will assume rigidity of the object, but our experience indicates that this assumption works well in practice even for deformable objects) using the distances between the contact points and COM ( $d_1$  and  $d_2$  in Figure 5). The distance to move outward is calculated as:

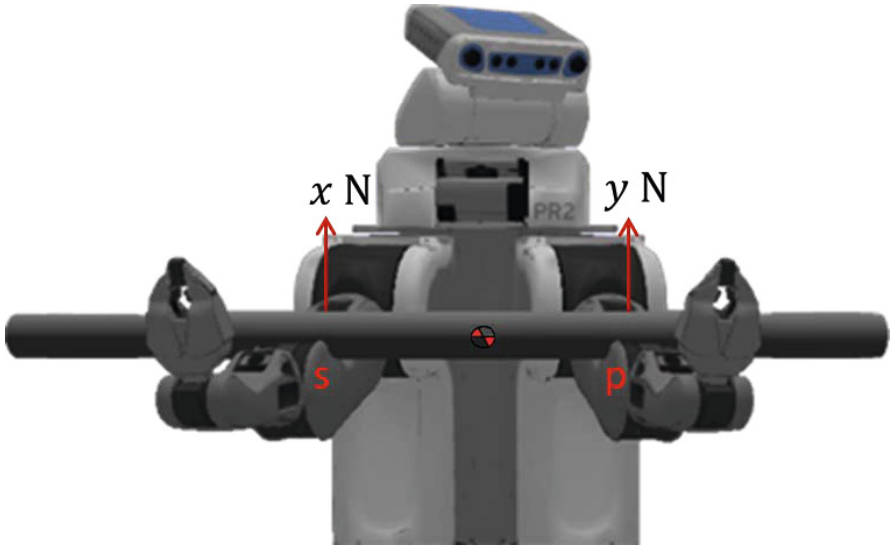
$$d = |d_2 - d_1| \quad (3)$$

The robot then executes the outward movement at a constant speed (*i.e.*, dependent on distance moved) to avoid excessive settling time, as described above.

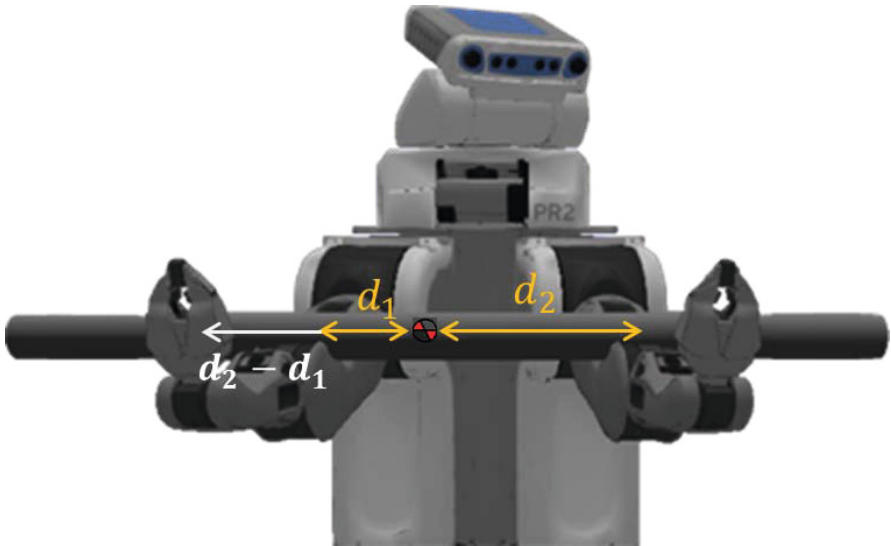
Finally, the robot releases the object from its grasp in the **RELEASE** state, and the forearm returns to its original position. This slow movement occurs over 3.5 seconds, again to avoid excessive settling time. An successful series of steps showing the robot performing this strategy can be seen in Figure 6.

---

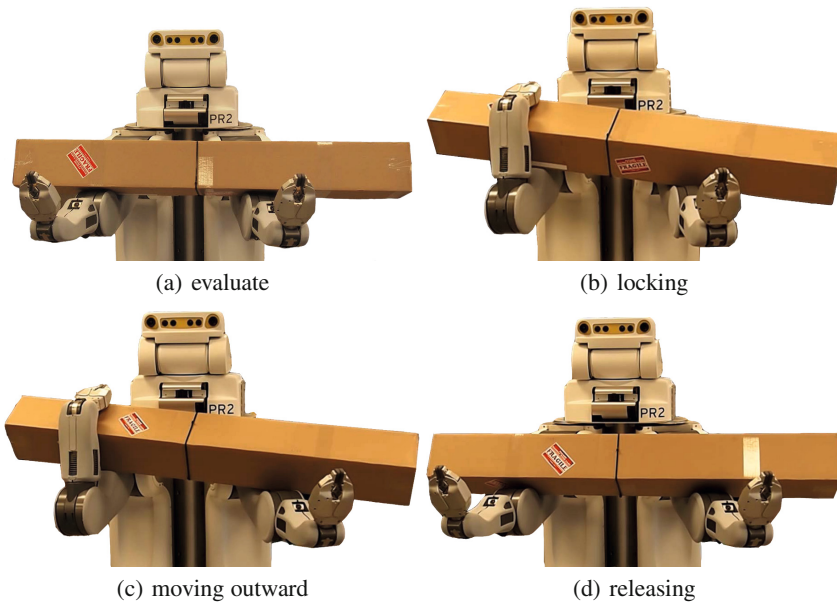
<sup>1</sup> According to Willow Garage, the force units are not calibrated to SI or any other measurement system.



**Fig. 4** Knowing (or estimating) the points of contact points ( $s, p$ ) and forces applied to the arms ( $xN$  and  $yN$ ) allows us to estimate the center-of-mass of manipulated objects (Equation 2)



**Fig. 5** Computing the distances between the contact points and the estimated center-of-mass allows us to determine how far outward we should move the arm with greater applied load (Equation 3)



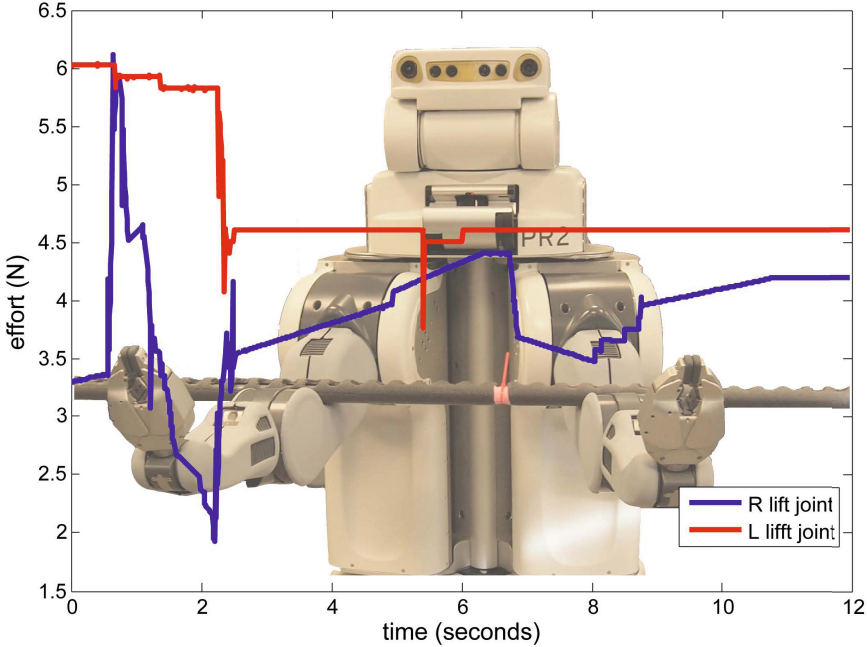
**Fig. 6** A series of photos depicting the robot performing *Lock-Move-Release* strategy

## 4 Experiments and Results

This section discusses a number of experiments conducted using the *Lock-Move-Release* strategy.

### 4.1 *Balancing a Load with COM Projection within the Support Polygon*

One of our first experiments with the *Lock-Move-Release* strategy tested whether the robot could equalize loads with centers-of-mass located non-equidistantly between the robot's arms yet with projections remaining within the support polygons. The transparent photo of the robot in Figure 7 shows the manipulated object used in the experiments in Sections 4.1–4.3: a 2.7m, 1.4kg iron pipe. That figure also depicts the object's geometric center (marked in pink near the robot's left arm); the geometric center is located near the pipe's COM. The data plotted in Figure 7 correspond to immobilizing the object with the robot's right arm (0–2.5) seconds), shifting the load (2.5–6 seconds), and releasing the object (6–9.5 seconds); the plot shows that the actuator force decreases when the object is grasped and increases on release. We have yet to observe a failure using the *Lock-Move-Release* strategy over numerous experimental trials with variable initial condition (location of the pipe's geometric center within the support polygon).



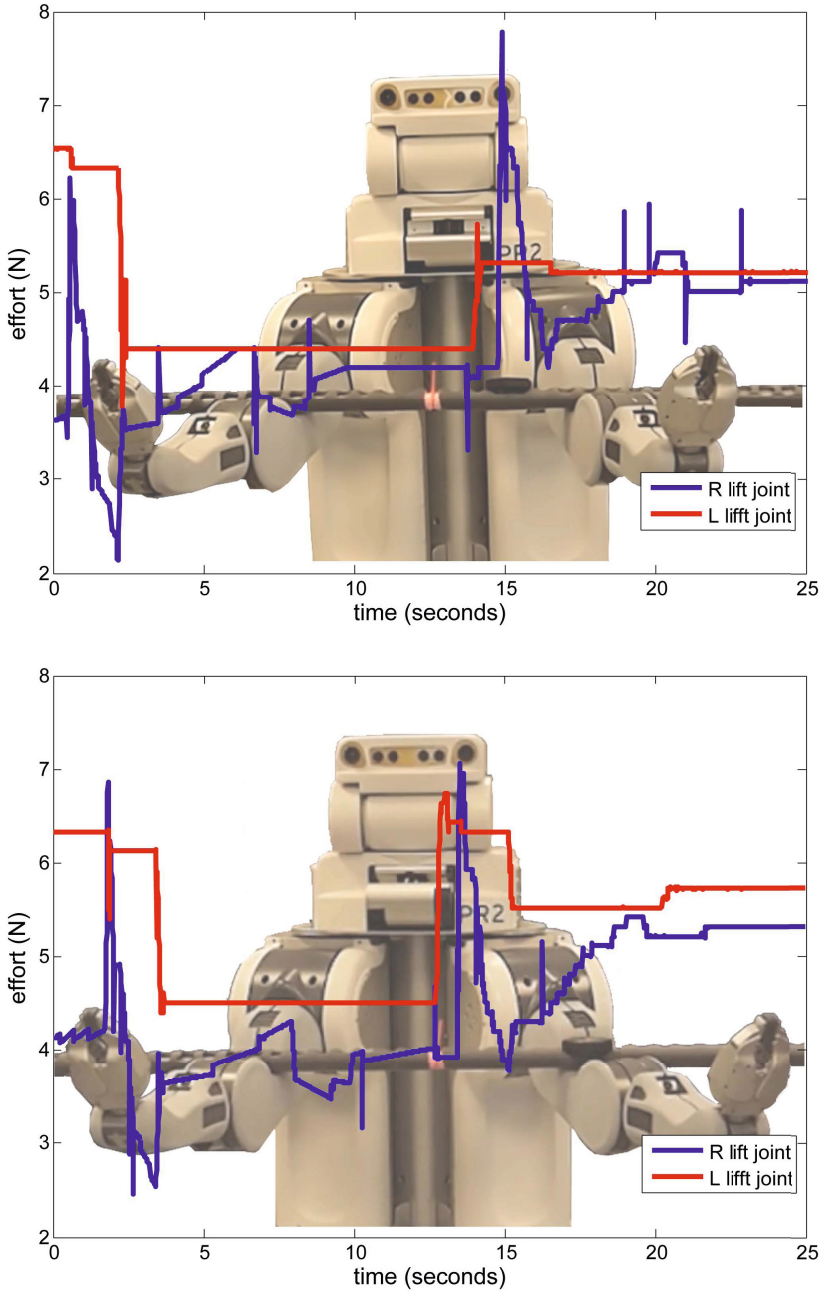
**Fig. 7** Plot of the “shoulder lift” motor torques during the load equalization process corresponding to a trial of the first experiment: as seen in the photo, the geometric center of the pipe (marked in pink near the robot’s left arm) lies within the object’s support polygon.

### 4.2 Equalizing Time Varying Loads

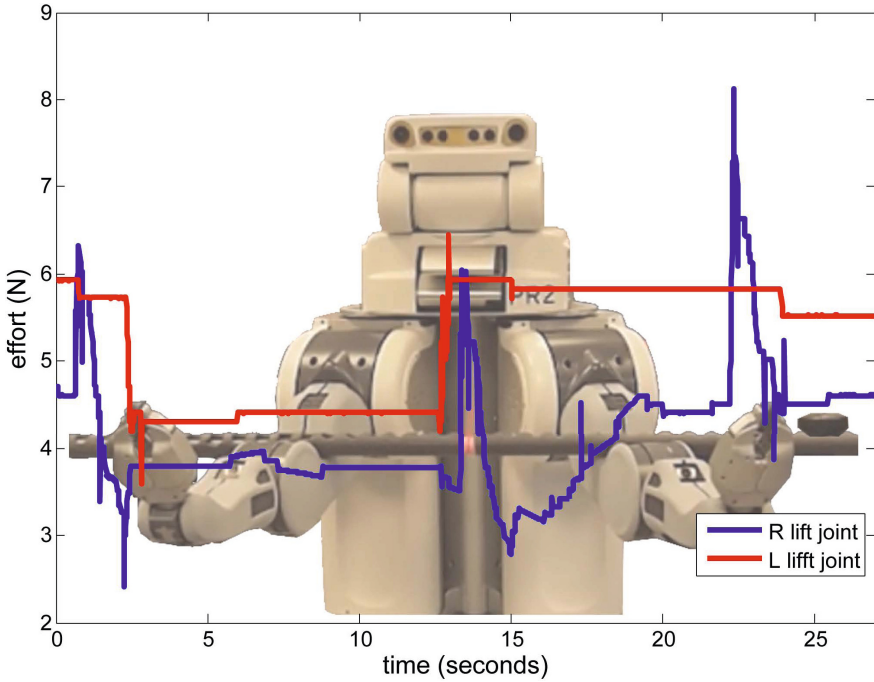
A second series of experiments tested the ability of our strategy to handle time varying loads (as if objects were continually added to the robot). The first ten seconds of the plots in Figure 8 correspond to the robot equalizing loads from only the iron pipe (identically to the conditions described in the previous section). After each load was equalized, we added an additional load (0.6–0.85kg) to the pipe at a point with projection near or at the inside edge of the support polygon (depicted by the photos in Figure 8). The plots at approximately fifteen seconds correspond to the robot beginning to re-equalize the loads. The re-equalization process requires fewer than ten seconds to complete.

### 4.3 Equalizing Time Varying Loads Added Outside the Support Polygon

A third series of experiments (depicted in Figure 9) was designed to stress the ability of our strategy to handle loads near the margin of dynamic instability. The experimental setup for these trials was identical to those described above but with the



**Fig. 8** Plots of the “shoulder lift” motor torques during the load equalization process corresponding to two trials of the second experiment: as shown in the photos, magnetic weights have been added to the iron pipe near (top) and directly above (bottom) the robot’s left elbow to effect a change to the load

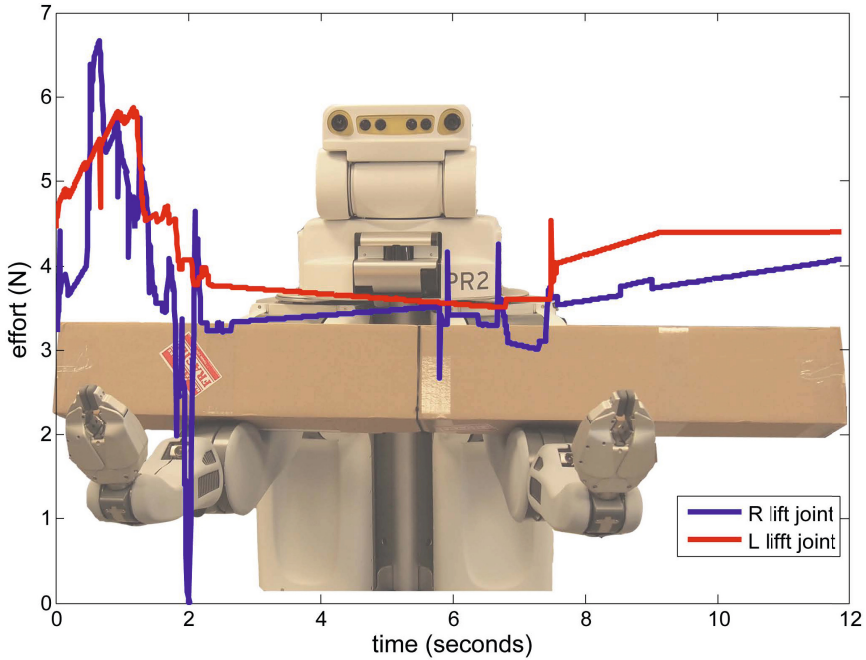


**Fig. 9** Plot of the “shoulder lift” motor torques during the load equalization process corresponding to a trial of the third series of experiments: as seen in the transparent photo (near the right elbow), a magnetic weight has been added well outside the support polygon of the iron pipe to simulate a modification to an equalized load

additional load (a 0.6kg weight) placed well outside of the support polygon. We tested various positions of the weight—including at the end of the 2.7m iron bar—and still observed complete success of our strategy. The equalization process for the added load generally required two iterations (this was the case with the trial depicted in Figure 9): the first iteration (13–20 seconds) nearly equalized the load, while the second iteration yielded full equalization. As with the plots in Figure 8, the plot in Figure 9 shows that both the detection and equalization of the added load was conducted quickly.

#### 4.4 Equalizing a Load from a Box

We also experimented with the *Lock-Move-Release* strategy on objects of different shapes, such as boxes (one of which is depicted in Figure 10). For the data plotted in Figure 10, the box’s position shifted during the lock and release phases of the strategy, thereby introducing significant noise into the process (as evident in the plot). Nevertheless, no failures were encountered; Figure 10 corresponds to an initially



**Fig. 10** Plot of the “shoulder lift” motor torques during the load equalization process corresponding to the robot equalizing a load from a cardboard box

unequal load from a box that is equalized in fewer than ten seconds using only a single iteration of our strategy.

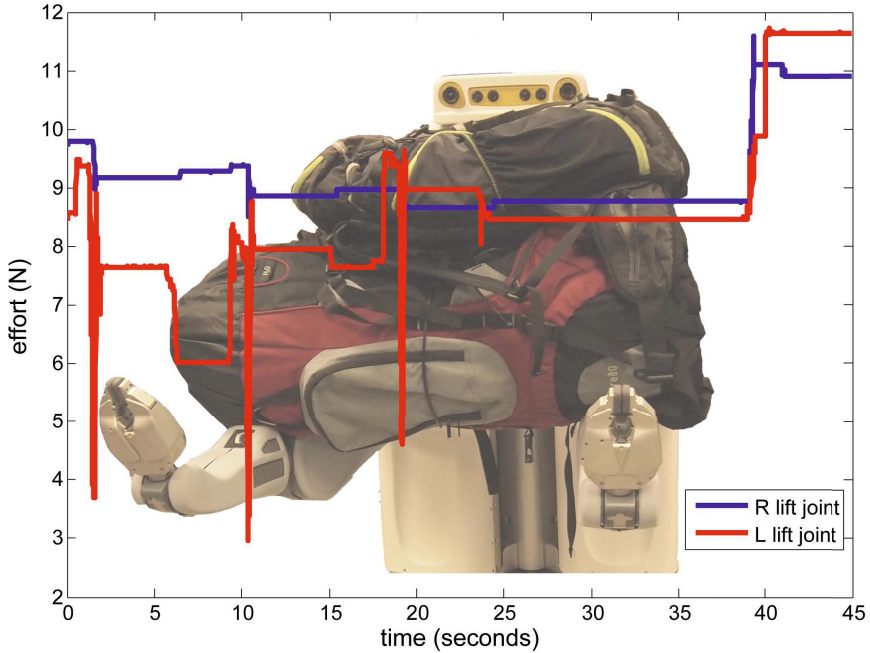
#### 4.5 Load Equalization with Non-rigid Objects

We conducted a final series of experiments on bulky non-rigid objects (camping backpacks with frames removed; see Figure 11) with non-uniformly distributed loads. These experiments entailed having the robot equalize one backpack placed on the robot’s arms followed by a second backpack placed on top of the first one. Again, no failures were encountered, though the equalization process required three iterations.

#### 4.6 Miscellaneous Testing

We also performed a number of tests of our strategy using very light objects (0.25 kg)—though one would expect that load equalization would not be in great need for such objects on current robots with bimanual manipulation capabilities. Again, we never observed a failure of the strategy.





**Fig. 11** Plot of the “shoulder lift” motor torques during the load equalization process conducted on non-rigid bodies (camping backpacks with frames removed and non-uniformly distributed loads)

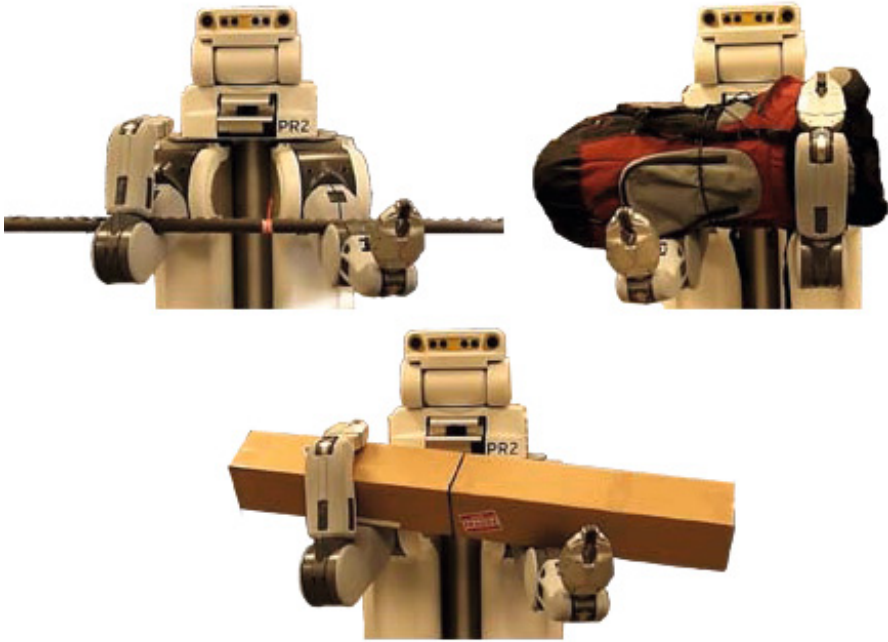
## 5 Main Experimental Insights

This section discusses distinct insights gleaned from the development of the *Lock-Move-Release* strategy and the performance of the robotic platform during the experiments with respect to both primary (*i.e.*, load equalization) and ancillary (*i.e.*, grasping) tasks.

### 5.1 Ability to Equalize Loads

The PR2 can very robustly equalize loads applied by various objects using a simple reactive strategy (this is likely explains much of its robustness), minimal sensory data, few assumptions on object geometry or mass distribution, and little modeling. More sophisticated strategies might yield faster equalization, though we sense that we are near the PR2’s performance envelope: dynamically equalizing shifting loads does not seem possible with this robot. Due to the minimal assumptions used in our work and the simplicity of our strategy, we predict that our approach will transfer to other manipulation platforms without difficulty.





**Fig. 12** Sample grasps of various objects manipulated by the robot’s arms during the course of our experiments

### ***5.2 Insights from Use of Purely Proprioceptive Sensory Data***

The PR2 is slightly better able to detect uneven loading for heavier objects than for lighter ones; it can be challenging to distinguish an uneven load from noise (hence our unit threshold for triggering the equalization strategy). This finding corresponds with an informal experiment conducted by the authors: each of us was able to accurately compare masses for only relatively heavy objects (greater than several kilograms). Thus, while proprioceptive sensory data is likely sufficient for heavier objects, lighter objects may require the addition of visual or tactile sensory data.

### ***5.3 Insights from Arm Grasping***

The PR2 seems quite capable of grasping objects placed onto various points on its arms. We tested assorted objects, including planks, broomsticks, pipes, boxes, and backpacks (among others), during the course of our experiments. The PR2 never failed to grasp an object with its arms and seemed fairly insensitive to the positioning of the objects. We would be curious to see a study gauging effectiveness of grasping with the arms against grasping with the grippers (at least, but not limited to, the PR2). Sample arm grasps realized during performances of our strategy are depicted in Figure 12.

## References

1. Meeussen, W., Wise, M., Glaser, S., Chitta, S., McGann, C., Mihelich, P., Marder-Eppstein, E., Muja, M., Eruhimov, V., Foote, T., Hsu, J., Rusu, R.B., Marthi, B., Bradski, G., Konolige, K., Gerkey, B., Berger, E.: Autonomous door opening and plugging in with a personal robot. In: Proc. IEEE Intl. Conf. Robotics and Automation (ICRA), Anchorage (2010)
2. van den Berg, J., Miller, S., Goldberg, K., Abbeel, P.: Gravity-based robotic cloth folding. In: Hsu, D., Isler, V., Latombe, J.-C., Lin, M.C. (eds.) *Algorithmic Foundations of Robotics IX*. STAR, vol. 68, pp. 409–424. Springer, Heidelberg (2010)
3. Chen, T.L., Kemp, C.C.: A direct physical interface for navigation and positioning of a robotic nursing assistant. *Advanced Robotics* 25, 605–627 (2011)
4. Salisbury, K.: Whole arm manipulation. In: Proc. Intl. Symposium Robotics Research (ISRR), pp. 183–189 (1987)
5. Yeap, S., Trinkle, J.C.: Dynamic whole-arm dextrous manipulation in the plane. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* (1995)
6. Song, P., Yashima, M., Kumar, V.: Dynamics and control of whole arm grasps. In: Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA), Seoul, pp. 2229–2234 (2001)
7. Platt Jr., R., Fagg, A.H., Grupen, R.A.: Extending fingertip grasping to whole body grasping. In: Proc. of IEEE Intl. Conf. on Robotics and Automation (ICRA), pp. 2677–2682 (2003)
8. Hsiao, K., Lozano-Perez, T.: Imitation learning of whole-body grasps. In: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Beijing, pp. 5657–5662 (2006)
9. Watson, J., Payne, R., Chamberlain, A., Jones, R., Sellers, W.I.: The kinematics of load carrying in humans and great apes: Implications for the evolution of human bipedalism. *Folia Primatologica* 80(5), 309–328 (2009)
10. Liu, J., Lockhart, T.E., Granata, K.: Effect of load carrying on local dynamic stability. In: Proc. Human Factors and Ergonomic Society Annual Meeting, vol. 51(5), pp. 909–913 (October 2007)
11. Schaal, S., Atkeson, C.G.: Open loop stable control strategies for robot juggling. In: Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA), Atlanta, vol. 3, pp. 913–918 (1993)
12. Schaal, S., Atkeson, C.G.: Robot juggling: An implementation of memory-based learning. *Control Systems Magazine* 14(1), 15–71 (1994)
13. Schaal, S., Atkeson, C.G.: Assessing the quality of local linear models. In: Cowan, J.D., Tesauro, G., Alspecter, J. (eds.) *Proc. of Advances in Neural Information Processing Systems*, pp. 160–167 (1994)
14. Wolpert, D., Kawato, M.: Multiple paired forward and inverse models for motor control. *Neural Networks* 11, 1317–1329 (1998)

# Mapping Grasps from the Human Hand to the DEXMART Hand by Means of Postural Synergies and Vision<sup>\*</sup>

Fanny Ficuciello, Gianluca Palli, Claudio Melchiorri, and Bruno Siciliano

**Abstract.** This work aims at defining a suitable postural synergies subspace for the DEXMART Hand from observation of human hand grasping postures. Previous works were carried out on a preliminary prototype (the UB Hand IV), without neither proprioceptive integrated sensors nor external sensors, by means of a joint-to-joint mapping technique. Using an RGB camera and depth sensor for 3D motion capture, the human hand palm pose and fingertip positions have been measured for a reference set of grasping postures. The proposed method for the determination of the synergies subspace is based on the kinematics mapping from the human hand to the robotic hand using data from experiments involving five subjects. The subjects' hand configurations have been mapped to the robotic hand by matching the hand pose and fingertip positions and applying a closed-loop inverse kinematic algorithm. Suitable scaling factors have been used to adapt the DEXMART Hand kinematics to the subjects' hand dimension. By means of Principal Component Analysis (PCA), the kinematic patterns of the first three predominant synergies have been computed and a brief comparison with the previous method and kinematics is reported. Finally, a synergy-based control strategy has been used for testing the efficiency of the grasp synthesis method.

---

Fanny Ficuciello · Bruno Siciliano

DIETI - University of Naples Federico II, Via Claudio, 21, 80125 Napoli, Italy  
e-mail: {fanny.ficuciello,bruno.siciliano}@unina.it

Gianluca Palli · Claudio Melchiorri

DEI - University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy  
e-mail: {gianluca.palli,claudio.melchiorri}@unibo.it

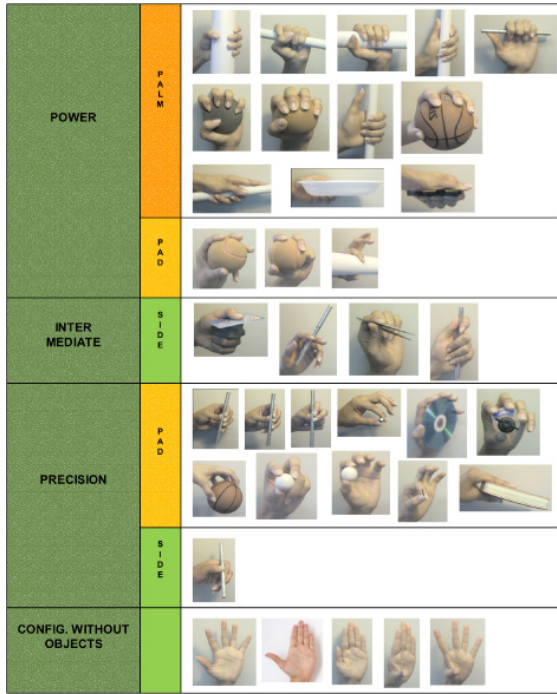
<sup>\*</sup> This research has been partially funded by the EC Seventh Framework Programme (FP7) under grant agreement no. 287513 for the IP SAPHARI (Safe and Autonomous Physical Human-Aware Robot Interaction).

## 1 Introduction

Recent studies on neuroscience and robotics have shown that imitating human prehension is a promising way to simplify and improve grasp planning and control issues related to high multiple Degree-of-Freedom (DoF) devices such as anthropomorphic robotic hands. In [22], the authors measure a set of static human hand postures by recording 15 joint angles and, by means of the PCA, they show that the first two principal components account for  $>80\%$  of the hand postures. Thus, the use of the principal components, also called postural synergies, holds great potential for robot hands control, implying a substantial reduction of the grasp synthesis problem dimension. Transferring human hand motion to a robotic hand is a quite challenging problem due to the complexity and variety of hand kinematics and the dissimilarity with the robotic hand. Indeed, in order to obtain a thorough human hand posture estimation, a reliable kinematic hand model and high-accurate motion tracking instrumentation are required. A synergies mapping from the human hand to the robotic hand has been addressed in [12]. The proposed mapping strategy between the synergies of a paradigmatic human hand and a robotic hand is carried out in the task space and it is based on the use of a virtual sphere. In [11] three synergies have been extracted from data on human grasping experiments and mapped to a robotic hand. Thus, a neural network with the features of the objects and the coefficients of the synergies has been trained and employed to control robot grasping. Recently, Kinect technology is increasingly used for hand tracking, since it interprets 3D scenes thanks to the depth sensor, consisting of an infrared laser projector, combined with an RGB camera. In [16] a model-based method for recovering and tracking the 3D position, the orientation and the full articulation of a human hand has been proposed from marker-less visual observations. In [10] a heuristic hand tracker has been developed for the animation of the hand avatar in the virtual reality and for the implementation of the force rendering in wearable haptics. In this work we propose a model-based method to map grasps from the human to the DEXMART Hand by means of vision. The Kinect sensor is used for 3D human hand fingertips detection and a closed-loop inverse kinematic algorithm is used for human grasp mapping. A reference set of grasps have been selected and five subjects have been involved in the experiments. The inverse kinematic algorithm is based on the DEXMART Hand kinematics that is linearly scaled according to the subjects' hand dimensions.

## 2 Technical Approach and Motivation

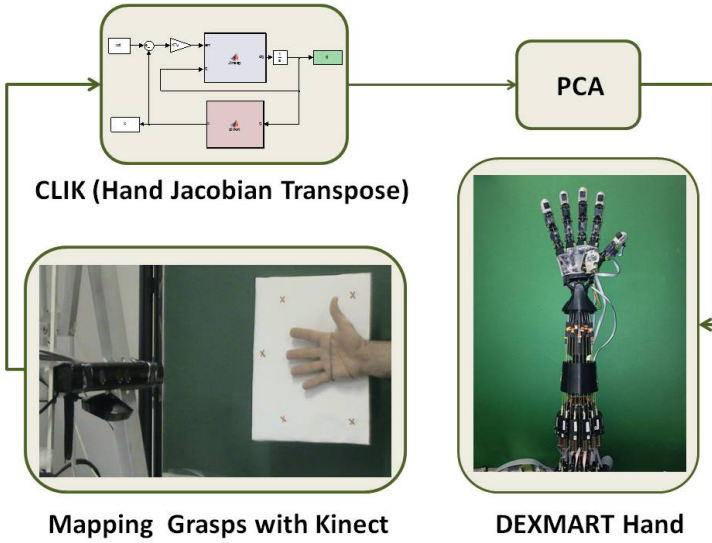
According to recent grasp taxonomy literature [21], in [8] and [9] a set of 36 human hand grasp configurations was selected and adopted for experiments carried out on a preliminary prototype of the DEXMART Hand [18], the UB Hand IV [2]. The mapping of the human grasps to the robotic hand was obtained by means of a joint-to-joint mapping technique, i.e. the fingers' joint position of the DEXMART Hand prototype were manually adjusted in order to imitate as much as possible



**Fig. 1** Reference set of comprehensive human grasps and open-hand configurations used for PCA

the human hand desired configuration. Obviously, this method strongly depends on the interpretation of the reference grasp by the operator and on his/her ability to reproduce the grasp with the robotic hand.

In this work, the same reference set of 36 human hand grasps, represented in Fig. 1, has been considered. Notwithstanding, a new method, based on fingertip mapping from human hand grasps to the robotic hand, has been adopted using measures in 3D space of the fingertip positions and inverse kinematics. A computer vision system, able to detect the position in 3D real world coordinates of each point in the scene, has been adopted in order to make the method independent of the characteristics and skills of the operator. As a matter of fact, the robotic hand grasping postures used for PCA are obtained from measurements performed directly on the hand of several subjects while they execute the grasps of the reference set. A commercial low-cost RGB+Depth (RGBD) camera, such as the Kinect from Microsoft Corp., has been adopted since this device achieves suitable precision (about 1mm) for space and distance in the range of interest of our measurements (an area of about  $0.6 \times 0.6$  m at a distance of about 0.5 m). Moreover, a large number of open-source software for both the use of the Kinect sensor and computer vision applications are freely available. Therefore, we produced very easily and quickly a customized application for the execution of the required measures. Instead, capturing the motion



**Fig. 2** Diagram of the methodological approach

of all the phalanges and the position of the joints during a grasp by means of a single camera is a very challenging task because of evident occlusion problems. Moreover, the reconstruction of the human hand joint angular positions implies the knowledge of its kinematics, not completely consistent in anatomical literature [13, 7], and may cause problems when the acquired information must be mapped on a different kinematic structure.

The DEXMART Hand is designed with a human-like kinematics and a total amount of  $h = 15$  joint angles describes the whole robot hand configuration. Therefore we directly map the human fingertip position measured in the palm frame to the robot hand kinematics by means of a suitable scaling of the robot hand dimensions. The fingertip positions and the palm pose in the Cartesian space, and the fingers length of the human hand are all the measurements we need to compute the 15 joint angles of the robotic hand by means of kinematic inversion. An inverse closed-loop kinematic algorithm has been adopted also to minimize the errors that affect the measures of the fingertip and palm positions, performed by the Kinect, due to noise and limited precision. Since significant differences exist in both the size and the kinematics of the human hands [15], the method has been applied to 5 different subjects. For each of the 36 grasps in the table (Fig.1), mapped from the five subjects to the robotic hand, the mean value has been computed and utilized for deriving the postural synergies by means of PCA. Finally, the grasps in the reference set has then been performed with the DEXMART Hand by controlling the whole motion during reach to grasp using a linear combination of the selected synergies. A schematic representation of the whole procedure for postural synergies subspace computation is represented in Fig. 2.

### 3 The DEXMART Hand

The DEXMART Hand [18] is an innovative anthropomorphic hand developed within the DEXMART project [1]. Based on a previous prototype, the so-called UB Hand IV (University of Bologna Hand, version IV) [2], the DEXMART Hand has been designed with an improved thumb kinematics. The new design aims at a human-like manipulation capabilities and mobility, allowing the opposition of the thumb with the other four fingers. For this purpose, the thumb kinematics has been differentiated with respect to the other fingers and provided of different joint limits. In particular, the adduction/abduction base joint and the first flexion joint have been inverted in the kinematic chain order. Furthermore, the Denavit-Hartenberg (D-H) parameters, reported in Tab. 1, are different for all the fingers in order to fit better with the human hand kinematics. The DEXMART Hand is also equipped with a wide set of sensors which comprise joint position sensors, tendon force sensors [20] and fingertip tactile sensors [6]. Moreover, the twisted string actuation concept [19] has been adopted for driving this robotic hand. As in the previous version, the maximum design simplification and reduction of the device's cost production and development has been held. To this end, the DEXMART Hand has been conceived by taking into account the following driving issues:

- The hand mechanics is based on an endoskeletal structure articulated by means of pin joints integrated into the phalanx body simply consisting in a plastic shaft which slides on a cylindrical surface [2, 5, 14].
- Remotely located actuators with tendon-based transmissions routed by sliding paths (*sliding tendons*) [17] have been adopted for the joints actuation.
- A purposefully designed soft cover mimicking the human skin [4, 3] has been introduced for improving the grasping capabilities of the hand.
- The mechanical structure of the hand has been manufactured adopting additive technologies (Fused Deposition Manufacturing).

Taking inspiration from the biological model and in order to reduce the complexity of the hand control, an internal non-actuated (passive) tendon has been introduced to couple the movements of the last two joints of each finger, i.e. the medial and the distal joint. Hence, only three angles are considered for the index, the middle, the ring and the little finger, i.e. the base (adduction/abduction) angle  $\theta_{1f}$ , the proximal angle  $\theta_{2f}$  and the medial angle  $\theta_{3f}$ . About the thumb, the angles are progressively: the base (proximal angle) angle  $\theta_{1t}$ , the adduction/abduction angle  $\theta_{2t}$  and the medial angle  $\theta_{3t}$ . Therefore, a total amount of  $h = 15$  joint angles is needed to describe the robotic hand configuration. The joint angles ranges for each finger are mechanically constrained by stroke limiters within the intervals:

$$\theta_{1f} \in [-10, 10], \quad \theta_{2f} \in [0, 90], \quad \theta_{3f} \in [0, 110] \quad [\text{deg}] \quad (1)$$

and for the thumb within the intervals:

$$\theta_{1t} \in [0, 90], \quad \theta_{2t} \in [0, 60], \quad \theta_{3t} \in [0, 90] \quad [\text{deg}]. \quad (2)$$



**Table 1** Denavit-Hartenberg parameters of the DEXMART Hand fingers

Link (Thumb)					Link (Index)				
	$d$ [mm]	$\theta$	$a$ [mm]	$\alpha$ [deg]		$d$ [mm]	$\theta$	$a$ [mm]	$\alpha$ [deg]
1	7.53	$\theta_1 + 85$	19.64	-110	1	40.75	$\theta_1$	82	-95
2	-42.5	$\theta_2 - 80$	18	-90	2	-2.91	$\theta_2 - 20$	18	90
3	-1.65	$\theta_3 + 10.62$	24.57	70.32	3	0	$\theta_3$	38	0
4	4.89	$\theta_4 - 3.61$	30	0	4	0	$\theta_4$	28	0
5	0	$\theta_4$	30	0	5	0	$\theta_4$	28.5	0

Link (Middle)					Link (Ring)				
	$d$ [mm]	$\theta$	$a$ [mm]	$\alpha$ [deg]		$d$ [mm]	$\theta$	$a$ [mm]	$\alpha$ [deg]
1	14.34	$\theta_1$	86	-86	1	-11.16	$\theta_1$	82	-80
2	-4.91	$\theta_2$	18	90	2	-1.93	$\theta_2 - 5$	18	90
3	0	$\theta_3$	40	0	3	0	$\theta_3$	38	0
4	0	$\theta_4$	28	0	4	0	$\theta_4$	28	0
5	0	$\theta_4$	28.5	0	5	0	$\theta_4$	28.5	0

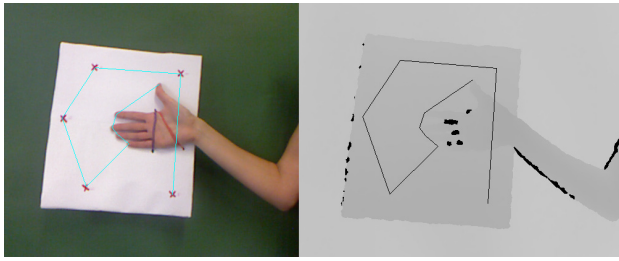
  

Link (Little)				
	$d$ [mm]	$\theta$	$a$ [mm]	$\alpha$ [deg]
1	-36.1	$\theta_1$	68	-75
2	4.24	$\theta_2 + 15$	18	90
3	0	$\theta_3$	35	0
4	0	$\theta_4$	28	0
5	0	$\theta_4$	28.5	0

## 4 Mapping the Human Hand Grasps

The Kinect RGBD camera has been used to provide the RGB color image and the corresponding depth image of the hand for 3D fingertip positions and palm pose detection. After the calibration [24], it is possible to reconstruct the 3D real world coordinates (in meters) with respect to the camera frame. This functionality is embedded in the OpenCV library that has been used for the Kinect image elaboration, whereas the freenect driver has been used for low-level communication with the RGBD camera. For the purpose of this work, we developed an application that allows collecting and saving the information coming from the Kinect. The points to be detected are selected by clicking with the mouse on the RGB image and by matching information from the RGB data and depth data. To obtain the fingertip positions with respect to the hand reference frame (the palm frame), the hand position and orientation (the palm pose) with respect to the RGBD camera has to be known. As a matter of fact, the measures provided by the sensor are represented in the camera frame, thus, for inverse kinematic computation, the homogeneous transformation between the camera frame and the palm frame has to be known. This goal is reached by measuring a set of points fixed to the palm. Since the palm can be hidden by the grasped object, a rigid panel with five reference points has been attached to the opisthenar (the back of the hand). Thus, before starting the mapping procedure, each subject was instructed first in wearing the rigid support and then in assuming an open hand posture corresponding to the DEXMART Hand configuration of zero joint angles. The distance of the fingertips from the palm base in the open hand configuration was measured with the aim of defining suitable scaling factors between the subject hand and the robot hand size. The finger lengths of the 5 subjects are reported in Table 2.





**Fig. 3** Open Hand Configuration corresponding to the DEXMART Hand joint angles set to zero

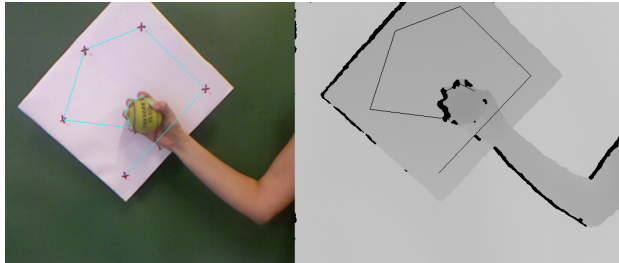
**Table 2** Finger lengths (in mm) of the five subjects and of the DEXMART Hand

Finger	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	DEXMART Hand
Thumb	115	120	130	125	140	126
Index	165	170	173	180	190	217
Middle	170	175	180	185	200	216
Ring	160	165	173	175	185	210
Little	140	140	155	150	165	198

As shown in Fig. 3, ten points have been detected in the open configuration: the fingertips and the five points on the support. The obtained fingertip positions, expressed in the camera frame, have been then matched with the ones of scaled version of the robot hand kinematics expressed in the robot hand reference frame (the palm frame), obtaining in this way the affine transformation  $T_i$  between the camera and the hand reference frame for the  $i$ -th subject. The position of the panel reference points  $p_{i,j_{ref}}$ ,  $j = 1, \dots, 5$  with respect to the hand frame has been then computed on the basis of the transformation  $T_i$ .

Once  $T_i$  and  $p_{i,j_{ref}}$  have been determined, each subject has been asked to achieve each of the 36 reference postures using the right hand and following the approach depicted in Fig.1. The fingertip position and the position of the panel reference points have been acquired as shown in Fig. 4. By matching the position of the panel reference points measured by the Kinect with  $p_{i,j_{ref}}$  the new affine transformation  $T_{i,k}$ ,  $k = 1, \dots, 36$  between the camera and the hand reference frame during the execution of each grasp can be computed together with the position of the subject fingertips  $p_{i,j}$  expressed in the hand frame.

The mapping from the human hand fingertip positions  $p_{i,j}$  and the robotic hand joint positions is then performed by scaling the robot hand link dimensions to fit with the human subject and by inverting the robot hand kinematics through a closed-loop algorithm. For each subject, the DEXMART Hand kinematics is differently scaled for each finger by multiplying the linear D-H parameters (Tab. 1) for the ratio between the human finger length and the corresponding robotic finger length (Tab. 2). A closed-loop inverse kinematic (CLIK) algorithm [23], based on the transpose of the DEXMART Hand Jacobian matrix, has then been implemented in order to map



**Fig. 4** RGB and depth image from Kinect

the subject fingertip position to the robotic hand joint configuration. For each grasp of the reference set (Fig. 1), the average of the five robotic hand configurations  $\mathbf{c}_i \in \mathbb{R}^{15}$  mapped from the five subjects performance has been computed and stored in a configuration matrix  $\mathbf{C}$ . Once the mapping procedure has been completed, the PCA analysis has been computed on the obtained DEXMART Hand configuration matrix.

## 5 Postural Synergies Subspace of the DEXMART Hand

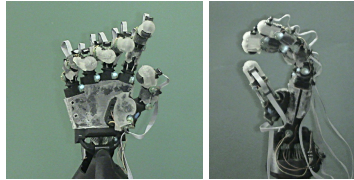
The procedure reported in [9] has been adopted for deriving the postural synergies of the DEXMART Hand on the basis of the data obtained by mapping human hand grasps to the robot hand kinematics, as reported in the previous section. Once the matrix  $\mathbf{C} = \{\mathbf{c}_i \mid i = 1, \dots, 36\}$  of the DEXMART Hand configurations has been built, the vector  $\bar{\mathbf{c}}$  representing the mean hand position in the grasp configurations space (the zero-offset position) and the matrix  $\mathbf{F} = \{\mathbf{c}_i - \bar{\mathbf{c}} \mid i = 1, \dots, 36\}$  of the grasp offsets with respect to the mean configuration have been computed. The PCA has then been performed on the matrix  $\mathbf{F}$  and a base matrix  $\mathbf{E} = \{\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3\}$  of the postural synergies subspace has been found selecting the three predominant components from the PCA. In the following, the three synergies are described, with further considerations about the differences with respect to the previous results obtained with the previous prototype reported in [9], especially for the motion of the thumb.

### 5.1 Kinematics Assessment and Comparison

We refer to the minimum and maximum configuration of each synergy as the robot hand configurations spanned by  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  and  $\mathbf{e}_3$  by means of, respectively, the minimum and maximum value of the corresponding synergy weights without violating the joint limits reported in (2) and (1). When the weights of the synergies are zero, the hand posture corresponds to the zero-offset position (mean position)  $\bar{\mathbf{c}}$ , see Fig. 5 where a frontal and a lateral view of the robot hand mean position are shown. In Figs. 7, 8 and 9 the minimum and maximum configurations in frontal and lateral views of the three postural synergies are represented. The vectors of the

**Table 3** First three eigenpostures and zero offset (data in degrees) vectors of the DEXMART Hand postural synergies subspace

		$e_1$	$e_2$	$e_3$	$\bar{c}$ [deg]
Thumb	adduction/abduction	0.1270	0.1078	-0.3166	18.7185
	proximal	-0.0224	-0.1410	0.4633	35.2796
	medial	0.0363	0.0086	0.6401	38.4707
Index	adduction/abduction	-0.0190	-0.0037	-0.1267	-1.1206
	proximal	0.1610	-0.0471	0.1775	20.6411
	medial	0.2960	0.2785	0.3858	45.6014
Middle	adduction/abduction	0.0428	0.0272	-0.0714	-1.9326
	proximal	0.5254	-0.1617	-0.0477	49.3695
	medial	0.1751	0.4205	0.1212	29.1440
Ring	adduction/abduction	0.0599	0.0283	-0.0978	-1.4200
	proximal	0.5674	-0.2750	-0.1368	56.5125
	medial	0.1724	0.5216	-0.0366	26.2536
Little	adduction/abduction	0.0345	-0.0105	-0.0794	-3.8122
	proximal	0.3789	-0.3453	0.0410	44.4781
	medial	0.2440	0.4659	-0.1390	31.0406

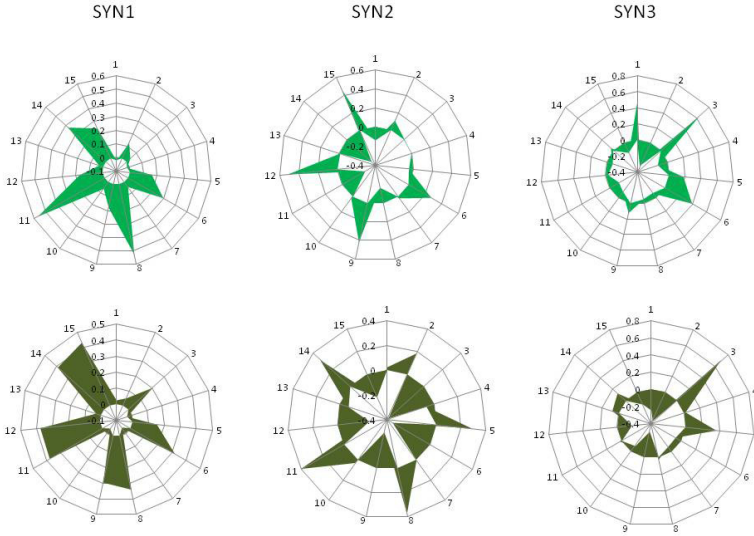


**Fig. 5** Mean position of the DEXMART Hand configuration

three synergies and the zero-offset vector are reported in Tab. 3. The synergies subspace derived in [9] on the basis the previous robot hand prototype by means of the joint-to-joint mapping is now compared with the results obtained with the DEXMART Hand kinematics and the mapping method reported in this paper by means of appropriate graphical tools. The circular graphs in Fig. 6 are a graphical representation of the postural synergy vectors  $e_1$ ,  $e_2$  and  $e_3$  reported in Tab. 3 and identify the joints whose rotations are more involved in each synergy. From left to right, the joint position variation is represented for the first, the second and the third synergy respectively. On the top of the figures, the graphs corresponding to the DEXMART Hand are reported, whereas on the bottom the ones corresponding to the UB Hand IV prototype are shown for comparison. The three fundamental synergies derived for the DEXMART Hand are here briefly described.

*First Synergy*

With reference to the first postural synergy (column  $e_1$  in Tab. 3, Fig. 7), in the minimum configuration the proximal and medial flexion joint angles of all the fingers



**Fig. 6** Graphical representation of the postural synergy vectors  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  and  $\mathbf{e}_3$ . The adduction/abduction, proximal and medial flexion joints are indicated from 1 to 3 for the thumb, from 4 to 6 for the index finger, from 7 to 9 for the middle finger, from 10 to 12 for the ring finger and finally from 13 to 15 for the little finger. On the top of the figure the graphs are referred to the DEXMART Hand, on the bottom to the UB Hand IV prototype.

are all almost zero and increase their value during the motion toward the maximum configuration. The adduction/abduction movements of the four fingers are not very involved in this synergy.

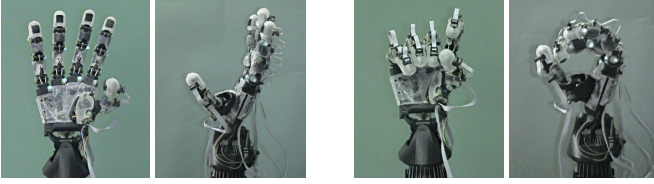
### *Second Synergy*

The second postural synergy (column  $\mathbf{e}_2$  in Tab. 3, Fig. 8) is characterized by a movement in opposite directions of the proximal and medial flexion joints. The adduction/abduction movements of the four fingers are not very involved also in this synergy.

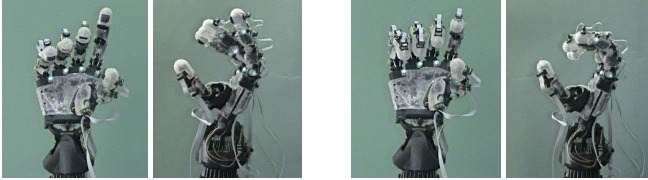
### *Third Synergy*

In the third postural synergy (column  $\mathbf{e}_3$  in Tab. 3, Fig. 9) the movement involves especially the index and the thumb. The movement of adduction/abduction of the thumb and of the index increases with respect to the first two synergies. This characteristic is important for correct index/thumb opposition that allows increasing the grasp accuracy, and thus achieving more stable grasps.

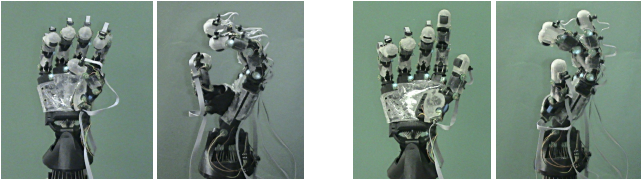
By observing the circular graphs it emerges that the basic characteristics of the three synergies are quite similar for both the kinematics and methods. Nevertheless,



**Fig. 7** From left to right, front and lateral view of the first postural synergy in the minimum and the maximum configuration



**Fig. 8** From left to right, front and lateral view of the second postural synergy in the minimum and the maximum configuration



**Fig. 9** From left to right, front and lateral view of the third postural synergy in the minimum and the maximum configuration

with reference to the kinematics of the DEXMART Hand, even if the excursion of the thumb motion is greater in the third synergy, the differences with the previous two synergies are less evident if compared to the results on the previous robot hand prototype. Since the three predominant postural synergies derived for the DEXMART Hand account for  $>86\%$  of the hand postures, i.e. the percentage of the total variance of the data described by the first 3 principal components of the  $\mathbf{F}$  covariance matrix, it is expected that a control strategy that uses these synergies for robot hand motion allows obtaining very good grasping performance in a configuration space of highly reduced dimensions with respect to the DoFs of the robot hand itself.

## 5.2 Control with Postural Synergies

As in our previous work [9], the first three postural synergies have been used to control the hand in order to perform selected grasps from the reference set. Each hand grasp posture  $\mathbf{c}_i$  can be approximated, by a suitable selection of the postural

synergy weights  $[\alpha_1 \ \alpha_2 \ \alpha_3]^T \in \mathbb{R}^3$ , as the projection  $\hat{\mathbf{c}}_i$  on the postural synergies subspace

$$\hat{\mathbf{c}}_i = \bar{\mathbf{c}} + \mathbf{E} \begin{bmatrix} \alpha_{1,i} \\ \alpha_{2,i} \\ \alpha_{3,i} \end{bmatrix}. \quad (3)$$

Thus, the value of the three eigenpostures weights  $[\alpha_1 \ \alpha_2 \ \alpha_3]^T$  are computed from the desired grasp posture as

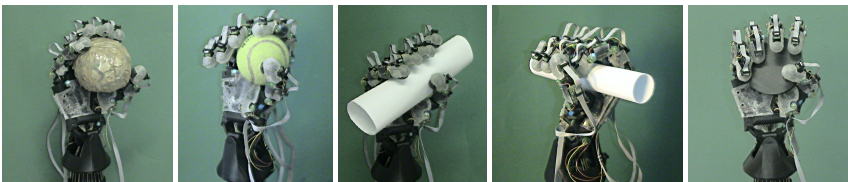
$$\begin{bmatrix} \alpha_{1,i} \\ \alpha_{2,i} \\ \alpha_{3,i} \end{bmatrix} = \mathbf{E}^\dagger (\mathbf{c}_i - \bar{\mathbf{c}}) \quad (4)$$

where  $\mathbf{E}^\dagger$  is the Moore-Penrose pseudo-inverse of the base matrix  $\mathbf{E}$ . The temporal value of the weights  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$  during grasp operations has to be chosen in such a way that, starting from the zero-offset position  $\bar{\mathbf{c}}$  (i.e.  $\alpha_1 = \alpha_2 = \alpha_3 = 0$ ), the hand opens during the reach in preparation for object grasp, and then closes reaching a suitable shape determined from (4) and depending on the original grasp configuration  $\mathbf{c}_i$ . In the open-hand configuration, namely  $\mathbf{c}_0$ , all the flexion joint angles are close to zero, and the corresponding values of  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  can be determined from (4) by posing  $\mathbf{c}_i = \mathbf{c}_0$ .

The intermediate values of the synergy weights have been determined by assuming a suitable time interval for the grasp operation (six seconds for the whole reach to grasp phase, three seconds for both the opening and closing phases) and by linear interpolation of the  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  values in the three reference configurations  $\{\bar{\mathbf{c}}, \hat{\mathbf{c}}_0, \hat{\mathbf{c}}_i\}$ .

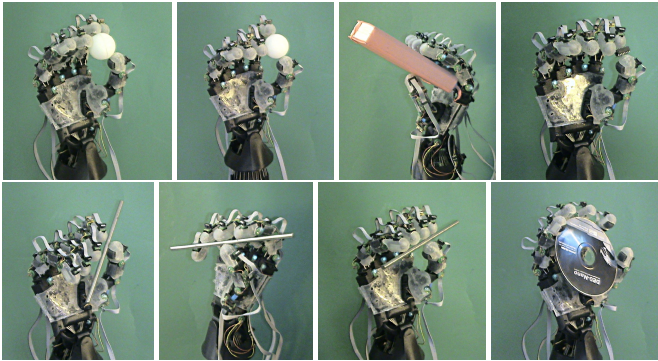
## 6 Experimental Evaluation of the Synergy-Based Grasp Control

In the experiments, starting from the zero-offset position, the hand moves continuously in the synergies configuration subspace and goes in an open-hand configuration. Then, it closes reaching a configuration that depends on the particular grasp to be performed. During the closing phase, the weights of the three postural synergies are obtained by linear interpolation from those corresponding to the open-hand

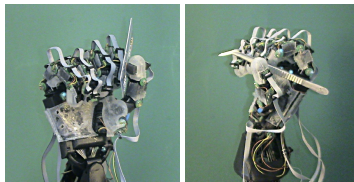


**Fig. 10** Reproduced power grasps from the reference set of postures using the first three synergies





**Fig. 11** Reproduced precision grasps from the reference set of postures using the first three synergies



**Fig. 12** Reproduced intermediate side grasps from the reference set of postures using the first three synergies

**Table 4** Synergy weights of the grasps from the reference set of postures

Conf.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
$\alpha_1$	6.853	46.23	49.31	59.63	25.91	-22.48	23.33	46.92	-15.63	29.30	40.92	53.38
$\alpha_2$	-38.23	-9.027	-27.78	-16.81	89.32	94.38	-38.39	34.18	-31.73	-3.989	-21.40	-22.79
$\alpha_3$	9.587	13.77	13.45	-13.61	-1.344	38.53	21.57	-18.98	13.45	-23.41	-26.13	-12.12
Conf.	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24
$\alpha_1$	0.379	23.39	-63.11	63.99	23.05	51.46	64.84	10.45	24.91	37.85	-0.793	-71.15
$\alpha_2$	-13.23	11.83	1.973	26.72	-0.912	10.02	29.73	-3.443	-5.238	6.500	-3.208	9.748
$\alpha_3$	6.449	-2.747	23.69	-5.729	-26.83	-3.781	-14.49	10.40	3.79	-4.464	9.452	11.09
Conf.	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35	C36
$\alpha_1$	0.552	-31.042	12.68	-2.231	-37.17	6.858	49.08	106.4	-102.5	-94.19	-100.9	-103.6
$\alpha_2$	-30.58	-3.642	-1.075	-7.637	8.701	-50.77	8.869	-5.661	3.314	10.97	-2.110	-8.587
$\alpha_3$	-6.857	18.39	4.446	8.478	25.41	12.38	-6.857	-32.20	-33.54	-32.88	2.810	1.624

configuration to those suitable values unique for each object and computed using (4). The linear combination of the three synergies allows a power grasp of both cylinders and spheres of different dimensions by means of suitable opposition of the thumb, see Fig. 10.

In Fig. 11, the performance corresponding to precise grasp operations are reported considering both prismatic and circular objects and grasps involving from two to five fingers.

The reproduced intermediate side grasps from the reference set of postures is depicted in Fig. 12. Table 4 reports the first three synergy weights computed by projection of the reference set of postures in the synergies subspace, and indicates the contribution rate of each synergy to achieve the final configuration.

## 7 Conclusion

In this work, a model-based method for mapping grasps from the human hand to the DEXMART Hand has been presented. The method comprises fingertips detection, using a commercial low-cost RGB+Depth (RGBD) camera such as the Kinect sensor, and a closed-loop inverse kinematics algorithm, that is based on the DEXMART Hand kinematics. The robotic hand kinematics is linearly scaled according to the hands dimension of five subjects involved in the experiments. The kinematic patterns of the first three postural synergies of the DEXMART Hand are computed. A comparison with the synergies corresponding to the UB Hand IV prototype, obtained using a joint-to-joint mapping, reveals that the basic characteristics of the three synergies are quite similar for both the kinematics and methods. Experimental evaluation of the computed synergies subspace has been performed testing the synergy-based control during reach to grasp. The results of the experiments reveals that hand succeeded on grasping various objects throughout a complete taxonomy.

## References

1. DEXMART Project website, <http://www.dexmart.eu/>
2. Berselli, G., Borghesan, G., Brandi, M., Melchiorri, C., Natale, C., Palli, G., Pirozzi, S., Vassura, G.: Integrated mechatronic design for a new generation of robotic hands. In: Proc. IFAC Symposium on Robot Control, Gifu, Japan (2009)
3. Berselli, G., Piccinini, M., Palli, G., Vassura, G.: Engineering Design of Fluid-Filled Soft Covers for Robotic Contact Interfaces: Guidelines, Nonlinear Modeling, and Experimental Validation. *IEEE Trans. on Robotics* 27(3), 436–449 (2011)
4. Berselli, G., Vassura, G.: Differentiated layer design to modify the compliance of soft pads for robotic limbs. In: Proc. IEEE Int. Conf. on Robotics and Automation, Kobe, Japan, pp. 1285–1290 (2009)
5. Biagiotti, L., Lotti, F., Melchiorri, C., Palli, G., Tiezzi, P., Vassura, G.: Development of UB Hand 3: Early results. In: Proc. IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, pp. 4488–4493 (2005)
6. De Maria, G., Natale, C., Pirozzi, S.: Force/tactile sensor for robotic applications. *Sensors and Actuators A: Physical* (2012), doi:10.1016/j.sna.2011.12.042
7. Drenckhahn, D., Benninghoff, A.: Anatomie: Makroskopische Anatomie, Embryologie und Histologie des Menschen. Zellen-und Gewebelehre, Entwicklungsbiologie, Bewegungsapparat, Herz-Kreislauf-System, Immunsystem, Atem-und Verdauungsapparat. Anatomie: Makroskopische Anatomie, Embryologie und Histologie des Menschen. Urban & Schwarzenberg (1994)



8. Ficuciello, F., Palli, G., Melchiorri, C., Siciliano, B.: Experimental evaluation of postural synergies during reach to grasp with the UB Hand IV. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, San Francisco, pp. 1775–1780 (2011)
9. Ficuciello, F., Palli, G., Melchiorri, C., Siciliano, B.: Planning and control during reach to grasp using the three predominant ub hand iv postural synergies. In: Proc. IEEE Int. Conf. Robotics and Automation, Saint Paul, MN, pp. 1775–1780 (2012)
10. Frati, V., Prattichizzo, D.: Using kinect for hand tracking and rendering in wearable haptics. In: IEEE World Haptics Conference, Istanbul (2011)
11. Geng, T., Lee, M., Hulse, M.: Transferring human grasping synergies to a robot. *Mechatronics* 21(1), 272–284 (2011)
12. Gioioso, G., Salvietti, G., Malvezzi, M., Prattichizzo, D.: Mapping synergies from human to robotic hands with dissimilar kinematics: An object based approach. In: Proc. IEEE Int. Conf. on Robotics and Automation, Workshop on Manipulation Under Uncertainty, Shanghai (2011)
13. Kapandji, I., Honoré, L.: *The Physiology of the Joints: The upper limb. The Physiology of the Joints*, Churchill Livingstone (2007)
14. Lotti, F., Vassura, G.: A novel approach to mechanical design of articulated finger for robotic hands. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robot and Systems, vol. 2, pp. 1687–1692 (2002)
15. Grebenstein, M., et al.: A method for hand kinematics designers 7 billion perfect hands. In: Proc. of 1st International Conference on Applied Bionics and Biomechanics, Venice, Italy (2010)
16. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: Proc. 13th International Conference on Computer Vision, Barcelona, pp. 1260–1264 (2011)
17. Palli, G., Borghesan, G., Melchiorri, C.: Modeling, identification and control of tendon-based actuation systems. *IEEE Trans. on Robotics* 28(2), 277–290 (2012)
18. Palli, G., Melchiorri, C., Vassura, G., Berselli, G., Pirozzi, S., Natale, C., De Maria, G., May, C.: Innovative technologies for the next generation of robotic hands. In: Siciliano, B. (ed.) *Advanced Bimanual Manipulation. STAR*, vol. 80, pp. 173–217. Springer, Heidelberg (2012)
19. Palli, G., Natale, C., May, C., Melchiorri, C., Wurtz, T.: Modeling and control of the twisted string actuation system. *IEEE/ASME Transactions on Mechatronics* 18(2), 664–673 (2013), doi:10.1109/TMECH.2011.2181855
20. Palli, G., Pirozzi, S.: Miniaturized optical-based force sensors for tendon-driven robots. In: Proc. IEEE Int. Conf. on Robotics and Automation, Shanghai, China, pp. 5344–5349 (2011)
21. Romero, J., Feix, T., Kjellstrom, H., Kragic, D.: Spatio-temporal modelling of grasping actions. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Taipei, pp. 2103–2108 (2010)
22. Santello, M., Flanders, M., Soechting, J.: Postural hand synergies for tool use. *Journal of Neuroscience* 18(23), 10,105–10,115 (1998)
23. Siciliano, B., Khatib, O. (eds.): *Springer Handbook of Robotics*. Springer (2008)
24. Zhang, C., Zhang, Z.: Calibration between depth and color sensors for commodity depth cameras. In: Proc. Int. Conf. on Multimedia and Expo, Barcelona, Spain, pp. 1–6 (2011)

# Manipulation with Multiple Action Types<sup>\*</sup>

Jennifer Barry, Kaijen Hsiao, Leslie Pack Kaelbling, and Tomás Lozano-Pérez

**Abstract.** We present DARRT, a sampling-based algorithm for planning with multiple types of manipulation. Given a robot, a set of movable objects, and a set of actions for manipulating the objects, DARRT returns a sequence of manipulation actions that move the robot and objects from an initial configuration to a final configuration. The manipulation actions may be non-prehensile, meaning that the object is not rigidly attached to the robot, such as push, tilt, or pull. We describe a simple extension to the RRT algorithm to search the combined space of robot and objects and present an implementation of DARRT on the Willow Garage PR2 robot.

## 1 Introduction

Consider a robot trying to move a plate that is lying flat on a cluttered table to another table. The robot cannot grasp the plate while the plate lies flat on the table, so it has to first maneuver the plate to the edge of the table and then grasp the plate in a way that enables it to later place the plate. This task requires at least three different types of manipulation: push, pick and place.

The classic “pick and place” tasks use only two types of manipulation: transit, in which the robot moves alone, and rigid-transfer, in which the robot moves a rigidly attached object. These two types of manipulation are usually planned separately, connected only by the grasp. Human manipulation, however, is not limited to picking and placing but instead combines many different types of manipulation: pushing,

---

Jennifer Barry · Leslie Pack Kaelbling · Tomás Lozano-Pérez

MIT, Cambridge, MA

e-mail: {jbarry, lpk, tlp}@csail.mit.edu

Kaijen Hsiao

Willow Garage, Menlo Park, CA

e-mail: hsiao@willowgarage.com

<sup>\*</sup> This material is based upon work supported by the National Science Foundation Grant No. 1122374.

pulling, flipping, etc. To extend robotic manipulation capabilities towards those of humans we must be able to plan for and control sequences of diverse actions.

In this paper we outline an algorithm for planning with multiple types of manipulation. The algorithm has the structure of a rapidly exploring random tree (RRT) searching the combined configuration space of the robot and objects. The RRT algorithm attempts to directly connect points in configuration space and, if that fails, samples possible intermediate points and connects those. In many spaces this “connection” is straightforward, usually just a straight line in Euclidean space. Because objects cannot move by themselves, however, such a connection is not viable in the combined space of the robot and objects. Instead, we must use a path that reflects the underlying dynamics of the system. Because of the necessary interactions of the robot and objects during manipulation, this requires modifications to both the extension phase and the sampling phase of the RRT algorithm. We discuss these modifications and describe and analyze experiments run on the PR2 robot.

## 2 Related Work

There is a large body of work on manipulation actions for us to draw upon. Mason [13] discusses the mechanics of pushing an object, while Brost [2] and Dogar and Srinivasa [5] propose to combine pushing and grasping in a push-grasp, and Huang and Mason investigate striking or tapping objects [9]. However, these papers focus on describing and simulating the dynamics and control of a specific action. When they address planning, they emphasize working with a particular type of manipulation rather than combining it with other types. We take a different view; we assume that by building on this work, we can simulate the forward motion of the object and focus on planning given a diverse set of these actions.

There has also been work on problems that require the robot to manipulate multiple objects. Much of this work assumes only rigid grasping [1, 14, 17, 18, 19] or that each object moves only once [14, 17, 18, 19]. van den Berg et al. [1] relax this second assumption, but their approach relies on describing connected components of a robot’s configuration space, which is intractable for high-dimensional configuration spaces. Cosgun et al. [3] discuss trying to place an object on a cluttered surface. They assume only the object to be placed is grasped, but that this object can push other objects out of the way. Multiple objects can be moved at once, but this still incorporates only a single manipulation action. Dogar and Srinivasa [6] consider the problem of trying to move an object in clutter and have a library of manipulation actions, including non-prehensile actions, but assume each object or piece of clutter is moved only once using a single manipulation action. In contrast, we are interested in using multiple types of manipulation to manipulate a single object.

The re-grasping problem [12, 16], especially as framed by Siméon et al., is an example of planning with two manipulation actions. Siméon et al. [16] take a hierarchical approach to the problem, first finding a high-level sequence of transits (motions for the robot alone) and rigid-transfers (motions in which the robot rigidly grasps an object) and then planning each in the robot’s configuration space.

Unfortunately, their method relies on the grasped object being able to move instantaneously in any direction, which does not hold for non-prehensile manipulation.

The problem of manipulation with multiple actions is a multi-modal planning problem. Hauser [7] defines a multi-modal planning problem as one in which the system moves among configurations and also among a set of *modes*. The mode space is part of the problem description and each mode describes a set of configurations that all satisfy certain mode-specific constraints. For example, a mode might be the set of configurations in which the robot and object are in a specific grasp. In his initial work Hauser focused on problems with discrete mode spaces, but low-dimensional mode transitions. He showed how to create a two-level roadmap of modes and configurations using interspersed mode and configuration sampling. Later Hauser [8] extended this work to domains like manipulation where the mode space is continuous and used interleaved intra-mode and inter-mode planning to find paths for a walking robot pushing an object on a table. However, that work required the implementation of complicated mode samplers and a number of heuristics, some of which took substantial pre-processing time. Here we show how to solve the problem of manipulation with multiple actions as an RRT with no pre-processing.

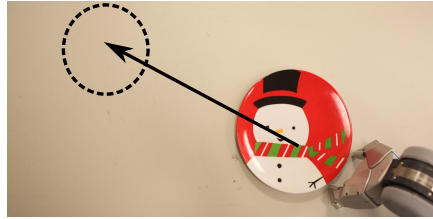
### 3 Problem Definition

We address problems in which we have a robot, a set of movable objects, and a set of diverse, possibly non-prehensile manipulation actions. The input is the configuration space (c-space) of the robot and movable objects, a set of fixed obstacles, a set of manipulation primitives, a starting configuration, and a set of goal configurations. The goal set may be infinite in size. For example, in manipulation, goal positions are often specified only for objects. The goal set is then any configuration in the combined space in which the objects are in their goal positions.

A *manipulation primitive* is a function that takes an initial configuration of the robot and objects and a displacement of the robot's configuration and returns a final configuration of the robot and objects. A *primitive instance* is an instantiation of the primitive with a specific initial configuration and displacement. A *solution* is a sequence of primitive instances that takes the initial configuration into the goal set.

Any type of manipulation can be represented as a manipulation primitive provided it is possible to describe the effect of the primitive on any given configuration of the robot and objects. For complicated primitives it is possible that this would require computational integration of equations of motion, but we use simpler primitives. Throughout this paper we use the following primitives as examples:

- *Transit*: The robot moves alone whenever there is no collision between the robot and the objects or any obstacles in the world.
- *Rigid-transfer*: The robot moves an attached object whenever there is no collision between the robot or object and any obstacles or other objects in the world.
- *Pick*: The robot rigidly grasps an object and lifts that object from a support surface when the robot and object are in a feasible grasp configuration.



**Fig. 1** When the gripper and plate are in two-point contact, the robot can push the plate along the ray connecting its gripper to the plate’s center

- *Push*: The robot pushes an object when it has two-point contact between its gripper and the object. The push can only be along the ray connecting the center of the contact points with the center of mass of the object, as shown in Figure 1.

Given a  $D_R$  degree-of-freedom robot and  $n$  objects each with  $D_i$  degrees of freedom, we have a problem with  $D_R + \sum_{i=1}^n D_i$  degrees of freedom. In Section 4, we discuss our approach to solving this problem.

## 4 DARRT Algorithm

In Algorithm 1, we present the Diverse Action Rapidly Exploring Random Tree (DARRT) algorithm, a sampling-based algorithm for motion planning problems with diverse, non-prehensile manipulation actions. DARRT has the structure of a rapidly exploring random tree (RRT) with controls [11], but the indirect control of the objects, the high-level manipulation primitives, and the necessity of switching between primitives all require modifications to the classic state sampling, action sampling, and distance metrics. We describe DARRT’s EXTENDTOWARDS and SAMPLE methods in detail and discuss its distance metric approximation.

### 4.1 Extension

We first describe the method for extending from a configuration  $c_1$  towards a configuration  $c_2$ . Note that the canonical method of using a short straight line extension in Euclidean space is not applicable here. The Euclidean extension moves  $c_1$  a small amount towards  $c_2$  in each dimension. This moves the robot a short distance from its configuration in  $c_1$  to its configuration in  $c_2$ , but also moves each object a short distance from its configuration in  $c_1$  towards its configuration in  $c_2$ . Because objects cannot move by themselves, it is not possible to actually execute this extension. We need an extension method that reflects the actual dynamics of the system.

However, this is not just a case of planning in a non-holonomic system because manipulation usually requires specific relative configurations of the robot and objects. Thus, the subspaces in which primitives can be executed are usually lower-dimensional than the full configuration space. For example, in Pick, the robot must

**Algorithm 1**

*Input:*  $M$ :  $c$ -space of movable components, robot  $R$  and objects  $\{o_1, \dots, o_n\}$ ,  $B$ : fixed obstacles,  $A$ : manipulation primitives,  $c_I$ : initial configuration,  $G$ : goal set

*Output:* Graph with a configuration in  $G$ .

---

DARRT( $M, B, A, c_I, G$ )

```

1  $V \leftarrow \{c_I\}$ 
2 while no configuration in  $V$  is in  $G$ 
3    $s \leftarrow \text{SAMPLE}(M)$ 
4    $t \leftarrow \arg \min_{v \in V} \text{Distance}(v, s, M, A)$ 
5    $\{c_1, \dots, c_l\} \leftarrow \text{EXTENDTOWARDS}(t, s, M, B, A)$ 
6    $V \leftarrow V \cup \{c_1, \dots, c_l\}$ 
7 return  $V$ 
```

SAMPLE( $M$ )

```

1  $\{m_1, \dots, m_j\} \leftarrow \text{randomSubset}(\{R, o_1, \dots, o_n\})$ 
2  $r \leftarrow \text{random configuration for each } m_i$ 
3 return  $r$ 
```

EXTENDTOWARDS( $c_1, c_2, M, B, A$ )

```

1  $e \leftarrow \text{PATH}(c_1, c_2, M, A)$ 
2  $\{e_1, \dots, e_l\} \leftarrow \text{Discretize}(e)$ 
3 for  $e_i$ , if collision( $e_i, M, B$ ), return  $\{e_1, \dots, e_{i-1}\}$ 
4 return  $\{e_1, \dots, e_l\}$ 
```

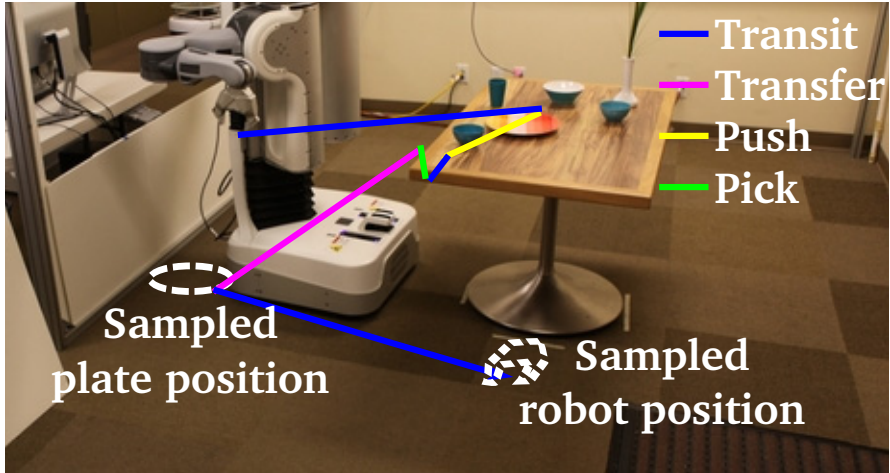
PATH( $c_1, c_2, M, A$ )

```

1 if  $c_1 = c_2$  or no useful primitives, return  $\{\}$ 
2  $p \leftarrow \text{randomUsefulPrimitive}(c_1, c_2, M, A)$ 
3  $P \leftarrow \text{propagate}(p, c_1, c_2, M)$ 
4  $c \leftarrow \text{state after applying } P \text{ to } c_1$ 
5 return  $P \cup \text{PATH}(c, c_2, M, A)$ 
```

---

be holding the object in a feasible grasp; for Push, the robot's gripper must be in two-point contact with the object. Both of these primitives require configurations that have zero probability of being sampled at random from the full configuration space. Moreover, non-prehensile manipulation constrains the space in which the object can move. For example, an object that is being pushed can only be moved along a single ray as shown in Figure 1. Thus, even if  $c_1$  is a configuration in which the gripper and object are in two-point contact, the ray along which the object can move must also be "towards"  $c_2$ . Therefore, we not only need to pass through configurations in the subspace in which the primitive is executable, we must be in the particular part of that subspace in which the object can be moved towards  $c_2$ .

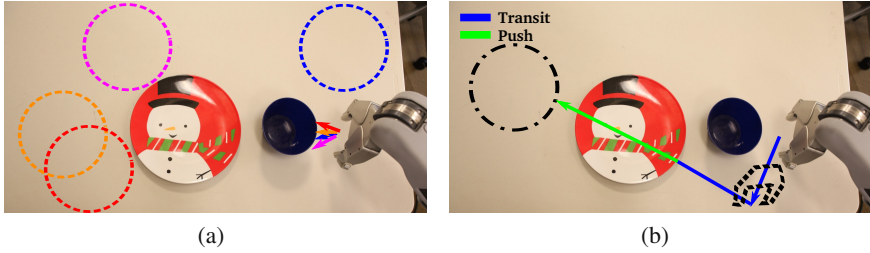


**Fig. 2** An extension from the state shown in the photograph towards the sample shown with the white dashed lines. This sequence first transits the robot to a pushing configuration (blue), pushes the plate towards the edge of the table (yellow), transits the robot to a grasp (blue), picks up the plate (green), transfers it to its sampled position (magenta), and finally transits the robot to its sampled position (blue).

The classic method for extending an RRT with actions is to use a single, short application of some action to move a short distance from  $c_1$  towards  $c_2$ . However, it is difficult to ensure that such short applications can reach and then remain in the subspaces that must be traversed to reach  $c_2$ . Therefore, rather than extend a short way towards  $c_2$ , we try to find a sequence of actions that moves all the way from  $c_1$  to  $c_2$ . Attempting to extend all the way towards the sampled configuration is the version of the RRT described in Lavelle Chapter 5 (2006).

The problem of manipulation with multiple actions is particularly difficult because the presence of obstacles may require complex paths that use a large number of primitives. Without obstacles, finding a path between two configurations is usually easy. There are a number of ways to implement a search for such a path. In our implementation, we required that a primitive  $p$  implement *useful* and *propagate* functions. Propagating  $c_1$  towards  $c_2$  returns a sequence of primitives that applied to  $c_1$  result in a state nearer to  $c_2$  than  $c_1$  is in the subspace in which the primitive operates. A primitive is useful if propagating  $c_1$  towards  $c_2$  will result in a state closer to  $c_2$  than  $c_1$  is. For example, propagating  $c_1$  towards  $c_2$  using Transit results in a state in which the objects are in their positions in  $c_1$  and the robot is in its position in  $c_2$ . Therefore Transit is useful if all objects are in the same position in  $c_1$  and  $c_2$  but the robot is not. Push is useful when an object is on a support surface in  $c_1$  and in a different position in  $c_2$ . Propagating  $c_1$  towards  $c_2$  using Push returns





**Fig. 3** A robot arm pushes a plate on a table. (a) If there are obstacles between the robot and the plate, every direct path (solid colored) will be truncated at the obstacle regardless of the sampled pose of the plate (dashed colored). The colors of the paths added to the tree in this figure correspond to the colors of the sample; i.e. if the plate is sampled in the position shown by the red dashed line, the small red arrow is all that is added to the tree. (b) By sampling robot and object configurations separately, we will eventually sample a configuration for the robot (dashed) that allows a direct path to the plate. A subsequent sample for the plate (dash-dot) results in a much longer extension.

two primitives: Transit to the pushing configuration and Push from the object’s position in  $c_1$  to the object’s position in  $c_2$  or the point on the edge of the support surface nearest the object’s position in  $c_2$ . Pseudo-code is shown in the PATH function in Algorithm 1. Figure 2 shows an example of a path. Note that the path not only includes configurations for executing pushing and grasping, but also ensures that the configuration used for the non-prehensile Push primitive is one that can move the plate towards its sampled position.

Therefore we extend  $c_1$  towards  $c_2$  by finding a sequence of primitive instances that, in the absence of obstacles, takes  $c_1$  to  $c_2$ . We check this path for collisions, truncating it to the first collision, and then add the truncated path to the tree.

### 4.2 Sampling the Space

Although extending with a sequence of actions rather than a single action as described in Section 4.1 finds configurations that lie in lower dimensional subspaces, it creates another problem because we have subspaces with different controllability. We are able to fully control the robot, but can only move objects when the robot can immediately manipulate them. Thus, the robot’s first action along an extension will always be to move directly towards a position from which it can manipulate an object. Sampling random configurations from the space does not in fact sample random movements for the robot.

To illustrate, consider a simple world in which a robot arm is pushing a plate on a table as shown in Figure 3. If we simply sample configurations in this world, there is zero probability that we will sample a configuration with the plate in its initial position. Since the plate cannot move on its own, the robot must first move the plate into its new position. A valid pushing path requires that the gripper be in two-point



contact with the plate so the first part of the extension will always be a direct robot transit into two-point contact with the plate. If all such direct movements intersect an obstacle, as shown in Figure 3(a), the planning will fail.

To alleviate this problem, rather than sample an entire configuration, we first sample a (possibly proper) subset of the movable components (objects and robot)  $O = \{m_1, \dots, m_j\}$ . We then generate a partially specified sample  $s$  in which only the configurations of those components in  $O$  are specified. The distance from a fully specified configuration  $c$  to  $s$  is the distance to the nearest configuration to  $c$  such that each of the components in  $O$  are in the configurations specified in  $s$ . An extension from  $c$  to  $s$  is any path that results in the components in  $O$  being in the configurations specified in  $s$ . By using partially specified sample configurations, we allow the robot and objects to take up new positions relative to each other as shown in Figure 3(b).

Sampling partially specified configurations is all that is necessary to reposition the robot and subsets of objects. However, in experimentation, we found that sampling configurations for the robot and objects together tended to be unhelpful because the path is usually truncated before the segment in which the robot moves to its position. Therefore, we either generate a sample that specifies only configurations of objects or one that specifies only a configuration of the robot.

As is common practice when implementing an RRT, some fixed fraction of samples are from the goal set. When sampling from the goal set, we do not explicitly split the sample into robot and object subspaces but we can take advantage of goals for which that split is natural. In many manipulation problems the goal only specifies positions for the objects, in which case a goal sample fits well into this framework.

### 4.3 Distance Metric

We must also define the distance between two configurations,  $c_1$  and  $c_2$ . Because the configuration space contains subspaces that are not directly controllable, the sum of Euclidean distances in each subspace is a significant underestimate of the actual distance between configurations. The correct distance from  $c_1$  to  $c_2$  is the length of the shortest path traveled by the robot that moves each movable component from its position in  $c_1$  to its position in  $c_2$ .

Evaluating the correct distance function is, of course, intractable. Thus, we first simplify it by ignoring any obstacles in the world; however, the remaining problem is still hard. The robot must “visit” the action of moving each object exactly once, so this is a version of the Traveling Salesman Problem. In our implementation, we used a greedy algorithm to find the Cartesian distance from  $c_1$  to  $c_2$  by finding a path from  $c_1$  to  $c_2$  in which the robot always moves the closest object first.

## 5 Results

We implemented DARRT on the Willow Garage PR2 robot and ran experiments in several domains with a variety of manipulation primitives.

## 5.1 Implementation

We planned for one of the PR2's seven degree-of-freedom arms, its base, and a single rigid object, for a total of sixteen dimensions in the state space. The implementation is built on top of the Open Motion Planning Library (OMPL) [4], which allows the user to define a custom state space, control space, distance metric, sampling algorithm, and extension algorithm for an RRT.

Because we have much finer control over the arms of the robot than we do over the base, we implemented separate transit and rigid-transfer primitives for the arm and the base. At present the planner does not reason about the precision of the primitives, but we hope to add that capability in future work. In total, we implemented eight primitives for the PR2:

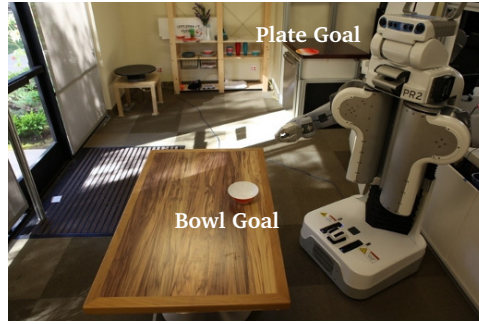
- *Arm-Transit*: The arm moves towards a joint goal in a straight line in joint space.
- *Straight-Line-Arm-Transit*: The arm moves the gripper in a straight line in Cartesian space. This was used for approaching and retreating from objects.
- *Arm-Rigid-Transfer*: The arm moves an attached object towards a goal pose using a straight line in joint space.
- *Base-Transit*: The base moves towards a goal pose.
- *Base-Rigid-Transfer*: The base moves an attached object towards a goal pose.
- *Pick*: When the gripper is in a valid grasp pose, the object is attached to the robot's gripper and lifted in a straight line in Cartesian space.
- *Place*: The object attached to the gripper is set down on a support surface in a straight line in Cartesian space and detached from the gripper.
- *Push*: When the gripper contacts the perimeter of a round object on a support surface, the object is pushed along the ray connecting the gripper's center to the object's center. This primitive is shown in Figure 1.

We detected objects at the start of planning using a point cloud from a Microsoft Kinect by segmenting the cloud above the plane of the table. We did not re-detect the objects at any time during execution. We used a three-dimensional map of the environment for collision checking, as shown in Figures 5 and 6. During execution, we used Monte Carlo localization to localize the base of the robot.

## 5.2 Domains and Problems

We ran experiments on four problems in two different domains, shown in Figures 4-6: Plate1, Plate2, Plate3, and Bowl. In the Plate domain, we had the robot manipulate a flat plate that it could not grasp while the plate was sitting on a table. In the Bowl domain, we demonstrated that the planner works with a bowl that can be directly picked up from a flat surface. In all problems the goal allowed any orientation around the object's  $z$  axis, because our objects were symmetric around this axis. To illustrate the difference in execution successes between pushing and rigidly grasping, we did not include the Push primitive in the Bowl domain.

We used two metrics in our experiments. We looked at planner performance on the problems by evaluating the time it took to plan paths for each problem. We also



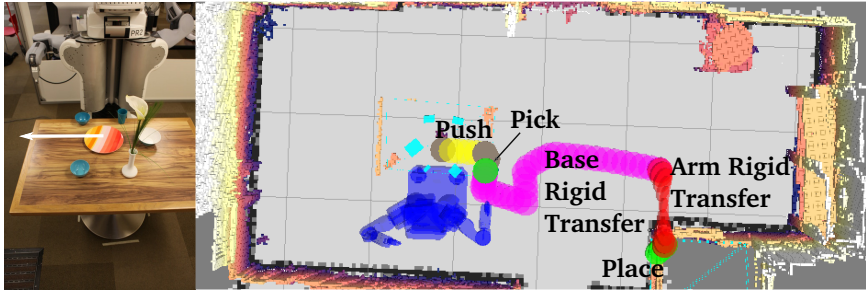
**Fig. 4** The world in which we ran DARRT. For each plate problem, the plate started on the central table and the goal was the corner of the side table. The bowl started on the shelf and the goal was on the central table.

**Table 1** Performance of the unoptimized planner and execution performance on the PR2 for each of the four problems. The Execution Successes are the fraction of the trials in which the robot was able to successfully execute the plan in the real world. We also show planning time for a version of Plate3 in which the goal does not involve a place (Plate 3 (No Place)) and a simple world in which there are no obstacles on the table (Simple Plate). The restart time is the amount of time the planner was given before restarting from the initial state. Times were averaged over 10 trials.

Problem Name	Planning Time (s)	Restart Time (s)	Execution Successes on PR2 Robot
Plate1	257	60	2/5
Plate2	461	200	4/5
Plate3	480	200	3/5
Bowl	66	30	5/5
Plate3 (No Place)	82	60	–
Simple Plate	19	30	–

measured the ability of a real robot to execute the plans returned by our planner. An execution was considered a “success” if the object (plate or bowl) was placed in the goal position without disturbing the rest of the environment (i.e. knocking anything off the table). The robot executed these plans “open-loop” in that it sensed the object’s position once before planning and never again during execution.

The planner was able to find solutions for all four problems. Running times and execution success fractions are given in Table 1. In this table we also give results for a problem identical to Plate3 except that the goal was in the center of the environment rather than on a table. This problem is included to emphasize how difficult it is to plan to place a plate. For comparison’s sake, we also include planning time



(a) Plate1



(b) Plate2



(c) Plate3

**Fig. 5** The plate domain problems. The 3D map of the obstacles present in every domain (walls and tables) is shown as a colored grid while obstacles that we added and removed from the environment (bowls, cups, etc) are shown as blue boxes. For each problem, we show the starting state of the robot and obstacles, possible trajectories for pushing the plate to the edge of the table (white arrows), and the planned trajectory for the plate color-coded by primitive. Videos are on our website<sup>1</sup>.

for a problem in which the goal is in the center of the environment and there are no obstacles placed on the table around the plate. Videos of the trajectories executed by the PR2 in each of the domains are on our website<sup>1</sup>.

<sup>1</sup> <http://people.csail.mit.edu/jbarry/pr2/darrrt>

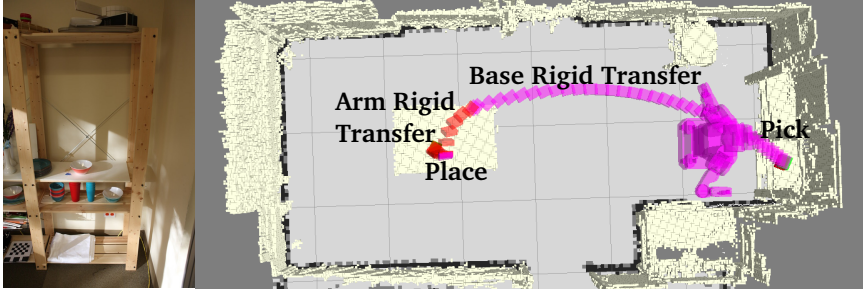


Fig. 6 The Bowl domain. Videos are on our website<sup>1</sup>.

## 6 Performance Analysis

The running times given in Table 1 are for an unoptimized version of the planner. Optimizing the planner will significantly decrease these times. However, we include them because they give a good metric of the relative difficulty of different problems.

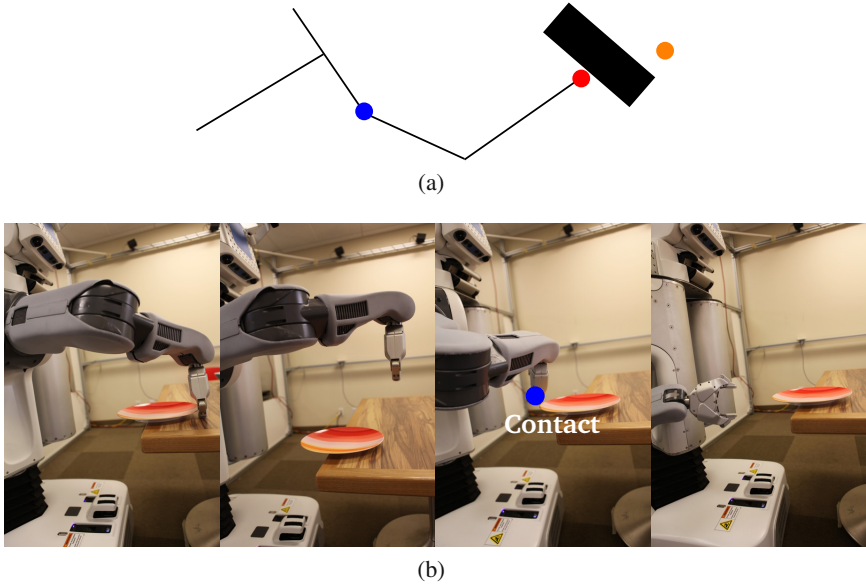
### 6.1 Problem Difficulty

It is clear from the running times in Table 1 that some problems are easier for the planner than others. In the Plate1 problem (Figure 5(a)), for instance, the plate can be moved to the edge of the table in a straight line parallel to the robot's torso. This is an easy manipulation for the robot and requires only three primitive instances (Arm-Transit, Straight-Line-Arm-Transit, and Push), making it relatively easy to find a plan in this domain. Similarly, the Bowl domain (Figure 6) is an easy domain with no pushing at all. We included this domain to show that the algorithm can work easily with other types of objects.

The Plate2 and Plate3 problems, however, require longer plans. In the Plate2 problem (Figure 5(b)), it is possible to move the plate to the edge of the table by pushing the plate back towards the robot using a single push or to the left edge of the table using a minimum of two pushes to move the plate around the small bowl. However, it is not possible to find an arm trajectory that can push the plate towards the robot without first moving the robot's base. In the Plate3 problem (Figure 5(c)) we removed all possible straight line paths to the edge of the table, forcing the planner to plan at least two pushes. This problem shows that the algorithm can find plans requiring multiple instances of a non-prehensile primitive.

### 6.2 Planning Time Analysis

We also analyzed our experiments to find the bottlenecks in planning. As with most sampling-based algorithms, the majority of the planning time is spent finding paths around obstacles. This is a problem common to almost all RRT implementations



**Fig. 7** For most RRT based algorithms the majority of the time is spent finding paths around obstacles. (a) An example in the two dimensional case with a single obstacle (black rectangle). The initial state is shown in blue, a sample in orange, and the nearest point to the sample in red. (b) A similar example in our domain. When the plate is at the edge of the table the robot can grasp it. However, in trying to move from the pushing configuration (left) to the approach to the grasp (right), the gripper usually contacts the plate.

because the algorithm is greedy in its choice for the nearest state in the tree. For example, as shown in Figure 7(a), we usually quickly grow an RRT all the way towards an obstacle. Subsequent samples beyond the obstacle (shown in orange in the figure) find the point near the obstacle (shown in red in the figure) as the “nearest” point in the tree, but this point cannot be extended towards the sample. It takes a large number of samples to find a path around the obstacle. In our experiments, this problem was most evident in two situations: transitioning from pushing to grasping and placing on the table. We describe these scenarios in detail below.

When the plate is at the edge of the table, the robot can grasp it. However, in moving the plate to the edge of the table, the robot must have used the push primitive, which puts its gripper on the far side of the plate from the table edge, as shown in Figure 7(b). During the transition to the grasp, the robot retreats upwards from the push and then moves in a straight line in joint space to the approach to the grasp. In many cases, there is a collision between the plate and the robot’s gripper along this line. Subsequent samples for the plate will almost all be nearest to this state because both the plate and the robot are close to a state in which the robot can approach a grasp for the plate. However, the state in the tree cannot be extended directly towards the approach to the grasp because of the gripper-plate collision. In order to



move around the plate, the gripper must first move off the direct line to the approach to the grasp, around the plate, and then to the approach to grasp configuration. This requires a large number of samples in the robot's subspace.

The table on which we placed the plate is approximately one meter tall, which makes it too high for the robot to manipulate upon easily, especially as the gripper must be angled when placing flat plates. In addition, this table is about 25 centimeters higher than the table from which we picked the plate, which means that if the robot does a base transfer directly from the pick to the place, the plate will hit the side of the table as the robot moves it towards the place position. This creates a state in which the plate is near the place position but cannot be moved directly there.

To understand the relative difficulty of placing versus transitioning from pushing to grasping, we also considered a problem in which the robot had to push the plate off of the table and transfer it to a specific position in the environment but not place it. One might expect pushing to be harder than placing, because the relative configurations of the robot and object are more constrained. However, because of the particulars of our test environment, the placements are more constrained. In particular, the central table where the pickup happens is one under which the robot can move its base whereas the table for the place has a solid base. Without placing, the planning time was not much over one minute.

## 7 Future Work: Accounting for Uncertainty

The planner does not take into account uncertainty in the world. Moreover, the execution is open-loop in that we sense the plate's position once at the beginning of the planning and never again throughout execution. This gives us two main sources of uncertainty: uncertainty in the initial detection and uncertainty in the robot's motion. Slightly incorrect gripper positions originating from one of these tended to compound so that domains requiring more pushes (Plate3) or domains requiring long pushes (Plate1) tended to have execution failures<sup>2</sup> more often.

We are considering several approaches to dealing with uncertainty. For uncertainty created by the robot's motion, we can use a re-planning strategy [10], detecting when we have deviated from our planned path and updating the path accordingly. For perception error in the initial detection, we can use closed-loop control, checking the pressure sensor on the robot's fingertips for contact with the plate and adjusting. With more sensor input during the execution, we can also use strategies that plan in belief space [15] to try to bias the planner towards actions that also gain information about the environment.

Another method for minimizing uncertainty in execution is to choose primitives to reduce the uncertainty as much as possible. For example, we use two point pushing rather than single-point pushing. By using two point pushing, placing one side of the gripper on each of the center of friction, the robot is able to control the plate much more effectively [13]. We also implemented base and arm movement separately because the uncertainty models for the PR2 arms and base are very different,

<sup>2</sup> [http://people.csail.mit.edu/jbarry/pr2/darrrt#failure\\_modes](http://people.csail.mit.edu/jbarry/pr2/darrrt#failure_modes)

making it difficult to accurately execute synchronous trajectories. In general, separating primitives with different uncertainty models will also allow us to re-plan after executing primitives we know are likely to result in an uncertain state.

We have shown that the DARRT algorithm can solve problems requiring the use of many types of manipulation in a sixteen degree-of-freedom space. However, we also found that the trajectories returned by the planner could be successfully executed only about sixty percent of the time. We have given both an analysis of the planner and directions for future work in reducing and planning for uncertainty.

## References

1. van den Berg, J., Stilman, M., Kuffner, J., Lin, M., Manocha, D.: Path Planning among Movable Obstacles: A Probabilistically Complete Approach. In: Chirikjian, G.S., Choset, H., Morales, M., Murphey, T. (eds.) *Algorithmic Foundation of Robotics VIII*. STAR, vol. 57, pp. 599–614. Springer, Heidelberg (2009)
2. Brost, R.C.: Automatic Grasp Planning in the Presence of Uncertainty. *IJRR* 7(1) (1988)
3. Cosgun, A., Hermans, T., Emeli, V., Stilman, M.: Push Planning for Object Placement on Cluttered Table Surfaces. In: *IROS* (2011)
4. Şucan, I.A., Moll, M., Kavraki, L.E.: The Open Motion Planning Library. *IEEE Robotics & Automation Magazine* (2012) (to appear)
5. Dogar, M., Srinivasa, S.: Push- Grasping with Dexterous Hands: Mechanics and a Method. In: *IROS* (2010)
6. Dogar, M.R., Srinivasa, S.S.: A Framework for Push-Grasping in Clutter. In: *RSS* (2011)
7. Hauser, K.: Motion Planning for Legged and Humanoid Robots. Ph.D. thesis, Stanford University (2008)
8. Hauser, K., Ng-Throw-Hing, V.: Randomized Multi-Modal Motion Planning for a Humanoid Robot Manipulation Task. *IJRR* 30(6) (2011)
9. Huang, W., Mason, M.T.: Experiments in Impulsive Manipulation. In: *ICRA*, vol. 2 (1998)
10. Kaelbling, L.P., Lozano-Pérez, T.: Pre-Image Backchaining in Belief Space for Mobile Manipulation. In: *ISRR* (2011)
11. LaValle, S.M.: *Planning Algorithms*. Cambridge University Press (2006)
12. Lozano-Pérez, T., Jones, J.L., Mazer, E., O'Donnell, P.A.: *Handey: A Robot Task Planner*. MIT Press, Cambridge (1992)
13. Mason, M.T.: *Mechanics of Robotic Manipulation*. MIT Press, Cambridge (2001)
14. Okada, K., Haneda, A., Nakai, H., Inaba, M., Inoue, H.: Environment Manipulation Planner for Humanoid Robots Using Task Graph That Generates Action Sequence. In: *IEEE/RSJ IROS* (2004)
15. Platt, R., Tedrake, R., Kaelbling, L., Lozano-Pérez, T.: Belief Space Planning Assuming Maximum Likelihood Observations. In: *RSS* (2010)
16. Siméon, T., Laumond, J.P., Cortés, J., Sahlbani, A.: Manipulation Planning with Probabilistic Roadmaps. *IJRR* 23(7-8) (2004)
17. Stilman, M., Kuffner, J.: Navigation Among Movable Obstacles: Real-Time Reasoning in Complex Environments. In: *HUMANOIDS* (2004)
18. Stilman, M., Kuffner, J.: Planning Among Movable Obstacles with Artificial Constraints. *IJRR* 27(11-12) (2008)
19. Stilman, M., Schamburek, J.U., Kuffner, J., Asfour, T.: Manipulation Planning Among Movable Obstacles. In: *ICRA* (2007)



# A Constraint-Aware Motion Planning Algorithm for Robotic Folding of Clothes

Karthik Lakshmanan, Apoorva Sachdev, Ziang Xie, Dmitry Berenson, Ken Goldberg, and Pieter Abbeel

**Abstract.** Motion planning for robotic manipulation of clothing is a challenging problem as clothing articles have high-dimensional configuration spaces and are computationally expensive to simulate. We present an algorithm for robotic cloth folding that, given a sequence of desired folds, outputs a motion plan consisting of a sequence of primitives for a robot to fold the cloth. Previous work on cloth folding does not take into account the constraints of the robot, and thus produces plans which are often infeasible given the kinematics of robots like the Willow Garage PR2. In this paper we introduce a class of motion primitives that start and end in a subset of the cloth's state space. To find a sequence of primitives that achieves all required folds, the algorithm takes advantage of the partial ordering inherent in folding, and produces a time-optimal motion plan (given the set of primitives) for the robot if one exists. We describe experiments with a general purpose mobile robotic platform, the PR2, folding articles that require dragging and base motion in addition to folding. Our experiments show that (1) many articles of clothing conform well enough to the assumptions made in our model and (2) this approach allows our robot to perform a wide variety of folds on articles of various sizes and shapes.

## 1 Introduction

Robotic manipulation of 2D deformable objects is a difficult problem largely because such objects typically have infinite-dimensional configuration spaces and are too computationally expensive to simulate in the inner-loop of a motion planner.

The problem we address is as follows: Given a robot model, the shape of a piece of cloth in a spread-out configuration on a horizontal table, and a final folded configuration specified by a sequence of g-folds, output a sequence of robot motions that achieves the final folded configuration or report that none exists.

---

Karthik Lakshmanan · Apoorva Sachdev · Ziang Xie · Dmitry Berenson · Ken Goldberg · Pieter Abbeel

University of California, Berkeley

e-mail: {karts, apoorvas, ziang.xie, berenson}@berkeley.edu, goldberg@berkeley.edu, pabbeel@cs.berkeley.edu

The state-of-the-art approach is g-folding [24]. At the core of this approach is the definition of a cloth model that allows reasoning about the geometry rather than the physics of the cloth in relevant parts of the state space. Given the geometry of the cloth, their algorithm computes how many grippers are needed and prescribes motions for these grippers to achieve the final configuration, specified as a sequence of g-folds—folds that can be achieved while staying in the subset of the state space to which their geometric model applies. G-folding, however, has severe practical limitations: due to robot and environmental constraints, the gripper motions produced by g-folding are often infeasible. When that happens, the top-down approach followed by g-folding fails.

This paper presents a motion planning approach that plans directly at the level of robotic primitives rather than the gripper motions used in [24]. Given a sequence of g-folds that take the cloth from the initial to the final configuration, our approach determines whether a sequence of motion primitives exists that results in the successful execution of all specified g-folds. If so, the algorithm outputs the robot motion which brings the cloth to the final configuration.

To restrict the (otherwise unmanageably large) search space we restrict our search to a class of motion primitives, called *g-primitives*. This class of primitives consists of all motions that begin and end with the cloth in a *g-state* — a configuration of the cloth where all parts of the cloth are either horizontal or vertical. This is a broader class of motion than was allowed in [24] because the intermediate states of the cloth during execution of primitives are unrestricted. The primitives used in our experiments are of three types: performing a g-fold (including folds that allow the article to hang from the table), dragging the article along the table, and robot base motion.

To find a plan to fold a given article using a given set of g-folds, we associate a cost with each g-primitive and search for an optimal solution, returning failure if no solution exists. Our algorithm for searching over primitives consists of (1) generating a graph that captures dependencies between the folds specified in a g-fold sequence and (2) using A\* to search for a sequence of primitives that accomplishes all folds in the graph.

In our experiments the cost we associated with each g-primitive is the time it takes to execute. Hence not only does our approach allow us to find solutions in cases where g-folding would simply fail, it also enables finding better (i.e., time-optimal) solutions. Our experiments show that (1) many articles of clothing conform well enough to the assumptions made in our model and (2) this approach allows our robot to perform a wide variety of folds on articles of various sizes and shapes.

## 2 Related Work

Our work builds on g-folding [24], but does account for kinematic restrictions posed by real robots. Our work also draws from the work of Bell and Balkcom [4, 5], which deals with computing the grasp points needed to immobilize a polygonal non-stretchable piece of cloth. Relevant work in cloth folding includes dynamic towel

folding [1], cloth perception [21], and strategies to bring the cloth into a spread-out configuration where folds can be applied [20, 10, 6].

Our work is also related to methods in motion planning that search over primitives to construct robot trajectories [13, 7, 17, 23, 9, 22, 15]. Similar search-based approaches have also been developed in the graphics literature [19, 18]. However, to our knowledge no previous work has applied a search over motion primitives to the cloth folding problem.

Other relevant work has been done in physical simulation of cloth [3, 8, 12], origami folding [2], carton folding [16, 12], and metal bending [14], which use similar material models to the one presented here.

### 3 Problem Statement

The problem we address is as follows: Given a robot model, the shape of a piece of cloth in a spread-out configuration on a horizontal table, and a final folded configuration specified by a sequence of g-folds, output a sequence of robot motions that achieve the final folded configuration or report that none exists.

We assume a coordinate frame in which gravity acts in the downward vertical ( $-z$ ) direction and a rectangular table in the horizontal ( $xy$ ) plane. We assume the article of clothing can be fully described by a 2D polygon (convex or non-convex) with  $n$  vertices, initially lying on the horizontal surface in a fully spread-out, known configuration. Except for the table surface, there are no obstacles in the environment.

We define  $S$  as the state space of the system (i.e. the robot and cloth). A state  $s \in S$  consists of a robot state and a cloth state.

We are also given a robot model, which specifies the number and width of grippers and the inverse kinematics for each gripper. It is assumed that the grippers are not capable of distinguishing between layers of cloth and will grasp all layers of the stack at each grip location. For example, the Willow-Garage PR2 we use in our experiments has 2 grippers.

In this paper we focus on robots with a mobile base and two manipulators (like the PR2), though our framework is not limited to this type of robot. Focusing on this robot model, a robot configuration  $q$  consists of the  $(x, y, \theta)$  of the base, a set of joint angles for each arm  $\{\alpha_1, \dots, \alpha_{2N}\}$  (where each arm has  $N$  joints), and booleans  $q_{grip}^L, q_{grip}^R$  indicating whether each gripper is open or closed. The configuration of the cloth is represented as a set of polygons.

Since we are building on the g-folding work in [24], we inherit the assumptions of the g-folding model:

1. The cloth has infinite flexibility. There is no energy contribution from bending.
2. The cloth is non-stretchable. No geodesic path lengths can be increased.
3. The cloth has infinite friction with itself.
4. The cloth has zero thickness.
5. The cloth is subject to gravity.
6. The cloth has no dynamics, i.e. it is moved quasi-statically.

7. If the cloth is held by a number of grippers, and one or more grippers release the cloth, no point of the cloth will move upwards as a result of gravity and internal forces within the cloth. (The *downward tendency assumption* from [24], which works well for most everyday articles of clothing. However, it does not hold for a family of objects known as pinwheels, as shown in [4].)

We extend the model used in g-folding to allow the robot to perform more types of motions (such as dragging and letting parts of articles hang off the table). We add the following assumptions:

8. The weight and density of the cloth are known.
9. There is a known uniform coefficient of static friction  $\mu$  between the cloth and the surface on which it lies. The robot can exert enough force on the cloth to overcome this friction. (Note that this assumption is different from [24], which assumes that this friction is infinite.)
10. If a part of the cloth on a horizontal surface is gripped at one or more points, and all such points are moved horizontally at the same velocity, then the distance of any point of the cloth from a gripper, measured in the  $xy$  plane, will not decrease. Intuitively, this means that when a cloth is dragged across a surface, no point of the cloth will move towards a gripper due to internal forces.

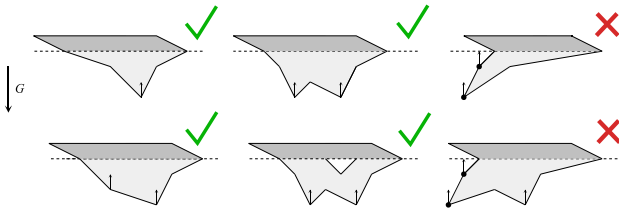
## 4 g-States

The key to our search strategy is to restrict the movements of the robot to start and end in a *g-state*. A state  $s \in S$  is a g-state if and only if a part (possibly empty) of the cloth is horizontal and a part (possibly empty) is vertical, and the cloth is not in motion and will not move independently. We define a *baseline* as a line that separates a horizontal part of the cloth from a vertical part.

Given the problem description above, there are two ways for parts of the cloth to be vertical if the cloth is in a g-state: 1) The cloth is held up by robot grippers and/or 2) the cloth is hanging off the edge of the table. In the first case, the cloth will not move as long as Theorem 1 from [24] (restated below for convenience) holds:

**Theorem 1.** *In our material model, a vertically hanging cloth polygon is immobilized when every convex vertex of the cloth at which the negative gravity vector does not point into the cloth polygon is fixed (i.e. be held by a gripper or be part of the baseline).*

In the second case, as long as the net horizontal force on the part of the cloth lying on the table surface can be countered by friction, points lying on the table surface (including those on the baseline) will not move. This is because such points cannot move in the plane due to friction, and they will not move upward per the downward-tendency assumption. Thus to determine if the cloth will slip over the edge, we check the following two conditions:



**Fig. 1** Examples of parts of cloth hanging off the table in various configurations. The dotted line is the table edge (baseline). In order for the cloth to be hung, all convex vertices not at the baseline should have the negative gravity vector (small arrows) point into the cloth. These vertices are indicated by dots. In the configurations shown on the right, there are convex vertices where the negative gravity vector does not point into the cloth and which are not on the baseline, hence the cloth is not immobilized.

1. The friction between the table surface and the part of the cloth on the table must be sufficient to counter the force exerted by gravity on the vertically hanging parts of the cloth.
2. All convex vertices of the hanging part at which the negative gravity vector does not point into the cloth are part of the table edge (which is also the baseline), from which the part of the cloth hangs. This prevents the vertical part of the cloth from moving, according to Theorem 1. Examples are shown in Figure 1.

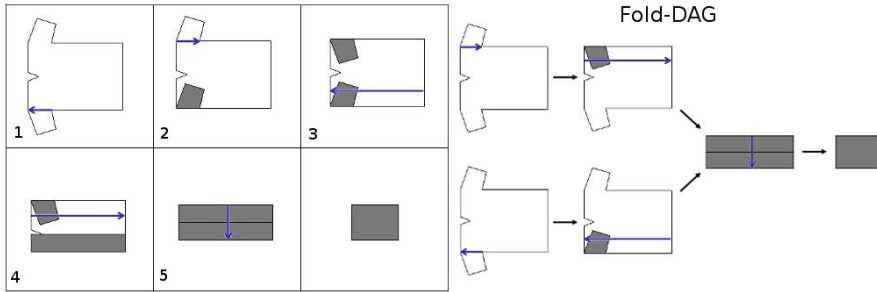
A *convex vertex* is defined as a vertex of the cloth polygon that is also a vertex of its convex hull.

## 5 Algorithm Formulation

We use a search-based procedure to find a sequence of folding primitives that results in the successful completion of all the requested g-folds, if possible. Our planning algorithm is comprised of two components: (1) The creation of a Fold-DAG. (2) A search over motion primitives.

### 5.1 Fold-DAG

The user inputs a desired sequence of g-folds. However, it may not be necessary to execute the folds in the same order as given, as some folds may be performed independent of the previous folds in the sequence. For example, consider the g-fold sequence shown in Figure 2. If the robot were to perform the folds in the same order, it might have to, for example, complete fold 1, then move around to complete fold 2 before moving back to complete fold 3. However, we can see that either the g-fold in part 1 or the one in part 2 can be performed first. The g-fold in part 3 only needs part 1 to be completed, and the g-fold in part 4 only needs part 2. Capturing the dependencies between the requested g-folds can potentially lead to



**Fig. 2** The Fold-DAG generated from a set of user defined g-folds. Left: A user defined sequence of g-folds for a T-shirt. Right: The sequence when it has been transformed into a Fold-DAG. The corresponding T-shirt configurations are also shown for clarity.

quicker solutions. For example, the robot can choose to complete 1 and 3 before moving around to complete 2 and 4.

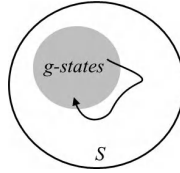
We use a directed acyclic graph (DAG), which we call the Fold-DAG, to capture dependencies between the folds specified in a folding sequence, i.e. which g-folds need to be completed before a particular g-fold can be performed. The nodes  $V$  of this graph  $G = (V, E)$  are the specified folds and the edges  $E$  specify dependencies.  $E$  is produced in two steps: First, we add an edge from node  $v_a \in V$  to node  $v_b \in V$ ,  $e(v_a, v_b)$  if the folded part of the cloth specified in node  $v_b$  overlaps the folded part of the cloth specified in  $v_a$  for all pairs of nodes in the graph. We then remove redundant dependencies by eliminating any  $e(v_a, v_b)$  if there is a path from  $v_a$  to  $v_b$  that consists of more than one edge. For example, Fig 2 shows an input g-fold sequence and its Fold-DAG.

When executing a sequence of folds, we are implicitly traversing the Fold-DAG. We can determine a list of *available* g-folds for each cloth configuration depending on the g-folds that have been performed previously. A node  $v$  is *available* if all nodes in  $V_i = \{v_i \in V | e(v_i, v) \in E\}$  have been executed, i.e. if all nodes pointing to  $v$  have already been performed.

### 5.2 Motion Primitives

The goal of the search process is to produce a series of motion primitives that achieves the given sequence of g-folds. If we disregard the kinematic constraints of the robot, this process is trivial since all g-folds can be performed directly. However, taking into account the constraints, a given g-fold may not be possible due to, for instance, the grip locations being too far from the robot. The search process searches over a given set of motion primitives, attempting to achieve the given g-fold sequence. The search algorithm uses the Fold-DAG to guide the search to the goal configuration (where all requested g-folds have been performed).

In order to make the search process computationally feasible, we must ensure that the search space we consider is not unmanageably large. Allowing any arbitrary



**Fig. 3** The set of states  $S$  and the subset of  $g$ -states. The black path represents the sequence of states traversed by an example  $g$ -primitive.

motion to be considered would cover  $S$ , but it would also produce an unmanageable search space and would require simulations of the cloth dynamics (which are computationally expensive and sometimes inaccurate). We thus restrict the search to a class of motion primitives we call *g-primitives*.

Intuitively, a  $g$ -primitive is any motion primitive that both starts and ends with the system in a  $g$ -state. Formally, a *motion primitive* is a function  $f : S \rightarrow T$ , where  $T$  is the set of all robot trajectories.

**Definition 1.** A *g-primitive* is a motion primitive whose domain and range are restricted:  $f : S' \rightarrow T'$ , where  $S' \subseteq S$  is the set of all  $g$ -states and  $T' \subseteq T$  is the set of all trajectories that end with the system in a  $g$ -state.

Using  $g$ -primitives allows us to search over a manageable subspace of  $S$  while also allowing us to perform practical tasks with the cloth. It is important to note that there are no restrictions on the intermediate states of the robot and cloth when performing a  $g$ -primitive (i.e. the cloth and robot can move arbitrarily) as long as the start and end conditions are met (see Figure 3).  $g$ -primitives are also a broader class of motion than was allowed in [24], where all intermediate states of a fold were required to be  $g$ -states.

The  $g$ -primitives used in this paper are described in Section 6, however these are only examples of what possible primitives could be used, and by no means constitute the entirety of the class of  $g$ -primitives.

### 5.3 Search over Primitives Using $A^*$

Given vertices of the cloth, the robot model, and the Fold-DAG we can compute a time-optimal motion sequence of primitives for the robot to execute in order to reach the desired final configuration. We define a time-optimal sequence as one which takes the least time to execute on the robot.

We used the  $A^*$  search algorithm to find a sequence of  $g$ -primitives, which is a heuristic-guided search that guarantees solutions of lowest cost. To apply  $A^*$ , we require functions that compute whether a state is a goal, the successors of a state, the cost of reaching a state, and a heuristic estimate of the cost to reach the goal from a given state.

We define a goal state as a state which has no  $g$ -folds remaining in the DAG. For a given state, we generate the successors by applying all the primitives in our

set of primitives. If a given primitive is infeasible (due to, for instance, reachability constraints), it does not generate a successor. Since we want the search algorithm to return the plan of least execution time, the sum of costs along the path starting at the starting state is a measure of the time taken by the robot to perform all primitives that constitute the path. The primitive-specific cost functions are given in Section 7.

Our A\* heuristic uses a relaxation of the problem which ignores robot constraints and approximates the time taken for the robot to perform all remaining folds in the DAG at the given state. Since robot constraints are ignored, only gripper trajectories are considered and the method of computation is identical to that used to find the time taken to move the grippers in the cost calculation. This also ensures that the heuristic is admissible.

## 6 Primitives for Robotic Cloth Manipulation

Any primitive that starts and ends in a g-state is allowed. We provide three examples of primitives; g-fold, g-drag, and base motion, including the motivation for and a formal description of each.

### g-Fold

As defined in [24], a *g-fold* is specified by a directed line segment in the plane whose endpoints lie on the boundary of the cloth polygon. The segment partitions the polygon into two parts, one to be folded over another. A g-fold is successfully achieved when the part of the polygon to the left of the directed line segment is folded across the line segment and placed horizontally on top of the other part, while maintaining the following property:

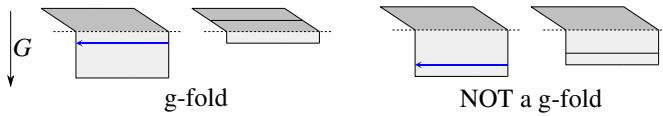
*At all times during a folding procedure, every part of the cloth is either horizontal or vertical, and each vertical part of the cloth is held either by grippers or by the edge of the table such that it is immobilized.* (See Figure 2 in [24].)

The definitions of “blue” and “red” g-folds from [24] remain unchanged—a blue g-fold is specified by a line segment partitioning the polygon formed by the silhouette of the stacked geometry into two parts, and is successfully achieved by folding the (entire) geometry left of the line segment. A red g-fold is similarly specified, but only applies to the (potentially stacked) geometry that was folded in the previous g-fold.

The g-fold primitive is the execution of a user specified g-fold (“red” or “blue”). g-folds involving parts of the cloth that are not hanging are performed in the same way as described in section 4 of [24]. A fold for parts of the cloth that are hanging can be performed when the following conditions hold true:

1. Theorem 1 must hold true, and for this the robot must have a sufficient number of grippers, which must be able to reach and grip the necessary vertices throughout the motion required to complete the fold.



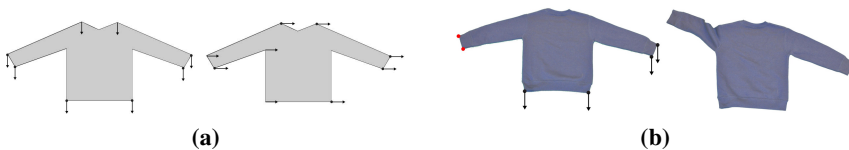


**Fig. 4** Left: A valid g-fold performed on the hanging part. The dotted line shows the table edge. The hanging part is highlighted in light gray. The cloth does not move in the resulting configuration due to friction between the cloth and table. Right: A fold on the hanging part of the cloth which is not a g-fold because the cloth will move after it is released.

2. A g-fold parallel to the table edge along which the cloth hangs is permitted only if the resulting cloth configuration is a g-state (see Fig. 4 for an example where this condition is violated).

### g-Drag

Kinematic restrictions on the robot arms may yield points on the cloth that need to be gripped in order to perform a g-fold unreachable from the robot’s current pose. This motivates the action of dragging the article across the table, in order to make such points reachable. The g-drag primitive is defined as the translation of the part of the article on the  $xy$  plane of the table. A g-drag is determined by the direction of dragging and distance through which the article is dragged. See Figure 5 for examples of g-drags.



**Fig. 5** (a) Two example g-drags of a long sleeve shirt in two directions. For translation without deformation of the article all convex vertices on the table surface at which the negative friction vector (small arrows) does not point into the cloth must be grasped. Grasp locations are indicated by the dots. (b) A drag of a long sleeve shirt which is not a g-drag. Red convex vertices are not gripped, which results in deformation of the left sleeve (shown on the right).

In order to ensure that the part of the cloth on the surface of the table is translated without any deformation, the following theorem must hold true:

**Theorem 2.** *In our material model, a cloth polygon resting on a horizontal surface can be translated without deformation if (1) every convex vertex of the cloth at which the negative vector of friction (between the table and the cloth) does not point into the cloth polygon is held by a gripper, and (2) all such vertices move at the same velocity.*

*Proof.* Our proof is based on the proof of Theorem 1 in [24], where van den Berg et al. prove that a vertically hanging cloth polygon can be immobilized by fixing every convex vertex of the cloth at which the negative gravity vector does not point into the cloth. Consider a cloth polygon being dragged across a horizontal surface with uniform friction. The motion does not induce deformation of the cloth if at any time during the drag, all points of the polygon have the same velocity as the gripped points. Let us pick one of these gripped points, and, without loss of generality, consider a frame of reference whose origin is at this point, such that the table surface is the  $xy$  plane, and the positive  $x$  axis is the direction along which the cloth is dragged. The origin moves with the gripped point as the cloth is dragged. Uniform friction then acts along the negative  $x$  axis. Using assumption 10, we know that no point of the cloth will move in the positive  $x$  direction due to internal forces.

Let us define a *leading string* of the polygon as a maximal sequence of edges of which the extreme vertices are convex vertices of the polygon, and no part of the polygon has a greater  $x$ -coordinate than these edges. A given polygon  $P$  can have multiple leading strings, but has at least one.

The proof now proceeds in a similar fashion to the proof in [24], which presents a recursive proof where at every step in the recursion, all *upper strings* of a polygon  $P$  are proven immobilized, and thereby every point of  $P$  that can be connected to an upper string by a vertical line segment that is fully contained within  $P$  is proven immobilized. The notion of “upward” in their case is analogous to the  $+x$  direction in our case, and the “downward” force of gravity can be replaced by friction in the  $-x$  direction. Leading strings take the place of “upper strings.” Then, in a similar fashion, we can prove that all leading strings are immobilized. The proof in [24] uses the downward tendency assumption to show that 1) non-convex vertices of a gripped upper string will not move upward, 2) every point of  $P$  that can be connected to an upper string by a vertical line segment that is fully contained within  $P$  cannot move upward. Assumption 10 serves the same purpose in our case, ensuring that the corresponding points do not move in the  $+x$  direction. The rest of the proof follows accordingly.  $\square$

The possible distances an article can be dragged form a continuous space, but we discretize them to meet the requirements of our search algorithm. More details are given in the experiment description.

## Move

The robot is allowed to move around the table, which makes different portions of the article reachable. The space of robot base poses is continuous and we manually select a set of base poses from this space that provides good coverage of the table surface for the arms. The allowed poses for our robot’s base are shown in Figure 7a.

## 7 Costs for the Primitives

The cost we associated with each *g*-primitive is the time it takes to execute. Our motion primitives can be decomposed into sequences of base motion and arm movement, and the cost is computed by summing the execution time for each motion.

A *g-fold* consists of moving the arm(s) to the grip configuration, performing the fold, and releasing the cloth (if blue fold), while moving the base backward and forward as necessary to reach the computed gripper locations.

A *g-drag* consists of moving from the initial configuration to the grip configuration, moving the base along the direction of the *g-drag*, and releasing the cloth (if the next fold is not a red fold).

For the above two primitives it is sometimes necessary for the robot to back away from the table in order to make the target grippoints reachable (this is due to the PR2's limited reachability for points close to its body). Thus if the grip points are too close to the robot, we include a backward base motion in the primitive's sequence of motions as well.

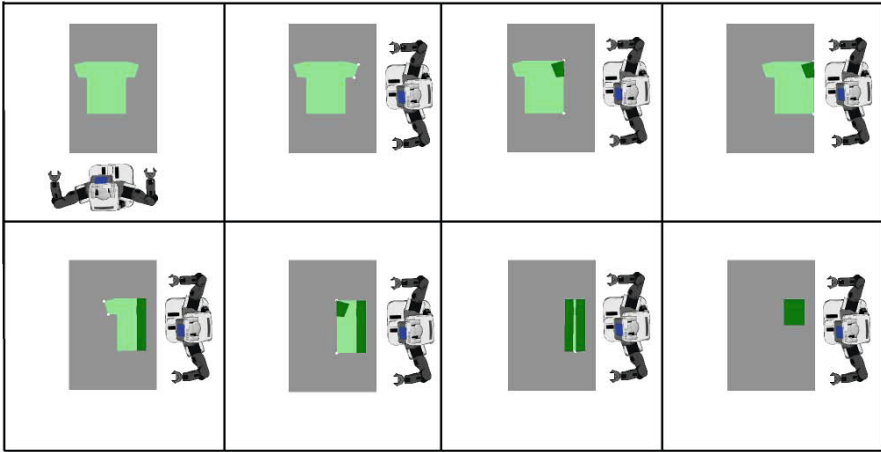
The *move* primitive moves the base between pre-specified positions by turning in place, driving to the target position, and turning in place again to reach the desired rotation.

The costs for the above primitives are computed by summing the execution time necessary to complete each component of the primitive (i.e. reaching and base movement). Details about setting the execution time for the arms is discussed in Section 8. The time necessary to execute base motions is computed assuming constant linear and angular speed.

## 8 Planning and Simulation Experiments

We used a Willow Garage PR2 robotic platform [15] and performed experiments both in simulation and on the physical robot. For the *Move* primitive, our robot could move between 5 predefined positions around the table as shown in Figure 7a. For the *Fold* primitive, the computed grip-points were often too close to the robot's body for it to reach them without moving its base. To overcome this, the robot was allowed to move its base backward until the points could be reached, thereby enhancing the robot's reachability along the *x*-axis, (as defined for each position in Figure 7a). For the *Drag* primitive, we only allowed the robot to drag the article towards itself. The drag distance was discretized, and drags in increments of 10 cm (up to the length of the article) were considered. Grip points were computed by first computing grip locations as prescribed by the primitives while assuming point grippers. Then, following [24], we accounted for gripper width (and some cloth stiffness) to find a reduced number of required grip-points.

We used 3 seconds for the input execution time for all arm motions on the PR2. We found that going below this time decreases the path-following accuracy of the physical robot, thus resulting in imperfect folds.



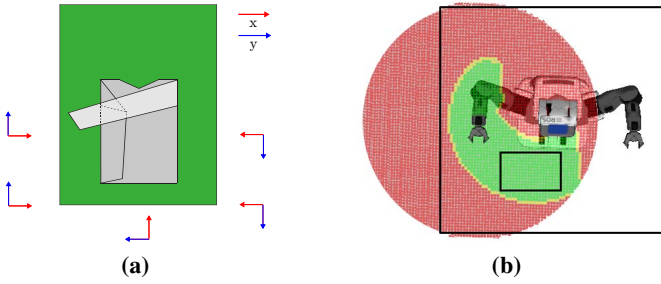
**Fig. 6** Solution for the T-shirt using the g-folds of Figure 2 as input

**Table 1** Simulation Results. L is the number of primitives in the solution path, C is the cost of the path in seconds. N expand/explore is the number of nodes expanded/explored. T is the total search time of algorithm.

Article	# Folds	L	C(s)	N Expand	N Explore	T(s)	IK time(s)	IK Queries	Overhead(s)
T-shirt	5	7	48.12	16	84	6.29	1.14	3512	0.007
Jeans	2	3	35.01	13	116	4.07	1.44	776	0.008
Shirt	7	9	76.94	84	491	28.29	3.96	5911	0.040
Tie	3	4	27.90	24	256	6.06	1.59	593	0.013
Scarf	2	3	18.90	14	177	4.94	1.37	593	0.011
Vest	2	2	24.62	8	61	1.91	0.41	383	0.004
Skirt	3	4	29.63	60	535	15.37	5.42	10087	0.026
Big Towel	3	4	27.91	22	232	8.47	3.01	6675	0.013
Hand Towel	3	3	27.35	18	161	5.04	1.90	400	0.007

We experimented with nine common clothing articles listed in Table 1 in simulation. The folding sequence is given as input to the algorithm. An example of the solution sequence for a T-shirt is shown in Figure 6.

We used the `ikfast` module provided by OpenRAVE [11] in order to determine if the robot can reach a given point. This can be a time consuming operation because it involves collision-checking and searching over the free joint of the 7 DOF manipulator, and IK queries are made often by the search algorithm. In order to decrease the number of times `ikfast` is called, we created a fast approximation to IK that we query before calling `ikfast`: offline, we define two bounding boxes for each robot arm such that any point within the inner box is reachable by that arm's wrist and any point outside the larger outer box is unreachable from the current robot base position. For each arm, the inner box is pre-computed by making IK queries



**Fig. 7** (a) Allowed robot base positions around the table for the Move primitive. (b) Bounding boxes used to decrease the number of IK queries on the PR2.

on a grid of 3D points around the robot with a 1cm discretization. We then check if the wrist can reach each point in the grid. We only consider points that are above the table and less than 1m above the floor, as this region encompasses most of the points whose reachability the planner needs to check. See Figure 7b. The shape of reachable 3D points is not cuboidal, but we inscribe a cuboid in this region to create the inner box for ease of checking during the search. The outer box is constructed based on the fact that the PR2 cannot reach any point that is further away than the length of its arm at full stretch. For our experiments, any point farther than 0.75m from the center of the base in any direction lies outside the outer box. Whenever the planner makes an IK query, we first check if the translation component of the query is within the inner box for the relevant arm. If so, we immediately return that the point is reachable. If the point is outside the outer box, we immediately declare that it is unreachable. If the point is contained in the outer box but is not in the inner box, we query `ikfast`. This enabled us to reduce the time spent on IK queries by about 50%, on average.

In order to make the generated plans robust to robot execution error (for example, dragging by less than the desired amount), we introduce the concept of “IK

**Table 2** Simulation Results with IK comfort radius of 3 cm. See symbol definitions in Table 1.

Article	# Folds	L	C(s)	N Expand	N Explore	T(s)	IK time(s)	IK Queries	Overhead(s)
T-shirt	5	7	48.12	40	319	55.08	39.77	9950	0.010
Jeans	2	3	35.01	13	60	1.64	1.25	272	0.006
Shirt	7	9	76.94	134	920	121.70	85.99	26867	0.034
Tie	3	4	27.90	26	329	20.75	11.89	4387	0.007
Scarf	2	3	18.90	16	197	18.71	14.01	4950	0.004
Vest	2	2	24.62	14	101	9.24	6.82	2073	0.007
Skirt	3	4	29.63	22	201	19.61	14.87	4432	0.006
Big Towel	3	4	27.91	24	284	35.63	26.23	6921	0.016
Hand Towel	3	3	27.35	17	175	20.80	15.25	4214	0.004

comfort.” We only declare a point reachable if both the point and four points on the circumference of a circle of a set radius, centered at the given point are reachable. Also, if the gripper fails to grab the cloth at the target point during execution on the robot, it tries to grab other points that lie within the comfort radius of the target before failing. For our trials, we found that an IK comfort radius of 3 cm resulted in robust execution.

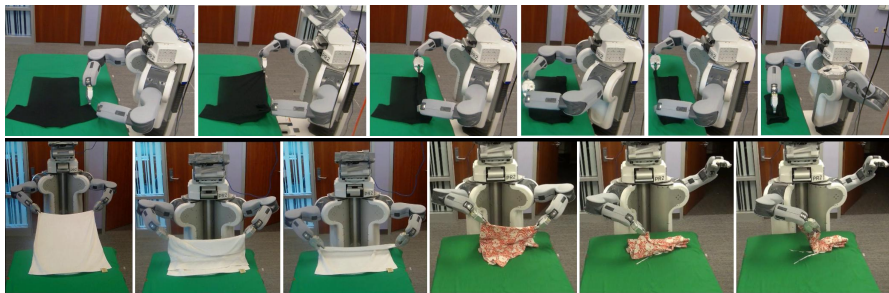
### 9 Physical Experiments

We used a rectangular table with a soft working surface, so that the relatively thick grippers can easily get underneath the cloth. At the beginning, the robot can always see the entire clothing article in a known, fully spread-out configuration. The robot detects the green table surface using hue segmentation. The cloth polygon is then detected as described in [21]. We used the PR2 2D Navigation package in ROS to move the robot between the 5 base positions. The robot performed Drags by moving the base backward, and then correcting for any undershoot in base motion by using its grippers to drag the cloth further.

For several of the articles, we executed the generated plans multiple times on the robot. Table 3 shows the results of our runs. Several snapshots from folding a T-shirt, towel, and skirt are shown in Figure 8. Videos of the executions are posted at <http://rll.berkeley.edu/iser2012-folding>.

**Table 3** Success rates and timing results of physical robot executions. Vision Time includes time for initial detection as well as for visual feedback during execution. The times are averaged over all successful runs.

Clothing item	Success Rate	Av. Execution Time	Av. Vision Time	Av. Total Time
T-shirt	3/3	218s	60s	278s
Large towel	2/3	78s	20s	98s
Skirt	3/3	91s	20s	111s
Jeans	3/4	150s	55s	205s



**Fig. 8** T-shirt, towel and skirt folding sequences executed by PR2

The PR2 uses visual feedback to correct for execution errors. The position of the current cloth model is updated after every g-drag or Move primitive, by fitting the model to the observed contour of the cloth on the table surface, assuming that the cloth only undergoes translation and not deformation. The expected translation of the cloth is used to initialize the position of the model, following which the optimization proceeds as described in [21]. After each g-fold, the performed fold is detected and the model is updated as described in [21].

As illustrated by our success rates on the various clothing articles, our method shows high reliability on real cloth, even if it does not perfectly conform to our assumptions. For example, jeans and towels clearly violate the zero thickness assumption and the pleats of the skirt are not taken into consideration by our cloth model. Regardless we are able to achieve high success rates on these articles.

Our failures typically arise from errors in robot execution, particularly base motion. If the base does not move by the desired amount, a grip point might become unreachable. The introduction of IK comfort along with making the arms correct for base motion undershoot during a drag greatly reduces the number of such failures.

Our experiments show that the planned execution times typically underestimate the real execution times observed with the PR2. This is because the costs for each primitive used in the planning phase are highly idealized. A large part of this discrepancy can be attributed to the Move primitive. The 2D navigation package causes the robot to stop multiple times during the move. The planner also ignores certain other behaviors of robot execution. For example, the PR2 sometimes fails to grab the cloth on the first attempt, and would need to move the gripper to regrasp the cloth. Additionally, the planner assumes a constant velocity for base movement, while the robot actually spends more time accelerating and decelerating than at its full speed.

We are currently investigating ways to overcome these limitations. If the PR2 fails to grasp the article on its first attempt, it will try to grab up to 8 other points within the IK comfort radius until it grasps the cloth. The discrepancy in the computed cost of Moves and the real times may be reduced by eliminating the constant velocity assumption and averaging the actual times taken to move from one base position to another over multiple runs.

## 10 Conclusion

In conclusion, we described a motion planning algorithm for robotic cloth folding, enabling us to avoid computationally expensive physics simulations while taking into account kinematic constraints. We presented examples of cloth manipulation primitives that allow the robot to perform a set of user defined g-folds using our simplified cloth model. Our search algorithm allowed the robot to choose a sequence of primitives to perform all given folds in the shortest possible time (given the available primitives). At the core of our method is the consideration of real robot limitations. Our experiments show that (1) many articles of clothing conform well enough to the assumptions made in our model and (2) this approach allows our robot to perform a wide variety of folds on articles of various sizes and shapes.

## References

1. Balaguer, B., Carpin, S.: Combining imitation and reinforcement learning to fold deformable planar objects. In: IROS (2011)
2. Balkcom, D., Mason, M.: Robotic origami folding. *IJRR* 27(5), 613–627 (2008)
3. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: SIGGRAPH (1998)
4. Bell, M.: Flexible object manipulation. PhD Thesis, Dartmouth College (2010)
5. Bell, M., Balkcom, D.: Grasping non-stretchable cloth polygons. *IJRR* 29, 775–784 (2010)
6. Bersch, C., Pitzer, B., Kammel, S.: Bimanual robotic cloth manipulation for laundry folding. In: IROS, pp. 1413–1419 (September 2011)
7. Branicky, M.S., Knepper, R.A., Kuffner, J.J.: Path and trajectory diversity: Theory and algorithms. In: ICRA (2008)
8. Bridson, R., Fedkiw, R., Anderson, J.: Robust treatment of collisions, contact, and friction for cloth animation. In: SIGGRAPH (2002)
9. Cohen, B.J., Chitta, S., Likhachev, M.: Search-based planning for manipulation with motion primitives. In: ICRA, pp. 2902–2908 (May 2010)
10. Cusumano-Towner, M., Singh, A., Miller, S., O'Brien, J., Abbeel, P.: Bringing clothing into desired configurations with limited perception. In: ICRA (2011)
11. Diankov, R., Kuffner, J.: Openrave: A planning architecture for autonomous robotics. Technical Report CMU-RI-TR-08-34, Robotics Institute, Pittsburgh, PA (July 2008)
12. Fahantidis, N., Paraschidis, K., Petridis, V., Doulgeri, Z., Petrou, L., Hasapis, G.: Robot handling of flat textile materials. *IEEE Robotics and Automation Magazine* 4(1), 34–41 (1997)
13. Green, C., Kelly, A.: Toward optimal sampling in the space of paths. In: ISRR (2007)
14. Gupta, S., Bourne, D., Kim, K., Krishnan, S.: Automated process planning for robotic sheet metal bending operations. *Journal of Manufacturing Systems* 17, 338–360 (1998)
15. Hauser, K., Bretl, T., Harada, K., Latombe, J.-C.: Using motion primitives in probabilistic sample-based planning for humanoid robots. In: Akella, S., Amato, N.M., Huang, W.H., Mishra, B. (eds.) *Algorithmic Foundation of Robotics VII*. STAR, vol. 47, pp. 507–522. Springer, Heidelberg (2006)
16. Kanade, T.: A theory of origami world. *Artificial Intelligence* 13(3), 279–311 (1980)
17. Knepper, R., Mason, M.: Empirical sampling of path sets for local area motion planning. In: ISER. Springer (2008)
18. Lau, M., Kuffner, J.J.: Behavior planning for character animation. In: SIGGRAPH (2005)
19. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. In: SIGGRAPH (2002)
20. Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., Abbeel, P.: Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In: ICRA (2010)
21. Miller, S., Fritz, M., Darrell, T., Abbeel, P.: Parametrized shape models for clothing. In: ICRA (2011)
22. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: learning and generalization of motor skills by learning from demonstration. In: ICRA (2009)
23. Pivtoraiko, M., Knepper, R.A., Kelly, A.: Mobile Robot Motion Planning. *Journal of Field Robotics* 26(3), 308–333 (2009)
24. van den Berg, J., Miller, S., Goldberg, K., Abbeel, P.: Gravity-based robotic cloth folding. In: Hsu, D., Isler, V., Latombe, J.-C., Lin, M.C. (eds.) *Algorithmic Foundations of Robotics IX*. STAR, vol. 68, pp. 409–424. Springer, Heidelberg (2010)



# **Part IX: ISER Session Summary on “Applications to the Life Sciences”**

Jaydev P. Desai

University of Maryland, College Park

## **Session Summary**

The session on “Applications to the Life Sciences” consisted of papers from several different areas ranging from biological microrobots to robotic system for breast biopsy under continuous magnetic resonance imaging (MRI). There were a total of five papers presented in this session after Robert D. Howe provided the introduction to this session. Yang et al. in the paper titled: “Towards the Development of a Master-Slave Surgical System for Breast Biopsy under Continuous MRI”, describe their initial work on the development of a teleoperated master-slave surgical system under MRI guidance. MRI compatibility results of the slave robot are also presented in this paper and the slave robot has been shown to be MRI compatible. Kesner and Howe’s paper on “Motion Compensated Catheter Ablation of the Beating Heart Using Image Guidance and Force Control” discusses the implementation of a robotic catheter system, which uses force control and 3D ultrasound for image guidance for interventional procedures. They show that the robotic catheter with the ablation electrode was able to apply the force more consistently on the target compared to a manual catheter. Shojaei-Baghini and Sun in the paper on “Robotic Micropipette Aspiration of Biological Cells”, present their work on mechanical characterization of single cells through micropipette aspiration. The micropipette aspiration process is automated using visual servoing techniques and they show that the system is efficient and independent of the operator expertise. The paper on “Quantitative Analysis of Locomotive Behavior of Human Sperm Head and Tail”, by Liu et al., discusses an algorithm to track multiple sperms, including their head and tail, instead of just the sperm head. They found that there was a relationship between the sperm head velocity and the sperm tail beating amplitude and that the sperms with higher velocity generally did bind to hyaluronic acid. Finally, in the paper by Khalil et al., on “Characterization and Control of Biological Microrobots”, characterization and control of magnetotactic bacterium (MTB) is presented. They show that they are able to achieve position tracking of MTB.

# Towards the Development of a Master-Slave Surgical System for Breast Biopsy under Continuous MRI

Bo Yang, U-Xuan Tan, Alan McMillan, Rao Gullapalli, and Jaydev P. Desai

**Abstract.** Magnetic Resonance Imaging (MRI) provides superior soft-tissue contrast. But the strong magnetic field inside the MRI bore and the limited scanner bore size restricts direct means of breast biopsy under real-time imaging. Current blind targeting approach based on MR images obtained a priori sometimes requires multiple needle insertions if the tool tip position is compromised. A MRI-compatible robot that can be teleoperated is thus desired to take advantage of the real-time MR imaging and avoid multiple needle insertions. In this paper, we present our initial work on the development of a master-slave surgical system. The MRI-compatible slave robot is actuated with five pneumatic cylinders and one piezo motor and operates inside the MRI bore. The master robot provides an intuitive manipulation platform for the physician. The MRI experiment shows that the slave robot does not induce visually-detectable distortion in the MR images and can be safely operated inside the MRI.

## 1 Introduction

### 1.1 Motivation and Related Works

Nearly one in eight American women will be diagnosed with breast cancer during their lifetime and that accounts for nearly one in three cancers diagnosed in

---

Bo Yang · Jaydev P. Desai  
Robotics, Automation, and Medical Systems (RAMS) Laboratory,  
Maryland Robotics Center, Institute for System Research,  
University of Maryland, College Park, MD, USA  
e-mail: {yangbo, jaydev}@umd.edu

U-Xuan Tan  
Singapore University of Technology and Design, Singapore  
e-mail: uxuan\_tan@sutd.edu.sg

Alan McMillan · Rao Gullapalli  
University of Maryland School of Medicine, Baltimore, MD, USA  
e-mail: {amcmillan, rgullapalli}@umm.edu

American women, excluding cancers of the skin [1]. Breast biopsy is the gold standard to confirm if the suspicious tissue is cancerous or not; however, it requires a reliable and accurate means of guidance so that the tissue sampled is actually from the suspicious region. Magnetic Resonance Imaging (MRI) provides superior soft-tissue contrast to identify the tumor boundary while avoiding harmful ionizing radiation of Computed Tomography (CT) or the poor image quality of Ultrasound. Studies have shown that MRI is able to detect cancer in the contralateral breast that is otherwise missed by mammography and clinical examination [2] and it can improve the ability to diagnose Ductal Carcinoma in situ (DCIS) with high nuclear grade [3]. Hence, MRI has become a popular diagnostic tool for breast cancer and is a competitive imaging candidate for use in image-guided interventions.

Though MRI provides superior soft-tissue contrast compared to other imaging modalities, it severely restricts the instruments allowable in the MRI scanner due to the high magnetic field it employs. Furthermore, the limited MRI bore size (generally 70cm in diameter) prohibits the physician from performing biopsy procedures while the patient is being scanned inside the MRI bore. Current breast biopsy procedures using MRI guidance require a needle to be inserted outside the MRI scanner bore based on MR images acquired a priori to place the guiding insulating sheath for the biopsy needle/gun [4]. Once the needle insertion is completed, the patient is scanned again to check if the needle has been inserted to the correct location for biopsy. Such blind targeting approach can induce needle tip positioning error thereby requiring repeated needle insertion. This is one of the primary drawbacks of the current MR-guided breast biopsy procedures. Hence, it is desired to develop a teleoperated master-slave surgical system that can perform needle insertion under continuous MRI guidance to improve needle positioning accuracy and thereby reducing sampling errors.

With similar motivation, MRI-compatible robots for other cancer treatments have also been developed and some of them have focused heavily on prostate therapy [5, 6, 7, 8, 9, 10]. MRI-compatible robotic systems for other applications have also been reported such as for brain tumor removal developed by Ho et al [11]. For breast cancer treatment, only a few attempts have been made to help perform breast biopsy. A 6 degree-of-freedom (DOF) robotic system has been developed in [12] that can carry out breast biopsy under MRI though its needle approach direction is limited. Larson et al have implemented a MRI-compatible robotic device for interventions in the breast that is remotely controlled by ultrasonic motors but suffers from excessive backlash error [13]. Kokes et al [14] have proposed a teleoperated needle driver robot with haptic feedback for radiofrequency ablation of breast tumors with only 1 DOF. All these works are mainly on the development of the MRI-compatible slave robot, instead of implementing a complete master-slave surgical system.

## ***1.2 Technical Approach***

To perform needle insertion for breast biopsy under continuous MRI guidance, a teleoperated master-slave surgical system is to be developed. The slave robot should

be MRI-compatible to operate inside the MRI bore, which is ensured by employing non-magnetic materials and MRI-compatible actuation techniques. Non-metallic materials such as plastic and ceramic are more preferable for applications inside the MRI, and non-magnetic metal such as brass, aluminum, and titanium can also be used for parts that require higher stiffness and strength, as long as they are of small volume and placed distant away from the scanning center to minimize interference from the induced eddy current during MR imaging. Various MRI-compatible actuation techniques have been reviewed in [15], and pneumatics stands out since it is cleaner and easier to maintain compared to hydraulics and leads to negligible MR image signal-to-noise ratio (SNR) loss compared to the piezo motor. The slave robot implemented employs hard plastic (Delrin<sup>®</sup>) as the primary constructing material and brass (Alloy 360) and aluminum (Alloy 3601) for certain structural parts. Pneumatic cylinders are used as the primary actuators and one small piezo motor is adopted to provide rotary actuation with a small footprint. The piezo motor is placed at a distance from the scanning center to minimize the SNR loss due to the electrical noise introduced by its electronic driver.

The slave robot consists of a three-link parallel mechanism, an *X-Y* stage, and a needle driver to achieve the desired arbitrary needle orientation and position configuration under the space of the breast coil. Fig. 1 shows the needle (which is a substitute for a biopsy needle) attached to the fiber-optic force sensor [16]. The parallel mechanism uses three pneumatic cylinders as both structural components and actuators and provides two needle rotational DOFs and one translational DOF. The *X-Y* stage uses two pneumatic cable cylinders and allows two orthogonal translational DOFs, thereby enabling the placement of the needle in 3D. These five pneumatic cylinders are actuated through long transmission lines (up to 9 m), so that the non MRI-compatible valves and electronics can be placed inside the control room. The needle driver uses the small piezo motor to implement the needle insertion DOF.

The master robot has a similar kinematic structure as the slave robot and provides to the physician an intuitive manipulation platform for the needle driver and the parallel mechanism of the slave robot. The needle driver of the master is fully actuated and provides needle tissue interaction force feedback. The parallel mechanism provides an intuitive way of needle orientation adjustment and can be locked in place to help the physician maintain a stable needle orientation while performing needle insertion. The *X-Y* stage of the slave robot mounting the parallel mechanism is intuitive to operate and is directly controlled with a keyboard. The master robot communicates with the slave robot through the control PCs using a local dedicated router. With similar kinematic structures, actuator level one-to-one map can be established and the configuration information of each robot can be exchanged directly to create virtual connections, which establishes the teleoperated master-slave system.

Continuing on the prior work that demonstrated the 4-DOF slave robot [4], this paper presents the master-slave surgical system, with a 6-DOF MRI-compatible slave robot that can be operated under continuous MRI guidance. The paper is organized as follows. In Sect. 2, the current design of the slave robot as well as its control is presented, and the design of the master robot is covered in Sect. 3. Section 4 describes the system integration and control scheme of the master-slave system. The

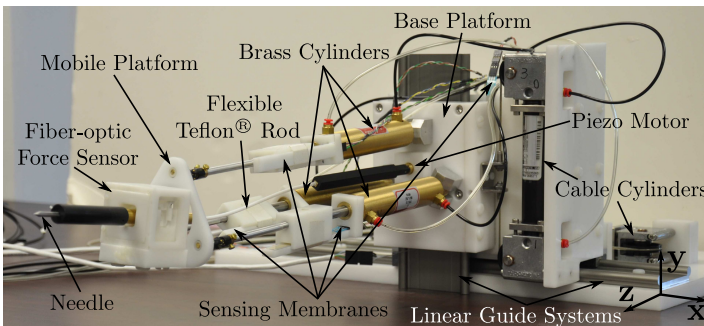
results of teleoperation and the slave robot MRI-compatibility evaluation are presented in Sect. 5 and concluding remarks are made in Sect. 6.

## 2 Design and Implementation of the MRI-Compatible Slave Robot

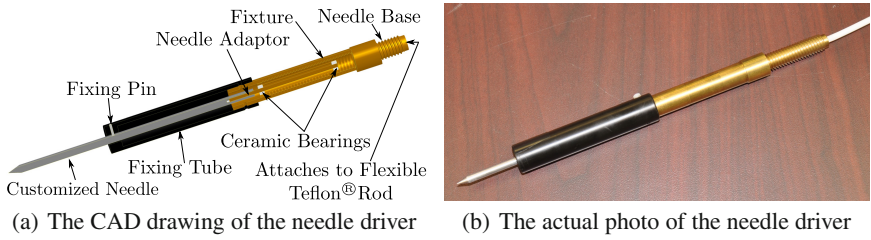
The MRI-compatible slave robot is shown in Fig. 1. It can operate inside the MRI bore and is built with brass (Alloy 360), aluminum (Alloy 3601), and Delrin<sup>®</sup>. Other parts including cylinders and bearings that have to be purchased are carefully chosen to be as non-magnetic as possible. The detailed design of its needle driver, parallel mechanism, and X-Y stage is described in the following subsections.

### 2.1 Needle Driver

The needle driver mechanism drives the needle by converting the rotary motion of the piezo motor (PiezoMotors, LEGS-R01NM-10) into translational motion with a screw structure. A CAD illustration of the design is shown in Fig. 2(a) and the actual mechanism is shown in Fig. 2(b). The rotational power generated by the distal piezo motor is transmitted via a flexible Teflon<sup>®</sup> rod to the outer threaded needle base to achieve translational motion. Two ceramic bearings (VXB, Kit8707) are press-fitted into the needle base and hold the needle adaptor to isolate the rotation of the screw motion at the needle base. The needle adaptor holds the customized needle (or needles of other dimensions) with screw threads. The friction force between the needle and the tissue during needle insertion is sufficient to prevent the needle from rotating and this leads to pure translational motion of the needle. To further ensure needle rotation elimination, the plastic fixing tube is attached to the outside of the fixture, into which the fixing pin is press-fitted and it mates with the customized groove on the needle.



**Fig. 1** Photo of the actual MRI-compatible slave robot



**Fig. 2** The needle driver that can advance the needle without rotation

The needle driver attaches to the parallel mechanism at its fixture through a previously developed MRI-compatible fiber-optic force sensor [16] that is used to sense the needle insertion force. This force sensor is mounted on the mobile platform of the parallel mechanism, as shown in Fig. 1.

### 2.2 Parallel Mechanism

The parallel mechanism consists mainly of three 76.2mm (3”) stroke pneumatic cylinders connecting the base platform with pin joints and the mobile platform with kinematically equivalent ball joints (serial connection of a universal joint and a roller bearing), since a small MRI-compatible ball joint was not available. The cylinders are primarily made of brass (Allentair, C-7/8x3-BU-L-SZ); and the base and the mobile platforms (See Fig. 1) are made of plastic instead of metal to avoid the interference of the induced eddy current, which can be generated inside these bulky parts when the slave robot is in motion or during scanning inside the MRI bore. These three cylinders are spaced evenly in 120° increments along circles on the two platforms with identical radius. By changing the cylinder lengths one translational and two rotational DOFs are achieved with coupled motion [17, 18, 4]. Using the world coordinate frame shown in Fig. 1 and the ZYZ Euler angle definition of the needle orientation, the  $x$  and  $y$  coordinates to the center point of the three universal joint centers and the needle rotation angle  $\gamma$  can be computed if the needle orientation angles,  $\alpha$  and  $\beta$ , are known. They are related by the expression [4]:

$$x_o = -\frac{r}{2}(1 - c_\beta)c_{2\alpha}, \quad y_o = \frac{r}{2}(1 - c_\beta)s_{2\alpha}, \quad \gamma = -\alpha \tag{1}$$

where  $r$  is the radius of the circles along which the three cylinders are spaced. The  $z$  coordinate of the center point of the three universal joint centers,  $z_o$ , can be specified independently. With the spatial coordinate of that center point, the coordinates of the universal joint centers can be computed and hence the lengths of the cylinders. The motion of the cylinder rod is controlled by mass flow rate valves inside the control room through long pneumatic transmission lines. Detailed information on the design of the parallel mechanism can be found in [4].

### 2.3 *X-Y Stage*

The *X-Y* stage holds the aforementioned parallel mechanism (Fig. 1) at its base platform and provides in the horizontal ( $x$ ) and vertical ( $y$ ) directions larger motion range compared to a traditional parallel mechanism. It can help position the needle at the appropriate location prior to engaging the parallel mechanism and the needle driver mechanism, and offset according to (1) the translational motion of the mobile platform while adjusting the needle orientation. To achieve the desired planar motion, two linear guide systems from igus<sup>®</sup> (DryLin<sup>®</sup> WK-10-80-20-01-450 and WK-10-80-10-01-250) have been adopted and are actuated with one 228.6 mm (9") stroke and one 101.6 mm (4") stroke cable cylinder (Tolomatic, 10760032 SK9 and 10760032 SK4, nonmagnetic version of CC07 series), respectively, which provide larger stroke in the limited space of the MRI bore. Position information for both directions is measured by the sensing membranes (spectrasymbol, TSP-L-0150-103-1%-RH and TSP-L-0300-103-1%-RH) as are used in the parallel mechanism.

With the *X-Y* stage and the parallel mechanism, the needle can be positioned at any location and oriented in any direction within the workspace, after which the needle can be inserted into the tissue.

### 2.4 *Pneumatic Control of the Slave Robot*

The pneumatic cylinders of the slave robot are actuated with long transmission lines, which is a challenging task and our prior work has demonstrated position control with good positioning accuracy using pressure valves [19] or mass flow rate valve [4]. The cylinders in the current system are all actuated with 9 m transmission lines (1.5875 mm (1/16") inner diameter) by 3/5 mass flow rate proportional valves (Festo, MPYE-5-1/4-010-B) that are placed inside the control room to avoid any electromagnetic interference with the magnetic field.

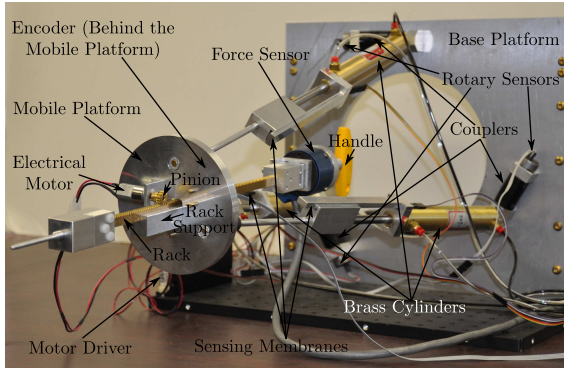
The control law implemented is PD control and Coulomb friction compensation has been introduced to reduce the static position error. The compensated friction force is intentionally set to be smaller than that observed in the experiment to avoid instability due to the slow pressure transient caused by the long transmission lines.

## 3 Design and Implementation of the Master Robot

Fig. 3 shows the master device which has a similar kinematic structure as the slave robot and is an intuitive and passive platform for the physician to adjust the needle orientation and perform needle insertion. It consists of a needle driver using a rack and pinion structure and a parallel mechanism similar to that of the slave device. The needle driver is actuated with an electrical motor and the needle position and interaction forces are measured by an encoder and a force sensor, respectively. Two control modes have been implemented based on proportional control: 1) position control with a proportional force feed-forward term that locks the needle in place while its orientation is being adjusted and, 2) force control that provides force



feedback to the physician based on the needle and soft-tissue interaction force. To enable the physician to comfortably hold the handle to manipulate the master device a larger diameter hole cut-out is present and hence the three brass cylinders are placed on a larger circle radius as shown in Fig. 3. Through appropriate modeling actuator level one-to-one map can be achieved. The parallel mechanism can be locked as needed to facilitate the physician for performing the needle insertion task by avoiding involuntary needle orientation change during needle insertion using a combination of six piloted check valves to close the cylinder chambers and a pair of pressure valves to adjust the chamber pressures.



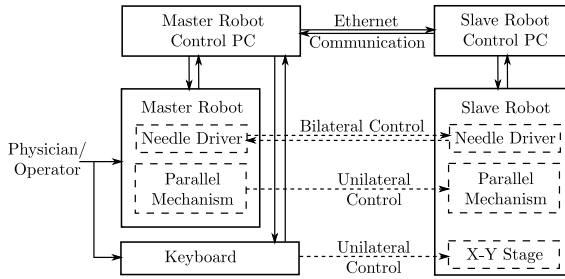
**Fig. 3** Photo of the actual master robot

## 4 Integration and Control of the Master-Slave System

The slave and the master robot have a dedicated control PC each with data acquisition cards (Sensoray, Model 626) installed. With similar kinematic structures that give actuator level one-to-one map, a master-slave system can be achieved by communicating the two control PCs and exchanging the configuration information of each pair of actuators directly, as shown in Fig. 4. The physician/operator manipulates the master robot to adjust the needle orientation and perform the needle insertion, while the  $X$ - $Y$  stage of the slave robot can be controlled using the keyboard of the master robot control PC. Bilateral control is implemented for the needle driver to help the physician perform the needle insertion task with force feedback, and unilateral control is applied to the other five DOFs.

The communication is implemented with Ethernet communication. With two control PCs equipped with Intel® Core™ i5 3.1 GHz CPU, 4 GB memory, and Intel® Gigabit Network Adaptors connected to a dedicated local router, the time delay was measured to be  $0.219 \pm 0.121$  ms under TCP protocol and  $0.204 \pm 0.120$  ms under UDP protocol, which is very small compared to the 500Hz sampling rate. It is hence neglected in the control system. TCP protocol has been chosen due to its implementation simplicity and reliability.





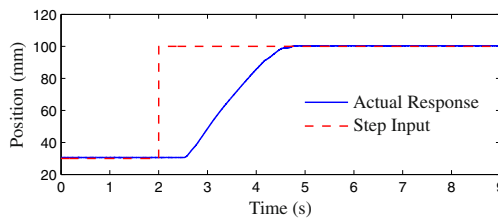
**Fig. 4** The control scheme of the master-slave system. The dashed lines indicate virtual connections established through the Ethernet communication.

## 5 Experimental Results

Experiments have been conducted to verify the pneumatic actuation of the slave robot, the teleoperation capability of the master-slave system, as well as the MRI-compatibility of the slave robot.

### 5.1 Pneumatic Actuation of the Slave Robot

Fig. 5 shows the actual response of the cable cylinder that actuates the slave robot in the y direction. It carried the 4kg parallel mechanism and was provided a step input from 30 mm to 100 mm at  $t = 2$  s. Smooth response curve was achieved with a position accuracy of 1 mm and long reaction time was observed due to the process in the long pneumatic transmission lines to build the pressures needed to overcome the friction force. All other cylinders showed similar response.

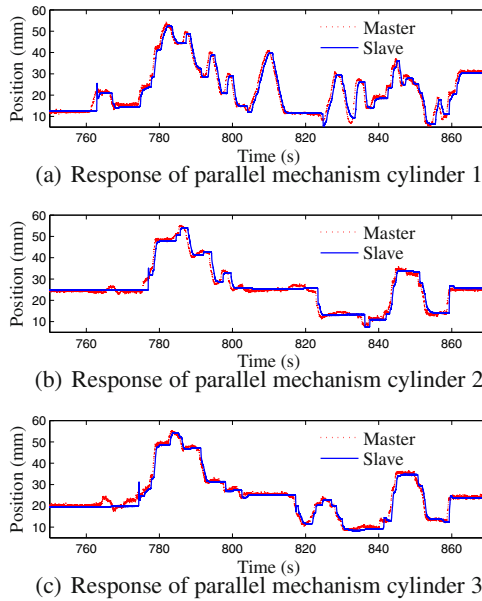


**Fig. 5** Experimental step response of the y-direction cable cylinder controlled with a mass flow rate valve through long transmission lines

### 5.2 Teleoperation of the Master-Slave System

A test was also conducted on the integrated master-slave system to verify its functionality by manipulating the master arbitrarily and collecting the response curves of both robots for orientation manipulation and needle insertion as shown in Fig. 6

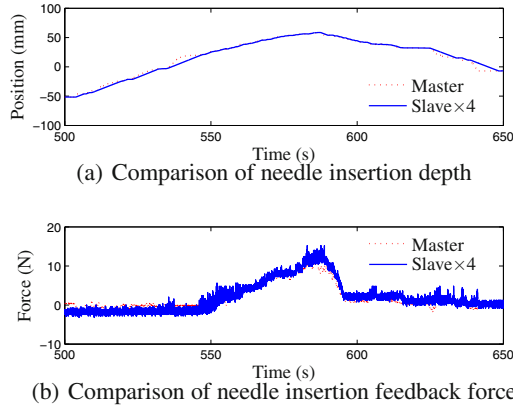
and Fig. 7, respectively. The commands for the  $X$ - $Y$  stage cable cylinders were set by keyboard and sent to the slave robot with negligible time delay; hence, their responses were similar to that when directly actuated and were omitted. Fig. 6(a) through Fig. 6(c) show the slave response curves in response to the master commanding signal. As can be seen from the plots, the slave robot is able to follow the master robot's configuration. A needle insertion task was also performed on a piece of foam and the resultant curves are shown in Fig. 7(a) and Fig. 7(b). The slave insertion depth was scaled down by 4 since precise insertion depth is desired while the feedback force was scaled up by 4 to enhance the force feel during needle insertion.



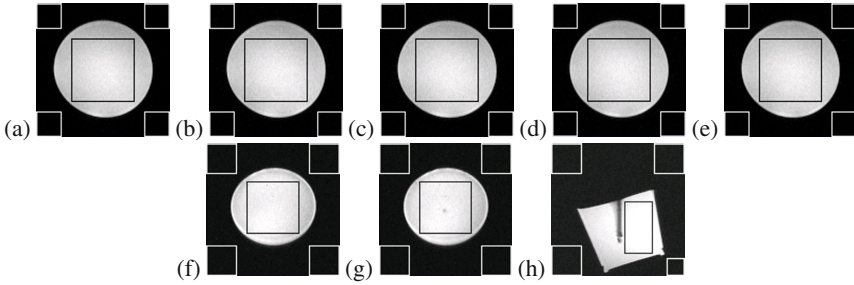
**Fig. 6** Experimental results of the master-slave system under free manipulation for needle orientation adjustment

### 5.3 MRI-Compatibility Test of the Slave Robot

Two MRI tests were conducted on the slave robot to verify its MRI-compatibility. The first test studied the MRI-compatibility of the robot when it was stationary and the influence of the pneumatic actuation on the MR image quality, while the second test studied the MR image quality variation due to piezo motor actuation. Testing was performed using a 3 T Tim Trio MR scanner (Siemens Medical Solutions; Malvern, PA) with a head coil. In the first test, a standard cylindrical bottle phantom was imaged using a gradient echo acquisition with  $T_E/T_R = 1.63/354.24$  ms, flip angle =  $20^\circ$ , bandwidth = 870 Hz/pixel, FoV =  $160 \times 160$  mm, matrix =  $128 \times 128$ .



**Fig. 7** Experimental results of the master-slave system under free manipulation for needle insertion



**Fig. 8** MR images of the two MRI tests. (a)–(e) are the images for the first test: (a) was taken without the robot as ground truth; (b) was taken with the robot unactuated; (c)–(e) were taken when the  $x$ ,  $y$  cylinders, and the parallel mechanism were actuated, respectively. (f)–(h) are the images for the second test: (f) was taken with the robot unactuated; (g) was taken when the piezo motor was actuated and the needle was approaching to the phantom; (h) was taken when the piezo motor was actuated and the needle was inserted into the phantom.

The second test used a homemade phantom consisting of a mixture of 200 ml boiling water with 7 g gelatin powder (Knox gelatin, Kraft Foods Global Inc.) and 2 ml Omniscan™ (gadodiamide – 287 mg/ml) (GE Healthcare; Waukesha, WI) to increase the  $T_1$ -weighted signal intensity. The imaging sequence used was a gradient echo acquisition with identical parameters as the first test except  $T_E/T_R = 1.81/388.8$  ms and  $FoV = 120 \times 120$  mm. The dynamic MR images are shown in Fig. 8.

**First Test:** Fig. 8(a) shows the image of the standard cylindrical bottle phantom while the slave robot was outside the MRI bore as the ground truth. Then the slave

robot was put into the MRI bore without any actuation to obtain Fig. 8(b). Fig. 8(c)–(e) were taken when the slave robot was actuated with the  $x$ ,  $y$  cylinders and with the three brass cylinders of the parallel mechanism, respectively.

**Second Test:** Fig. 8(f) shows the image of the gelatin phantom when the robot was put inside the MRI bore but without any actuation. Fig. 8(g) was taken when the piezo motor was actuated to advance the needle towards the phantom and the black point in the center indicates the needle tip. Fig. 8(h) was taken when the piezo motor was actuated and the needle was inserted into the phantom. All images can be clearly depicted with no visually-detectable distortion.

To quantify the effect of the robot on the MR image quality, the SNR, defined as the ratio of the mean pixel value of signal to the standard deviation of the pixel value of background noise, was calculated. The regions of interest (ROIs) used for this calculation are shown in Fig. 8 and the calculated SNRs are summarized in Table 1. It is worth noting that the SNRs for the two tests are not comparable due mainly to differences in the  $T_1$  of the bottle and gelatin phantoms. The SNR values show that when the robot was placed inside the MRI bore a small SNR loss was observed. When the slave robot was actuated pneumatically, the SNR value varied by a very small amount. This is likely due to the false signal generated by the metal parts. In the second test when the robot was actuated with piezo motor, a slight SNR increase was also observed. The significant SNR increase of Fig. 8(h) compared to Fig. 8(f) and 8(g) is largely due to its imaging plane residing inside the phantom, while the other two images were taken at the edge of the phantom to monitor the contact of the needle with the phantom, where the 3 mm thick imaging plane slice resided partially outside of the phantom.

**Table 1** SNRs of the MR images shown in Fig. 8, in which the ROIs used are marked

Test 1	(a)	(b)	(c)	(d)	(e)	Test 2	(f)	(g)	(h)
SNR	78.58	73.91	76.03	77.00	74.65	SNR	75.97	77.48	104.10

The acquired MR images and the SNR analysis show that the slave robot does not induce image distortion which cause significant degradation in the image quality and can be safely operated inside the scanner with only minimal losses in SNR.

## 6 Conclusions

This paper presents the design and implementation of a teleoperated master-slave surgical system that can potentially be used to perform breast biopsy under continuous MRI. The slave robot is built with MRI-compatible materials and actuators, and MRI tests showed that it could operate under continuous MR imaging with no visually-detectable distortion and minimal losses in SNR. The master robot is

designed with a similar kinematic structure as the slave to help the physician operate the system intuitively. The evaluation of the integrated system shows that the slave robot can follow the configuration of the master robot with satisfactory performance and force feedback can be provided on the master robot reliably. This shows the functionality of the developed teleoperation system and its potential to allow the physician to correct any needle insertion error during the procedures under continuous MRI guidance. Future work will focus on graphical user interface development and further refinement of the performance of the entire system.

**Acknowledgements.** This work is funded by NIH grant 1R01EB008713.

## References

1. American Cancer Society: Breast cancer facts & figures 2011–2012. American Cancer Society, Inc., Atlanta, Georgia (2011)
2. Lehman, C.D., Gatsonis, C., Kuhl, C.K., Hendrick, R.E., Pisano, E.D., Hanna, L., Peacock, S., Smazal, S.F., Maki, D.D., Julian, T.B., DePeri, E.R., Bluemke, D.A., Schnall, M.D.: MRI evaluation of the contralateral breast in women with recently diagnosed breast cancer. *The New England Journal of Medicine* 356(13), 1295–1303 (2007)
3. Kuhl, C.K., Schrading, S., Bieling, H.B., Wardelmann, E., Leutner, C.C., Koenig, R., Kuhn, W., Schild, H.H.: MRI for diagnosis of pure ductal carcinoma in situ: a prospective observational study. *The Lancet* 370(9586), 485–492 (2007)
4. Yang, B., Tan, U.X., McMillan, A., Gullapalli, R., Desai, J.P.: Design and implementation of a pneumatically-actuated robot for breast biopsy under continuous MRI. In: *IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 674–679 (May 2011)
5. Elhawary, H., Zivanovic, A., Rea, M., Davies, B., Besant, C., McRobbie, D., de Souza, N., Young, I., Lampérth, M.: The feasibility of MR-image guided prostate biopsy using piezoceramic motors inside or near to the magnet isocentre. In: Larsen, R., Nielsen, M., Sporning, J. (eds.) *MICCAI 2006*. LNCS, vol. 4190, pp. 519–526. Springer, Heidelberg (2006)
6. Stoianovici, D., Song, D., Petrisor, D., Ursu, D., Mazilu, D., Mutener, M., Schar, M., Patriciu, A.: “MRI Stealth” robot for prostate interventions. *Minimally Invasive Therapy* 16(4), 241–248 (2007)
7. Goldenberg, A.A., Trachtenberg, J., Kucharczyk, W., Yi, Y., Haider, M., Ma, L., Weersink, R., Raoufi, C.: Robotic system for closed-bore MRI-guided prostatic interventions. *IEEE/ASME Transactions on Mechatronics* 13(3), 374–379 (2008)
8. Fischer, G.S., Iordachita, I., Csoma, C., Tokuda, J., DiMaio, S.P., Tempany, C.M., Hata, N., Fichtinger, G.: MRI-compatible pneumatic robot for transperineal prostate needle placement. *IEEE/ASME Transactions on Mechatronics* 13(3), 295–305 (2008)
9. Krieger, A., Iordachita, I., Song, S.E., Cho, N.B., Guion, P., Fichtinger, G., Whitcomb, L.L.: Development and preliminary evaluation of an actuated MRI-compatible robotic device for MRI-guided prostate intervention. In: *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, pp. 1066–1073 (May 2010)
10. Su, H., Zervas, M., Cole, G.A., Furlong, C., Fischer, G.S.: Real-time MRI-guided needle placement robot with integrated fiber optic force sensing. In: *IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 1583–1588 (May 2011)

11. Ho, M., Koltz, M., Simard, J.M., Gullapalli, R., Desai, J.P.: Towards a MR image-guided SMA-actuated neurosurgical robot. In: IEEE International Conference on Robotics and Automation, Shanghai, China, pp. 1153–1158 (May 2011)
12. Kaiser, W.A., Fischer, H., Vagner, J., Selig, M.: Robotic system for biopsy and therapy of breast lesions in a high-field whole-body magnetic resonance tomography unit. *Investigative Radiology* 35(8), 513–519 (2000)
13. Larson, B.T., Erdman, A.G., Tsekos, N.V., Yacoub, E., Tsekos, P.V., Koutlas, I.G.: Design of an MRI-compatible robotic stereotactic device for minimally invasive interventions in the breast. *Journal of Biomechanical Engineering* 126(4), 458–465 (2004)
14. Kokes, R., Lister, K., Gullapalli, R., Zhang, B., McMillan, A., Richard, H., Desai, J.P.: Towards a teleoperated needle driver robot with haptic feedback for RFA of breast tumors under continuous MRI. *Medical Image Analysis* 13(3), 445–455 (2009)
15. Elhawary, H., Zivanovic, A., Davies, B., Lampérth, M.: A review of magnetic resonance imaging compatible manipulators in surgery. *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of Engineering in Medicine* 220(3), 413–424 (2006)
16. Tan, U.X., Yang, B., Gullapalli, R., Desai, J.P.: Triaxial MRI-compatible fiber-optic force sensor. *IEEE Transactions on Robotics* 27(1), 65–74 (2011)
17. Lee, K.M., Shah, D.K.: Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE Journal of Robotics and Automation* 4(3), 354–360 (1988)
18. Pfreundschuh, G.H., Sugar, T.G., Kumar, V.: Design and control of a three-degrees-of-freedom, in-parallel, actuated manipulator. *Journal of Robotic Systems* 11(2), 103–115 (1994)
19. Yang, B., Tan, U.X., McMillan, A., Gullapalli, R., Desai, J.P.: Design and control of a 1-DOF MRI compatible pneumatically actuated robot with long transmission lines. *IEEE/ASME Transactions on Mechatronics* 16(6), 1040–1048 (2011)

# Motion Compensated Catheter Ablation of the Beating Heart Using Image Guidance and Force Control

Samuel B. Kesner and Robert D. Howe

**Abstract.** Cardiac catheters allow physicians to access the inside of the heart and perform therapeutic interventions without stopping the heart or opening the chest. However, conventional manual and actuated cardiac catheters are currently unable to precisely track and manipulate the intracardiac tissue structures because of the fast tissue motion and potential for applying damaging forces. This paper addresses these challenges by proposing and implementing a robotic catheter system that use 3D ultrasound image guidance and force control to enable constant contact with a moving target surface in order to perform an interventional procedure, in this case tissue ablation. The robotic catheter system, consisting of a catheter module, ablation and force sensing end effector, drive system, and image-guidance and control system, was commanded to apply a constant force against a moving target using a position-modulated force control method. As compared to a manual catheter system, the robotic catheter was able to apply a more consistent force on the target while maintaining ablation electrode contact with 97% less RMS contact resistance variation. These results demonstrate that the 3D ultrasound guidance and force control allow the robotic system to maintain better contact with a moving tissue structure, thus allowing for more accurate and repeatable tissue ablation procedures.

## 1 Introduction

Advances in cardiac catheter technology allow physicians to treat a range of conditions inside the beating heart while avoiding both the invasiveness of opening the chest and the cognitive impairment risks associated with cardiopulmonary bypass [1-3]. However, the majority of catheters currently used for cardiac interventions only allow for slow manual motions of the catheter tip and are

---

Samuel B. Kesner · Robert D. Howe  
Harvard School of Engineering and Applied Sciences,  
Cambridge, MA, USA  
e-mail: {skesner, howe}@seas.harvard.edu

unable to control the forces applied to the tissue surfaces. Commercially available robotic catheter systems, such as the Artisan Control Catheter (Hansen Medical, Mountain View CA, USA) or the CorPath Vascular Robotic System (Corindus Vascular Robotics, Natick MA, USA), achieve manual catheter manipulation speeds while allowing the operator to utilize robotic teleoperation to reduce radiation exposure [4, 5]. However, neither the manual nor the commercial robotic catheter systems are able to compensate for the fast cardiac motion or regulate the forces applied to the tissue surface.

The goal of our work is to enable a robotic catheter to track the fast motions of the heart while controlling the forces applied by the catheter end effector to the tissue in order to improve the safety and efficacy of medical procedures. This objective is achieved through the use of 3D ultrasound (3DUS) guidance, active motion compensation, and catheter tip force control. The medical application selected for this project is the radiofrequency (RF) ablation of cardiac tissue. Ablation is used by interventional cardiologists and cardiac surgeons to destroy cardiac conduction pathways that contribute to arrhythmias, or heart beat abnormalities [6]. The outcome success of this procedure is dependent on the electrode contact with the tissue and force application, and therefore can benefit from the robotic system proposed here [7-10].

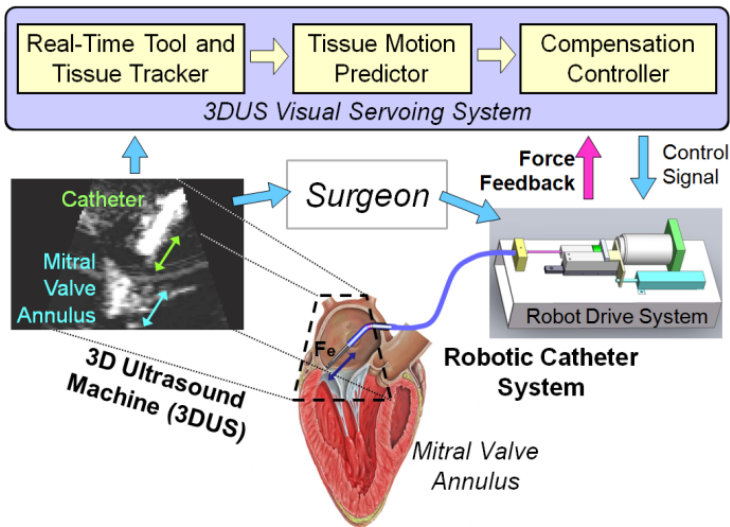
In previous work, we have demonstrated *in vivo* the ability of the robotic catheter system to compensate for the fast motion of the heart [11]. A custom catheter tip force sensor was developed to enable the catheter to maintain a constant force relative to a target. However, the force control system has to date only been evaluated on the bench top using noise-free simulated position signals without actual ultrasound image-derived signals [12, 13]. Other work in cardiac motion compensation has focused primarily on interacting with the exterior of the beating heart [14-16]. In addition, the previous work in robotic catheters has primarily focused on teleoperation and position control [4, 5]. To the authors' knowledge, the work presented here represents the first time 3DUS image guidance and force control has been used to enable a robotic catheter to accurately interact with a moving target. Furthermore, this paper presents the first application of the 3DUS-guided and motion compensated catheter system to a clinical procedure, improving the therapeutic efficacy of ablation on the beating heart.

The following paper presents the robotic catheter system, the force sensing and ablation end effector, and the force control method. Next, the paper presents the system evaluation method, the experimental results, and finally concludes with a discussion of the implications and limitations of the results. This work demonstrates the potential benefits of integrating motion-compensation and force control with cardiac intervention catheters.

## 2 Technical Approach

The goal of the robotic catheter system is to use real-time 3DUS to measure the target tissue motion and then drive a robotic catheter to synchronize with the motion and apply a constant force to the tissue with a RF ablation end effector. The system (Fig. 1) is composed of three main modules: the drive system that actuates

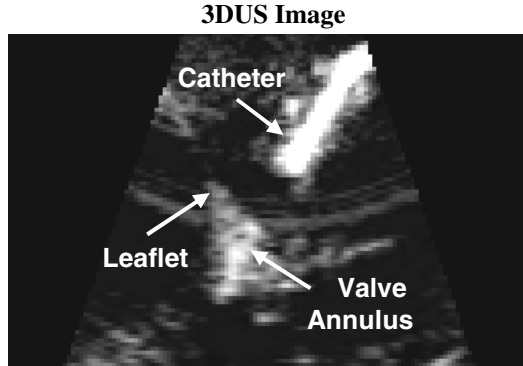




**Fig. 1** The robotic system serves the catheter using 3DUS guidance and force feedback

the catheter, the catheter module that is inserted through the vasculature into the heart, and the 3D ultrasound visual servoing system that tracks the tissue and commands the catheter to follow the motion. The drive system contains a linear voice coil actuator and a position sensor that are able to rapidly adjust the catheter position. The catheter module is composed of a stainless steel coil guidewire inside of a nylon sheath. The sheath is positioned inside of the vasculature to guide the actuated catheter into the heart and the guidewire is servoed by the drive system to compensate for the heart motion. Finally, the visual servoing system utilizes a 3DUS machine (Fig. 2, SONOS 7500 with X4 Ultrasound Transducer, Philips Healthcare, Andover, MA, USA) and a tissue tracking and motion prediction system to determine the real-time motion of the cardiac tissue and control the catheter [17-20]. A more detailed description of the mechanical design of the robotic catheter system is provided in [11].

A novel integrated force sensing and ablation end effector is presented here for the first time (Fig. 3). The design goal of the ablation tool design is to enable the catheter system to apply RF energy to the fast-moving tissue inside the heart while applying a constant normal force. The functional requirements of the ablation end effector are to sense forces, to ablate tissue using a clinical RF generator with the same efficacy as conventional ablation catheters, and to be robust enough to operate in the intracardiac environment. The device consists of a force sensor described in [12], a stainless steel electrode, and a fine wire that runs through the catheter to the RF current generator. The current prototype is approximately 5 mm in diameter and is created using rapid prototyping 3D printing technology. The size of the end effector can be further reduced with improved fabrication such as laser micromachining and metal laser sintering. See [12] for a more detail description of the 3D printed force sensor technology. Fig. 3b presents examples of



**Fig. 2** Ultrasound image showing the catheter, mitral valve annulus, and mitral valve leaflets

the RF ablation lesions created with this tool on porcine skeletal muscle tissue (RF generator: Stockert 70, Biosense Webster, Diamond Bar, California, USA).

The objective of the control system is to apply a desired force on a fast moving target with the robotic catheter end effector. A standard error-based force control approach will not work for the robotic catheter system because of the limitations identified in [11], including backlash and friction in the catheter transmission system [21, 22]. These limitations prevent a standard force regulator from correctly responding to the force tracking error in a stable manner because the internal dynamics of the catheter obstruct the controller action from being accurately transmitted from drive system to the catheter tip. To overcome these issues, we propose a method that uses the force error term to modulate the commanded position trajectory of the catheter. This approach is similar to the inner position loop force control approaches used to implement force control on high-friction industrial manipulators [23]. In addition to improved system stability, the use of an inner position loop also allows the controller to directly compensate for the catheter friction and backlash as these limitations are position and velocity dependent [11]. See Fig. 4 for a block diagram of the control system.

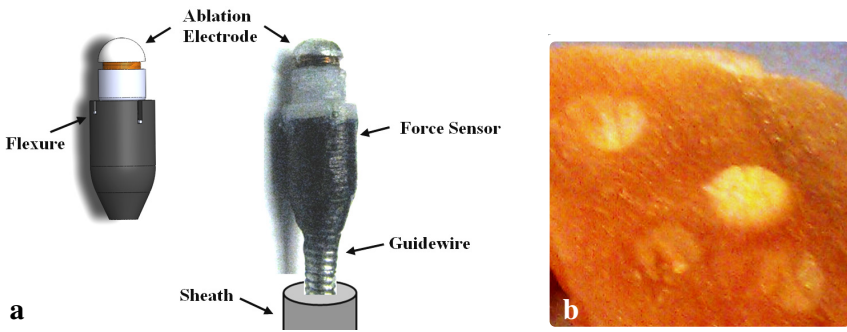
In this force control approach, the drive system is commanded to follow a desired position,  $x_d$ , that is the sum of the position of the moving target,  $x_e$  and the position offset required to maintain the desired force,  $x_f$

$$x_d = x_e + x_f \quad (1)$$

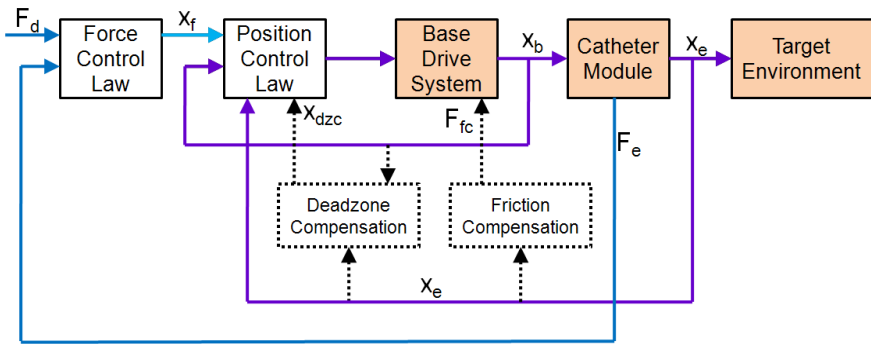
The force modulation term is

$$x_f = K_f (F_d - F_e) + K_{fi} \int (F_d - F_e) dt \quad (2)$$

where  $F_d$  is desired force,  $F_e$  is the force applied to the environment, and  $K_f$  and  $K_{fi}$  are controller gains. This control law is similar to the method presented by Villani et al in [24].



**Fig. 3** a) Ablation end effector solid model and prototype. b) Tissue sample (porcine skeletal muscle) ablated with the RF ablation end effector. Lesions are approximately 4 mm in diameter.



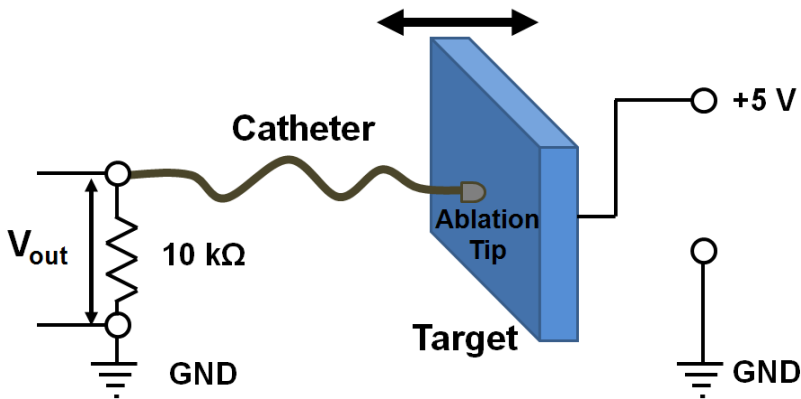
**Fig. 4** The force control system block diagram. The blue lines indicate force values and the purple lines indicate position values.

### 3 Experimental Evaluation

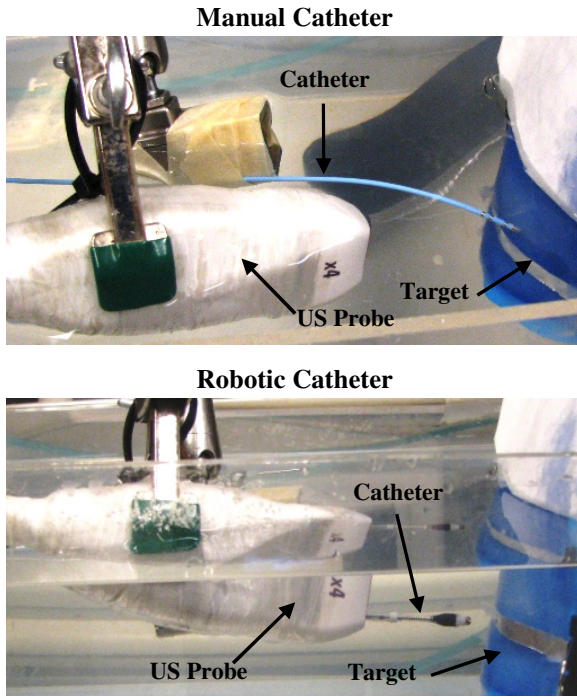
The robotic ablation catheter system was evaluated in a water tank experiment to examine the ability of the system to maintain good RF ablation electrode contact against a moving surface while applying a constant force. A number of studies have demonstrated that cardiac ablation efficacy is directly related to the forces applied by the catheter tip and the quality of the electrode-tissue contact [7-10]. Manually operated catheters do not adequately ablate tissue if they are bouncing or sliding on the tissue surface, in poor contact due to low forces, or creating tissue perforations due to large contact forces [9, 10]. The objective of this evaluation was to demonstrate that the robotic catheter system can improve ablation quality by maintaining good contact while accurately controlling the force.

The system was evaluated by commanding the catheter to maintain a constant contact force against a moving target. The target was composed of a conductive pad used as the current return path electrode in clinical ablation and electrocautery procedures (REM Polyhesive II Patient Return Electrode, Tyco Healthcare, Gosport, UK) backed with compliant foam (thickness: 25 mm, approximate stiffness: 0.1 N/mm). The target was translated with a 12 mm amplitude at a frequency of approximately 1 Hz (60 beats per minute). Two motion patterns were tested: a sinusoidal trajectory and a human mitral annulus trajectory [25]. The ablation quality was evaluated by measuring the electrical resistance between the catheter tip electrode and the return electrode pad using an instrumented voltage divider (Fig. 5). The water tank environment was used to allow the 3DUS guidance system to visualize the catheter and target.

The evaluation experiment was conducted using both the robotic catheter and a commercial manual ablation catheter (RF Marinr MCXL, Medtronic, Minneapolis, Minnesota, USA) for comparison. A manual catheter was select for comparison to demonstrate the limitations of the current technology due to a lack of motion compensation and force control. For the manual catheter, a load cell was also added to the target to record the forces applied by the catheter tip (LCFD-1KG, Omega Engineering, Stamford, CT, USA; range: 10 N, accuracy:  $\pm 0.015$  N). The robotic catheter was instrumented with the force-sensing ablation end effector and was operated under force control with 3DUS guidance. Both catheters were rigidly braced 100 mm from the ablation tip at orientations perpendicular to the plane of the moving target. The manual catheter was positioned so its ablation electrode was able to remain in contact during the entire target trajectory (Fig. 6).



**Fig. 5** The catheter ablation experimental setup. The moving target was connected to a 5 V DC signal and the catheters were instrumented with a voltage divider to measure the ablation resistance. Resistance measurements were used to evaluate tip contact quality for both a manual catheter and the robotic catheter system.

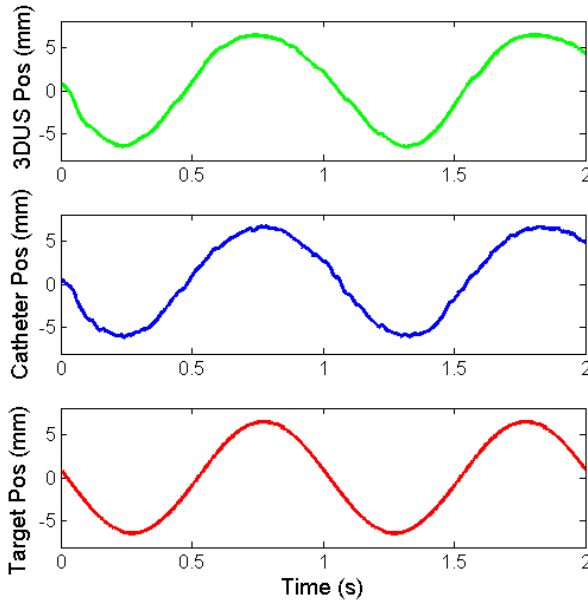


**Fig. 6** The water tank setup for the manual catheter (top) and the robotic catheter (bottom). Both images show the catheters, the white 3DUS imaging probe, and the blue motion target.

## 4 Results

Fig. 7 presents the position trajectories of the 3DUS tracking system, robotic catheter, and motion simulator during the experiments. Fig. 8 presents typical results of the ablation experiment on the sinusoidal motion target. Both the manual and robotic catheters were in contact with the moving target for over 5 s during each trial, sufficient time to perform ablation. The manual catheter was not able to apply a constant force or maintain a constant resistance.

The reason for the manual catheter's poor performance was because the motion of the target caused the manual catheter ablation tip to slide and tilt relative to the target surface as the motion simulator pushed on the catheter and caused it to buckle. Compliance is a desirable feature in manual catheters because it prevents them from applying large forces and perforating cardiac tissue. However, this bending compliance makes it challenging to achieve reliable ablation performance. As shown in Fig. 8, the manual catheter generated peak-to-peak resistance variations of over 20 kOhm and peak-to-peak force variations of 0.4 N.

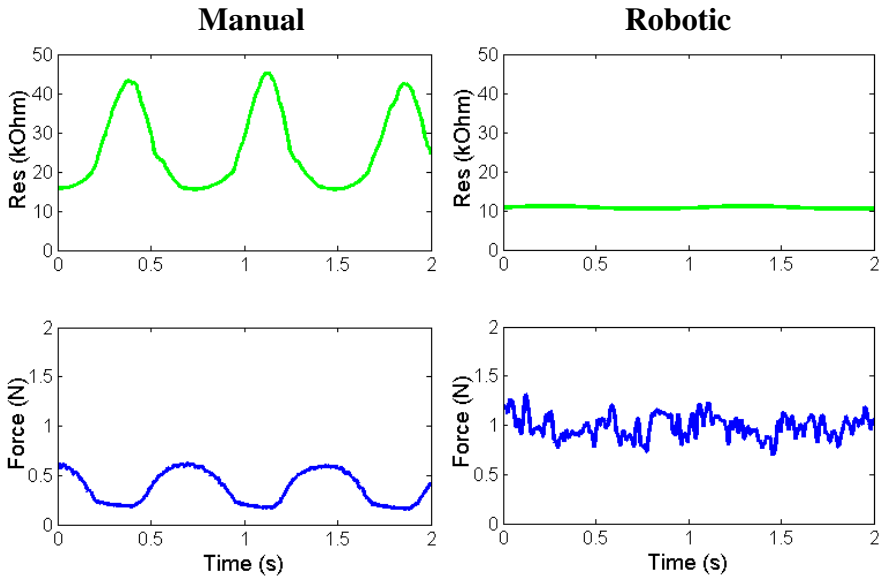


**Fig. 7** The position trajectories during the robotic catheter experiment: (top) The 3DUS tracking system; (middle) The robotic catheter position commanded by the force control system; (bottom) The actual target trajectory.

The robotic catheter, in contrast to the manual catheter, achieved almost constant resistance values while maintaining a desired force of 1 N with a force tracking error of 0.11 N RMS. The RMS variation of the resistance value for the robotic catheter was 0.25 kOhms, 97% less than the RMS variation of 9.88 kOhm for the manual catheter system. The robotic catheter was able to achieve this level of performance because the 3DUS-guided motion compensation system and the force control algorithm enabled the ablation tip to maintain consistent contact with the target despite the fast motion (Fig. 8). Similar force tracking results were obtained using the human mitral valve trajectory.

## 5 Discussion

These results demonstrate that image-guided motion compensation and force control can improve key parameters that determine ablation quality. This confirms that our robotic approach has the potential to increase clinical efficacy of intracardiac procedures. The system was able to apply a constant force while maintaining a constant ablation resistance with the ablation end effector on a moving target. In contrast, the force and electrical contact provided by the manual catheter in the same experimental setup varied greatly. This variation can primarily be attributed to the buckling, sliding, and tilting behavior of the manual catheter



**Fig. 8** A comparison of the electrical resistance and interaction forces between a conductive target and a manual catheter (left) and the robotic catheter system (right). The manual catheter applies a force and resistance that vary with the motion of the target. In contrast, the robotic catheter achieved consistent contact with the moving target while applying a constant force.

tip due to the target motion. The 3DUS motion tracking enabled the robotic catheter to compensate for the target motion and maintain good ablation electrode contact without the buckling behavior of the manual catheter.

One insight from this work is that multiple forms of sensor information are required to command a catheter to safely and effectively interact with the moving target. Force sensing alone is not sufficient for the catheter to track the target motion, as described in [13]. This is due to the fact that the catheter performance limitations of backlash and friction prevent the system from responding fast enough to the quick tissue motion using only force feedback. Motion tracking must also be used to overcome these limitations and maintain system stability [13]. The image guidance provides the desired position trajectory for the tip of the catheter and the force feedback allows for minor adjustments in the tip position to regulate and maintain the tool-tissue interactions forces. Without either 3DUS guidance or force sensing, the catheter would be unable to maintain the consistent ablation electrode contact and could either penetrate or retract from the target surface.

Although the experimental results demonstrate that the robotic catheter system is able to apply a constant force while maintaining a consistent ablation contact, there are a number of limitations in this initial validation study due to the challenges of accurately simulating *in vivo* cardiac ablation in a laboratory setting. First, measuring the DC resistance of the contact does not consider the electrical frequency response of the cardiac tissue at the 500 kHz frequency used by the RF

energy generator. In addition, the system was tested in water instead of electrically-conductive blood or saline, which alters the electrode conduction properties. Finally, the experimental setup did not accurately approximate the exact mechanics of intracardiac ablation, including the compliance of the vessels in the heart and the tool orientation relative to the moving tissue structures. The manual catheter performance depends on its orientation with respect to the moving tissue target, although similar fluctuations in force and resistance would have resulted for other orientations. We anticipate that these issues will not impair the demonstrated advantages of the robotic system because of the known properties of the ablation process and the success of previous *in vivo* tests of the image guidance systems [11, 26, 27].

## 6 Conclusions

This paper presents the experimental evaluation of the robotic catheter system for cardiac ablation. The system uses motion compensation and force feedback to maintain a constant force and ablation resistance on a moving target. The experimental results presented here demonstrated that the robotic system is able to maintain consistent ablation electrode contact with a translating motion simulator with a 97% reduction in RMS resistance variation over a manual catheter. The result can be explained by the fact that a compliant manual catheter slides and buckles while in contact with a quickly moving structure, such as the actively contracting heart wall.

Future work in this project will focus on the demonstration and evaluation of the technology in an *in vivo* setting. While the motion compensation and robotic catheter system has been demonstrated previously *in vivo* [11, 26], the force control ablation system has not yet been tested inside a beating heart. One possible challenge the system will encounter *in vivo* is how to respond when the tissue stiffness changes over the course of the heart cycle. In addition, safety issues such as system stability and preventing tissue collisions will need to be further investigated. The project objective is to enable a range of beating heart surgical procedures with a catheter, and the ablation procedure presented here is a first step toward this ultimate goal.

## References

- [1] Baim, D.S. (ed.): *Grossman's Cardiac Catheterization, Angiography, and Intervention*. Lippincott Williams & Wilkins, Philadelphia (2005)
- [2] Murkin, J.M., Boyd, W.D., Ganapathy, S., Adams, S.J., Peterson, R.C.: Beating heart surgery: why expect less central nervous system morbidity? *Annals of Thoracic Surgery* 68, 1498–1501 (1999)
- [3] Roach, G.W., Kanchuger, M., Mangano, C.M., Newman, M., Nussmeier, N., Wolman, R., Aggarwal, A., Marschall, K., Graham, S.H., Ley, C.: Adverse cerebral outcomes after coronary bypass surgery. *N. Engl. J. Med.* 335, 1857–1864 (1996)



- [4] Camarillo, D.B., Milne, C.F., Carlson, C.R., Zinn, M.R., Salisbury, J.K.: Mechanics Modeling of Tendon-Driven Continuum Manipulators. *IEEE Transactions on Robotics* 24, 1262–1273 (2008)
- [5] Beyar, R.: within the heart and vessels in clinical practice. *Ann. N. Y. Acad. Sci.* 1188, 207–213 (2010)
- [6] Huang, S.K., Huang, S.K.S., Wilber, D.J.: *Radiofrequency Catheter Ablation of Cardiac Arrhythmias: Basic Concepts and Clinical Applications*. Wiley-Blackwell (2000)
- [7] Okumura, Y., Johnson, S., Packer, D.: An analysis of catheter tip/tissue contact force-induced distortion of three-dimensional electroanatomical mapping created using the Sensei robotic catheter system. *Heart Rhythm* 4, S318 (2007)
- [8] Kalman, J.M., Fitzpatrick, A.P., Olgin, J.E., Chin, M.C., Lee, R.J., Scheinman, M.M., Lesh, M.D.: Biophysical characteristics of radiofrequency lesion formation in vivo: dynamics of catheter tip-tissue contact evaluated by intracardiac echocardiography. *Am. Heart J.* 133, 8–18 (1997)
- [9] Shah, D.C., Lambert, H., Nakagawa, H., Langenkamp, A., Aeby, N., Leo, G.: Area Under the Real - Time Contact Force Curve (Force–Time Integral) Predicts Radiofrequency Lesion Size in an In Vitro Contractile Model. *J. Cardiovasc. Electrophysiol.* 21, 1038–1043 (2010)
- [10] Shah, D., Lambert, H., Langenkamp, A., Vanenkov, Y., Leo, G., Gentil-Baron, P., Walpoth, B.: Catheter tip force required for mechanical perforation of porcine cardiac chambers. *Europace* 13, 277 (2011)
- [11] Kesner, S.B., Howe, R.D.: Position Control of Motion Compensation Cardiac Catheters. *IEEE Transactions on Robotics* 27, 1045–1055 (2011)
- [12] Kesner, S.B., Howe, R.D.: Design Principles for Rapid Prototyping Forces Sensors Using 3-D Printing. *IEEE/ASME Transactions on Mechatronics* 16, 866–870 (2011)
- [13] Kesner, S.B., Howe, R.D.: Force control of flexible catheter robots for beating heart surgery. In: *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1589–1594 (2011)
- [14] Ginhoux, R., Gangloff, J., de Mathelin, M., Soler, L., Sanchez, M.M.A., Marescaux, J.: Active filtering of physiological motion in robotized surgery using predictive control. *IEEE Transactions on Robotics* 21, 67–79 (2005)
- [15] Bebek, O., Cavusoglu, M.: Intelligent control algorithms for robotic assisted beating heart surgery. *IEEE Transactions on Robotics* 23, 468–480 (2007)
- [16] Nakamura, Y., Kishi, K., Kawakami, H.: Heartbeat synchronization for robotic cardiac surgery. In: *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2014–2019 (2001)
- [17] Novotny, P.M., Stoll, J.A., Vasilyev, N.V., Del Nido, P.J., Dupont, P.E., Zickler, T.E., Howe, R.D.: GPU based real-time instrument tracking with three-dimensional ultrasound. *Med. Image Anal.* 11, 458–464 (2007)
- [18] Novotny, P.M., Stoll, J.A., Dupont, P.E., Howe, R.D.: Real-time visual servoing of a robot using three-dimensional ultrasound. In: *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2655–2660 (2007)
- [19] Yuen, S.G., Novotny, P.M., Howe, R.D.: Quasiperiodic predictive filtering for robot-assisted beating heart surgery. In: *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3875–3880 (2008)
- [20] Yuen, S.G., Kettler, D.T., Novotny, P.M., Plowes, R.D., Howe, R.D.: Robotic motion compensation for beating heart intracardiac surgery. *The International Journal of Robotics Research* 28, 1355 (2009)

- [21] Eppinger, S., Seering, W.: Understanding bandwidth limitations in robot force control. In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 904–909 (1987)
- [22] Townsend, W., Salisbury Jr., J.: The effect of coulomb friction and stiction on force control. In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 883–889 (1987)
- [23] Maples, J., Becker, J.: Experiments in force control of robotic manipulators. In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 695–702 (1986)
- [24] Chiaverini, S., Siciliano, B., Villani, L.: A survey of robot interaction control schemes with experimental comparison. IEEE/ASME Transactions on Mechatronics 4, 273–285 (1999)
- [25] Kettler, D.T., Plowes, R.D., Novotny, P.M., Vasilyev, N.V., del Nido, P.J., Howe, R.D.: An active motion compensation instrument for beating heart mitral valve surgery. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1290–1295 (2007)
- [26] Yuen, S.G., Kesner, S.B., Vasilyev, N.V., del Nido, P.J., Howe, R.D.: 3D ultrasound-guided motion compensation system for beating heart mitral valve repair. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) MICCAI 2008, Part I. LNCS, vol. 5241, pp. 711–719. Springer, Heidelberg (2008)
- [27] Kesner, S.B., Yuen, S.G., Howe, R.D.: Ultrasound servoing of catheters for beating heart valve repair. In: Navab, N., Jannin, P. (eds.) IPCAI 2010. LNCS, vol. 6135, pp. 168–178. Springer, Heidelberg (2010)

# Robotic Micropipette Aspiration of Biological Cells

Ehsan Shojaei-Baghini and Yu Sun

**Abstract.** This paper presents a system for mechanically characterizing single cells using automated micropipette aspiration. Using vision-based control and position control, the system controls a micromanipulator, a motorized translation stage, and a custom-built pressure system to position a micropipette ( $4\ \mu\text{m}$  opening) to approach a cell, form a seal, and aspirate the cell into the micropipette for quantifying the cell's elastic and viscoelastic parameters as well as viscosity. Image processing algorithms were developed to provide controllers with real-time visual feedback and to accurately measure cell deformation behavior on line. Experiments on both solid-like and liquid-like cells demonstrated that the system is capable of efficiently performing single-cell micropipette aspiration and has low operator skill requirements.

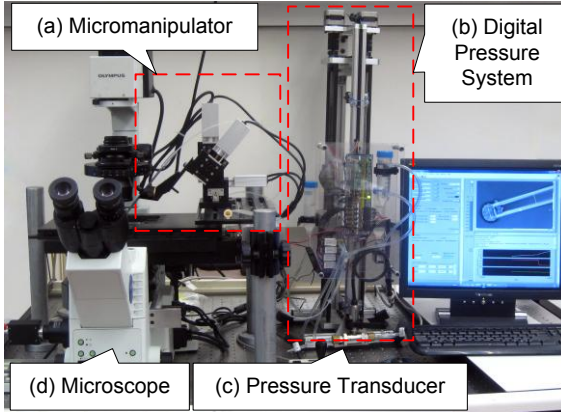
## 1 Introduction

The study of living cell mechanics can help gain insight in cell structures and functions [1]. It has been demonstrated that single-cell mechanical characterization can also be useful for investigating disease mechanisms and progression [2, 3], since the biomechanical properties of the pathological cells can differ from healthy ones. Mechanical characterization of cells from the pleural fluids of patients revealed that the Young's modulus of cancer cells is considerably less than in benign cells [3].

Population-based cell experiments do not permit a thorough examination of the stochastic processes involved in regulating cellular function at the single cell level [4]. The need of investigating biomechanics at cellular level calls for single cell manipulation techniques for studying the complex and dynamic behavior of

---

Ehsan Shojaei-Baghini · Yu Sun  
Advanced Micro and Nanosystems Laboratory,  
University of Toronto,  
Toronto, ON M5S 3G8, Canada  
e-mail: {ehsan, sun}@mie.utoronto.ca



**Fig. 1** System for automated single-cell micropipette aspiration. (a) Micromanipulator with encoder feedback. (b) Custom-built computer-controlled pressure system. (c) Differential pressure transducer measuring the pressure difference between the delta and reference tanks. (d) Inverted microscope with motorized focus.

individual cells. Over the years, many methods have been developed to study mechanical properties of single cells. In Atomic Force Microscopy (AFM), mechanical deformation is induced using the tip of a cantilever. The deflection of the cantilever is used to calculate the applied force. Using this technique, a map of cell stiffness across the cell surface can be generated [1, 5–7]. In Magnetic Twisting Cytometry (MTC), magnetic beads are attached to the cell surface. Twisting moment induced by the application of a magnetic field causes the cell to deform. Both elastic and viscoelastic properties of cells can be measured using this technique [6, 8–10]. In optical tweezers, laser beams trap high-refractive-index dielectric beads attached to cells and hence, exert forces to the cells [1, 2, 5, 6, 11]. The applied force in optical tweezers is typically limited to few hundred of pN. More experimental single-cell mechanical characterization techniques were reviewed in [1].

This paper focuses on the micropipette aspiration technique. Micropipette aspiration uses a fine hollow needle to aspirate a portion of a cell with vacuum pressure. Geometric changes in cell shape are measured to determine the elastic and viscoelastic properties, and flow resistance of the cell [1, 5, 12–17]. Conventionally, micropipette aspiration requires well-trained operators to look into the eyepieces of a microscope, skillfully operate multiple devices to micromanipulate single cells, control vacuum at proper timings to form a seal between the cell and micropipette opening, synchronized pressure and video recording, and accurately measure recorded images to extract cell deformation parameters. After cell experiments, the operator determines cell deformations by performing manual measurements on a high number of recorded images.

Micropipette aspiration is a powerful technique and has low infrastructure requirement. However, the high skill requirements and the efforts required for tedious, time-consuming post processing of data make the technique difficult and

**Table 1** Valve states in the digital pressure system

State	Description	V1	V2	V3	V4
1	Sense Pressure	C	O	O	O
2	Sense and apply pressure	O	C	O	O
3	Route high pressure to micropipette only	O	O	C	C
4	Route high pressure to Delta tank only	C	O	O	C

Note: O and C represent Open and Close, respectively.

less appealing to use. In contrast, AFM over the past decade has been largely automated. The automated operation of AFM ('point-and-click') is part of the reason why AFM has become a more widely used research tool in cell mechanics.

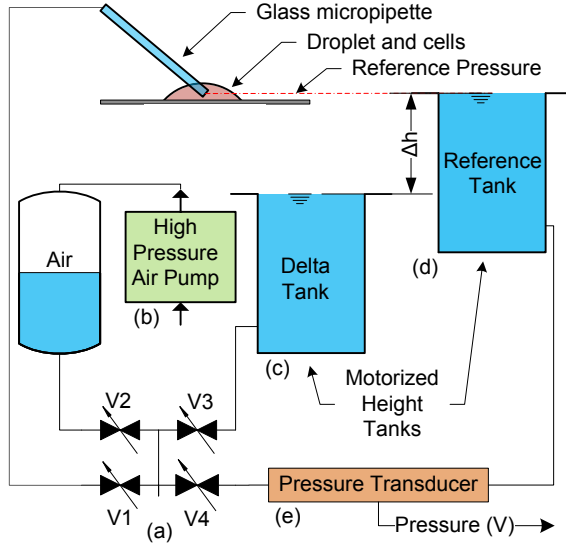
This study aims to prove the feasibility of achieving 'point-and-click' operation in micropipette aspiration through the use of automation techniques. In [18] and our previous work [19], image processing algorithms were developed to measure cell deformation behavior. This paper reports on an automated system that performs the complete micropipette aspiration procedure via computer mouse clicking. The system controls motion control devices based on visual servoing and position control and processes images in real time for tracking features and measuring cell shape changes.

## 2 System and Control

The system (Fig. 1) consists of an inverted microscope (Olympus IX81) with a CMOS camera (Basler A601F), a motorized micromanipulator (Siskiyou, 7600 series) with encoder feedback for positioning the micropipette, an XY translation stage (Prior Scientific ProScan II) for positioning cell samples, and a custom-developed pressure system for generating computer-controlled vacuum levels using a differential pressure transducer (Omega *PX409-10WDWU5V*) to obtain pressure feedback. The host computer runs LabVIEW (National Instruments) programs to control the hardware and to process pressure and image data.

The suction pressure ( $\Delta P = \rho g \Delta h$ ) is generated by creating a height difference between the reference and delta tanks (Fig. 2). Solenoid valves (Fig. 2(a)) are used to route the fluid for various functions. For instance, when the operator mounts a new micropipette using the back filling method [18], the trapped air could be forced out by routing the high-pressure pump (Fig. 2(b)) to the micropipette tip without damaging the transducer (Fig. 2(e)) or affecting the water levels in reservoirs. Table 1 describes four functions of valves used in this system. The positions of the two water tanks are controlled by a standard proportional integral derivative (PID) controller ( $\pm 2$  Pa error) with pressure feedback from the differential pressure transducer as illustrated in Fig. 3(a).

The micromanipulator is controlled via an image-based look-and-move visual servoing algorithm [20]. The internal closed-loop controller uses encoders as input

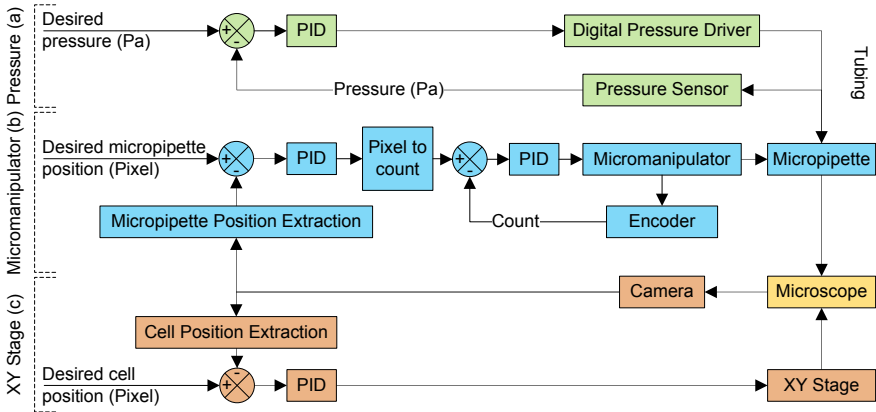


**Fig. 2** Digital pressure pump. (a) Solenoid valves are used in routing fluid, (b) High pressure air pump is used to remove trapped air from a newly mounted micropipette, (c) Delta Tank moves vertically to generate pressure. (d) Reference Tank moves vertically to balance the pressure in Delta Tank. (e) Differential pressure transducer outputs the pressure difference between Delta and Reference Tanks in Volts.

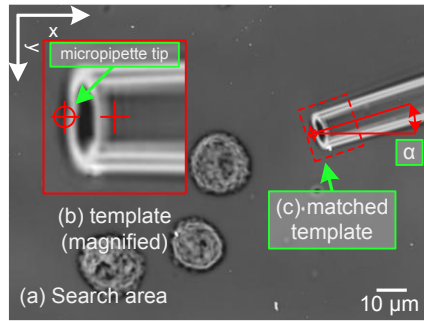
with a sampling rate of 1 kHz, whereas the visual servoing controller has a low sampling rate of 30 Hz (Fig. 3(b)). Two sets of PID gains for position errors larger than  $3 \mu\text{m}$  and within the proximity of the cell are used [21] to control the micromanipulator in order to produce smooth motion and minimize micropipette vibrations. Cells in a standard Petri dish are placed on the motorized XY translation stage that is also controlled via image-based look-and-move visual servoing, as shown in Fig. 3(c). The PID controller sends velocity commands to the stage amplifier.

### 3 Micropipette Tracking

The micropipette tip is visually tracked using an optimized rotation-invariant normalized cross-correlation (NCC) method. The basic NCC algorithm was described in [22]. A template of the micropipette tip (Fig. 4(a)) is cross-correlated with the current frame of image to locate the best match within a search area. The selection of a template is required only when a new micropipette is mounted on the micromanipulator in the beginning of an experiment. Once the template is selected, the search area is limited to the proximity of the last known location of the micropipette from the previous frame. This method reduces the search time. The normalized cross-correlation algorithm determines the location and the angle of the matched template  $(x, y, \alpha)$ , as shown in Fig. 4(c).



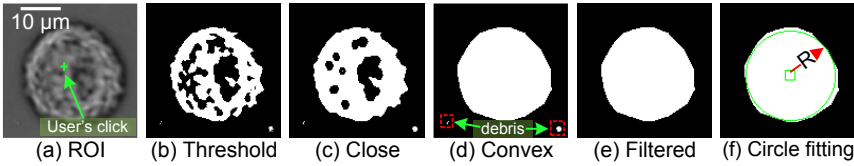
**Fig. 3** System control architecture. (a) PID position controller for controlling the pressure system. (b) Image-based look-and-move visual servoing controller for controlling the micromanipulator. (c) PID position controller for controlling the motorized XY translation stage.



**Fig. 4** Micropipette tracking. (a) Search with the normalized cross-correlation algorithm within the search area. (b) User selects a template containing the micropipette tip. (c) Matched template location and angle ( $x, y, \alpha$ ).

## 4 Cell Tracking

There are typically several cells present within a field of view. Thus, the cell tracking algorithm requires the user to indicate which cell is the target for characterization via one computer mouse click anywhere on top of the target cell (see green cross in Fig. 5(a)). The cell tracking algorithm uses binary morphology techniques [23] to determine the position of the cell. The system then forms a region of interest (ROI) that is binarized with the Otsu adaptive thresholding method (Fig. 5(b)). Close binary operation is used to create a smooth object on the outside (Fig. 5(c)). Convex

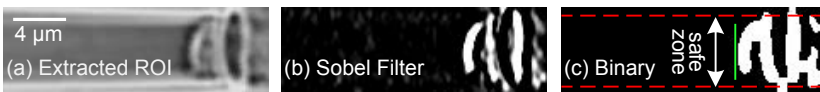


**Fig. 5** Cell tracking. (a) ROI formed around user's click (green cross). (b) Binarized image with the Otsu adaptive thresholding method. (c) Morphological close operation creates a smooth object. (d) Morphological convex operation fills the holes and create an envelope of the cell. (e) Noise removal based on object area. (f) Least square fitting to obtain a circle  $(x, y, R_c)$ .

hull is used to obtain the cell's envelope (Fig. 5(d)), and objects with small areas are removed (Fig. 5(e)). The system determines the position and radius of the cell,  $(x, y, R_c)$  (Fig. 5(f)) using least squares fitting.

## 5 Aspiration Length Tracking

When a cell is aspirated into the micropipette, an ROI is formed to include the inside of the micropipette, as shown in Fig. 6(a). The extracted ROI is then convolved with a standard  $5 \times 5$  West Sobel kernel to eliminate the vertical edges and enhance the horizontal contours (Fig. 6(b)). The result of the gradient filter is binarized using the Otsu method, as shown in Fig. 6(c). After binary morphology close, the objects found in the image are filtered according to two criteria: (1) object is removed if the area is smaller than a threshold value; and (2) object is removed if the center of mass is not in the 'safe' zone (within the micropipette), as shown in Fig. 6(c). The first criterion filters out noises and small background debris. The second criterion ensures that the attached debris to the wall of the micropipette is not recognized as cell contour. The left most object is recognized as the protruded cell contour (see green line in Fig. 6(c)).

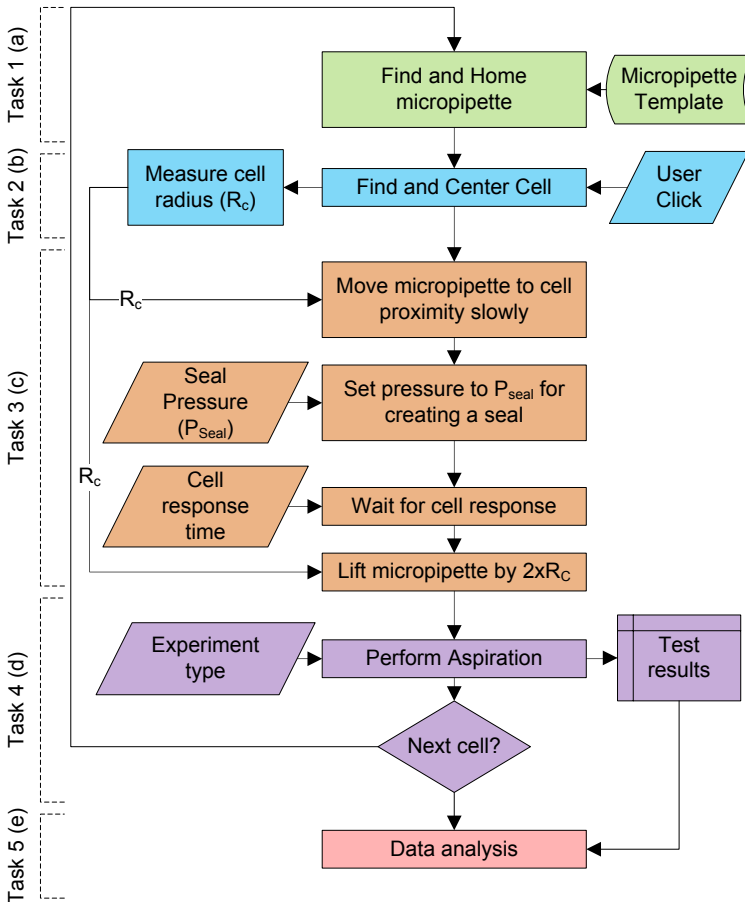


**Fig. 6** Cell aspiration length tracking. (a) ROI containing the inside of the micropipette. (b) Sobel filter is used to enhance the edges and remove vertical edges. (c) The tracked aspiration length is highlighted in green, and the area between the dashed red lines is termed 'safe zone'.

## 6 Experimental Results

The porcine aortic valve interstitial cells (PAVICs) used in the experiments were harvested from aortic valve leaflets from porcine. The PAVICs were resuspended





**Fig. 7** Automated micropipette aspiration flow chart. (a) Task 1 finds and park the micropipette in home position. (b) Task 2 selects a target cell. (c) Task 3 creates a seal between the mouth of the micropipette and the cell. (d) Task 4 generates a specific pressure profile based on the experiment type. (e) Task 5 performs data analysis and calculates biomechanical properties of the tested cell.

in standard tissue culture medium (DMEM supplemented with 10% FBS and 1% antibiotics). When the cultured cells were confluent, the cells were resuspended in standard DMEM using trypsinization method. The human promyelocytic leukemia cells (HL-60) used in the experiments were purchased from ATCC (CCL-240), which were obtained by leukopheresis from a 36-year-old female with acute promyelocytic leukemia. The cells were subcultured by replacement of medium (as recommended by ATCC) every two days.

The system performs cell aspiration according to the following protocol consisting of five steps (Fig. 7): (1) find and home micropipette based on a pre-defined micropipette template; (2) center a user-selected cell in the field of view; (3) create a seal between cell and micropipette tip based on a pre-defined pressure,  $P_{seal}$ ; (4) perform aspiration; and (5) process data. There are three types of experiments that a user can choose to conduct: (a) characterization of elastic properties of solid-like cells; (b) characterization of viscoelastic properties for solid-like cells; (c) viscous properties of liquid-like cells.

In our experiments, the system was used to measure the elastic and viscoelastic properties of PAVICs. It has been shown that the mechanical properties of aortic valves play an important role in the heart's functionality [24, 25]. The regulatory valve function is also affected by the valve cells. For testing liquid-like cells, HL-60 cells were used for viscosity characterization. Leukocytes are known to play a significant role in blood flow and oxygen delivery in micro-capillaries. Less deformable cells induce impairment to the circulation to the eye and the central nervous system causing damage to such organs [26]. The PAVIC and HL-60 cells were also chosen for evaluating our automated micropipette aspiration system's performance because the characterization results of these cells using manual micropipette aspiration are available in the literature.

## 6.1 Elastic Characterization

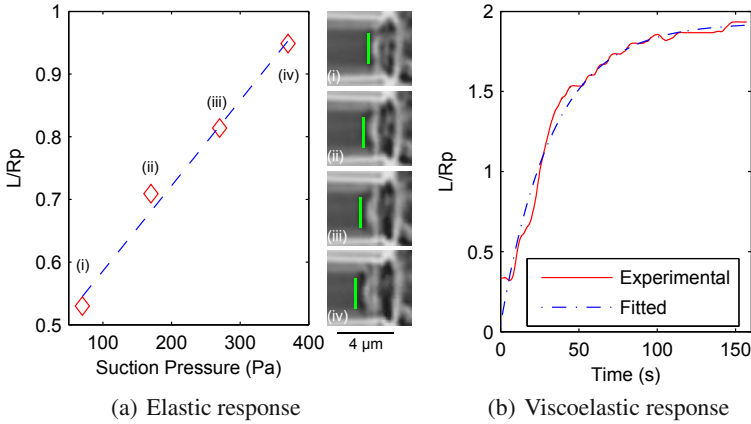
In order to measure the Young's modulus of the cells, the system applied a series of small pressure steps (4 steps of -100 Pa) to the cell with an interval of 100 seconds via the digital pressure system with an initial seal pressure of -70 Pa. The aspiration length ( $L$ ) and step pressure ( $\Delta P$ ) are related by the half space model [14]

$$\frac{L}{R_p} = \frac{3\Delta P}{2\pi E} \Phi_p(\eta) \quad (1)$$

where  $E$  is the Young's modulus,  $\Phi_p$  for our used micropipette is approximately 2.1, and  $\eta$  is the wall parameter. The aspiration length,  $L$  and suction pressure,  $\Delta P$  (Fig. 8(a)) at the end of each 100-second interval were used for linear regression to obtain the Young's modulus from the slope of the fitted line. A robust linear regression was used to eliminate the effect of outliers. The determined Young's modulus value for the tested PAVICs is  $358.7 \pm 192$  Pa ( $n=30$ ). This result is in agreement with our previous results ( $354.8 \pm 142$  Pa) from manual testing [19] and agrees well with the results reported in [25].

## 6.2 Viscoelastic Characterization

We also used the system for quantifying viscoelastic properties of the PAVICs. In viscoelastic characterization, the system applied to the cell a large step pressure (-400 Pa) for 150 seconds. In this period, the system on-line measured the cell aspiration length,  $L(t)$ , which is related to step pressure input ( $h$ ) by the Kelvin model [15]



**Fig. 8** Elastic and viscoelastic responses of PAVICs.  $L$  is the aspiration length and  $R_p$  is the radius of the micropipette. (a) Elastic response of a PAVIC, where (i)-(iv) are equilibrium points at the end of 100-second intervals. (b) Viscoelastic response of a PAVIC. A step pressure of  $-400$  Pa was applied for 150 seconds.

$$L(t) = L_s \left[ 1 - \frac{k_2}{k_1 + k_2} \exp\left(\frac{-t}{\tau}\right) \right] h(t) \tag{2}$$

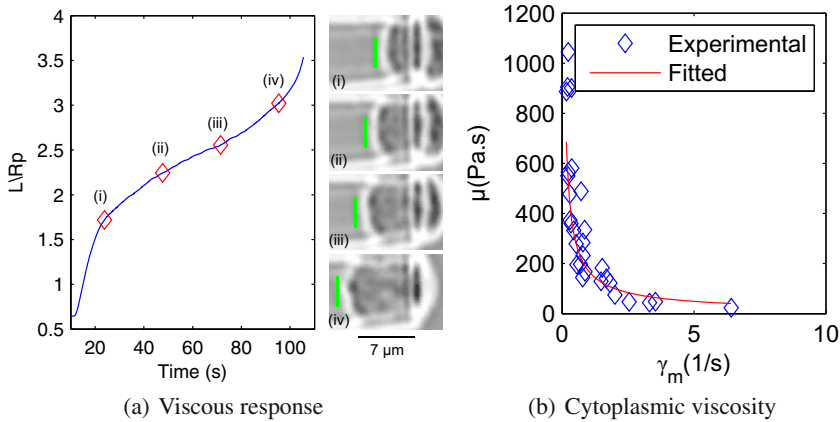
where  $L_s$  is the aspiration length at the end of the interval. Using least-squares nonlinear fitting,  $L(t)$  was fitted to the recorded data to determine  $k_1$ ,  $k_2$ , and  $\tau$  (Fig. 8(b)), where  $k_1$  and  $k_2$  are the elastic constants of a standard linear body, and  $\tau$  is the time constant. The results are summarized in Table 2 together with the previously reported values from manual testing [15].

**Table 2** Determined viscoelastic parameters of PAVIC cells

Viscoelastic Parameters	$k_1$	$k_2$	$\tau$
Curve fitting results ( $n = 25$ )	$189.6 \pm 71.3$	$270.4 \pm 68.6$	$40.0 \pm 34.2$
Literature reported values [15]	$153.8 \pm 59.9$	$310 \pm 70.4$	$45.30 \pm 11.6$

### 6.3 Viscosity Characterization

To quantify viscosity of liquid-like cells, the automated micropipette aspiration system applied step pressures ranging from  $-700$  Pa to  $-220$  Pa for 150 seconds to HL-60 cells. In this period, the system automatically measured aspiration lengths,  $L(t)$ , as shown in Fig. 9. Using a power-law fluid model [26, 27], the shear rate dependence of cytoplasmic viscosity is



**Fig. 9** Viscous properties of HL-60 cells. (a) Viscous response of an HL-60 cell to a step pressure of  $-270$  Pa. (i)-(iv) show the cell at various time instances.  $L$  is the aspiration length and  $R_p$  is the radius of the micropipette. (b) Cytoplasmic viscosity as a function of mean shear rate  $\gamma_m$ . Step pressure varied from  $-700$  Pa to  $-220$  Pa ( $n = 30$ ).

$$\mu = \mu_c (\gamma_m / \gamma_c)^{-b} \quad (3)$$

where  $\mu$  is the cytoplasmic viscosity,  $\gamma_m$  is the mean shear rate during cell entry into the micropipette,  $\mu_c$  is the characteristic viscosity at the characteristic shear rate  $\gamma_c$ , and  $b$  is a material coefficient. In this study,  $\gamma_c$  was set to  $1 \text{ s}^{-1}$ . Characterization results are  $\mu_c = 171.35 \pm 13.04 \text{ Pa.s}$ , and  $b = 0.78 \pm 0.08$  for HL-60 cells ( $n = 30$ ). These values are in agreement of the previously reported value [26].

## 7 Discussion

For comparison purposes, manual micropipette aspiration was also performed by a proficient operator. In manual operation, the user looked into the eyepieces of the microscope and controlled the micropipette positions using a joystick. Comparing the speed of locating and homing the micropipette, the system was at least two times faster than the proficient operator (19 vs. 60 seconds). The system was also at least twice as fast in forming a seal between a cell and the micropipette (23 vs. 52 seconds). The time taken in the cell aspiration step was comparable since sufficient waiting time was required for cells to respond; however, we believe that the pressure profile was generated more accurately by the closed-loop controlled pressure pump in the automated system. It was observed that the cells experience a consistent seal forming experience due to use of accurately controlled pressure profile.

It would take much longer for less trained users in these steps in manual operation, while the automated system's performance and operation speed are user independent. In image processing, the system determined cell deformation parameters during cell aspiration in real time without the need for tedious, lengthy post

processing. In comparison, it took the proficient user averagely 2 minutes per cell to manually make measurements in recorded images for elastic measurements and approximately 2 hours for viscoelastic characterization due to the high number of images captured.

## 8 Conclusion

This paper presented an automated micropipette aspiration system. The system performance was evaluated by characterizing elastic and viscoelastic properties of PAVICs and HL-60 cells representing solid and liquid-like cell models. Experiments demonstrate that the system enables single-cell micropipette aspiration with a higher efficiency and higher accuracy. System performance is operator skill independent and is achieved by 'point-and-click'. Currently the system tests a single cell per user intervention (click). However the user can select several cells initially after which the system sequentially tests the selected cells without need of further user input. Automated cell type recognition will add value by allowing unsupervised investigation of a large number of cells in an area of interest. Currently the system assumes that the motion of the micropipette does not affect the location of the target cell. This assumption holds true in a clean sample in which only cells are suspended in the medium. However, when other particles present in the medium, new control algorithms should be developed for micropipette positioning.

**Acknowledgements.** The authors acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada and Canada Research Chairs Program.

## References

1. Kim, D.H., Wong, P.K., Park, J., Levchenko, A., Sun, Y.: *Annual Review of Biomedical Engineering* 11, 203 (2009)
2. Lee, G.Y.H., Lim, C.T.: *Trends in Biotechnology* 25(3), 111 (2007)
3. Cross, S.E., Jin, Y.S., Rao, J., Gimzewski, J.K.: *Nature Nanotechnology* 2(12), 780 (2007)
4. Lu, Z., Moraes, C., Ye, G., Simmons, C.A., Sun, Y.: *PloS One* 5(10), 13542 (2010)
5. Suresh, S.: *Acta Biomaterialia* 3(4), 413 (2007)
6. Lim, C., Zhou, E., Li, A., Vedula, S., Fu, H.: *Materials Science and Engineering: C* 26(8), 1278 (2006)
7. Haupt, B.J., Pelling, A.E., Horton, M.A.: *The Scientific World Journal* 6, 1609 (2006)
8. Fabry, B., Maksym, G., Butler, J., Glogauer, M., Navajas, D., Fredberg, J.: *Physical Review Letters* 87(14), 1 (2001)
9. An, S.S., Fabry, B., Trepatt, X., Wang, N., Fredberg, J.J.: *American Journal of Respiratory Cell and Molecular Biology* 35(1), 55 (2006)
10. Bao, G., Suresh, S.: *Nature Materials* 2(11), 715 (2003)
11. Mills, J.P., Qie, L., Dao, M., Lim, C.T., Suresh, S.: *Mechanics & Chemistry of Biosystems: MCB* 1(3), 169 (2004)
12. Hochmuth, R.M.: *Journal of Biomechanics* 33(1), 15 (2000)

13. Evans, E., Yeung, A.: *Biophysical Journal* 56(1), 151 (1989)
14. Theret, D.P., Levesque, M.J., Sato, M., Nerem, R.M., Wheeler, L.T.: *Journal of Biomechanical Engineering* 110(3), 190 (1998)
15. Sato, M., Theret, D.P., Wheeler, L.T., Ohshima, N., Nerem, R.M.: *Journal of Biomechanical Engineering* 112(3), 263 (1990)
16. Needham, D., Hochmuth, R.M.: *Journal of Biomechanical Engineering* 112(3), 269 (1990)
17. Shao, J.Y., Hochmuth, R.M.: *Microcirculation (New York, N.Y.: 1994)* 4(1), 61 (1997)
18. Heinrich, V., Rawicz, W.: *Langmuir: the ACS Journal of Surfaces and Colloids* 21(5), 1962 (2005)
19. Liu, X., Wang, Y., Sun, Y.: *IEEE Transactions on Automation Science and Engineering* 6(3), 536 (2009)
20. Hashimoto, K.: *Advanced Robotics* 17(10), 969 (2003)
21. Rugh, W.J., Shamma, J.S.: *Automatica* 36(10), 1401 (2000)
22. Lewis, J.P.: *Vision Interfacem. Canadian Image Processing and Pattern Recognition Society*, vol. 10, pp. 120–123. Citeseer (1995)
23. Dougherty, E.R., Lotufo, R.A.: *Hands-on Morphological Image Processing*. SPIE, 1000 20th Street, Bellingham, WA 98227-0010 USA (2003)
24. Simmons, C.A.: *Journal of the American College of Cardiology* 53(16), 1456 (2009)
25. Merryman, W.D., Youn, I., Lukoff, H.D., Krueger, P.M., Guilak, F., Hopkins, R.A., Sacks, M.S.: *American Journal of Physiology. Heart and Circulatory Physiology* 290(1), H224 (2006)
26. Tsai, M.A., Waugh, R.E., Keng, P.C.: *Biophysical Journal* 70(4), 2023 (1996)
27. Tsai, M.A., Frank, R.S., Waugh, R.E.: *Biophysical Journal* 65(5), 2078 (1993)

# Quantitative Analysis of Locomotive Behavior of Human Sperm Head and Tail

Jun Liu, Zhe Lu, Clement Leung, and Yu Sun

**Abstract.** Sperm selection plays a significant role in *in vitro* fertilization (IVF). Approaches for assessing sperm quality include non-invasive techniques based on sperm morphology and motility as well as invasive techniques for checking DNA integrity. In 2006, a new device using hyaluronic acid (HA) coated dish for sperm selection was cleared by the Food and Drug Administration (FDA) and entered IVF clinics. In this technique, only sperms with DNA integrity bind to the HA droplet, after which these bound sperm stop revealing head motion and their tail movement becomes more vigorous. However, selecting a single sperm cell from among HA-bound sperms is *ad hoc* in IVF clinics. Different from existing sperm tracking algorithms that are largely limited to tracking sperm head only and are only able to track one sperm at a time, this paper presents a multi-sperm tracking algorithm that tracks both sperm heads and low-contrast sperm tails. The tracking results confirm a significant correlation between sperm head velocity and tail beating amplitude; demonstrate that sperms bound to HA generally have a higher velocity (before binding) than those sperms that are not able to bind to HA microdots; and quantitatively reveal that HA-bound sperms' tail beating amplitudes are different among HA-bound sperms.

## 1 Introduction

The mechanisms of natural sperm selection are not well understood. It is accepted, however, that the sperm selection mechanisms play a great significant role for the inheritance of superior health traits such as disease resistance, offspring survival and fecundity [1] [2]. In natural human conception, sperm selection occurs as a healthy

---

Jun Liu · Zhe Lu · Clement Leung · Yu Sun  
University of Toronto,  
5 King's College Road, Toronto, ON, Canada, M5S3G8  
e-mail: {ljun, sun}@mie.utoronto.ca,  
{zhe.lu, clement.leung}@utoronto.ca

sperm actively seeks out and fertilizes an egg. However, for couples having infertility issues, assisted reproduction technologies are required to address their reproductive needs. For instance, in intracytoplasmic sperm injection (ICSI), an embryologist selects a single sperm and injects it into an oocyte (i.e., egg cell) to overcome issues such as male infertility [3]. This IVF procedure bypasses the physiologic and biologic barriers for sperm selection and demands the operator to select high-quality sperms.

The criteria for sperm assessment provided by the World Health Organization are vitality, morphology, and motility [4]. In IVF, sperm selection is commonly based on sperms' motility and morphology attributes. A widely used method for sperm selection is motile sperm organelle morphology examination [5] [6]. Sperm motility is also a widely accepted criterion for sperm quality assessment. The past few decades have witnessed the development of computer-assisted sperm analysis (CASA) methods for measuring both sperm morphology and motility [7].

Recently, an emerging methodology was introduced for selecting viable sperms with a high level of DNA integrity. In order to non-invasively select a healthy sperm, Huszar's group proposed the use of a hyaluronic acid (HA) assay [8]. HA is a linear polysaccharide in the extracellular matrix of cumulus oophorus around the oocyte and plays an important role in natural human fertilization [9]. A series of studies on HA-based sperm selection confirmed that the HA assay is able to select healthy sperms with no DNA damage [10] [11], and has received FDA approval. In the HA assay, sperms that bind to HA microdots are proven to have a higher level of DNA integrity compared to those unbound sperms. When a sperm binds its head to an HA microdot, the sperm loses its progressive movement and the tail beating motion becomes more vigorous. The sperm tail beating amplitude becomes the only parameter to differentiate the HA bound sperms from each other, calling for techniques to quantify HA-bound sperms' tail beating motion.

Several algorithms have been developed to track sperm trajectories, measure sperm velocities, and evaluate sperm energetics [12]-[14]. Shi *et al.* reported a single-sperm tracking algorithm based on a four-class thresholding method to extract a single sperm in a small region of interest [15]. The method is limited to tracking a single sperm and is incapable of multi-sperm tracking. Nafisi *et al.* demonstrated a template matching algorithm for sperm tracking. The algorithm is insensitive to image acquisition conditions [16]. However, this algorithm relies on user input to obtain the sperm's initial position and cannot track multiple sperms. Existing algorithms for sperm tracking are largely limited to sperm head tracking. The small size ( $\leq 1\mu\text{m}$  in thickness) and low contrast of sperm tails under optical microscopy make sperm tail tracking challenging. In our previous study [17], a maximum intensity region algorithm was developed for sperm tail tracking. The tracking algorithm, without proper filtering, can be susceptible to disturbances, such as overlapping of the target sperm with other sperms or debris and changes in lighting conditions.

In this paper, we report an approach for tracking both sperm head and tail. We use a motion template method to detect and track multiple moving sperms, and integrate a Kalman filter to the maximum intensity region algorithm to locate the sperm



tail's position. With the positions of the sperm head and tail detected, the sperm's velocity and tail beating amplitude were measured. Experimental results demonstrate that there is a significant correlation between the sperm velocity and its tail beating amplitude. We also analyzed sperm motility and the tail beating movement on HA coated dishes. We found sperms with a higher level of motility are more likely to bind to the HA microdots, and the sperm tail beating amplitude significantly increases after a sperm binds to the HA microdots. Quantitative analysis of sperm tail's beating amplitude can provide useful information for sperm selection.

## 2 Experiment Setup and Design

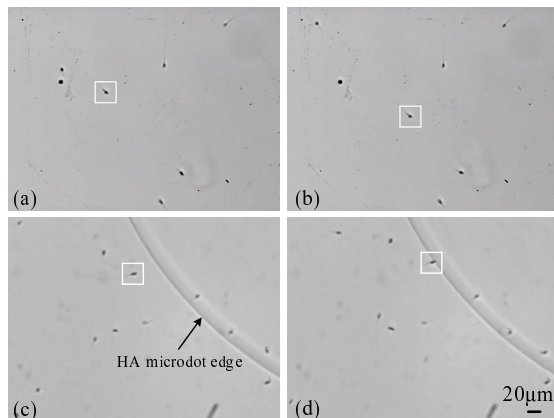
### 2.1 Experiment Setup

Human sperm samples were tested under a standard inverted microscope (bright field imaging, Nikon TE2000-S). A 20 $\times$  objective with a numerical aperture of 0.45 was used (CFI Plan Fluor ELWD, Nikon). A CMOS camera (601f, Basler; resolution: 640 $\times$ 480) was connected to the microscope to capture images at a frame rate of 30 frame/second.

### 2.2 Experiment Design

We first investigated the correlation between sperm velocity and sperm tail beating amplitude. In this experiment, human sperms were placed in a Petri dish containing a standard medium (SpermCatch, NidaCon International). Mineral oil was used to cover the medium to prevent evaporation. Fig. 1(a) and 1(b) show a sperm changed its position between the two image frames. Sperms that had linearly progressive movement for at least two seconds were taken into consideration. To evaluate the linearity of the sperm motion, sperm's average curvilinear velocity and straight line velocity were calculated by measuring the sperm's head position in each frame.

**Fig. 1** Experiment design. (a)(b) Experiment 1: a sperm changed its position between two frames. (c)(d) Experiment 2: a sperm was moving towards an HA microdot and then bound to the HA microdot.



Additionally, the sperm tail beating amplitude was measured using the sperm tail tracking algorithm.

The second experiment was designed to investigate the difference of sperm velocity and tail beating amplitude between HA bound sperms and those unbound ones. Human sperms were analyzed in a PICSi dish (MidAtlantic diagnostic, Mount Laurel, NJ, USA) with HA microdots coated on the dish bottom. The microdots were first hydrated by placing 10  $\mu$ l droplets of Human Tubal Fluid, and were added with 10  $\mu$ l SpermCatch. Human sperm was then placed on the HA microdots, which were covered with mineral oil to prevent the culture medium from evaporation. The experiment was conducted at room temperature. Some sperms were observed to bind to the microdots after 5 minutes. Fig. 1(c) shows a sperm was moving towards an HA microdot, and Fig. 1(d) shows the same sperm bound to the HA microdot. When a sperm bound to the HA microdot, its head motion stopped and its tail beat vigorously. In contrast, sperms that might have DNA defects, swam freely on top of the HA microdots without binding. In this experiment, the head velocity of HA bound sperms before binding and the head velocity of unbound sperms were measured and compared. The increase of sperm tail beating amplitude after a sperm binds to an HA microdot was also measured.

### 3 Sperm Tracking

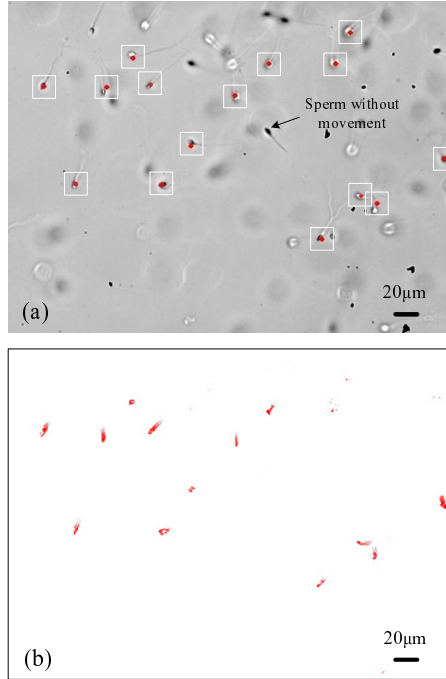
#### 3.1 Overview

In both experiments described in the above section, the sperm velocity and tail beating amplitude are calculated in every frame of image. The algorithm consists of three steps. The first step tracks the sperm head and uses its position to calculate the sperm curvilinear velocity (VCL), straight line velocity (VSL), and linearity of the sperm's moving path. In the second step, the sperm tail region of interest (STROI) is extracted. STROI extraction is an extrapolation process that calculates the region in which the sperm tail is located by using information from the first step. The STROI is used to capture the tail tip region of the sperm. Once the STROI is found, the maximum intensity region (MIR) algorithm is used to locate a point on the sperm tail within the STROI. Finally, a Kalman filter is used to improve the accuracy of the located point on the sperm tail.

#### 3.2 Sperm Head Tracking

There are typically multiple sperms moving randomly within a field of view. In order to detect the sperm heads simultaneously, a multi-target tracking algorithm was developed. For a specific single sperm, a silhouette image of this sperm is obtained by subtracting two consecutive frames. The silhouette image is then binarized by applying a threshold to suppress the background noise. When this sperm cell moves, new silhouettes are captured and overlaid to the old silhouette. The sequentially fading silhouettes record the motion history of this sperm. Using this method, a motion

**Fig. 2** Multi-sperm head tracking. (a) Sperm head tracking results. (b) Corresponding motion history image.



history image (MHI) of all the moving objects in the same field of view is obtained, as shown in Fig. 2(b). The position of each moving object is calculated from its central moment in the MHI. Among these moving objects, there are some objects with very little motion caused by those sperms with extremely low motility or by Brownian motions of debris. These objects are excluded by applying morphological transformations (i.e., erosion and dilation) to the MHI. Fig. 2(a) shows the detected moving sperms.

To track multiple sperms, the position history of each sperm is recorded and managed in the multiple sperm tracking algorithm, as summarized in Table 1. When  $J$  sperms are detected at frame  $i$ ,  $P_{ij}$  represents position of  $j$ -th sperm at frame  $i$ . When a sperm moves close to the edge of the image and its moving direction is towards the boundary, it is considered swimming out of the field of view. The algorithm then terminates the corresponding data column for this sperm. As shown in Table 1, the second column disappears after frame  $i + 1$ , which means the second sperm has moved out the field of view. On the other hand, when a sperm newly moves into the field of view, it is added into the data column. For instance, a new sperm moves into the field of view at frame  $i + 3$ . Correspondingly, a new column is added (see last column in Table 1).

To track an individual sperm, the direction vector of the sperm is used as a unique identifier to distinguish it from other sperms. By denoting the position of the  $j$ -th sperm at  $i + 1$  frame as  $P_{i+1,j}$ , the sperm's direction vector is  $D_{ij} = P_{i+1,j} - P_{i,o}$ , where  $o \in [1, J]$ . We found experimentally that the maximum moving distance of

**Table 1** Dynamic representation of sperm positions in the multi-sperm tracking algorithm

Position Frame	Sperm				
	1	2	...	$J$	$J + 1$
$i$	$P_{i1}$	$P_{i2}$	...	$P_{iJ}$	
$i + 1$	$P_{i+1,1}$	$P_{i+1,2}$	...	$P_{i+1,J}$	
$i + 2$	$P_{i+2,1}$		...	$P_{i+2,J}$	
$i + 3$	$P_{i+3,1}$		...	$P_{i+3,J}$	$P_{i+3,J+1}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$

fast sperms between two consecutive frames is approximate  $3.86 \mu\text{m}$  (i.e., 8 pixels). Therefore, if the distance between the  $j$ -th sperm's position at frame  $i + 1$  and the  $o$ -th sperm's position at frame  $i$  is less than 8 pixels (i.e.,  $\|D_{ij}\| < 8$ ), these two sperms are recognized as the same sperm (i.e.,  $o = j$ ). There may be instances in which two or more sperms at frame  $i + 1$  have a distance less than 8 pixels to the  $j$ -th sperm in frame  $i$ . The average movement direction in previous 30 frames,  $\overline{D}_j$ , is then used as a unique identifier to determine which sperm is the correct  $j$ -th sperm.

$$\overline{D}_j = \frac{1}{30} \sum_{k=1}^{30} D_{i-k,j} \tag{1}$$

The candidate sperm  $s$  that produces the minimum Euclidean distance value is considered the same  $j$ -th sperm in the previous frame  $i$ .

$$s = \min_{m \in [1,M]} \|D_m - \overline{D}_j\| \tag{2}$$

where  $M$  is in frame  $i + 1$  the total number of sperms close to the  $j$ -th sperm (i.e.,  $\|D_{ij}\| < 8$ ); and  $D_m$  is the distance vector between the candidate sperm and the  $j$ -th sperm.

Assume the  $j$ -th sperm enters the field of view at frame  $i$  and swims out of the field of view at frame  $i + N$ . With the sperm position detected in each frame, the travel distance of the  $j$ -th sperm between two consecutive frames can be determined from its direction vector,  $D_{ij}$ . The curvilinear velocity (*VCL*), which is the average velocity of the sperm head along its actual curvilinear path, is

$$VCL_j = \frac{1}{N} \sum_{k=0}^{N-1} D_{i+k,j} \tag{3}$$

The straight line velocity (*VSL*), which is the average velocity of the sperm head along the straight line between its first and last detected position, is

$$VSL_j = \frac{P_{i+N,j} - P_{ij}}{N} \quad (4)$$

The linearity ( $LIN$ ) of the sperm's curvilinear path is

$$LIN = \frac{VSL}{VCL} \quad (5)$$

where  $LIN$  is the linearity measure ( $0 \leq LIN \leq 1$ ). A higher  $LIN$  value means that the sperm's moving path is more linear. Healthy energetic sperms with progressive/linear movement are desired (vs. those traveling in circles for instance) in sperm selection. In our experiments, only those sperms having a  $LIN$  value greater than 0.9 were considered for further analysis.

### 3.3 Sperm Tail Tracking

After the sperm head position is detected, the sperm tail tracking algorithm extracts a sperm tail region of interest (STROI). As shown in Fig. 3(b), the STROI is determined using the sperm head position and the average direction vector of its movement. The average direction vector,  $\overline{D}_j$  is used instead of the direction vector  $D_{ij}$  because the sperm may exhibit abrupt changes in movement direction between two consecutive frames. By averaging the direction vectors of the sperm across a number of frames (e.g., 30 frames), the effect of abrupt changes in the sperm moving direction between frames are mitigated and the extraction of STROI becomes more robust.

The STROI's center position in the  $i$  frame,  $T_{ij}$ , is determined by subtracting a scaled value of the direction vector from the sperm head's centroid

$$T_{ij} = P_{ij} - a \cdot \frac{\overline{D}_j}{\|\overline{D}_j\|} \quad (6)$$

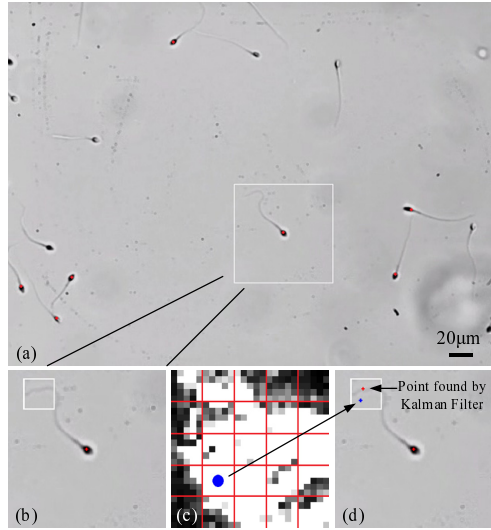
where  $a$  is a scalar value determined by the human sperm length. Under the  $20\times$  magnification, the average length of human sperms is approximately 90 pixels (i.e.  $a = 90$ ). After the center position is found, a  $25\times 25$  region of interest is taken as the STROI. The size of  $25\times 25$  provides a sufficient tail search area that takes into consideration a range of sperm tail length variations and sperm tail beating amplitudes.

After finding the STROI, the algorithm verifies that a tail is present in the STROI. The fundamental feature of flicker is extracted by taking the absolute difference between six consecutive inverted grayscale image frames.

$$f(i) = \sum_{k=0}^5 |I(i-k) - I(i-k-1)| \quad (7)$$

where  $f(i)$  is the flicker image extracted at frame  $i$ , and  $I$  represents the grayscale images containing the sperm of interest in frame  $i$  to frame  $i-5$ . Each pixel in the flicker image is squared to enhance the pixel values of areas in which the tail is

**Fig. 3** Sperm tail tracking. (a) Sperm head position is found. (b) STROI (sperm tail region of interest) is determined. (c)  $5 \times 5$  windows are scanned to locate the section with the highest intensity sum in the flicker image. The center point (blue dot in figure) of the section is considered the tail location. (d) Based on the blue dot position found in (c), Kalman filter is applied to improve the accuracy of located sperm tail position.



present. The sum of pixel value in the STROI of the  $f(i)$  image is used as a measure to determine the presence of a sperm tail. If the pixel sum is above a specified threshold value, a tail is considered present. The threshold value was found experimentally by comparing the pixel sum values of STROI images in which a tail exists against cases where no tail exists. An example flicker image is shown in Fig. 3(c). If the pixel sum is below a threshold value, no tail is found inside the STROI. This situation can occur when the sperm of interest moves out of focus, resulting in the disappearance of the sperm tail.

Once the sperm tail is determined to exist within the STROI, the MIR algorithm uses the flicker image to locate a point on the sperm tail. By extracting the flicker feature of the sperm tail, as shown in Fig. 3(c), the position of the sperm tail can be detected. This approach overcomes the challenges that arise from the low-contrast image of the sperm tail in a single frame. The algorithm first finds the location of maximum intensity within the  $25 \times 25$  STROI of the flicker image. This is accomplished by evaluating the sum of the intensity values inside a  $5 \times 5$  window at a spatial sampling interval of 5 pixels in both the  $x$  and the  $y$  coordinates of the STROI flicker image. The center position of the  $5 \times 5$  window with the highest intensity is considered the tail location (i.e., a point on the sperm tail).

The located point on the sperm tail is often inaccurate because the flicker image contains noises caused by some dark debris or by other sperms entering the STROI. Therefore, a Kalman filter is applied to correct the measured point on the sperm tail. In order to model the sperm tail motion, the sperm tail's location and velocity in the image coordinate are chosen as state variables (i.e.,  $X = [x \ y \ \Delta x \ \Delta y]^T$ ). The model of the sperm tail motion is

$$X_k = AX_{k-1} + w_k \quad (8)$$

where  $A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ ,  $w_k$  is noises affecting the actual state of the sperm and is

assumed to have a Gaussian distribution,  $(N(0, Q_k))$ .

The sperm tail's position is calculated according to

$$Z_k = HX_k + v_k \quad (9)$$

where  $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ ; and  $v_k$  is the measurement noise, which is also assumed to have a Gaussian distribution  $(N(0, R_k))$ .  $R_k$  is chosen based on the estimate of how accurately the sperm tails positions are detected using the MIR algorithm.

Based on the dynamic model, a priori estimate of the state is computed ( $X_{k|k-1} = AX_{k-1|k-1} + w_k$ ). The error covariance is denoted as  $P_{k|k-1}$ . The priori estimate for this covariance at time  $k$  is then determined by

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_k \quad (10)$$

With the priori estimate of the state  $X_{k|k-1}$  and the measurement  $Z_k$  (i.e., detected results using the MIR algorithm), the real state of sperm tail is optimized by

$$X_{k|k} = X_{k|k-1} + K(Z_k - HX_{k|k-1}) \quad (11)$$

where  $K$  is the Kalman gain and is given by

$$K = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R_k} \quad (12)$$

After the optimized sperm tail's position is found, the sperm tail beating amplitude inside the STROI is computed. The relative position inside the STROI in frame  $i$  is denoted by  $PT_i$ . The sperm tail beating amplitude,  $A$  is

$$A = \frac{1}{N} \sum_{i=1}^N \|PT_i - \overline{PT}\| \quad (13)$$

where  $\overline{PT}$  is the sperm tail's average position inside the STROI, and  $N$  is the number of frames until when the sperm tail is successfully detected.

## 4 Experimental Results and Discussion

The multi-sperm head tracking algorithm was evaluated by comparing the number of successfully tracked sperms to the actual number of swimming sperms in a 96-second long video clip captured under  $20\times$  magnification. The actual number of moving sperms in the video was carefully counted manually for verification. The experimental results show that the algorithm was able to successfully detect and

track all the 63 moving sperms before they swam out of focus. Sometimes a sperm returned to focus, then it was recognized as a new sperm.

The sperm tail tracking algorithm was evaluated by measuring the Euclidean distance error between the detected tail point and the actual sperm tail position. The actual position was carefully identified/input by a user for each frame via computer mouse clicking. Table 2 summarizes the average Euclidean distance error for five additional videos. The overall average Euclidean distance error for the MIR algorithm integrated with Kalman filtering is 1.43 pixels ( $0.69 \mu m$ ), while the error for the MIR algorithm without Kalman filtering is 1.95 pixels ( $0.95 \mu m$ ). This result indicates that Kalman filtering is effective in improving the sperm tail tracking accuracies of the MIR algorithm. When the MIR algorithm failed to track the sperm tail due to the overlap with other sperms or occlusion by debris, Kalman filtering was able to estimate the sperm tail's position based on the motion model.

To investigate the correlation between the sperm head velocity and the tail beating amplitude, we chose 30 sperms showing good progressive movements ( $LIN > 0.9$ ). The tracking algorithms described in Section III(B)(C) were used to measure their head velocity and the tail beating amplitude. We processed the measured data using a linear regression model. As shown in Fig. 4, there is a statistically significant correlation between the head velocity and sperm tail beating amplitude. Bivariate association between sperm velocity and sperm tail beating amplitude was evaluated by Pearson's correlation coefficient. Tracking data captured on the 30 sperms show that Pearson's correlation coefficient was 0.7225 and the  $p$ -value was lower than 0.0001. This result quantitatively demonstrates that the sperm's head velocity is proportional to its tail beating amplitude.

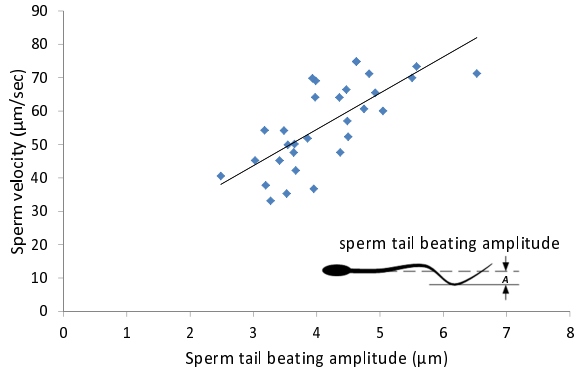
In the HA binding experiment, the heads of the HA bound sperms became stationary, and their tail movements became more vigorous. The process of sperm binding to the HA microdots was recorded at 30 frames/second. Fig. 5(a)(b) show sperms binding to the HA microdot at 5 minutes and 30 minutes. If a moving sperm was detected to stop in the region of HA microdots, this sperm was considered successfully binding to HA. In contrast, if a moving sperm passed the HA microdot and disappeared out of the image boundary, it was considered an HA-unbound sperm. During the process of sperm binding to the HA microdots, the head velocity of the HA-bound sperms before binding was measured. The HA-bound sperms were

**Table 2** Average Error of Sperm Tail Tracking

Video	V1	V2	V3	V4	V5	Average
Duration (sec)	3.83	6.67	3.83	4.60	3.87	4.56
MIR (pixels)	1.16	2.73	1.83	2.02	1.98	1.95
Kalman Optimized (pixels)	0.97	2.53	1.05	1.39	1.25	1.43



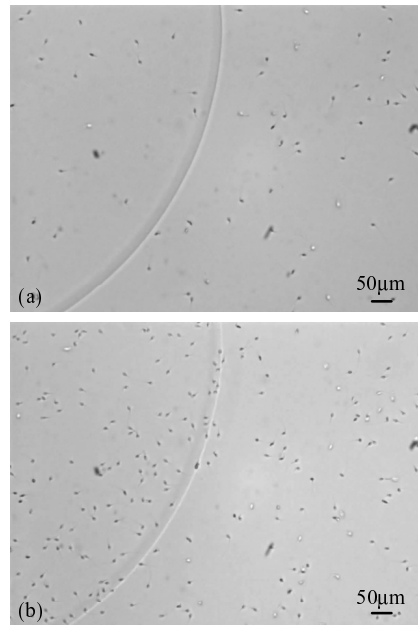
**Fig. 4** Correlation between sperm velocity and tail beating amplitude. Pearson's correlation coefficient  $r = 0.7225$ .



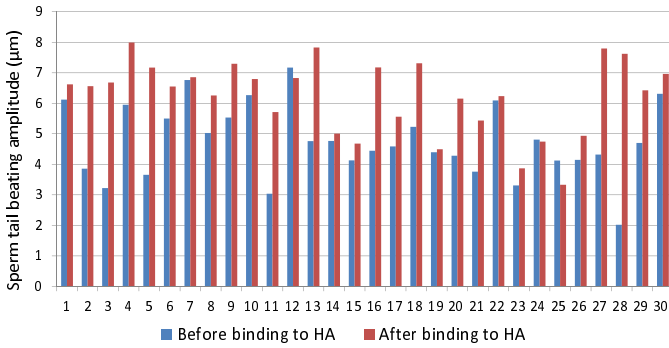
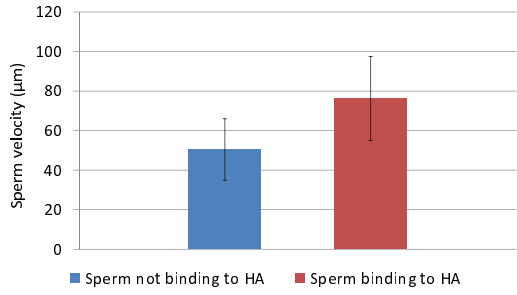
compared to those unbound ones in terms of their head speed. As in the first experiment, only those sperms that exhibited linearly progressive movement were analyzed. The result, as shown in Fig 6, demonstrates that the HA-bound sperms tended to have a higher head velocity than those unbound sperms. The HA-bound sperms had an average head velocity of  $76.28 \mu\text{m}/\text{sec}$  with a standard deviation of  $21.25 \mu\text{m}/\text{sec}$ , while the average head velocity of those unbound sperms was  $50.45 \mu\text{m}/\text{sec}$  with a standard deviation of  $15.52 \mu\text{m}/\text{sec}$ .

We also observed in the HA binding experiment that after the sperms bound to the HA microdots, their tails beat more vigorously. The tail beating amplitude was

**Fig. 5** Sperms on PICSI dish with HA microdots. (a) Sperms binding after five minutes. (b) Sperms binding after 30 minutes.



**Fig. 6** Velocity comparison between the HA-bound sperms ( $n_1 = 30$ ) and unbound sperms ( $n_2 = 30$ )



**Fig. 7** Tail beating amplitude of the same sperms before and after binding to HA microdots

measured on the same sperms before and after they bound to the HA microdots. Fig. 7 shows sperm tail beating amplitude before and after binding to the HA microdots. The average amplitude produced by these 30 HA-bound sperms was  $5.31 \mu m$  (before binding) and  $6.93 \mu m$  (after binding). These results, for the first time, quantitatively reveal increase in sperm tail beating amplitude before and after a sperm binds to an HA microdot. The results also quantifies differences in sperm tail beating amplitude across HA-bound sperms. The measurement of sperm tail beating amplitude can possibly be used as an additional criterion for sperm selection among HA-bound sperms.

### 5 Conclusion

This paper presented visual tracking algorithms for tracking both the head and tail of motile human sperms. The sperm head tracking algorithm is capable of tracking multiple moving sperms with a high success rate. Based on the sperm head's position and its motion direction vector, the sperm tail region of interest is located. In this region of interest, the MIR algorithm together with Kalman filtering determines the sperm tail position. The sperm head and tail tracking algorithms enabled a number of new findings. A significant correlation between sperm head velocity and tail beating amplitude was found, suggesting that stronger tail propelling

produces a higher velocity. The results also reveal that sperms bound to HA generally have a higher velocity (before binding) than those sperms that are not able to bind to HA microdots. This discovery ‘unifies’ the conventional sperm assessment criterion based on sperm velocity/motility and the most recent HA assay technique. Among the sperms bound to HA microdots, their tails produce different beating amplitudes. Measuring such amplitude differences quantitatively can possibly be used as a new, useful sperm selection criterion among HA bound sperms.

## References

1. Holt, W., Van Look, K.J.W.: Concepts in sperm heterogeneity, sperm selection and sperm competition as biological foundations for laboratory tests of semen quality. *Reproduction* 127(5), 527–535 (2004)
2. Ramlau-Hansen, C.H., Thulstrup, A.M., Olsen, J., Bonde, J.P.: Parental Subfecundity and Risk of Decreased Semen Quality in the Male Offspring: A Follow-up Study. *American Journal of Epidemiology* 167(12), 1458–1464 (2008)
3. Palermo, G., Joris, H., Devroey, P., Van Steirteghem, A.C.: Pregnancies after intracytoplasmic injection of single spermatozoon into an oocyte. *Lancet* 340(8810), 17–18 (1992)
4. Cooper, T.G., Noonan, E., Von Eckardstein, S., Auger, J., Gordon Baker, H.W., Behre, H.M., Haugen, T.B., Kruger, T., Wang, C., Mbizvo, M.T., Vogelsohn, K.M.: World Health Organization reference values for human semen characteristics. *Human Reproduction Update* 16(3), 231–245 (2009)
5. Bartoov, B., Eltes, F., Pansky, M., Lederman, H., Caspi, E., Soffer, Y.: Estimating fertility potential via semen analysis data. *Human Reproduction* 8(1), 65–70 (1993)
6. Berkovitz, A., Eltes, F., Yaari, S., Katz, N., Barr, I., Fishman, A., Bartoov, B.: The morphological normalcy of the sperm nucleus and pregnancy rate of intracytoplasmic injection with morphologically selected sperm. *Human Reproduction* 20(1), 185–190 (2005)
7. Larsen, L., Scheike, T., Jensen, T.K., Bonde, J.P., Ernst, E., Hjollund, N.H., Zhou, Y., Skakkebaek, N.E., Giwercman, A.: Computer-assisted semen analysis parameters as predictors for fertility of men from the general population. *Human Reproduction* 15(7), 1562–1567 (2000)
8. Huszar, G., Ozenci, C.C., Cayli, S., Zavaczki, Z., Hansch, E., Vigue, E.: Hyaluronic acid binding by human sperm indicates cellular maturity, viability, and unreacted acrosomal status. *Fertility and Sterility* 79(3), 1616–1624 (2003)
9. Petersen, C.G., Massaro, F.C., Mauri, A.L., Oliveira, J.B., Baruffi, R.L., Franco Jr., J.: Efficacy of hyaluronic acid binding assay in selecting motile spermatozoa with normal morphology at high magnification. *Reproductive Biology and Endocrinology* 8(1), 149–155 (2010)
10. Jakab, A., Sakkas, D., Delpiano, E., Cayli, S., Kovanci, E., Ward, D., Revelli, A., Huszar, G.: Intracytoplasmic sperm injection: a novel selection method for sperm with normal frequency of chromosomal aneuploidies. *Fertility and Sterility* 84(6), 1665–1673 (2005)
11. Yagci, A., Murk, W., Stronk, J., Huszar, G.: Spermatozoa Bound to Solid State Hyaluronic Acid Show Chromatin Structure with High DNA Chain Integrity: An Acridine Orange Fluorescence Study. *Journal of Andrology* 31(6), 566–572 (2010)
12. Shi, L., Nascimento, J., Chandsawangbhuwana, C., Botvinick, E., Berns, M.: An automatic system to study sperm motility and energetics. *Biomed. Microdevices* 10(4), 573–583 (2008)

13. Amann, R.P., Katz, D.F.: Reflections on CASA after 25 years. *Journal of Andrology* 25(3), 317–325 (2004)
14. Nafisi, V.R., Moradi, M.H., Nasr-Esfahani, M.H.: Sperm identification using elliptic model and tail detection. In: *Proc. 4th World Enformatika Conf., Istanbul, Turkey (2005)*
15. Shi, L.Z., Nascimento, J.M., Berns, M.W., Botvinick, E.L.: Computer-based tracking of single sperm. *Journal of Biomedical Optics* 11(5), 054009 (2006)
16. Nafisi, V.R., Moradi, M.H., Nasr-Esfahani, M.H.: A template matching algorithm for sperm tracking and classification. *Physiological Measurement* 26(5), 639–651 (2005)
17. Leung, C., Lu, Z., Esfandiari, N., Casper, R.F., Sun, Y.: Automated sperm immobilization for intracytoplasmic sperm injection. *IEEE Trans. Biomedical Engineering* 58(49), 35–942 (2011)

# Characterization and Control of Biological Microrobots

Islam S.M. Khalil, Marc P. Pichel, Lars Zondervan,  
Leon Abelmann, and Sarthak Misra

**Abstract.** This work addresses the characterization and control of Magnetotactic Bacterium (MTB) which can be considered as a biological microrobot. Magnetic dipole moment of the MTB and response to a field-with-alternating-direction are characterized. First, the magnetic dipole moment is characterized using four techniques, i.e., Transmission Electron Microscopy images, *flip-time*, *rotating-field* and *u-turn* techniques. This characterization results in an average magnetic dipole moment of  $3.32 \times 10^{-16}$  A.m<sup>2</sup> and  $3.72 \times 10^{-16}$  A.m<sup>2</sup> for non-motile and motile MTB, respectively. Second, the frequency response analysis of MTB shows that its velocity decreases by 38% for a field-with-alternating-direction of 30 rad/s. Based on the characterized magnetic dipole moment, the magnetic force produced by our magnetic system is five orders-of-magnitude less than the propulsion force generated by the flagellum of the MTB. Therefore, *point-to-point* positioning of MTB cannot be achieved by exerting a magnetic force. A closed-loop control strategy is devised based on calculating the position tracking error, and capitalizes on the frequency response analysis of the MTB. *Point-to-point* closed-loop control of MTB is achieved for a reference set-point of 60  $\mu\text{m}$  with average velocity of 20  $\mu\text{m/s}$ . The closed-loop control system positions the MTB within a region-of-convergence of 10  $\mu\text{m}$  diameter.

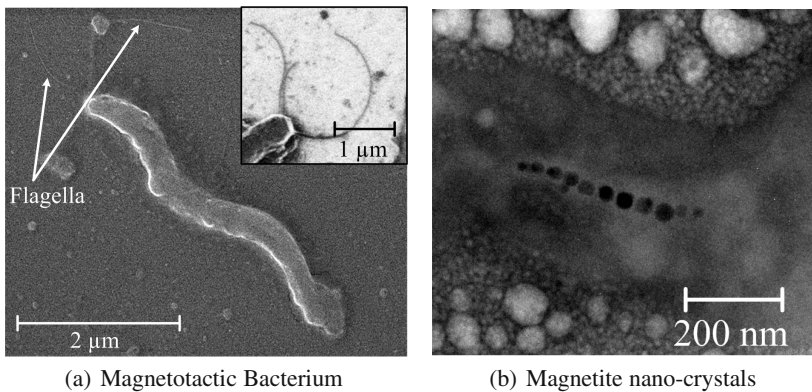
---

Islam S.M. Khalil · Marc P. Pichel · Sarthak Misra  
MIRA–Institute for Biomedical Technology and Technical Medicine,  
University of Twente,  
7500 AE Enschede,  
The Netherlands  
e-mail: i.s.m.khalil@utwente.nl

Lars Zondervan · Leon Abelmann  
MESA+ Institute for Nanotechnology,  
University of Twente,  
7500 AE Enschede,  
The Netherlands

## 1 Introduction

Recently, considerable progress has been made in the area of medical microrobots, with dimensions in the range of  $100\ \mu\text{m}$  or less [1]. These microrobots have the potential to perform targeted drug delivery and actuation of micro-objects [2, 3]. Two strategies are being proposed for propulsion: extracting energy from an external magnetic field [4], or extracting energy from the surrounding liquid and using the external magnetic field for steering only [5]. In the latter case, control of the microrobot is a challenge and to study this control issue, we benefit from Magnetotactic Bacterium (MTB), which uses the earth's magnetic field to orient itself [6, 7]. MTB contains magnetic nano-crystals enveloped by an organic membrane as shown in Fig. 1. These nano-crystals are arranged as a chain along the longitudinal axis of the MTB and provide a magnetic dipole moment which enables the MTB to passively orient itself along the magnetic field lines [8]. Often MTB, which can be considered as a biological microrobot, possesses flagella at its ends which provide self-propulsion along the field lines. Therefore, MTB or a swarm of magnetotactic bacteria can be utilized to transport and steer micro-objects using the forces generated by the flagella and the torque exerted on the magnetic nano-crystal chain, respectively [9]. However, due to limitations on generating relatively large field gradients, velocity of the MTB cannot be controlled and its closed-loop control cannot be achieved by pulling the MTB using field gradients [10]. Therefore, a frequency response-based closed-loop control strategy is presented to control the MTB without relatively large field gradients.



**Fig. 1** Scanning/Transmission Electron Microscopy (SEM/TEM) images of the Magnetotactic Bacterium (MTB), i.e., *Magnetospirillum magnetotacticum* (MS-1). (a) SEM image of the MTB. MTB propels itself by rotating its helical flagella. The flagella are illustrated by the inset taken by a SEM. (b) TEM image of a chain of magnetite nano-crystals. The nano-crystal chain is enveloped by an organic membrane and the interaction between the magnetic dipole moment and the external field allows the MTB to be oriented along the field lines. The nano-crystals have a cuboctahedral morphology and an edge length of  $\sim 36\ \text{nm}$ .

In this work, we characterize the magnetic dipole moment and the frequency response of MTB. First, magnetic dipole moment of MTB is characterized using its TEM images. The *flip-time* of non-motile MTB during field reversal is used to characterize its magnetic dipole moment. In addition, based on motion analysis, the magnetic dipole moment of motile MTB is estimated by analyzing its behavior under the influence of rotating fields and field reversals [11, 12]. Second, response of the MTB to a field-with-alternating-direction is investigated and provides a strategy to perturb the vibrational and rotational modes of its flagella, hence decreasing its velocity. The estimated magnetic dipole moment is used in the realization of the magnetic torque-current map, which is necessary for the open-loop control system. Furthermore, the force-current map is utilized along with the frequency response of the MTB to realize the closed-loop control system.

The remainder of this paper is organized as follows: In Section 2 we shall discuss the theoretical background pertaining to the general characterization of the magnetic dipole moment of MTB using non-motile techniques (TEM images and *flip-time*) and motile techniques which depends on motion analysis (*rotating-field* and *u-turn*). In addition, frequency response of MTB is characterized. Open- and closed-loop control strategies of MTB are analyzed in Section 3. Discussion about the presented control strategies, characterization techniques, directions for future work and conclusions are provided in Section 4.

## 2 Characterization of Magnetotactic Bacterium

Under the influence of a magnetic field, the magnetic torque ( $\mathbf{T}(\mathbf{P}) \in \mathbb{R}^{3 \times 1}$ ) and force ( $\mathbf{F}(\mathbf{P}) \in \mathbb{R}^{3 \times 1}$ ) experienced by an MTB located at position ( $\mathbf{P} \in \mathbb{R}^{3 \times 1}$ ) are given by

$$\mathbf{T}(\mathbf{P}) = \mathbf{m} \times \mathbf{B}(\mathbf{P}) \quad \text{and} \quad \mathbf{F}(\mathbf{P}) = (\mathbf{m} \cdot \nabla)\mathbf{B}(\mathbf{P}), \quad (1)$$

where  $\mathbf{m} \in \mathbb{R}^{3 \times 1}$  and  $\mathbf{B}(\mathbf{P}) \in \mathbb{R}^{3 \times 1}$  are the magnetic dipole moment of the MTB and the induced magnetic field, respectively. The flagellum of the MTB is modeled as a helical propeller with length, thickness, diameter (of the MTB) and pitch of 12  $\mu\text{m}$ , 20 nm, 0.5  $\mu\text{m}$  and 2  $\mu\text{m}$ , respectively. The propulsion force generated by this helical propeller is calculated by a model provided in [13], and results in  $2 \times 10^{-12}$  N at linear velocity of 20  $\mu\text{m/s}$ . In order to determine the magnetic torque and force, we have to estimate the magnetic dipole moment of the MTB.

### 2.1 Magnetic Dipole Moment Characterization

TEM images can be used to determine the magnetic dipole moment of motile and non-motile MTB. The *flip-time* technique allows for calculating the magnetic dipole moment of non-motile MTB, whereas the *rotating-field* and the *u-turn* techniques provide magnetic dipole moment of motile MTB based on its motion analysis.

### 2.1.1 Transmission Electron Microscopy Images

The TEM image of the MTB indicates that a nano-crystal chain is fixed in its organic membrane as shown in Fig. 1(b). These nano-crystals have a cuboctahedral morphology. Using the number of nano-crystals and the cuboctahedron edge length ( $\sim 36$  nm), the magnetic dipole moment has an upper limit given by

$$|\mathbf{m}| = \sum_{i=1}^k M_s V_i, \quad (2)$$

where  $M_s$  is the saturation magnetization of magnetite ( $4.3 \times 10^5$  A.m<sup>-1</sup>). Further,  $k$  and  $V_i$  are the number and volume of the  $i$ th magnetite nano-crystal, respectively.

### 2.1.2 Flip-Time Technique

Non-motile MTB undergoes flip-turn when the magnetic field is reversed. The elapsed-time of the flip-turn ( $\tau$ ), i.e., *flip-time*, is given by

$$\tau = \frac{\alpha}{|\mathbf{m} \parallel \mathbf{B}(\mathbf{P})|} \ln \left( \frac{2 |\mathbf{m} \parallel \mathbf{B}(\mathbf{P})|}{kT} \right), \quad (3)$$

where  $k$  and  $T$  are the Boltzmann constant and the temperature of the medium in which MTB navigates, respectively [11]. Further,  $\alpha$  is the rotational drag coefficient given by

$$\alpha = \frac{\pi \eta L^3}{3} \left[ \ln \left( \frac{L}{d} \right) + 0.92 \left( \frac{d}{L} \right) - 0.662 \right]^{-1}, \quad (4)$$

where  $\eta$  is the medium dynamic viscosity. Further,  $L$  and  $d$  are the length and diameter of the MTB, respectively [14]. The *flip-time* can be determined experimentally, then the magnetic dipole moment can be calculated by solving (3).

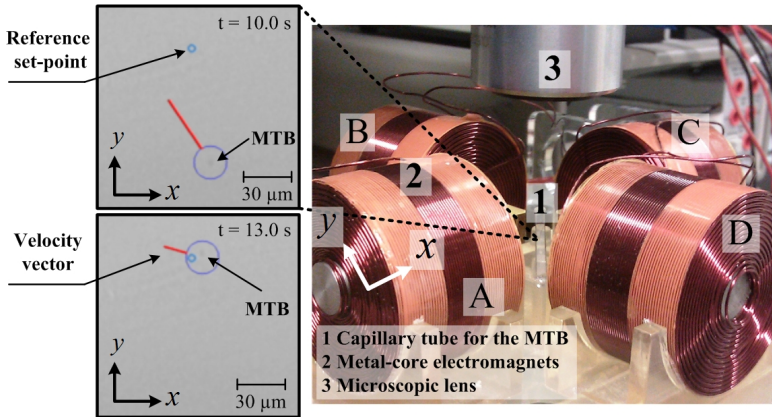
### 2.1.3 Rotating-Field Technique

Under the influence of a rotating magnetic field [17], the relation between the magnetic torque and the angular velocity of the MTB ( $\omega$ ) is

$$|\mathbf{m} \parallel \mathbf{B}(\mathbf{P})| \sin \beta + \alpha \omega = 0, \quad (5)$$

where  $\beta$  is the angle between the induced magnetization field and the magnetic dipole moment. Characterization of the magnetic dipole moment requires determination of its boundary frequency ( $\omega_b$ ). This frequency can be determined by gradually increasing the frequency of the rotating field and observing the frequency after which MTB can no longer follow the field, i.e.,  $\omega = \omega_b$ , when  $\sin \beta = 1$ . Therefore, (5) can be written as





**Fig. 2** Magnetic system developed for characterization and control of Magnetotactic Bacterium (MTB) by the magnetic fields generated at each of the electromagnets. The system consists of a microscope equipped with a vision system mounted on the top of an array of electromagnets surrounding a flat capillary tube which incubates a culture of magnetotactic bacteria [15]. The capillary tube has an inner thickness, an inner width and length of 0.2 mm, 2 mm and 50 mm, respectively (VitroTubes 3520-050, VitroCom, Mountain Lakes, USA). The insets show a closed-loop controlled MTB (tracked and marked by the large blue circle) moving towards a reference set-point by the magnetic fields and its self-propulsion. The electromagnets are labeled with the letters A, B, C and D.

$$|\mathbf{m} \parallel \mathbf{B}(\mathbf{P})| + \alpha\omega_b = 0. \quad (6)$$

The boundary frequency can be determined experimentally and used in (6) to calculate the magnetic dipole moment.

#### 2.1.4 U-Turn Technique

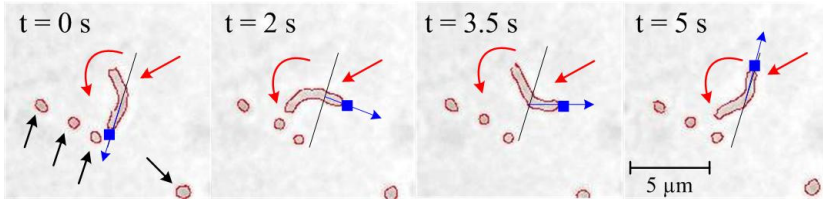
Under magnetic field reversals, MTB undergoes u-turn trajectories [11]. The u-turn diameter ( $D$ ) is given by

$$D = \frac{\alpha\pi v}{|\mathbf{m} \parallel \mathbf{B}(\mathbf{P})|}, \quad (7)$$

where  $v$  is the linear velocity of the MTB. The u-turn diameter can be determined experimentally and used in (7) to calculate the magnetic dipole moment. It is worth noting that (3) represents the u-turn elapsed-time for motile MTB. The elapsed-time can be determined and used in (3) to calculate the magnetic dipole moment.

## 2.2 Frequency Response Characterization

Velocity of the MTB depends on the rotational and vibrational modes of its flagella. Our MTB, i.e., *Magnetospirillum magnetotacticum* (MS-1), provides propulsion by



**Fig. 3** Non-motile Magnetotactic Bacterium (MTB) undergoing counter-clockwise flip-turn when it is subjected to magnetic field reversal. The red arrow indicates the flipping MTB, whereas the black arrows indicate non-flipping and non-motile magnetotactic bacteria in its vicinity. The curved red arrow indicates the direction of rotation. The blue mark indicates one end of the MTB. The MTB changes the curvature of its membrane while its rotating, as shown in the second and third frames. The magnetic dipole moment is calculated for 15 magnetotactic bacteria, and results in an average of  $3.32 \times 10^{-17}$  A.m<sup>2</sup> for magnetic field and average *flip-time* of 7.9 mT and 5.8 s, respectively. Magnetic dipole moment is calculated using (3).

rotating its helical flagella at  $\sim 628.3$  rad/s [14]. Alternating the direction of the field lines could perturb the desirable modes of the MTB, and hence decreases its velocity. This allows for devising a closed-loop control strategy which is partially based on alternating the direction of the field to decrease the velocity of the MTB.

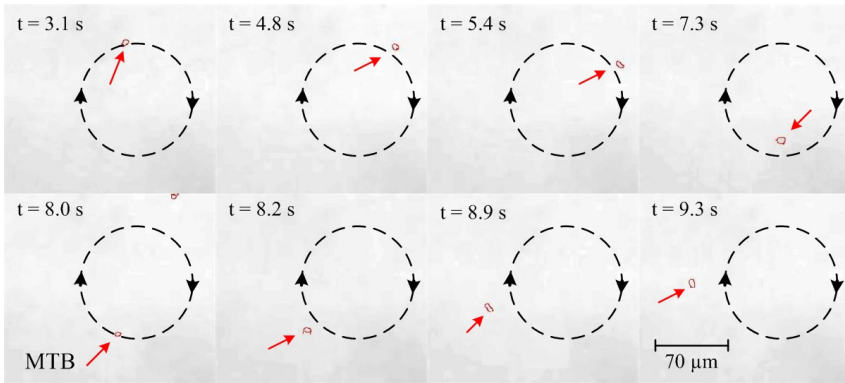
### 2.3 Characterization Results

Characterization of the magnetic dipole moment is conducted on a magnetic system with four orthogonal metal-core electromagnets. The system is capable of providing magnetic fields and field gradients of 15 mT and 60 mT/m, respectively [15]. The electromagnets array surrounds a capillary tube as shown in Fig. 2. The capillary tube incubates a culture of magnetotactic bacteria in 0.02 ml of growth medium. The bacterial density ranges from  $10^6$ /ml to  $10^7$ /ml. The *Magnetospirillum magnetotacticum* (MS-1) cultures utilized in our work are grown according to the protocol provided in Bertani *et al.* [16].

First, the TEM images of 15 magnetotactic bacteria are used to determine the number and volume of the magnetite nano-crystals illustrated in Fig. 1(b). The calculated magnetic dipole moment using (2), has an average of  $3.0 \times 10^{-16}$  A.m<sup>2</sup>.

MTB undergoes flip-turns during field reversals, The *flip-time* is determined experimentally and used in (3) to calculate the magnetic dipole moment. An electromagnet (electromagnet D in Fig. 2) is utilized to generate uniform magnetic field of 7.9 mT. The *flip-time* is calculated from the initiation time of the field reversal until MTB completes a 180 degrees turn. The field reversal is initiated in the first frame of Fig. 3, and the non-motile MTB completes a 180 degrees turn in 5 s. The *flip-time* is used in (3) to calculate the magnetic dipole moment for 15 magnetotactic bacteria, and results in an average of  $3.32 \times 10^{-16}$  A.m<sup>2</sup>.

In order to determine the magnetic dipole moment of motile MTB using the *rotating-field* technique, two orthogonal electromagnets (electromagnets A and D)



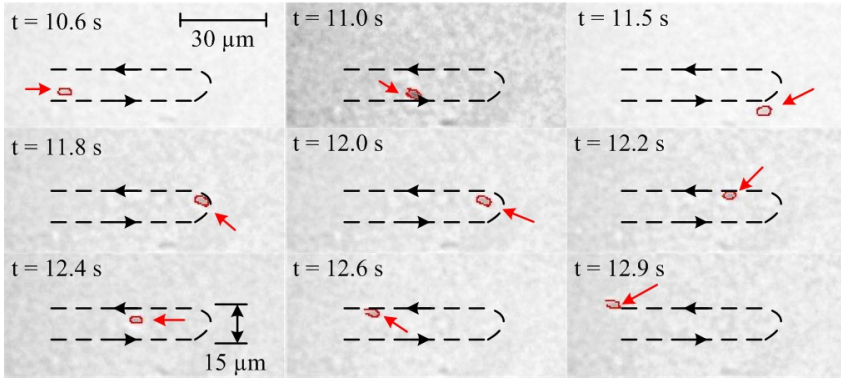
**Fig. 4** Motion of the Magnetotactic Bacterium (MTB) under the influence of a rotating magnetic field with gradually increasing frequency from 1 rad/s to 10 rad/s. The black-dashed circle illustrates the path and direction of the MTB. The red arrow indicates the MTB. The images are processed to detect the edges of the MTB. As we gradually increase the rotating field frequency, the MTB can no longer follow the fields, as shown in the last frame of the top row at,  $t = 7.3$  s and field frequency of 9.5 rad/s. The estimated magnetic dipole moment using the *rotating-field* technique is  $4.34 \times 10^{-16}$  A.m<sup>2</sup> at magnetic field of 7.9 mT and boundary frequency of 9.5 rad/s. Magnetic dipole moment is calculated using (6).

are used to generate rotating fields at 7.9 mT. The MTB follows the rotating fields up to a specific frequency (the boundary frequency) after which it can no longer follow the fields, and deviates from its circular trajectory. The frequency of the applied rotating field is increased gradually from 1 rad/s to 10 rad/s. We observed from the off-line motion analysis that MTB has boundary frequency of 9.5 rad/s as shown in the last frame of the top row of Fig. 4. The length and diameter of the MTB are determined by the TEM image of Fig. 1(a), and used along with the boundary frequency ( $\omega_b$ ) to calculate the magnetic dipole moment using (4) and (6). The average magnetic dipole moment is  $4.34 \times 10^{-16}$  A.m<sup>2</sup> for 15 magnetotactic bacteria.

Magnetic dipole moment of motile MTB is determined using the *u-turn* technique. An electromagnet (electromagnet A) is used to generate magnetic field of

**Table 1** Characterized magnetic dipole moment by the Transmission Electron Microscopy (TEM) images, *flip-time*, *rotating-field* and *u-turn* techniques. The average magnetic dipole moment represents the culture which includes both non-motile and motile magnetotactic bacteria. The average is calculated for 15 magnetotactic bacteria and from 15 TEM images.

Applied field magnitude	7.9 mT
Average magnetic dipole moment calculated from the TEM images	$3.00 \times 10^{-16}$ A.m <sup>2</sup>
Average magnetic dipole moment - <i>flip-time</i> technique	$3.32 \times 10^{-16}$ A.m <sup>2</sup>
Average magnetic dipole moment - <i>rotating-field</i> technique	$4.34 \times 10^{-16}$ A.m <sup>2</sup>
Average magnetic dipole moment - <i>u-turn</i> technique	$3.11 \times 10^{-16}$ A.m <sup>2</sup>



**Fig. 5** Motion of the Magnetotactic Bacterium (MTB) during magnetic field reversal. The MTB (length  $\sim 5 \mu\text{m}$ ) performs a u-turn trajectory marked with the black dashed line, during the field reversal. The red arrow indicates the MTB. The u-turn diameter ( $\sim 15 \mu\text{m}$ ) is used to estimate the MTB magnetic dipole moment. The elapsed-time can be determined starting from the field reversal initiation time until the MTB aligns itself along the field lines. The average magnetic dipole moment for 15 magnetotactic bacteria is  $3.11 \times 10^{-16} \text{ A}\cdot\text{m}^2$  at magnetic field of 7.9 mT. Magnetic dipole moment is calculated using (7).

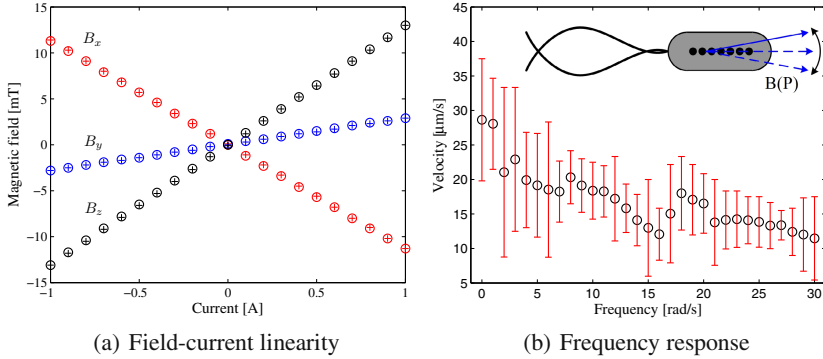
7.9 mT. The MTB undergoes u-turn trajectory when the field is reversed as shown in Fig. 5. The u-turn diameter is determined from the off-line motion analysis of the u-turn trajectory. Using (7), the average magnetic dipole moment is  $3.11 \times 10^{-16} \text{ A}\cdot\text{m}^2$  for 15 magnetotactic bacteria. Results of the magnetic dipole moment characterization are summarized in Table 1.

A magnetic field-with-alternating-direction is applied to investigate the effect of the field frequency on the velocity of the MTB. The frequency of the applied field is gradually increased from 1 rad/s to 30 rad/s, then the velocity is measured at each frequency for 15 magnetotactic bacteria. The frequency response experiment is conducted using two metal-core electromagnets (electromagnets B and D). Sinusoidal current inputs are simultaneously supplied at each of the electromagnets. Linearity of the current-field map ensures that the generated magnetic fields have the same frequency of the applied currents as shown in Fig. 6(a). Increasing the field frequency affects the velocity of the MTB as shown in Fig. 6(b). The average velocity of the MTB is decreased by 38% at field frequency of 30 rad/s.

### 3 Control of Magnetotactic Bacterium

MTB can be oriented towards a desired trajectory using the following torque-current or field-current maps:

$$\mathbf{T}(\mathbf{P}) = \hat{\mathbf{m}}\tilde{\mathbf{B}}(\mathbf{P})\mathbf{I} \text{ and } \mathbf{B}(\mathbf{P}) = \tilde{\mathbf{B}}(\mathbf{P})\mathbf{I}, \quad (8)$$



**Fig. 6** Frequency response analysis of Magnetotactic Bacterium (MTB). (a) Experimental validation of the current-magnetic field linearity and hysteresis. Magnetic field components at a representative point (10 mm along the centerline and from the side of electromagnet A) within the system's workspace (50 mm × 2 mm × 0.2 mm) are plotted versus the increasing (denoted by  $\circ$ ) and decreasing (denoted by  $+$ ) currents. The red, blue and black colors represent the magnetic field components ( $B_x$ ,  $B_y$  and  $B_z$ ) along  $x$ -,  $y$ - and  $z$ -axis, respectively. The fields are measured by a calibrated three-axis Hall magnetometer (Sentron AG, Digital Teslameter 3MS1-A2D3-2-2T, Switzerland). (b) Velocity of MTB versus the field frequency. The velocity is measured for 15 magnetotactic bacteria at each field frequency. The black circles represent the average velocity of the MTB, whereas the red bars represent the velocity error. Magnetic field-with-alternating-direction of frequency ranging from 1 rad/s to 30 rad/s is generated (represented by the blue solid and dashed arrows in the inset) by two metal-core electromagnets (B and D) at a magnetic field of 4 mT.

where  $\mathbf{I} \in \mathbb{R}^{n \times 1}$  is a vector of the current inputs at each of the  $n$ -electromagnets. Further,  $\hat{\cdot}$  is the cross-product operator ( $3 \times 3$  skew-symmetric matrix). The input current vector is mapped onto magnetic field by the matrix ( $\hat{\mathbf{B}}(\mathbf{P}) \in \mathbb{R}^{3 \times n}$ ). The magnetic force can be represented by

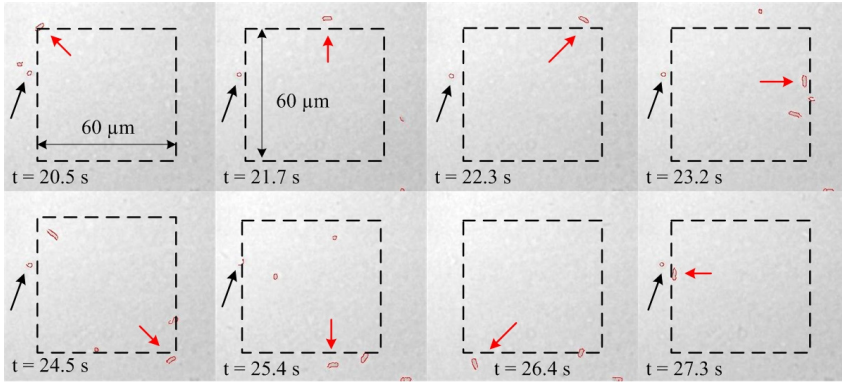
$$\mathbf{F}(\mathbf{P}) = \left[ \frac{\partial \mathbf{B}(\mathbf{P})}{\partial x} \quad \frac{\partial \mathbf{B}(\mathbf{P})}{\partial y} \quad \frac{\partial \mathbf{B}(\mathbf{P})}{\partial z} \right]^T \mathbf{m} = \left[ \frac{\partial \hat{\mathbf{B}}(\mathbf{P})\mathbf{I}}{\partial x} \quad \frac{\partial \hat{\mathbf{B}}(\mathbf{P})\mathbf{I}}{\partial y} \quad \frac{\partial \hat{\mathbf{B}}(\mathbf{P})\mathbf{I}}{\partial z} \right]^T \mathbf{m}. \quad (9)$$

Using (8), MTB will perform flagellated swim along the field lines, this is sufficient to realize the open-loop control system. However, to realize the closed-loop control system, we calculate the position and velocity tracking errors along  $x$ - and  $y$ -axis,

$$\mathbf{e} = \mathbf{P}_{ref} - \mathbf{P} \quad \text{and} \quad \dot{\mathbf{e}} = \dot{\mathbf{P}}_{ref} - \dot{\mathbf{P}}, \quad (10)$$

where  $\mathbf{e}$  and  $\dot{\mathbf{e}}$  are the position and velocity tracking errors, respectively. Further,  $\mathbf{P}_{ref}$  is a reference set-point. The desired force ( $\mathbf{F}_{des}(\mathbf{P})$ ) can be calculated by

$$\mathbf{F}_{des}(\mathbf{P}) = \mathbf{K}_1 \mathbf{e} + \mathbf{K}_2 \dot{\mathbf{e}}, \quad (11)$$



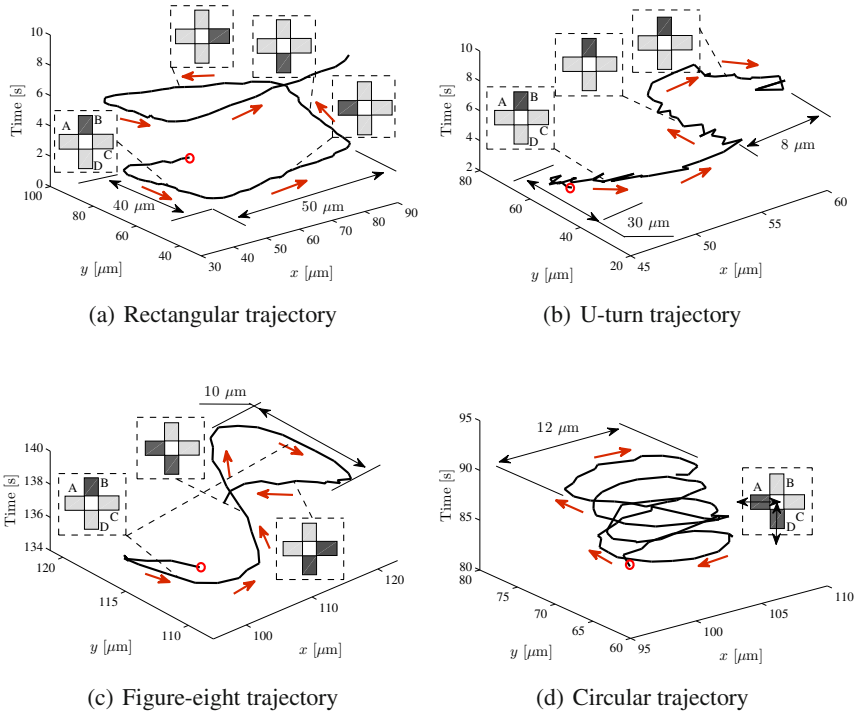
**Fig. 7** Experimental open-loop control result of the Magnetotactic Bacterium (MTB). The red arrow indicates the position of the controlled MTB. The square trajectory illustrated with dashed black lines represents the reference trajectory (length  $\sim 60 \mu\text{m}$ ). The black arrow shows a non-motile MTB. The average velocity of the MTB during this trajectory is  $32 \mu\text{m/s}$ .

where  $\mathbf{K}_1$  and  $\mathbf{K}_2$  are the controller positive definite gain matrices. The desired force ( $\mathbf{F}_{des}(\mathbf{P})$ ) is provided by solving (9) for the current ( $\mathbf{I}$ ) for each of the electromagnets [2]. The control law (11) will locate the MTB within the vicinity of the reference set-point, i.e., region-of-convergence.

Open-loop control is achieved experimentally by controlling the fields using (8), while closed-loop control is achieved by using (9), (10) and (11). In order to examine the effect of the field-with-alternating-direction on the velocity of the controlled MTB, closed-loop control experiments are conducted in the presence and absence of this field. In each case, velocity of the MTB and the size of the region-of-convergence are evaluated.

### 3.1 Open-Loop Control

In order to examine the open-loop control of MTB, we devised five trajectories, namely square, rectangular, u-turn, circular and figure-eight trajectories. Experimental result of the square trajectory is illustrated in Fig. 7, where MTB tracks a square trajectory of  $60 \mu\text{m}$  length. In this experiment, the field lines are oriented towards the square trajectory coordinates. As soon as one or more magnetotactic bacteria approach the square coordinates, uniform fields are generated parallel to the trajectory by a single electromagnet at a time. The average velocity of the MTB is  $32 \mu\text{m/s}$ . It is important to note that the field strength does not have any effect on the velocity of the MTB. Similarly, an MTB is controlled to follow a rectangular trajectory of  $40 \mu\text{m}$  width and  $50 \mu\text{m}$  length as shown in Fig. 8(a), at average velocity of  $36 \mu\text{m/s}$ . Experimental result of the u-turn trajectory of  $30 \mu\text{m}$  length and  $8 \mu\text{m}$  diameter is illustrated in Fig. 8(b), at average velocity of  $17 \mu\text{m/s}$ . Fig. 8(c) illustrates the experimental result of tracking figure-eight trajectory of  $25 \mu\text{m}$  length



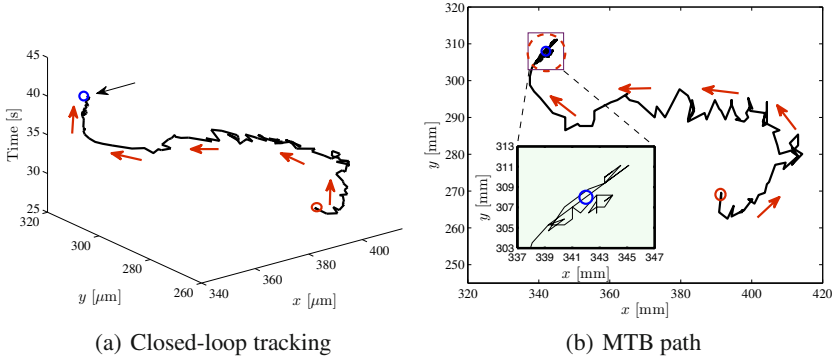
**Fig. 8** Open-loop control of the Magnetotactic Bacterium (MTB) under the influence of magnetic fields. The black line represents the actual trajectory taken by the MTB, whereas the red arrows represent its direction. The red circle indicates the starting position and time of the trajectory. The insets indicate the active electromagnets (darker shade) that are synchronized by the open-loop control system to generate the necessary fields. (a) MTB follows a rectangular trajectory of  $40\ \mu\text{m}$  width and  $50\ \mu\text{m}$  length at average velocity of  $36\ \mu\text{m/s}$ . (b) MTB follows u-turn trajectory of  $30\ \mu\text{m}$  length and  $8\ \mu\text{m}$  diameter at average velocity of  $17\ \mu\text{m/s}$ . (c) MTB follows figure-eight trajectory of  $25\ \mu\text{m}$  length at average velocity of  $23\ \mu\text{m/s}$ . (d) MTB follows a circular trajectory of  $12\ \mu\text{m}$  diameter at average velocity of  $28\ \mu\text{m/s}$ . Electromagnets A and D provide sinusoidal fields illustrated with the black arrows.

at average velocity of  $23\ \mu\text{m/s}$ . Similarly, Fig. 8(d) illustrates the tracking result of a circular trajectory of  $12\ \mu\text{m}$  diameter at average velocity of  $28\ \mu\text{m/s}$ . The insets included in Fig. 8 indicate the active (darker shade) electromagnets of the system.

### 3.2 Closed-Loop Control

In order to examine the frequency response-based closed-loop control strategy, we investigate the *point-to-point* positioning of motile MTB in the presence and absence of a field-with-alternating-direction. Position and velocity tracking errors



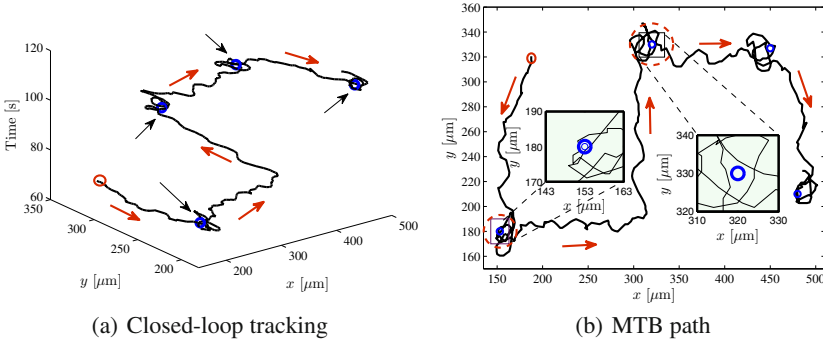


**Fig. 9** Closed-loop control of the Magnetotactic Bacterium (MTB) under the influence of the magnetic fields in the presence of field-with-alternating-direction at frequency of 30 rad/s. The black line represents the actual trajectory taken by the MTB, whereas the red arrows represent its direction. The red and blue circles indicate the starting position and the reference set-point, respectively. The entries of the diagonal controller gain matrices ( $\mathbf{K}_1$ ) and ( $\mathbf{K}_2$ ) are 15.0 and 15.5, respectively. (a) Motion of the MTB towards a reference set-point (indicated by the black arrow). (b) The inset indicates the effect of the closed-loop control on the motion of the MTB at the reference set-point. The average velocity of the MTB is 20  $\mu\text{m/s}$ . The red dashed circle represents a region-of-convergence for the MTB of 10  $\mu\text{m}$  diameter.

along  $x$ - and  $y$ -axis are calculated using (10). These errors are used to direct the fields towards the reference set-point using (9) and (11). First, ( $\mathbf{F}_{des}(\mathbf{P})$ ) is calculated based on the tracking error (10), then used in (9) along with the characterized magnetic dipole moment ( $4.34 \times 10^{-16}$  A.m<sup>2</sup>) to solve for the currents for each of the electromagnets. Field-with-alternating-direction at frequency of 30 rad/s is applied along with the fields generated using (11). The closed-loop control enforces the MTB to stay within the vicinity of the reference set-point. The trajectory taken by the MTB towards a reference set-point is shown in Fig. 9(a). The inset in Fig. 9(b) shows the response of the MTB within the vicinity of the reference set-point. The closed-loop control system enforces the MTB to stay within the vicinity of the reference set-point. The average velocity of the MTB during this experiment is 20  $\mu\text{m/s}$  and the region-of-convergence has 10  $\mu\text{m}$  diameter.

Closed-loop control is performed in the absence of the field-with-alternating-direction. Multiple reference set-points are tracked as shown in Figs. 10(a) and (b), with average velocity of 29  $\mu\text{m/s}$ . This result indicates the effect of the field-with-alternating-direction on the velocity of the MTB. The two insets in Fig. 10(b) illustrate the effect of the closed-loop control on the behavior of the MTB. Due to self-propulsion, closed-loop control can locate the MTB within the vicinity of the reference set-point but cannot achieve asymptotic convergence of the tracking error. As shown in Fig. 10(b), the region-of-convergence for the first and second set-points has diameter of 20  $\mu\text{m}$  and 26  $\mu\text{m}$ , respectively.





**Fig. 10** Closed-loop control of the Magnetotactic Bacterium (MTB) under the influence of the magnetic fields in the absence of a field-with-alternating-direction. The black line represents the actual trajectory taken by the MTB, whereas the red arrows represent its direction. The red and blue circles indicate the initial position and the reference set-points, respectively. The entries of the diagonal controller gain matrices ( $\mathbf{K}_1$ ) and ( $\mathbf{K}_2$ ) are 15.0 and 15.5, respectively. (a) Motion of the MTB towards reference set-points (indicated by the black arrows). (b) The insets indicate the effect of the closed-loop control on the motion of the MTB at the first and second reference set-points. The closed-loop control action locates the MTB within the vicinity of the given reference set-points. The average velocity of the MTB is  $29 \mu\text{m/s}$ . The red dashed circles represent circular regions-of-convergence for the MTB around the first and second reference set-points with diameter of  $20 \mu\text{m}$  and  $26 \mu\text{m}$ , respectively.

### 4 Discussion

In view of the practical limitations on the generation of magnetic field gradients, controlling the velocity of the MTB by exerting a pulling magnetic force cannot be achieved. This force ( $2.6 \times 10^{-17} \text{ N}$  for characterized magnetic dipole moment and field gradient of  $4.34 \times 10^{-16} \text{ A.m}^2$  and  $60 \text{ mT/m}$ , respectively) is five orders-of-magnitude less than the force generated by the MTB flagellum ( $2 \times 10^{-12} \text{ N}$ ). Frequency response of MTB shows that its average velocity decreases by 38% at field frequency of  $30 \text{ rad/s}$ . This observation allows for devising a strategy to decrease the velocity of the MTB, by alternating the direction of the field lines at this frequency. In addition, characterization of the magnetic dipole moment and frequency response are used in the realization of the closed-loop control strategy. This strategy requires inducing a field-with-alternating-direction along with the fields generated by the closed-loop control laws (9), (10) and (11). The closed-loop control system allows for positioning the MTB within the vicinity of a reference set-point along with decreasing its velocity. The closed-loop control system is further evaluated in the presence and absence of the field-with-alternating-direction. The closed-loop control system decreases the region-of-convergence and the velocity of the MTB in the presence of the field-with-alternating-direction as opposed to the same controller without this field.

## 4.1 Future Work

Future work should include accurate characterization of the MTB frequency response over a wider frequency range. A three-dimensional magnetic system with auto-focusing is essential in studying and controlling the MTB since it escapes into the third-dimension frequently. Therefore, our magnetic system will be redesigned to allow for visual tracking and control of the MTB in three-dimensional space. In addition, closed-loop control of a swarm of magnetotactic bacteria will be achieved.

## 4.2 Conclusions

A closed-loop control strategy for MTB is demonstrated experimentally. This strategy is based on characterizing the magnetic dipole moment and the frequency response of MTB. The characterized magnetic dipole moment using non-motile technique (*flip-time*) and motile techniques (*rotating-field* and *u-turn*) agrees with the result of the TEM images. Response of the MTB to a field-with-alternating-direction at different frequencies shows that its velocity decreases by 38 % at field frequency of 30 rad/s. These characterization results allow for devising a closed-loop control strategy based on generating two superimposed fields. The first is generated based on the position and velocity tracking errors of the MTB, whereas the second is a field-with-alternating-direction at frequency of 30 rad/s. These fields are superimposed and achieve position control of the MTB with a region-of-convergence of 10  $\mu\text{m}$  diameter and an average velocity of 20  $\mu\text{m/s}$ .

## References

1. Nelson, B.J., Kaliakatsos, I.K., Abbott, J.J.: Microrobots for minimally invasive medicine. *Annual Reviews of Biomedical Engineering* 12, 55–85 (2010)
2. Kummer, M.P., Abbott, J.J., Kartochovil, B.E., Borer, R., Sengul, A., Nelson, B.J.: OctoMag: an electromagnetic system for 5-DOF wireless micromanipulation. *IEEE Transactions on Robotics* 26(6), 1006–1017 (2010)
3. Martel, S., Tremblay, C.C., Ngakeng, S., Langlois, G.: Controlled manipulation and actuation of micro-objects with magnetotactic bacteria. *Applied Physics Letters* 89(23), 1–3 (2006)
4. Dong, L., Nelson, B.J.: Tutorial - Robotics in the small part II: nanorobotics. *IEEE Robotics and Automation Magazine* 14(3), 111–121 (2007)
5. Solovev, A.A., Mei, Y., Urena, E.B., Huang, G., Schmidt, O.G.: Catalytic microtubular jet engines self-propelled by accumulated gas bubbles. *Small* 5(14), 1688–1692 (2009)
6. Blackmore, R.: Magnetotactic bacteria. *Science* 190(4212), 377–379 (1975)
7. Faivre, D., Schuler, D.: Magnetotactic Bacteria and Magnetosomes. *Chemical Reviews* 108(11), 4875–4898 (2008)
8. Bahaj, A.S., James, P.A.B.: Characterisation of magnetotactic bacteria using image processing techniques. *IEEE Transactions on Magnetics* 29(6), 3358–3360 (1993)

9. Martel, S., Mohammadi, M.: Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to perform complex micro-assembly tasks. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Alaska, USA, pp. 500–505 (May 2010)
10. Martel, S., Mohammadi, M., Felfoul, O., Lu, Z., Pouponneau, P.: Flagellated magnetotactic bacteria as controlled MRI-trackable propulsion and steering systems for medical nanorobots operating in the human microvasculature. *International Journal of Robotics Research* 28(4), 571–582 (2009)
11. Bahaj, A.S., James, P.A.B., Moeschler, F.D.: An alternative method for the estimation of the magnetic moment of non-spherical magnetotactic bacteria. *IEEE Transactions on Magnetics* 32(5), 5133–5135 (1996)
12. Steinberger, B., Petersen, N., Petermann, H., Wiess, D.G.: Movement of magnetic bacteria in time-varying magnetic fields. *Journal of Fluid Mechanics* 273, 189–211 (1994)
13. Behkam, B., Sitti, M.: Design Methodology for Biomimetic Propulsion of Miniature Swimming Robots. *ASME Journal of Dynamic Systems, Measurement and Control* 128, 36–43 (2006)
14. Chemla, Y.R., Grossman, H.L., Lee, T.S., Clarke, J., Adamkiewicz, M., Buchanan, B.B.: A new study of bacterial motion: superconducting quantum interference device microscopy of magnetotactic bacteria. *Biophysical Journal* 76(6), 3323–3330 (1999)
15. Keuning, J.D., de Vries, J., Abelman, L., Misra, S.: Image-based magnetic control of paramagnetic microparticles in water. In: Proceedings of the IEEE International Conference of Robotics and Systems, San Francisco, USA, pp. 421–426 (September 2011)
16. Bertani, L.E., Weko, J., Phillips, K.V., Gray, R.F., Kirschvink, J.L.: Physical and genetic characterization of the genome of *Magnetospirillum magnetotacticum*, strain MS-1. *International Journal on Genes and Genomes* 264, 257–263 (2001)
17. Erglis, K., Ose, V., Zeltins, A., Sharipo, A., Janmey, P.A., Cebers, A.: Dynamics of magnetotactic bacteria in a rotating magnetic field. *Biophysical Journal* 93(4), 1402–1412 (2007)
18. Esquivel, D.M.S., Lins de Barros, H.G.P.: Motion of magnetotactic microorganisms. *Journal of Experimental Biology* 121, 153–163 (1986)

# **Part X: ISER Session Summary on “Planning and Control”**

Nikhil Chopra

University of Maryland, College Park

## **Session Summary**

In this session, five talks on various facets of planning and control were presented. The first paper by A. Lecours and C. Gosselin titled, “Computed-torque control of a four-degree-of-freedom admittance controlled intelligent assist device,” studies interaction control for cooperative human robot interaction. Admittance control is proposed under system constraints such as saturation, and the authors compare the computed torque and the PID controller in their experimental results. The second paper by O. Chuy et al. titled, “Sampling-Based Direct Trajectory Generation Using the Minimum Time Cost Function,” focuses on feedback motion planning under system and performance constraints. The proposed algorithm is experimentally studied on a one DOF robot under heavy load. The third paper titled, “Antagonistic Control of Multi-DOF Joint,” by K. Koganezawa studies rotary actuators with nonlinear elasticity. The author demonstrates using theory and experiments that the joint angle and stiffness can be independently controlled using the proposed approach. The fourth paper by P. Tallapragada and N. Chopra titled, “Lyapunov Sampling for Adaptive Tracking Control in Robot Manipulators: An Experimental Comparison,” studies adaptive sampling for tracking control in robotic manipulators. The authors present experimental comparison of two different tracking algorithms by using their feedback sampling algorithm. The last paper of the session by N. Dantam et al. titled, “Linguistic Composition of Semantic Maps and Hybrid Controllers,” formulates robot policies by exploiting the common underlying structure in semantic mapping and discrete event controllers. Experiments conducted in indoor environments using a mobile robot demonstrate the efficacy of the approach.

# Computed-Torque Control of a Four-Degree-of-Freedom Admittance Controlled Intelligent Assist Device

Alexandre Lecours and Clément Gosselin

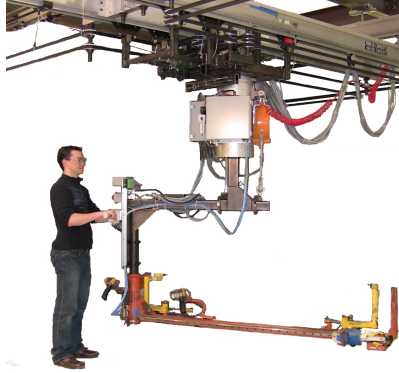
**Abstract.** Robots are used in different applications to enhance human performance and in the future, these interactions will become more frequent. In order to achieve this human augmentation, the cooperation must be very intuitive to the human operator. This paper proposes a computed-torque control scheme for pHRI using admittance control. The admittance model is first introduced. Then, the robot identification, the computed-torque approach and the saturation considerations are addressed. The intelligent assist device used for the experiments is then presented. Finally, experimental results that demonstrate the performance of the algorithm are provided.

## 1 Introduction

The main challenge for human augmentation systems is to perceive their environment and the human intentions and to respond to them adequately, intuitively and safely [1]. To this end, it is desired to enhance the control performances of such systems. On the other hand, computed-torque schemes are widely used in robotics for reference trajectory following since they decouple and linearize highly non-linear dynamical systems, thereby leading to better performances than simple PID control [2–8]. Such techniques are also used in haptics to improve impedance control performance and virtual environment rendering [9–11]. Computed-torque schemes were also very briefly introduced for admittance control in [12]. Admittance control is typically used for applications involving physical human-robot interactions (pHRI) with large payloads. When admittance control is used, a handle or a force/torque sensor is normally used to detect human intentions [13, 14].

---

Alexandre Lecours · Clément Gosselin  
Département de génie mécanique,  
Université Laval, Québec, Qc, Canada  
e-mail: alexandre.lecours.1@ulaval.ca, gosselin@gmc.ulaval.ca



**Fig. 1** Four-dof intelligent assist device prototype used in the experiments

With reference trajectory following, it is possible to design a smooth reference trajectory in order to avoid discontinuous acceleration or jerk profiles. However, obtaining smooth reference trajectories is less straightforward in human-robot collaboration since the trajectory is directed by the human operator. This paper presents a methodology to overcome these issues, along with experimental examples and results. Although the robot used in the experiments is decoupled and linear, it is shown that the computed-torque control leads to better performance. It is then possible to implement the control algorithms on a nonlinear robot where the performance improvement over PID control should be even greater.

This paper proposes a computed-torque control scheme for pHRI. The paper is structured as follows. The admittance model is first introduced. Then, the robot identification, the computed-torque approach and the saturation considerations are addressed. The intelligent assist device used for the experiments is then presented. Finally, experimental results that demonstrate the performance of the algorithm are provided.

## 2 Admittance Model

Two main classes of control schemes are used in haptic applications and pHRI, namely, impedance and admittance control. Because of the large inertia and significant friction of the intelligent assist device (IAD) used in this work, it would obviously be too hard for a human operator to impart a movement to the IAD, which makes impedance controllers not well adapted for the situation, even if a force sensor is used. An admittance controller is thus used as detailed in [15].

The one-dimensional admittance equation is written as:

$$f_H = m\ddot{x} + c\dot{x}. \quad (1)$$

where  $f_H$  is the interaction force, i.e., the force applied by the human operator,  $m$  the virtual mass,  $c$  the virtual damping and  $x$ ,  $\dot{x}$ ,  $\ddot{x}$  are respectively the position, velocity and acceleration.

The trajectory to be followed by the robot can be prescribed as a position  $x_d$  or as a desired velocity  $\dot{x}_d$ . For velocity control, the desired velocity can be written, in the Laplace domain, as:

$$\dot{X}_d(s) = \frac{F_H(s)}{ms + c} = \frac{F_H(s)/c}{\frac{m}{c}s + 1} = F_H(s)H(s). \tag{2}$$

where  $\dot{X}_d(s)$  is the Laplace transform of  $\dot{x}_d$ ,  $F_H(s)$  is the Laplace transform of  $f_h$  and  $s$  is the Laplace variable. Velocity control is used here, similarly to what was done in [15–17].

### 3 Model Identification

In order to use computed-torque control, a dynamic model of the robot is needed and is obtained using model identification techniques. In an identification scheme, a given variable is used as the dependent variable and the others as independent. The coefficients applied to the latter are the parameters to be identified.

#### 3.1 Model

The dynamic model of the robot is:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_f \tag{3}$$

where  $\mathbf{q}$  is the vector of joint displacements,  $\mathbf{M}(\mathbf{q})$  is the generalized inertia matrix,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$  is the vector of centripetal and Coriolis effects,  $\mathbf{g}(\mathbf{q})$  is the vector of gravitational effects and  $\boldsymbol{\tau}_f$  is the vector of joint friction.

Because the system used in the experiments is decoupled, each dof is identified separately for simplicity and only the  $X$  and  $Y$  axes are used in the experiments. The discrete-time equations are:

$$u(k) = m_I \ddot{x}(k) + \tau_{vI}(k) + \tau_{cI}(k) \tag{4}$$

where  $u$  is the command, expressed as a current with units of  $A$ , at time step  $k$ ,  $m_I$  is the inertial term expressed at the motor in  $As^2/m$ ,  $\tau_{vI}$  and  $\tau_{cI}$  are respectively the viscous and Coulomb friction and are simply modeled here as:

$$\begin{aligned} \tau_{cI} &= f_{cI} \text{sign}(\dot{x}) \\ \tau_{vI} &= f_{vI} \dot{x} \end{aligned} \tag{5}$$

where  $f_{cI}$  is the Coulomb friction coefficient, expressed in A, and  $f_{vI}$  is the viscous friction coefficient, in As/m. Although a more complex friction model could be used such as in [18–22], the chosen model is sufficient here as it will be shown in Section 4.3.

Defining:

$$\begin{aligned} y(k) &= u(k) \\ \phi_t &= [\ddot{x}(k), \dot{x}(k), -\text{sign}(\dot{x})]^T \\ \theta_t &= [m_I, f_{vI}, f_{cI}]^T \end{aligned} \quad (6)$$

the model is written as:

$$y_t(k) = \phi_t^T \theta_t + e_y(k) \quad (7)$$

where  $e_y$  stands for the error. Friction parameters could also be identified separately, as in [21, 22].

To solve this problem, simple least squares [21, 4, 23], recursive least squares [23] and recursive least squares with approximate maximum-likelihood [23] can be used. The latter technique was used in the experiments and since the parameters are identified off-line, the data is filtered with a non-causal FIR filter to obtain better convergence properties without creating phase shift. The velocity and acceleration are obtained from the position with a two-point Lagrangian derivative centred on the current point. The data is also normalized to obtain better numerical stability.

### 3.2 Results

The results obtained from the identification are first compared with the measured values and presented in Tab. 1. The mass was measured by summing all the component masses and by adding the motor inertia, transferred at the end effector. The Coulomb friction was measured by manually pushing the device with a dynamometer at a very low velocity. It should be noted that the comparison is an approximation aiming at providing ballpark figures.

Fig. 2 presents the comparison between the command signal used for the identification and the reconstructed command obtained with the identified parameters. The velocity and acceleration are filtered with a non-causal FIR filter to remove high-frequency noise in order for the comparison to be possible. The estimated torques do not match exactly the applied torques since the model is a simplification of reality but they are nevertheless fairly accurate.

Figure 3 shows an open-loop response to a human input, obtained by simply feeding the inverse dynamics to the four-dof assist device as:

$$\tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \tau_f \quad (8)$$



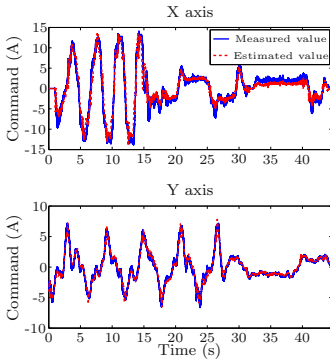
**Table 1** Identification results

	$M_x$ (kg)	$\tau_v x$ (Ns/m)	$\tau_c x$ (N)	$M_y$ (kg)	$\tau_v y$ (Ns/m)	$\tau_c y$ (N)
Measured	507	N/A	88	328	N/A	51
Identification	507	369	92	356	32	71

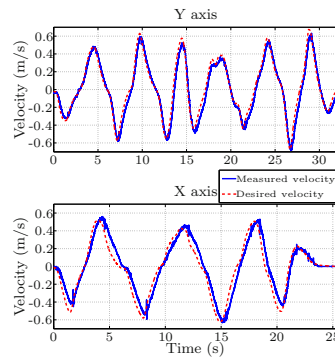
or in this simple case:

$$u(k) = m_I \ddot{x}_d + \tau_{vI} \dot{x}_d + \tau_{cI}. \tag{9}$$

More details on the implementation of this control are given in Section 4 (Subsections 4.3 and 4.4) for the friction compensation part. Even in an open-loop mode, it is easy for the operator to cooperate with the robot. The condition number (ratio between the maximal and minimal singular values) of the regressor matrix is 4.2 for the X axis and 4.1 for the Y axis. It is desired for this number to be close to 1 (and it should be below 100 [24]), for the regressor matrix to be well-conditioned and the estimation of the parameters to be reliable [21, 24].



**Fig. 2** Offline measured and estimated torque comparison



**Fig. 3** Open-loop reference velocity tracking in human-robot collaboration mode

## 4 Velocity Controller

In this section, the PID and computed-torque control used for the low-level velocity controller are explained. Even if the dynamics of the robot used for the experiments are linear, it will be shown that the computed-torque control leads to better results than the PID control. Additionally, it is possible

to implement the control algorithms with a nonlinear device, in which case the performance improvements over PID control should be greater. Friction compensation is also discussed.

#### 4.1 PID Control

First, PID control is considered for the velocity controller. The output command is:

$$\boldsymbol{\tau} = \mathbf{K}_P \mathbf{e} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_I \int \mathbf{e} \quad (10)$$

where  $\mathbf{e} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}$ ,  $\mathbf{K}_P$ ,  $\mathbf{K}_D$  and  $\mathbf{K}_I$  are respectively the proportional, derivative and integral gain matrices,  $\dot{\mathbf{q}}_d$  is the desired joint velocity vector and  $\dot{\mathbf{q}}$  is the measured joint velocity vector. This controller is applied to all joints independently. As pointed out in [15], it is not recommended to use a derivative gain since the signal is noisy [25] (acceleration signal) and no integral gain is used since the behaviour to an operator input would then depend on the error history. This introduces additional limitations and leads to a lack of flexibility of the PID controller.

#### 4.2 Computed-Torque Control

Computed-torque control is widely used in robotics. Its main advantage is to transform a complex nonlinear multi-input multi-output (MIMO) system into a very simple decoupled MIMO linear system [6, 2-5, 7, 8].

Numerous different approaches exist for the implementation of computed-torque control. A very popular approach [2, 3, 5] is written as

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_P \tilde{\mathbf{q}} + \mathbf{K}_V \dot{\tilde{\mathbf{q}}}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (11)$$

where  $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$ .

The PD+ approach [3, 5] is written as

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) + \mathbf{K}_P \tilde{\mathbf{q}} + \mathbf{K}_V \dot{\tilde{\mathbf{q}}} \quad (12)$$

while the non-adaptive version of the Slotine and Li controller [7, 6] is written as

$$\begin{aligned} \boldsymbol{\tau} = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \boldsymbol{\Lambda} \dot{\tilde{\mathbf{q}}}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})[\dot{\mathbf{q}} + \boldsymbol{\Lambda} \tilde{\mathbf{q}}] + \mathbf{g}(\mathbf{q}) \\ + \mathbf{K}_P \tilde{\mathbf{q}} + \mathbf{K}_V \dot{\tilde{\mathbf{q}}} \end{aligned} \quad (13)$$

where  $\boldsymbol{\Lambda} = \mathbf{K}_V^{-1} \mathbf{K}_P$ .

These approaches were widely applied for trajectory tracking. However, velocity control is considered here. Eqn. (11) is then modified accordingly, similarly to what was done in [26]. This leads to

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_P \dot{\tilde{\mathbf{q}}}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) \quad (14)$$

where there is no derivative nor integral gains for the reasons explained in the description of the PID controller.

The dynamics can also be only partially compensated for. The advantage of a partial compensation is to benefit from the advantages of the computed-torque technique while not relying too much on a model. It is also pointed out that the model part of the control is low-pass filtered in order to avoid high frequency command input.

### 4.3 Friction Compensation

Friction compensation can be included in the controller as follows:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_P \dot{\tilde{\mathbf{q}}}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_c + \boldsymbol{\tau}_v \quad (15)$$

where  $\boldsymbol{\tau}_c$  and  $\boldsymbol{\tau}_v$  are respectively the Coulomb and viscous friction vectors whose their  $i$ th components are written as:

$$\begin{aligned} \tau_c^i &= f_c^i \cdot \text{sign}(\dot{x}_d^i) \cdot (1 - e^{-\alpha^i |\dot{x}_d^i|}) \\ \tau_v^i &= f_v^i \dot{x}_d^i. \end{aligned} \quad (16)$$

The exponential term along with the  $\alpha$  parameter is used to reduce the chattering induced by friction compensation when the velocity is near zero. The desired velocity is used for viscous friction compensation in order to reduce the command noise and contribute to the command based on the human intention. A more complex model could have been used [18–20], including stiction for example, but as it will be shown, the simple friction compensation from eqn. (16) is sufficient in practice for the device used here.

### 4.4 Desired Velocity and Acceleration

Computed-torque is well adapted to classical position control trajectory following since the desired trajectory is known and can be designed to be smooth.

With admittance control, the trajectory is known (see eqn. (2)) and computed-torque control can then be used by inputting the desired velocity and acceleration in eqn. (15). However, the trajectory relies on the operator's intentions, which may not lead to smooth signals. The smoothness can however be controlled by a proper choice of admittance parameters (virtual damping and mass) and from force input signal pre-processing. Saturation must also be considered, as detailed in the next section.

With impedance control, desired velocity and acceleration cannot be used for computed-torque since they are unknown. Measured signals can be used but this requires very good position, velocity or acceleration sensors and signal processing.

## 5 Saturation Consideration

For safety reasons, the desired velocity and acceleration should be limited. However, such limitations can lead to abrupt variations in the acceleration or jerk profiles, which is undesirable since such variations result in abrupt variations in the command. The problem is amplified with computed-torque control since the command is directly related to the desired acceleration and velocity. Applying such commands to the robot can excite unmodelled dynamics and lead to vibrations.

### 5.1 Velocity Limits

If a simple velocity saturation is used, the acceleration at a saturation point will go from a given value to zero within one time step, leading to an abrupt acceleration profile. If the desired velocity is filtered, the desired acceleration will be smoother but at the expense of time delays. In order to alleviate this problem, the desired velocity is gradually limited as it approaches the saturation limit, using a third or fifth order polynomial as shown in Fig. 4. The third order polynomial is obtained by setting the output velocity equal to the input velocity for input velocities of  $(v_{sat} - \delta/\sqrt{2})$  and  $(v_{sat} + \delta)$  (where  $\delta$  is a tuning parameter) and by setting the first derivative for these same input velocities to one and zero respectively. For the fifth order polynomial, the second derivative is also set to zero for the same input velocities.

Fig. 5 presents the desired velocity, acceleration and jerk response to a human input sine force. Only the results obtained with the third order polynomial are shown for simplicity. It is shown that the desired acceleration is smoother with the third order polynomial than with basic saturation. This is verified with the desired jerk which is approximately  $270 \text{ m/s}^3$  for basic saturation and approximately  $40 \text{ m/s}^3$  with the third order polynomial. The saturation transition smoothness can be varied by changing the value of parameter  $\delta$ . In this example, the velocity limit is  $0.7 \text{ m/s}$  and  $\delta$  is in the order of 0.23.

### 5.2 Acceleration Limits

A common situation in which the acceleration can vary rapidly (other than a highly dynamic human force input) arises when the maximum allowed acceleration is not the same as the maximum allowed deceleration. However,

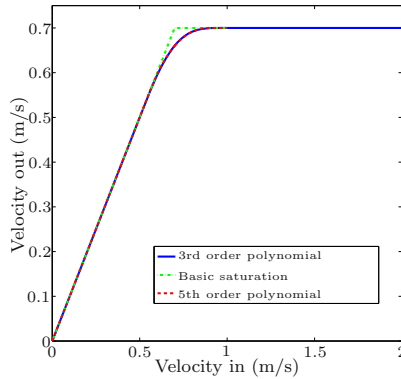


Fig. 4 Theoretical velocity saturation

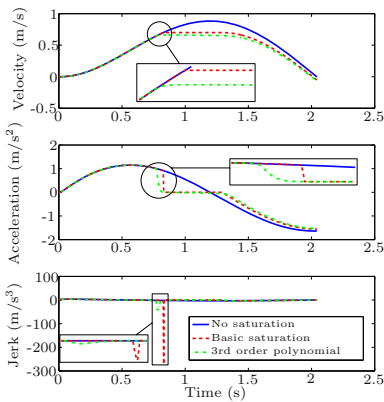


Fig. 5 Desired velocity, acceleration and jerk with velocity saturation for a human input sine force

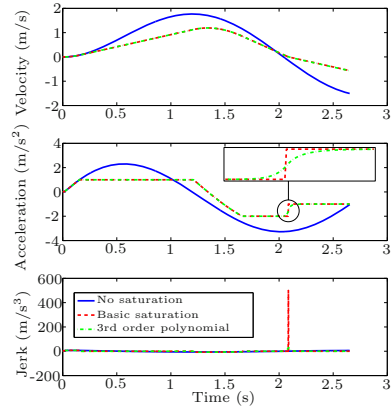


Fig. 6 Desired velocity, acceleration and jerk with acceleration saturation for a human input sine force

this situation is desirable for safety considerations: it must not be allowed to increase speed very rapidly but one should be able to stop rapidly. Then, when the device goes from a deceleration phase to an acceleration phase, the desired acceleration changes abruptly from the maximum allowed deceleration to the maximum allowed acceleration. Filtering could help but at the expense of time delays, which is not desirable. Similarly to what is done for velocity saturation, the desired acceleration is gradually limited using an exponential function with the desired velocity as a parameter. Indeed, the transition occurs at a desired velocity of zero and the desired acceleration is then gradually varied as a function of the desired velocity. The desired acceleration transition is then represented by

$$a_m = \frac{\ddot{x}_m^+ + \ddot{x}_m^-}{2} \text{sign}(\ddot{x}_d) - \frac{-\ddot{x}_m^+ + \ddot{x}_m^-}{2} \text{sign}(\dot{x}_d) \left(1 - e^{-|\gamma \dot{x}_d|}\right) \quad (17)$$

where  $\gamma$  is a smoothness parameter,  $a_m$  is the current maximum allowed acceleration/deceleration,  $\ddot{x}_m^+$  is the maximum allowed acceleration and  $\ddot{x}_m^-$  is the maximum allowed deceleration. Parameter  $\gamma$  should be high enough to obtain smoothness but not too high since it affects the maximal acceleration limit when the velocity is near zero.

Fig. 6 presents the results with a human input sine force. It is shown that with the gradually changing acceleration saturation, the transition is much smoother. This is verified with the desired jerk which is approximately equal to  $500 \text{ m/s}^3$  for basic saturation while it is reduced to  $30 \text{ m/s}^3$  with the proposed transition from eqn. (17). A jerk limiter could also be implemented if the force signal is not too noisy.

### 5.3 Virtual limits

A very intuitive means of implementing virtual limits is to make the desired velocity zero if the position is greater than a given limit and the velocity is directed toward the limit. To avoid very large required acceleration, an acceleration limiter can be implemented although the acceleration and jerk profile would remain abrupt. The proposed solution is to set the force to zero, with a rate limiter and with high and well chosen admittance parameters. The admittance parameters can also be increased according to the position and a stiffness term can also be added.

## 6 Prototype of a 4-DOF Intelligent Assist Device

The robot used for the experiments reported in this paper is a prototype of a 4-dof intelligent assist device (IAD), shown in Fig. 1, allowing translations in all directions ( $XYZ$ ) and a rotation ( $\theta$ ) about the vertical axis. In this prototype, the total moving mass is approximately  $500 \text{ kg}$  in the direction of the  $X$  axis and  $325 \text{ kg}$  along the  $Y$  axis. Additionally, the payload may vary between 0 and  $113 \text{ kg}$ . The horizontal workspace is  $3.3 \text{ m} \times 2.15 \text{ m}$  while the vertical range of motion is  $0.52 \text{ m}$ . The range of rotation about the vertical axis is  $120^\circ$ . Three different control modes are possible: autonomous motion, unpowered manual motion and interactive motion (cooperation). In this paper, only the latter is addressed. The controller is implemented on a real-time QNX computer with a sampling period of 2ms. The algorithms are programmed using Simulink/RT-LAB software.

## 7 Experimentation

In order to demonstrate the effectiveness of the proposed control algorithms, three experiments were performed. The first one consisted in simply moving the assist device to compare the PID and computed-torque velocity controllers. The second experiment is a drawing task and the third one consisted in asking the operator to trace imaginary circles in mid air.

### 7.1 Error and Noise Reduction

This experiment consisted in moving the intelligent assist device (see Fig. 1) back and forth to compare the magnitude of the error and the command noise between the PID and computed-torque velocity controllers. The parameters used in the PID controller are  $K_{Px} = 0.05$ ,  $K_{Py} = 0.06$ , and  $K_{Ix}$ ,  $K_{Iy}$ ,  $K_{Dx}$  and  $K_{Dy}$  are zero as previously explained. The closed-loop gains used in the computed-torque controller are  $m_{Ix}K_{Px} = 0.04$  and  $m_{Iy}K_{Py} = 0.04$  while the inertial and friction terms were taken to be 90% of the identified parameters (Tab. 1). With the PID control, it is not possible to increase the gains significantly since vibrations or instability occur. With the computed-torque control, it is possible to adjust the inertial and friction compensation while it is also possible to adjust the closed-loop gains. By increasing these gains, the error can be reduced, at the expense of command noise.

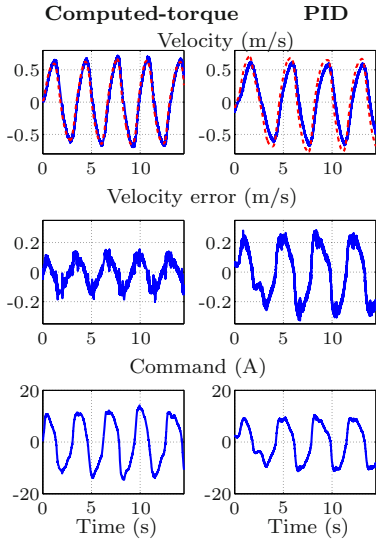
With the selected parameters, the error reduction from the PID controller to the computed-torque controller is about 50% for the  $X$  axis and 32% for the  $Y$  axis. The noise amplitude was approximately reduced by 20% for the  $X$  axis and by 33% for the  $Y$  axis. These results are shown in Figs. 7 and 8.

Fig. 9 compares both control laws for low velocity reference. It is clear that the results are much better with the computed-torque controller and this is clearly apparent to the human user as described in the subsequent experiments.

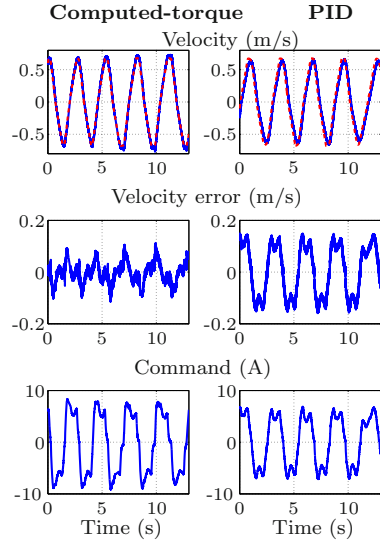
### 7.2 Drawing Task

The drawing task consisted in asking the operator to trace a simple maze, fixed to the ground, (shown in Fig. 11) with a pen mounted on the IAD at 1.4 metre from the operator. The instructions were to minimize the completion time and the overshoots in the maze. Experiments were performed with the PID and computed-torque controllers. The admittance parameters were fixed ( $c = 60Ns/m$  and  $m = 36kg$ ).

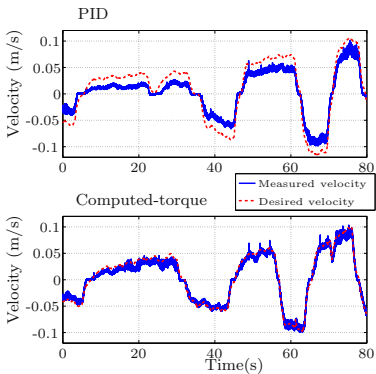
The experiment was performed by 6 subjects whose age ranged between 25 and 41. Task completion time, maze overshoots and subject comments were recorded. The subjects were allowed some practice before performing the task. Subjects were not told which control was set and the order was varied between subjects. Fig. 10 shows the task completion time along with the distance of



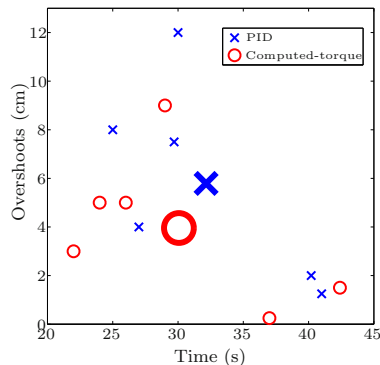
**Fig. 7** X axis velocity error and command noise comparison between computed-torque and PID control



**Fig. 8** Y axis velocity error and command noise comparison between computed-torque and PID control



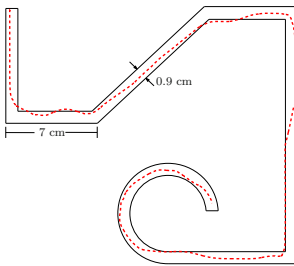
**Fig. 9** Y axis velocity tracking comparison at low velocity



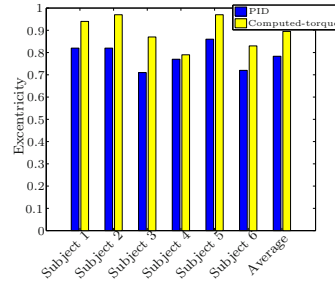
**Fig. 10** Distance of overshoots along with the time needed to complete the drawing task. The larger markers represents the group average.



overshoots (the total length of the curve outside the maze). The completion time is similar while the overshoot distance is about 32% lower with the computed-torque control. Subjects reported it was easier to perform high acceleration or deceleration, that it was easier to change direction and that the feeling was better at low velocities with the computed-torque control. The difference should be even more noticeable when the algorithms developed in this paper are applied to a more typical nonlinear robotic system for which the performance obtained with the computed-torque technique is usually much better than with PID control.



**Fig. 11** Maze trajectory with an example drawing



**Fig. 12** Comparison of the circle eccentricity

### 7.3 Circles

The last task consisted in asking the operator to trace imaginary circles in mid air. The average eccentricity (ratio of the short axis over the long axis) of the circles was measured from the position data. The results are shown in Fig. 12. In average, the eccentricity was 0.78 with PID control and 0.89 with computed-torque control, a 14% improvement. The subjects generally reported that it was easier to perform great circles with computed-torque control. Indeed, it is easier to change direction, especially because of friction compensation.

## 8 Conclusion

This paper presents computed-torque control adapted to admittance control. The admittance model, the robot identification, the computed-torque approach and the saturation considerations are presented. Finally, experimental results demonstrate the performance of the algorithm on a full-scale intelligent assist device prototype.

**Acknowledgement.** This work was supported by The Natural Sciences and Engineering Research Council of Canada (NSERC) as well as by the Canada Research Chair Program and General Motors (GM) of Canada.

## References

1. Dietsch, J.: People meeting robots in the workplace [industrial activities]. *IEEE Robotics Automation Magazine* 17(2), 15–16 (2010)
2. Chung, W., Fu, L.C., Hsu, S.H.: Motion control. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, Springer, Heidelberg (2008)
3. Kelly, R., Santibáñez Davila, V., Loria Perez, J.A.: *Control of Robot Manipulators in Joint Space*. Springer (2005)
4. Khalil, W., Dombre, E., Nagurka, M. (Reviewer): Modeling, identification and control of robots. *Applied Mechanics Reviews* 56(3), 37–38 (2003)
5. Lewis, L.F., Dawson, M.D., Abdallah, T.C.: *Robot Manipulator Control Theory and Practice*, 2nd edn. (Revised and Expanded). CRC Press (2004)
6. Slotine, J.J., Li, W.: *Applied Nonlinear Control*. Prentice Hall (October 1990)
7. Slotine, J.J.E., Li, W.: On the Adaptive Control of Robot Manipulators. *The International Journal of Robotics Research* 6(3), 49–59 (1987)
8. De Santis, A., Siciliano, B., De Luca, A., Bicchi, A.: An atlas of physical human-robot interaction. *Mechanism and Machine Theory* 43(3), 253–270 (2008)
9. Fiene, J., Kuchenbecker, K., Niemeyer, G.: Event-based haptic tapping with grip force compensation. In: 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 117–123 (March 2006)
10. Kuchenbecker, K., Fiene, J., Niemeyer, G.: Event-based haptics and acceleration matching: portraying and assessing the realism of contact. In: *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Orlando, USA, pp. 381–387 (March 2005)
11. Kuchenbecker, K., Niemeyer, G.: Improving telerobotic touch via high-frequency acceleration matching. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, USA (2006)
12. Erden, M., Tomiyama, T.: Human-intent detection and physically interactive control of a robot without force sensors. *IEEE Transactions on Robotics* 26(2), 370–382 (2010)
13. Kosuge, K., Kazamura, N.: Control of a robot handling an object in cooperation with a human. In: *Proceedings of the 6th IEEE International Workshop RO-MAN*, pp. 142–147 (1997)
14. Colgate, J.E., Peshkin, M., Klostermeyer, S.H.: Intelligent assist devices in industrial applications: a review. In: *Proceedings of the International Conference on Intelligent Robots and Systems*, Las Vegas, USA, pp. 2516–2521 (October 2003)
15. Lecours, A., Mayer-St-Onge, B., Gosselin, C.: Variable admittance control of a four-degree-of-freedom intelligent assist device. In: *IEEE International Conference on Robotics and Automation*, Saint Paul, USA, pp. 3903–3908 (May 2012)

16. Ikeura, R., Inooka, H.: Variable impedance control of a robot for cooperation with a human. In: Proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan (1995)
17. Tsumugiwa, T., Yokogawa, R., Hara, K.: Variable impedance control with regard to working process for man-machine cooperation-work system. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, USA, vol. 3, pp. 1564–1569 (October 2001)
18. Canudas de Wit, C., Nol, P., Aubin, A., Brogliato, B.: Adaptive Friction Compensation in Robot Manipulators: Low Velocities. *The International Journal of Robotics Research* 10(3), 189–199 (1991)
19. Cao, L., Schwartz, H.: Stick-slip friction compensation for pid position control. In: Proceedings of the American Control Conference, Chicago, USA, vol. 2, pp. 1078–1082 (June 2000)
20. Armstrong-Helouvry, B., Dupont, P., Wit, C.C.D.: A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica* 30(7), 1083–1138 (1994)
21. Kostic, D., de Jager, B., Steinbuch, M., Hensen, R.: Modeling and identification for high-performance robot control: an rrr-robotic arm case study. *IEEE Transactions on Control Systems Technology* 12(6), 904–919 (2004)
22. Kostic, D., Steinbuch, M., de Jager, B.: Modeling and identification for robot motion control. In: *Robotics and Automation Handbook* (2004)
23. Ljung, L., Soderstrom, T.: *Theory and Practice of Recursive Identification (Signal Processing, Optimization, and Control)*. The MIT Press (1983)
24. Hollerbach, J., Khalil, W., Gautier, M.: Model identification. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, Springer, Heidelberg (2008)
25. Belanger, P., Dobrovolny, P., Helmy, A., Zhang, X.: Estimation of angular velocity and acceleration from shaft-encoder measurements. *The International Journal of Robotics Research* 17(11), 1225–1233 (1998)
26. Moreno, J., Kelly, R.: Velocity control of robot manipulators: Analysis and experiments. *International Journal of Control* 76(14), 1420–1427 (2003)

# Sampling-Based Direct Trajectory Generation Using the Minimum Time Cost Function<sup>\*</sup>

Oscar Chuy, Emmanuel Collins, Damion Dunlap, and Aneesh Sharma

**Abstract.** This paper presents a methodology for computationally efficient, direct trajectory generation using sampling with the minimum time cost function, where only the initial and final positions and velocities of the trajectory are specified. The approach is based on a randomized A\* algorithm called Sampling-Based Model Predictive Optimization (SBMPO) that exclusively samples in the input space and integrates a dynamic model of the system. The paper introduces an *extended kinematic model*, consisting of the standard kinematic model preceded by two integrators. This model is mathematically a dynamic model and enables SBMPO to sample the acceleration and provide the acceleration, velocity, and position as functions of time that are needed by a typical trajectory tracking controller. A primary contribution of this paper is the development of an appropriate “optimistic A\* heuristic” (i.e, a rigorous lower bound on the chosen cost) based on the solution of a minimum time control problem for the system  $\ddot{q} = u$ ; this heuristic is a key enabler to fast computation of trajectories that end in zero velocity. Another contribution of this paper is the use of the extended kinematic model to develop a trajectory generation methodology that takes into account torque constraints associated with the regular dynamic model without having to integrate this more complex model as has been done previously. This development uses the known form of the trajectory following control law. The results are initially illustrated experimentally using a 1 degree of freedom (DOF) manipulator lifting heavy loads, which necessitates the development of trajectories with appropriate momentum characteristics. Further simulation results are for a 2 DOF manipulator.

---

Oscar Chuy · Emmanuel Collins · Damion Dunlap · Aneesh Sharma  
Center for Intelligent Systems, Control, and Robotics (CISCOR)  
and Department of Mechanical Engineering, Florida A&M - Florida State University,  
Tallahassee, 32310 USA  
e-mail: chuy@eng.fsu.edu, ecollins@fsu.edu,  
damion.d.dunlap@navy.mil, as10ac@my.fsu.edu

\* The funding for this research was provided by the National Science Foundation under award CMMI-1130286.

## 1 Introduction

The traditional approach to trajectory generation for robots is described by four steps [8]: 1) compute a collision free path, 2) smooth the path to produce a path that the robot can actually follow, 3) reparameterize the smoothed path to create a trajectory, and 4) design a feedback controller to track the trajectory. As discussed in [8], this paradigm can lead to problems since a later step might not succeed due to a choice made in an earlier step. Even when it does succeed it can lead to inefficient solutions. These problems have motivated recent research to consider the direct generation of trajectories such that steps 1 through 3 are integrated [5, 7, 11]. This paper uses sampling and A\* optimization to directly generate trajectories that minimize a total time cost function, take into account the torque constraints of the system, and avoids direct integration of the regular dynamic model of the system. The approach is based on a randomized A\* algorithm, called *Sampling-Based Model Predictive Optimization* (SBMPO) [4, 5]. In addition, the approach is based on the use of an *extended kinematic model*, consisting of the standard kinematic model preceded by two integrators. This model is mathematically a dynamic model and enables SBMPO to sample the acceleration and provide the acceleration, velocity, and position as functions of time that are needed by a typical trajectory tracking controller. A primary contribution of this paper is the development of an appropriate “optimistic A\* heuristic” (i.e., a rigorous lower bound on the chosen cost) based on the solution of a minimum time control problem for the system  $\ddot{q} = u$ ; this heuristic is a key enabler to efficient computation of trajectories that end in zero velocity. Another important contribution of this paper is the use of the extended kinematic model to develop a trajectory generation methodology that takes into account torque constraints associated with the regular dynamic model without having to integrate this more complex model as has been done previously.

The RRT approach of [9] is a sampling based algorithm for kinodynamic motion planning. However, unlike the randomized A\* approaches [3, 4], this approach is not based on optimization. RRT-based motion planners that do utilize optimization are discussed in [7, 6]. However, they do not use A\* optimization and do not appear to be able to take advantage of the efficiency obtained by using the prediction associated with the optimistic heuristic. The extended kinematic model and the methodology that uses it to take into account torque constraints while avoiding direct integration of the regular dynamic model can likely be extended to many of the kinodynamic planning algorithms. The optimistic A\* heuristic developed in this paper for a minimum time cost function should be applicable to the alternative randomized A\* algorithms.

In [5] the trajectory generation methodology was based on directly planning using the closed-loop model, i.e., the dynamic model that includes the tracking controller. This approach is based on a philosophy that differs from the standard approach described above and in [8] since the controller is designed prior to the development of the trajectory. The use of the closed-loop model is effective but can be computationally expensive. Hence, as mentioned previously, this research pursues an approach to the problem based on the use of an extended kinematic model that

is more computationally efficient, albeit more conservative since it must conservatively estimate the contribution of the feedback controller to the system torques and forces.

The results are illustrated in actual experimentation using a 1 degree of freedom (DOF) robot, in particular a 1 DOF manipulator, and subsequent simulation results are for a 2 DOF manipulator. It is assumed that the manipulator may have a load that is so heavy that it cannot be lifted quasistatically to the desired goal due to the torque limitations of the actuators. Hence, this problem requires that the trajectory have sufficient momentum to overcome the torque constraints. The problem of heavy lifting for manipulators has been considered in prior research [14]. However, the previous approach, which uses gradient based optimization, does not consider minimum time problems. In addition, the final time of the trajectory needs to be specified for [14] to work and this leads to multiple swings, even when they are not needed. In contrast, the approach presented in this paper will naturally develop the final time and number of swings. It is conjectured that even when applied to high DOF systems, the approach developed here has the potential to be computationally faster. This latter issue will be resolved after this approach is applied to more complex problems.

Previous studies in trajectory generation and minimum time control of robotic manipulator have been presented in [1, 12]. These studies consider the problem of determining manipulator control commands to move a manipulator in minimum time along a specified path subject to torque/force constraints. A major difference between [1, 12] and this study is that the approach presented in this paper does not need a path to create a trajectory. In fact, given the initial and final positions and velocities, the trajectory is directly generated subject to torque/force constraints as a result of the planning process. However, sampling, which is used for fast computations, always leads to suboptimality; hence the trajectories developed by the methodology presented in this paper are never truly optimal, although some of the trajectories are nearly optimal.

## 2 Description of Sampling-Based Model Predictive Optimization (SBMPO)

SBMPO is a sampling-based algorithm for motion planning with kinematic and dynamic models. It can plan using a variety of cost functions, including the standard sum of the squared error cost function used in Model Predictive Control (MPC) [10]. SBMPO was motivated by a desire to employ sampling and A\*-type optimization in place of the nonlinear programming that is commonly employed for optimization in MPC.

Fig. 1 shows the block diagram of a trajectory planning strategy that uses SBMPO. The model, cost evaluation, and heuristic are supplied by the user. It should be noted that in the SBMPO algorithm, a graph is created from start to goal and each vertex on the graph keeps track of the states of the system, the control input, and the cost associated with the state. The following are the main steps of SBMPO [4]:

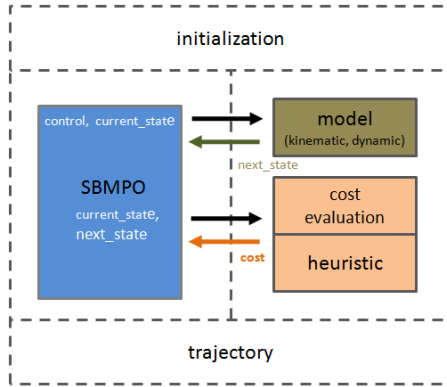


Fig. 1 Trajectory planning using Sampling-Based Model Predictive Optimization (SBMPO)

1. **Select the vertex with highest priority in the queue**
2. **Sample the input space**
3. **Add a new vertex to the graph**
4. **Evaluate the new vertex cost**
5. **Repeat 2-4 for B number of successors**
6. **Repeat 1-5 until one of the stopping criteria is true**

### 3 Trajectory Planning Using an Extended Kinematic Model

This section will discuss how the trajectory components (i.e., position, velocity, and acceleration) are generated. First, it is assumed that the robot is operating without saturating the control actuators. This is ensured in manipulator applications if the load is light enough to be lifted quasi-statically from its initial position to the final position. In this case the kinematic model can be used to develop a manipulator trajectory. The question is how do we use the kinematic model to develop the desired acceleration, velocity, and position (i.e.,  $\ddot{q}_d$ ,  $\dot{q}_d$ , and  $q_d$ ) needed by the trajectory tracking controller? This can be accomplished letting the model in the trajectory planning approach described in Fig. 1 be the “extended kinematic model,” of Fig. 2, where  $q_d \in R^n$ ,  $x \in R^m$ , and  $I_n$  is the  $n \times n$  identity matrix. In Fig. 1 and below  $T$  denotes the sample period for trajectory generation. It follows from the integration operations of Fig. 2 that

$$\dot{q}_d(kT) = \dot{q}_d(kT - T) + T\ddot{q}_d(kT), \tag{1}$$

$$q_d(kT) = q_d(kT - T) + T\dot{q}_d(kT). \tag{2}$$

Since  $\ddot{q}_d$  is the input to the extended kinematic model and SBMPO samples the input, SBMPO is able to generate  $\ddot{q}_d$  by sampling and  $\dot{q}_d$ , and  $q_d$  by model integration. Note that SBMPO must assume some bounds on  $\ddot{q}_d$ , which are given by

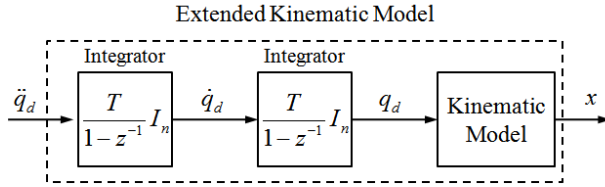


Fig. 2 Extended kinematic model

$$-a_i \leq \ddot{q}_{d,i} \leq b_i, \quad i = 1, \dots, n, \tag{3}$$

where  $a_i > 0$  and  $b_i > 0$ .

The heuristic in Fig. 1 is optimistic if it is chosen to be zero and the termination criteria may be chosen such that SBMPO only stops when it reaches the goal at a desired velocity. However, in this case SBMPO becomes very computationally inefficient. As is well known, A\*-type algorithms become efficient when a fairly non-conservative, optimistic heuristic is used. What is needed is an appropriate heuristic corresponding to the minimum time cost.

### 3.1 Development of a “Good” Minimum Time Heuristic

Motivated by (3), consider a system described by

$$\ddot{q} = u; \quad q(0) = q_0, \quad \dot{q}(0) = \omega_0, \tag{4}$$

where  $q$  and  $u$  are scalars and  $u$  is bounded by  $-a \leq u \leq b$ . The state space description of (4) is given by

$$\dot{q}_1 = q_2, \quad \dot{q}_2 = u; \quad q_1(0) = q_0 \triangleq q_{1,0}, \quad q_2(0) = \omega_0 \triangleq q_{2,0}, \tag{5}$$

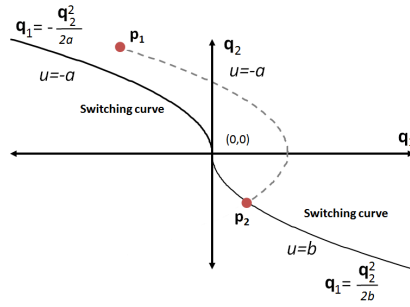
where  $q_1 = q$  and  $q_2 = \dot{q}$ . It is desired to find the minimum time needed to transfer the system from the original state  $(q_{1,0}, q_{2,0})$  to the final state  $(q_{1,f}, 0)$ , where  $q_{1,f} \triangleq q_f$ . Since the solution for transferring the system from  $(q_{1,0}, q_{2,0})$  to the origin  $(0, 0)$  is easily extended to the more general case by a simple change of variable, for ease of exposition we assume that  $q_{1,f} = 0$ .

The minimum time control problem described above can be solved by forming the Hamiltonian and applying the “Minimum Principle” (often referred to as “Pontryagin’s Maximum Principle”) as described in [2]. In fact, the above problem is solved in [2] for the case when the parameters  $a$  and  $b$  are given by  $a = b = 1$ . Generalizing these results yields that the minimum time is the solution  $t_f$  of

$$t_f^2 - \frac{2q_{2,0}}{a} t_f = \frac{q_{2,0}^2 + 2(a+b)q_{1,0}}{ab}, \quad \text{if } q_{1,0} + \frac{q_{2,0}|q_{2,0}|}{2b} < 0, \tag{6}$$

$$t_f^2 + \frac{2q_{2,0}}{b} t_f = \frac{q_{2,0}^2 - 2(a+b)q_{1,0}}{ab}, \quad \text{if } q_{1,0} + \frac{q_{2,0}|q_{2,0}|}{2a} > 0.$$





**Fig. 3** Illustration of bang-bang minimum time optimal control which yields the minimum time solution  $t_f$  of (6)

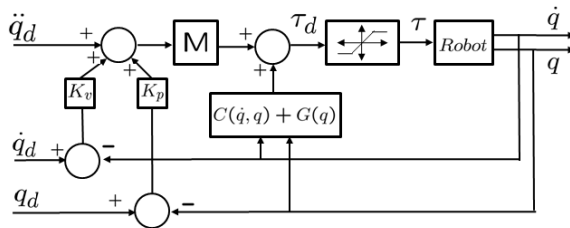
The minimum time  $t_f$  computed using (6) corresponds to the “bang-bang” optimal controller illustrated by Fig. 3, which shows switching curves that take the system to the origin using either the minimum or maximum control input (i.e.,  $u = -a$  or  $u = b$ ). Depending on the initial conditions, the system uses either the minimum or maximum control input to take the system to the appropriate switching curve. For example, if  $(q_{1,0}, q_{2,0})$  corresponds to point  $p_1$  in Fig. 3, then the control input should be  $u = -a$  until the system reaches point  $p_2$  on the switching curve corresponding to  $u = b$ . At this point the control is switched to  $u = b$ , which will take the system to the origin.

### 4 Trajectory Planning Using a Physics-Based Dynamic Model

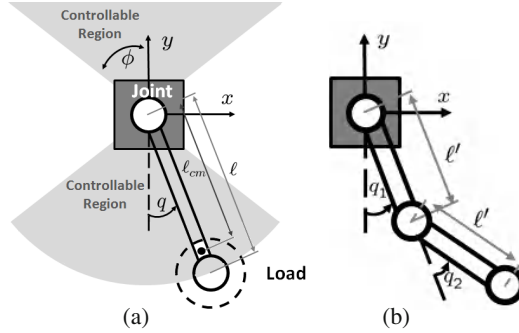
Consider a dynamic model of a robot system given by

$$M(q)\ddot{q} + C(\dot{q}, q) + G(q) = \tau, \tag{7}$$

where  $\ddot{q}$ ,  $\dot{q}$ , and  $q \in R^n$  are respectively the angular acceleration, velocity, and position,  $M \in R^{n \times n}$  is the inertia,  $C(\dot{q}, q) \in R^n$  is the friction term, and  $G(q) \in R^n$  is the gravity term, and  $\tau \in R^n$ .



**Fig. 4** Computed torque tracking controller



**Fig. 5** Manipulators used in the experiments (a) 1DOF manipulator (b) 2DOF manipulator

Using a computed torque tracking controller [13] as shown in Fig. 4, the inputs to the controller are the desired acceleration, velocity, and position (i.e.,  $\ddot{q}_d$ ,  $\dot{q}_d$ , and  $q_d$ ) and the desired torque  $\tau_d$  is calculated as

$$\tau_d = M(q)[\ddot{q}_d + K_v(\dot{q}_d - \dot{q}) + K_p(q_d - q)] + C(\dot{q}, q) + G(q), \tag{8}$$

where  $M(q)\ddot{q}_d$  is the feedforward term,  $C(\dot{q}, q) + G(q)$  are respectively the friction and gravity compensation terms,  $M(q)[K_v(\dot{q}_d - \dot{q}) + K_p(q_d - q)]$  is the feedback term, and  $K_v \in R^{n \times n}$  and  $K_p \in R^{n \times n}$  are the feedback gains.

The most direct method of taking into account the actuator limitations in planning is to simply choose the closed loop model as the SBMPO model [5]. The problem with this basic approach is that the controller is updated at a fast rate, e.g., 1 kHz, which must be the update rate for the closed-loop model. This leads to long planning times.

One alternative approach does not explicitly consider the tracking controller. Instead, it is assumed that the tracking controller is such that  $q(t) \approx q_d$ ,  $\dot{q}(t) \approx \dot{q}_d(t)$ , and results in to the feedback terms in (8) being small compared to the remaining terms. In this case, the feedback torque computed by the tracking controller, given by (8), is dominated by the feedforward and friction and gravity compensation portions of the controller such that it is assumed that

$$\tau_d(t) = M(q)\ddot{q}_d(t) + C(\dot{q}_d(t), q_d(t)) + G(q_d(t)), \tag{9}$$

where in the experiments of Section 5,  $C(\dot{q}_d(t), q_d(t))$  was set to zero. (Note that the  $q(t)$  and  $\dot{q}(t)$  in the  $C(\cdot)$  and  $G(\cdot)$  terms of (8) have been replaced by  $q_d(t)$  and  $\dot{q}_d(t)$  in (9).) Although (9) will sometimes underpredict the magnitude of  $\tau_d(t)$  and hence introduces the need for conservatism in the trajectory planning, it has the possibility of leading to fast computations. This new approach uses the extended

kinematic model, as in the previous section, but modifies the sampled (commanded acceleration) inputs to avoid saturation of the actuator torque.

Below we assume that  $t \in [NT, NT + T)$ , where  $N$  is some positive integer. During this interval  $\ddot{q}_d(t)$  is held constant at its sampled value  $\ddot{q}_d(NT)$ , i.e.,

$$\ddot{q}_d(t) = \ddot{q}_d(NT), \quad t \in [NT, NT + T). \tag{10}$$

It follows that for  $t \in [NT, NT + T)$

$$\dot{q}_d(t) = \dot{q}_d(NT) + \ddot{q}_d(t)t, \tag{11}$$

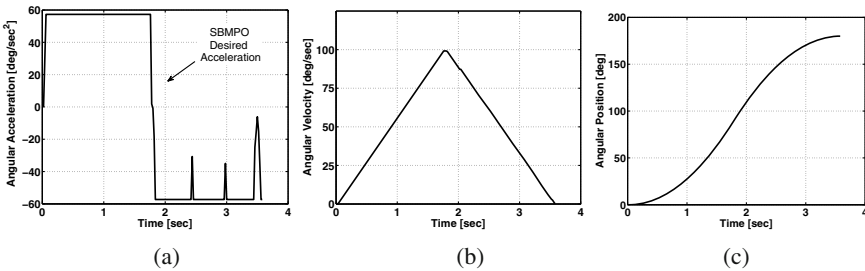
$$q_d(t) = q_d(NT) + \dot{q}_d(NT)t + \frac{1}{2}\ddot{q}_d t^2. \tag{12}$$

Ideally, one should check whether the torque  $\tau_d(t)$  given by (9) violated its saturation constraints ( $-\tau_{max,i} < \tau_{d,i}(t) < \tau_{max,i}$ ,  $i = 1, \dots, n$ ), in the entire interval  $t \in [NT, NT + T)$ . However, the torque will only be evaluated at the initial instant  $t = NT$ , which introduces potential analysis error since the torque can violate the saturation constraints later in the interval. To account for this error and that associated with neglecting the feedback terms in (9), the assumed saturation range is reduced such that for some  $\bar{\tau}_i < \tau_{max,i}$  it is determined whether

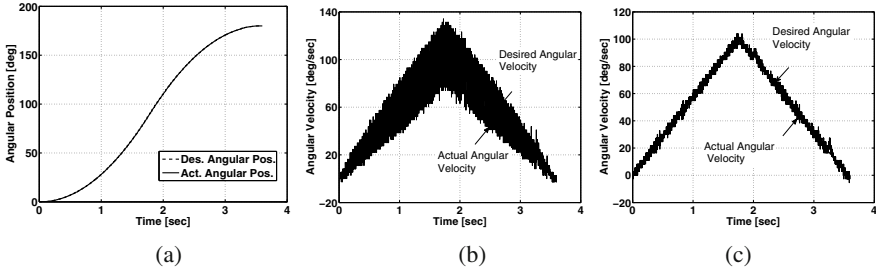
$$-\bar{\tau}_i < \tau_{d,i}(NT) < \bar{\tau}_i, \quad i = 1, \dots, n. \tag{13}$$

For each  $i \in \{1, \dots, n\}$  if  $\tau_{d,i}(NT) > \bar{\tau}_i$ , then  $\tau_{d,i} \leftarrow \bar{\tau}_i$  or if  $\tau_d(NT) < -\bar{\tau}$ , then  $\tau_{d,i} \leftarrow -\bar{\tau}_i$ . If torque saturation was detected in the above tests, then to choose a desired acceleration that does not violate the torque constraints, it follows from (9) that one can let

$$\ddot{q}_d(NT) \leftarrow M(q)^{-1}[\tau_d - C(\dot{q}_d(NT), q_d(NT)) - G(q_d(NT))] \tag{14}$$



**Fig. 6** Trajectory components generated based on the extended kinematic model of 1 DOF manipulator: (a) Angular acceleration command generated by SBMPO, (b) Angular velocity command from integration of the acceleration, (c) Angular position command from integration of the velocity



**Fig. 7** Manipulator trajectory tracking results for no load case with trajectory generated using the extended kinematic model: (a) Desired and actual angular positions, (b) Desired and actual angular velocities without interpolation, (c) Desired and actual angular velocities with interpolation

### 5 Experiments and Results Using the Extended Kinematic and Dynamic Models

This section discusses the experiments and results of the study where one and two degrees of freedom manipulators were used to evaluate the trajectories generated by SBMPO. A Core 2 Duo 2.93 GHz processor was used for the computations and SBMPO was implemented using C/C++. SBMPO was used to plan a trajectory with a 25 Hz sampling rate, i.e.,  $T = 0.04$  sec for planning using the extended kinematic model and the physics-based dynamic model. Fig. 5(a) and Fig. 5(b) show respectively the schematic of the 1 DOF and 2 DOF manipulators used in this study. Fig. 5(a) displays the controllable region for a heavy load; this region is symmetric with respect to both the x-axis and the y-axis and is completely characterized by angle  $\phi$ . Table 1 shows the dynamic parameters of the 1 DOF manipulator.

**Table 1** Key Parameters of 1 DOF Manipulator System with 2.27 kg. and 4.54 kg. Load

Parameter	Load 2.27 kg.	Load 4.54 kg.
arm mass $m$	0.81 kg.	0.81 kg.
total length $\ell$	0.56 m	0.56 m
maximum torque $\tau_{max}$	11.3 Nm	11.3 Nm
inertia $M$	0.92 kg-m <sup>2</sup>	1.6 kg-m <sup>2</sup>
center of mass $\ell_{cm}$	0.49 m	0.45 m
controllable region angle $\phi$	45°	23°

The 1 DOF manipulator is driven by a Maxon motor RE40 150 W coupled with a Maxon GP52 gearing system with a gear ratio of 66:1. Its angular position is sensed by an encoder, with 500 pulse per motor revolution, which is directly attached to the motor. The motor driver used in this study is configured in torque/current mode and has the ability to output the actual current of the system. The torque calculation, tracking controller implementation, angular position sensing, and velocity

calculation from the joint position are implemented using a PIII 900 MHz computer running the QNX realtime system with a 1 kHz sampling rate.

### 5.1 Planning and Experimental Results Using Extended Kinematic Model

In this experiment the 1 DOF manipulator is used and is unloaded. Referring to Fig. 5(a), the task is to move the manipulator from rest at a starting position (0m,-0.56m), where the end effector is vertically down, to rest at a goal position (0m,0.56m), where the end effector is vertically up. Preliminary experiments with the manipulator revealed that trajectories in which the angular acceleration is constrained to be within  $\pm 1 \text{ rad/sec}^2$  do not lead to saturation of the motor torque. This constraint does introduce a degree of conservatism to the trajectory planner, but is necessary since the torque constraints are not explicitly taken into account in the planning process. This is a general limitation of relying on only the kinematic model in planning.

The computational time was 0.020 sec, which is quite fast. It should be noted that when alternative optimistic heuristics were used the algorithm failed to converge due to memory limitations, indicating that the optimistic heuristic defined by (6) is vital. Fig. 6(a) shows the resulting desired acceleration (obtained by sampling), which is fairly close to the optimal bang-bang control law, and Figs. 6(b) and 6(c) show the corresponding velocity and position obtained via integration of the extended kinematic model. These three figures define the desired trajectory. Figs. 6(b) and 6(c) show that the desired velocity and position are smooth, which is a very desirable property.

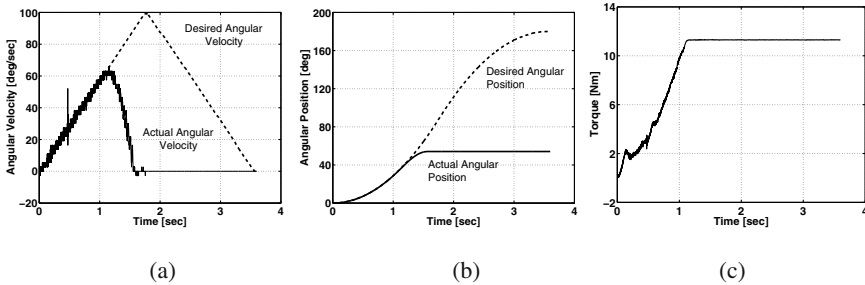
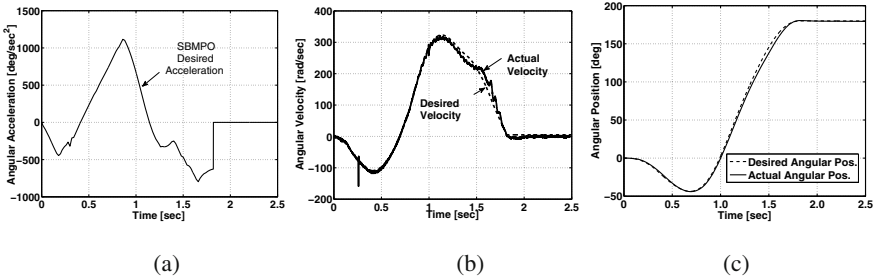


Fig. 8 Manipulator trajectory tracking results for a 2.27 kg. load with trajectory generated using the extended kinematic model: (a) Desired and actual angular velocities, (b) Desired and actual angular positions, (c) Applied torque

The trajectory components shown in Figs. 6(a), 6(b), and 6(c) were fed to the tracking controller of Fig. 4. Fig. 7(a) compares the desired and actual (experimental) angular position and it is seen that accurate tracking is achieved. The maximum absolute error norm between the desired and actual angular position is  $0.27^\circ$ . Despite this accurate position tracking, Fig. 7(b) reveals that accurate velocity tracking



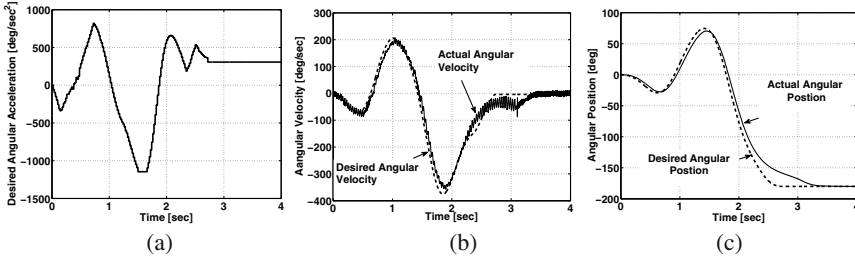
**Fig. 9** Manipulator generation and trajectory tracking results for 2.27 kg. load using the dynamic model and kinematic model: (a) Angular acceleration command generated by SBMPO, (b) Desired and actual angular velocities, (c) Desired and actual angular positions

is not achieved since the experimental velocity displays high frequency oscillations about the desired velocity. The undesired vibrations are due to the step changes in the accelerations that occur at each sample period; these steps serve as inputs to the manipulator system that excite unmodelled dynamics such as flexible modes or backlash in the gears, the latter of which are the unmodelled dynamics in the current experiment. The vibrations can be reduced by reducing the amplitudes of these step changes. This can be achieved by decreasing the trajectory sampling period  $T$  from 0.04 sec to 0.001 sec, the sample period of the tracking controller. However, this will greatly increase the planning time. An alternative and much less computationally costly solution is to linearly interpolate the desired commands in between time steps as discussed in [13]. Fig. 7(c) compares the resulting experimental velocity with the desired velocity. As expected, there is a very significant reduction in the vibrations such that this velocity much more closely matches the desired velocity.

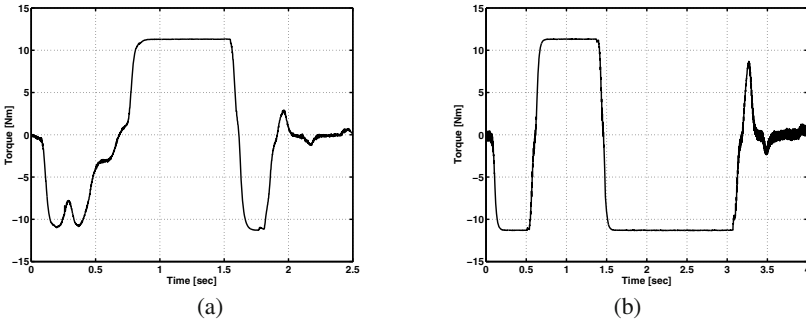
In the next experiment a 2.27 kg. heavy load is added to the manipulator. The SBMPO trajectory developed using the extended kinematic model (essentially the trajectory of Figs. 6(a), 6(b), and 6(c), modified slightly by the linear interpolation described above) was again fed to the tracking controller of Fig. 4. Fig. 8(a) and Fig. 8(b) respectively compare the desired and actual velocity and position. The system clearly fails to track the desired trajectory due to inadequate momentum when the motor torque saturates as shown in Fig. 8(c), illustrating the limitations of planning with only a kinematic model. Since the dynamic model includes a representation of the torque saturation, it is used in the planning of the next section.

### 5.2 Planning and Experimental Results Using Dynamic Model

In this experiment, the manipulator is loaded with a 2.27 kg. weight and the dynamic parameters are shown in Table 1 under the 2.27 kg. load. Referring to Fig. 5(a), the task is again to move the manipulator from rest at a starting position (0m,-0.56m) to rest at a goal position (0m,0.56m). The angular accelerations that are used to calculate the heuristic are set to  $\pm 10$  rad/sec<sup>2</sup>, such that in (4)  $a = b = 10$  rad/sec<sup>2</sup>. These



**Fig. 10** Manipulator trajectory tracking results for 4.54 kg. load using the dynamic model and kinematic model: (a) Angular acceleration command generated by SBMPO, (b) Desired and actual angular velocities, (c) Desired and actual angular positions



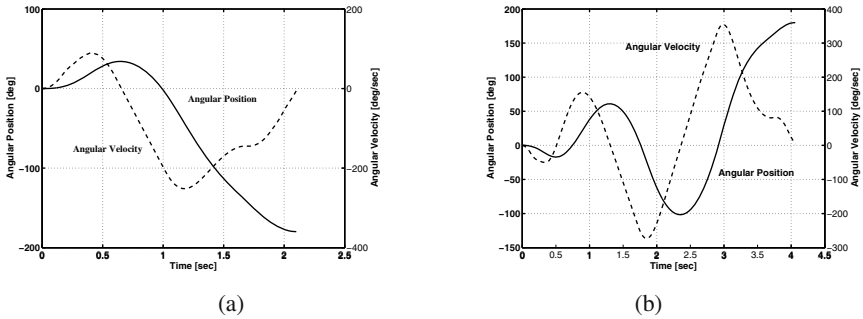
**Fig. 11** Torque applied to the 1 DOF manipulator in (a) experiment with 2.27 kg. load (b) experiment with 4.54 kg. load

bounds were determined by simulating the behavior of the robot when maximum torques were applied.

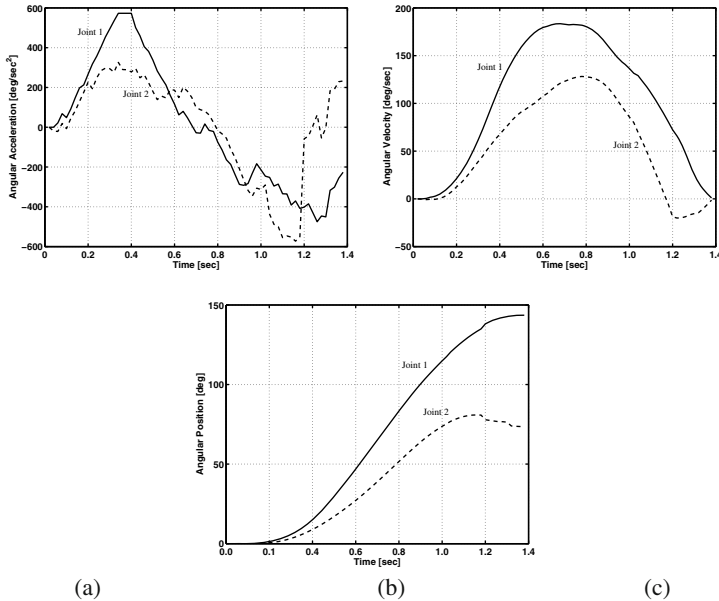
SBMPO computed a trajectory in 0.141 sec with a 25 Hz trajectory update rate (i.e.,  $T = 0.04$  sec). Fig. 9(a) shows the resulting acceleration commands, which are far from the bang-bang commands produced by using the extended kinematic model, while Figs. 9(b) and 9(c) respectively compare the desired and actual velocities and the desired and actual positions and Fig. 11(a) shows the applied torque. Figs. 9(b) and 9(c) show that the SBMPO trajectory reaches the desired position at zero velocity and is stabilized about this position by the trajectory tracking controller. This controller is able to track the trajectory despite the torque saturation observed in Fig. 11(a) due to the momentum associated with the trajectory. Notice from Figs. 9(b) and 9(c) that the manipulator moves in the opposite direction of the final swing to obtain the required momentum needed to pass the uncontrollable region as it swings to the vertical position.

Motion planning is next performed for the manipulator with a larger load of 4.54 kg. The corresponding dynamic parameters are shown in Table 1. SBMPO computed the trajectory in 0.110 sec and the results are shown in Figs. 10 and Fig. 11(b).

Figs. 10 (b) and (c) show that the manipulator needs to swing twice to gain enough momentum and reach the goal. The trajectory tracking is degraded due to the torque saturation seen in Fig. 11(b). This degradation can be reduced by decreasing the value of  $\bar{\tau}$  in (13).



**Fig. 12** Manipulator trajectory planning results in the presence of obstacle (a) joint angular position and velocity of 1 DOF manipulator with 2.27 kg. load and obstacle is located at (0.25m,0) (b) joint angular position and velocity of 1 DOF manipulator with 4.54 kg. load and obstacle is located at (-0.25m,0.25m)



**Fig. 13** Trajectory components generated based on the extended kinematic model of 2 DOF manipulator: (a) joint angular accelerations generated by SBMPO, (b) joint angular velocities from integration of the acceleration, (c) joint angular positions from integration of the velocity



### 5.3 Planning Using Dynamic Model in the Presence of Obstacles

The next planning tasks demonstrated SBMPO's ability to generate trajectories in the presence of obstacles. The same task was again given to the 1 DOF manipulator with the 2.27 kg. load and an obstacle was placed at (0.25m,0), which limited the maximum joint angle to be +90°. SBMPO computed a trajectory in 0.150 sec and Fig. 12 (a) shows the result. The generated trajectory avoided the obstacle in contrast to the trajectory shown in Fig. 9(c). The task was also repeated for 4.54 kg. load and the obstacle was placed at (-0.25m,0.25m), which limited the minimum joint angle to be -135°. SBMPO computed a trajectory in 0.123 sec and Fig. 12 (b) shows the result. Again, it can be noticed that the generated trajectory avoided the obstacle in contrast to the result shown in Fig. 10. The results clearly show the ability of SBMPO to generate trajectories in the presence of obstacles.

### 5.4 Planning Using Extended Kinematic Model for 2 DOF Manipulator

This section discusses the trajectory planning for a 2 DOF manipulator using the extended kinematic model. The joint accelerations were sampled and the extended kinematic models were employed to generate trajectory components for each joint. The computation of the time-to-goal ( $t_f$ ) is slightly modified to fit the 2 DOF manipulator. Although, sampling is done in the input space, SBMPO has the ability to use a cost metric computed based on the output space. The position and velocity in each of the x and y axes on the task space (see Fig. 5(b)) were used as the basis to compute  $t_f$ . Ideally, the time-to-goal (i.e., the heuristic) can be computed as  $t_f = \max(t_{f_x}, t_{f_y})$ , where  $t_{f_x}$  and  $t_{f_y}$  are the heuristics associated respectively with x and y movements of the end effector. However, the initial results showed that the approach leads to long planning times since there are several states that have same cost<sup>1</sup>. As an ad hoc approach, in the current implementation, time-to-goal is implemented as  $t_f = t_{f_x} + t_{f_y}$ . This approach gives nice results, however, it is not rigorous and not as fast as desired. A modification of SBMPO is currently being performed in which the heuristic is  $t_f = \max(t_{f_x}, t_{f_y})$  and an auxiliary time-to-goal  $t_{f_a} = \min(t_{f_x}, t_{f_y})$  is used as a tie breaker if there are states with the same cost. This appears to be the proper approach for using minimum time cost functions and will be detailed in a later paper.

In planning with the 2 DOF manipulator, a task is given to move the manipulator from rest at a starting position (0m,-0.5m), where the end effector is vertically down, to rest at a goal position (0m,0.4m). SBMPO computed a trajectory for both joints in 0.251 sec and Fig. 13 shows the result. Note that in Fig. 13(b), the joint angular velocities end at zero as desired.

<sup>1</sup> Development of heuristics for minimum time cost functions can lead to a loss of information that does not occur when developing heuristics for minimum distance cost functions.

## 6 Conclusions

This paper presented a methodology for computationally efficient, direct trajectory generation using sampling with the minimum time cost function, where only the initial and final positions and velocities of the trajectory are specified. It was shown that an extended kinematic model can be used to generate trajectories in applications where there is no danger of violating the actuator limitations. However, when actuator saturation is a serious concern, a physics-based dynamic model should also be used. This study also presented the construction of an optimistic “A\* heuristic” based on the solution of a simple minimum time control problem. This heuristic is a key enabler to fast computation of trajectories that end in zero velocity. The experimental results using a simple manipulator showed that for heavy manipulator loads, planning based only on kinematic models can lead to failure. This problem was solved by incorporating torque information from a physics-based dynamic model. These results apply in general to planning that requires the trajectories to satisfy certain momentum requirements. Although optimization with the minimum time cost function is used, the resulting trajectories are suboptimal; this is inherent to sampling, which permits a tradeoff between fast computations and optimality.

Ongoing work is focused on the implementation of the proposed approach on higher degree of freedom systems such as 6 DOF manipulators and autonomous space vehicles.

## References

- [1] Bobrow, J.E., Dubowsky, S., Gibson, J.S.: Time-Optimal Control of Robotic Manipulators Along Specified Path. *Int. Journal of Robotics Research*, 3–17 (1985)
- [2] Bryson, A.E., Ho, Y.: *Applied Optimal Control: Optimization, Estimation, and Control*. Hemisphere Publishing Corporation (1975)
- [3] Diankov, R., Kuffner, J.: Randomized Statistical Path Planning. In: *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 1–6 (2007)
- [4] Dunlap, D., Caldwell, C., Collins, E., Chuy, O.: Motion Planning for Mobile Robots Via Sampling-Based Model Predictive Optimization: Recent Advances in Mobile Robotics. *InTech* (2011)
- [5] Dunlap, D.D., Collins, E.G., Yu, W., Charmane, C.: Motion Planning for Steep Hill Climbing. In: *IEEE International Conference on Robotics and Automation* (2011)
- [6] Karaman, S., Frazzoli, E.: Incremental Sampling-based Optimal Motion Planning. In: *Robotics: Science and Systems, RSS* (2010)
- [7] Karaman, S., Frazzoli, E.: Optimal Kinodynamic Motion Planning Using Incremental Sampling-based Method. In: *IEEE Conference on Decision and Control*, pp. 7681–7687 (2010)
- [8] LaValle, S.: Motion Planning: Part II. *Robotics and Automation Magazine*, 108–118 (2011)
- [9] LaValle, S., Kuffner, J.: Randomized kinodynamic planning. *International Journal of Robotics Research*, 378–400 (2001)
- [10] Maciejowski, J.M.: *Predictive Control with Constraints*. Prentice-Hall (2002)

- [11] Plaku, E., Kavraki, L., Vardi, M.: Motion planning with dynamics by synergistic combination of layers of planning. *IEEE Transaction on Robotics*, 469–482 (2010)
- [12] Shin, K., McKay, N.D.: Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints. *IEEE Transactions on Automatic Control*, 531–541 (1985)
- [13] Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: *Robotics Modelling, Planning and Control*. Springer (2009)
- [14] Wang, C.Y.E., Timoszyk, W.K., Bobrow, J.E.: Payload Maximization for Open Chained Manipulators: Finding Weightlifting Motions for a Puma 762 Robot. *IEEE Transactions on Robotics and Automation*, 218–224 (2001)

# Antagonistic Control of Multi-DOF Joint

Koichi Koganezawa

**Abstract.** This paper presents a mechanical system that fundamentally mimics a human musculo-skeletal system aiming for using it in anthropomorphic robots or artificial limbs for disabled persons. At first, it introduces a mechanical module called ANLES (Actuator with Non-Linear Elasticity System). A new type of ANLES; rotary-type ANLES is introduced first in this paper, in addition to the formerly developed ANLES; linear type ANLES. They can be used like a voluntary muscle in a musculo-skeletal structure. Next it derives dual equations to independently control joint angle and joint stiffness, in which antagonistic alignment of the ANLEs similar to a musculo-skeletal system is premised. It follows to show an application of the two types of ANLES into a three DOF artificial joint arranged to use as a wrist joint of an anthropomorphic robot. The experimental results of the joint stiffness and joint angle control elucidates that the developed mechanism effectively mimics the human musculo-skeletal system.

**Keywords:** musculo-skeletal system, non-linear elasticity, antagonistic alignment, stiffness control.

## 1 Introduction

It is easily found that some dexterous motions of human articulations are due to the capability of regulating the stiffness in accordance with a task that he/she is about to do. The musculo-skeletal system of human articulations is able to regulate its stiffness mechanically rather than by efferent command from the CNS (central nervous system) using exteroceptive force feedback. The key mechanism for regulating the stiffness is antagonistic structure of the musculo-skeletal system; one agonist and its antagonist muscles counteractively drive one articulation. Simultaneous stretching of both muscles provides high stiffness of the articulation and both relaxing gives us the low stiffness. It is notified that the non-linear elasticity of the muscles is prerequisite for the agonist-antagonist alignment for regulating the stiffness. Some amount of displacement of joint angle requires a

---

Koichi Koganezawa  
Dept. of Mechanical Engineering, Tokai University  
4-1-1 Kitakaname Hiratsuka, Kanagawa 259-1292, Japan  
e-mail: kogane@keyaki.cc.u-tokai.ac.jp

small torque at the joint under the equilibrium state of low stretching of both muscles. On the other hand the equilibrium state under high stretching requires a respectively large torque to provide the same amount of angle displacement. So the stiffness is regulated according to magnitude of stretching of both muscles. It is obvious that linear elasticity does not provide such a stiffness change. A vast amount of physiological studies have elucidated skeletal muscles have such a non-linear elasticity [1][2][3][4].

Some studies for investigating the stiffness of human arms elucidate that the stiffness ellipsoid of the arm's endpoint is adjustable in its volume by stretching muscles[5], but its shape is roughly determined by the arm's posture [6].

Some studies in the field of robotics deal with the antagonistic control of joints [7][8][9][10][11][12] and pointed out the importance of the non-linear characteristics of the elastic elements to control the stiffness of the joint [9][10][11], but there have been few papers that propose effective method of stiffness control, although some theoretical approach for stiffness control provides valuable insights [11][13][14].

This study assumes an artificial joint that is controlled by at least two actuator units having a elastic characteristic similar to human voluntary muscles. It is called the *antagonistically driven joints* (ADJ).

There have been some approaches to comprise the ADJ using linear actuators that works like muscle. The most successful approach developed so far will be those of using the McKibben type pneumatic actuator [15][16]. Although the pneumatic rubber actuator inherently has non-linear elasticity, it has some drawbacks such as, the difficulty of designing the non-linear elasticity, the heat sensitivity, large volume for external air-compressor, etc. There are some recent another approaches to develop the non-linear elastic module used to control the stiffness of ADJ [17][18][19], which presented ingenious mechanical devices to design the non-linear elasticity.

Recent ardency for developing ADJ introduced above suggests its importance and potentiality of its nearly future application. Because a forthcoming "*Personal Care Robot*" will be anticipated to be *inherently safe* when it comes to interact with external objects, especially with human body [20] It requires mechanical resiliency and adaptability rather than those of virtually fabricated with a lot of sensory feedback loops. Our study is on the line of research tendency shown above to compose ADJ.

Recently various types of mechanisms classified in the variable stiffness actuator (VSA) have been proposed [21] [22][23][24], of which pioneering work will be the MIA (mechanical impedance adjuster) developed by Morita and Sugano[25]. The VSA approach aims to endow robots with an intrinsically safe property for using them in a human-robot interactive environment. The VSA is an actuator unit that has adjustable elastic element between a rotary joint and a rotary actuator. Therefore it does not aim to constitute an ADJ like a musculo-skeletal system.

This paper proposes the alternative mechanism used as an artificial muscle for an ADJ. It is on the same line of [18][19] in the sense of composing the non-linear elasticity through converting the force generated by a normal linear spring on the process of its transmission[26][27][28]. It aims to develop an intrinsically safe device as same as the VSA approach.

In the following section, two types of ANLES (actuator with non-linear elasticity system); the linear type ANLES (*l*-ANLES) and the rotary type ANLES

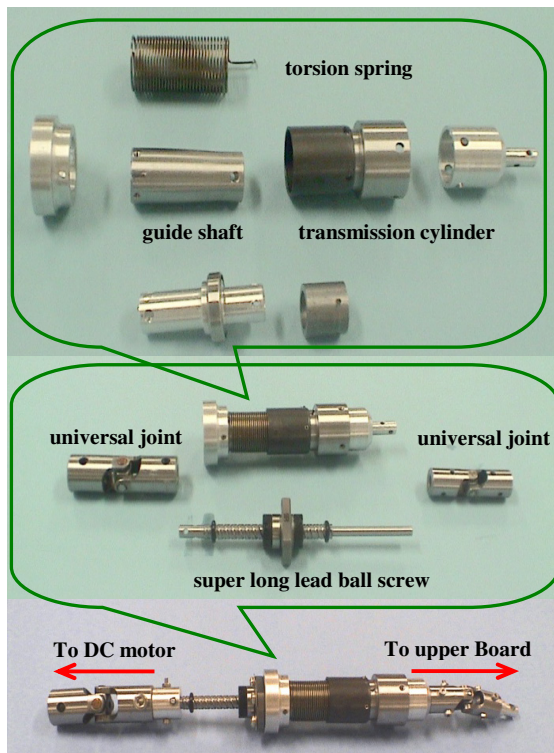
(*r*-ANLES) that is recently developed are briefly introduced. Both ANLESeS are used as voluntary muscles to control an ADJ like a musculo-skeletal system. Next it derives dual equations to independently control joint angle and joint stiffness, in which antagonistic alignment of the ANLESeS similar to a musculo-skeletal system is premised. The fourth section shows an application of both types of ANLESeS into a three DOF artificial joint arranged to use as a wrist joint for an anthropomorphic robot. It has three axes, each of which is controlled by a pair of ANLESeS (four *l*-ANLESeS for dorsi/plantar flexion and radial/ulnar flexion and two *r*-ANLESeS for pronation/supination) to individually adjust the stiffness of three rotary axes. The fourth section shows the experimental results of the joint stiffness and the joint angle control. It clearly elucidates that the joint stiffness and the joint angle can be separately controlled. The last section is devoted to some conclusive remarks.

## 2 Actuator with Non-linear Elasticity System

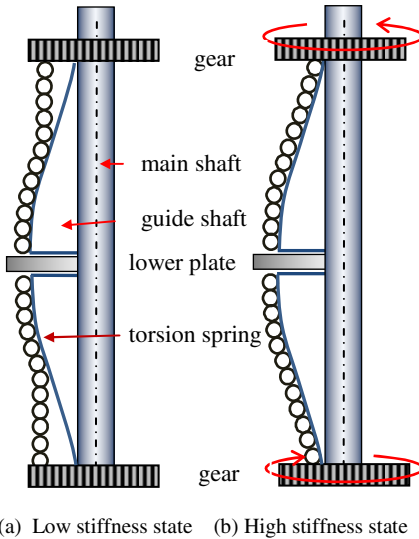
ANLES has been developed for controlling the ADJ in our studies [26][27][28]. It works like a voluntary muscle in a musculo-skeletal system of human articulation. Therefore a pair of ANLES is used to control one rotary axis. There are two types of mechanical configuration in the ANLES; the *linear-type ANLES (l-ANLES)* and the recently developed *rotary-type ANLES (r-ANLES)* as individually explained below.

### 2.1 Structure and Design of the *l*-ANLES

Fig.1 shows the structural parts and their assembled appearance of *l*-ANLES. It consists of a super long-lead ball screw, guide-shaft, torsion-spring and transmission cylinder (pulley). The torque generated by a DC-motor rotates the ball screw rod that brings about the rotation or translation of the guide shaft that embeds the ball screw nut. The rotation of the guide-shaft induces the twisting of the torsion-spring. The diameter of the guide shaft smoothly thins down along the rotation axis so that the torsion-spring twists around the guide shaft from the edge part of



**Fig. 1** Parts and the assembled appearance of *l*-ANLES



(a) Low stiffness state (b) High stiffness state

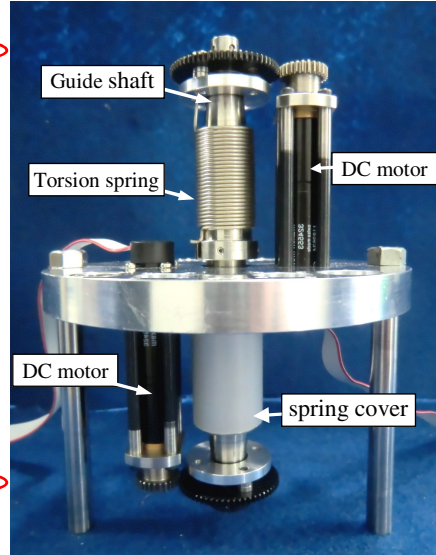


Fig. 3 Appearance of the r-ANLES developed for the wrist joint

Fig. 2 Outline of the stiffness change of the rod axis

the wide diameter, which yields non-linear elasticity as described below. This mechanism needs a transmitter to transform the rotation to the translation, and also vice versa with minimum transmission loss. We therefore employ the super long lead ball screw ( $\phi 6$  diameter rod with 6 mm lead) as shown in Fig. 1.

### 2.2 Structure and Design of the r-ANLES

A pair of guide-shaft and torsion spring is allocated counteractively along the main shaft as shown in Fig.2. The torsion springs are coiled by the individual DC motors via the gear and wrap around the guide shaft (Fig.3). Therefore this type of ANLES is identical to the *l*-ANLES but lacking the rotate/translate conversion.

### 2.3 Design of Non-linear Elasticity

The non-linear elasticity of *l*-ANLES and *r*-ANLES can be rigorously designed by designing the shape of the guide shaft. Now let us assume the torsion spring is wrapped on the guide shaft by the axial position  $x$  from the left edge as shown in Fig.4. In this state coiling the spring furthermore by an infinitesimal torsion angle  $\Delta\phi$  requires additional torque,

$$\Delta T_g(x) = (EI/l_r(x))\Delta\phi. \tag{1}$$

where  $l_r(x)$  is the expansion length of the spring wire (the expansion of the axial portion of  $L - x + \Delta L$  in Fig.4) that actually works as a spring at location  $x$ .

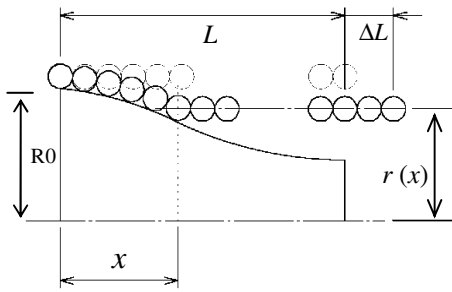


Fig. 4 Model of the guide-shaft

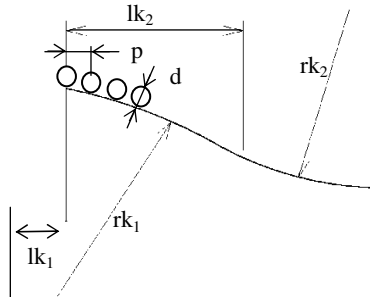


Fig. 5 Parameters for designing the guide shaft

$E$  is the modulus of longitudinal elasticity and  $I$  is the second moment of area of the torsion spring wire. (1) leads the spring coefficient as a function of  $x$ ,

$$K(x) = \frac{\Delta T_g(x)}{\Delta \phi} = \frac{EI}{l_r(x)} \quad (2)$$

We can obtain the relation between the torsion angle  $\phi(x)$  and the torque  $T_g(x)$ . Hence the  $T_g(x)$  and

also  $K(x)$  may be denoted by  $T_g(\phi)$  and  $K(\phi)$  respectively in lieu of using the intermediate parameter  $x$ . Now we have a free-hand to obtain the function  $T_g(\phi)$  through designing  $r(x)$ ; the radius function along the axis. We propose some configuration parameters for designing the guide-shaft as shown in Fig.4 in which  $r(x)$  consists of two curvatures having radius  $rk_1$  and  $rk_2$  that are smoothly connected at the location  $lk_2$ .

In Table 1 the design parameters of the guide shaft for  $l$ -ANLES are listed and its outcome of the non-linear elasticity of the  $l$ -ANLES is illustrated in Fig.6 accompanied by the measured data of torque-torsion angle relation. The error bar indicates the variation

Table 1 Designed parameters of the guide shaft of the  $l$ -ANLES shown in Fig.1

$lk_1$	Position of the center of the first curvature along axis	-5 mm
$lk_2$	Changing position of curvatures	25.2 mm
$rk_1$	Radius of the first curvature	450 mm
$rk_2$	Radius of the second curvature	400 mm
$d$	Diameter of the spring wire	0.8 mm
$p$	Pitch of the spring	1.0 mm
$n$	Winding number of the spring	28
$R_0$	Onset radius of the spring	8.0 mm

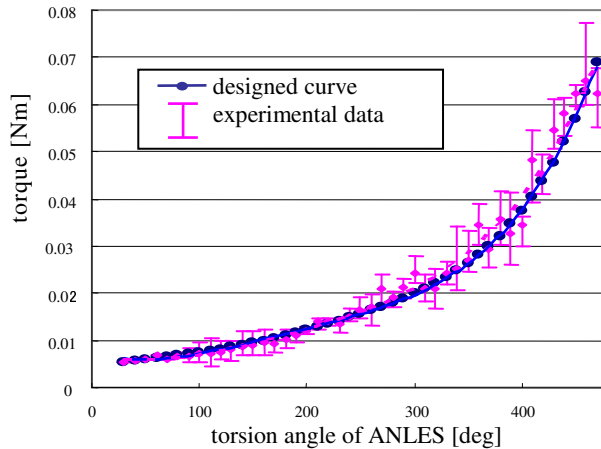


Fig. 6 Non-linear elasticity of ANLES shown in Fig.1



of the data of 10 trials. The non-linear elasticity shown in Fig.6 is arranged according to the expected stiffness variation of the wrist joint described below.

### 3 Basic Formula for Controlling Joint Stiffness and Joint Angle

In this section we derive dual equations to calculate torsion angles of ANLESEs that determine the joint angles and the joint stiffness of ADJ.

#### 3.1 Kinetics of a Multiple DOF Joint System Driven by Multiple Tendons

Let us consider the joint having three rotation axes (roll, yaw, pitch) that are driven by  $m$  tendons as shown in Fig.7. Each tendon is stretched by the individual ANLES.

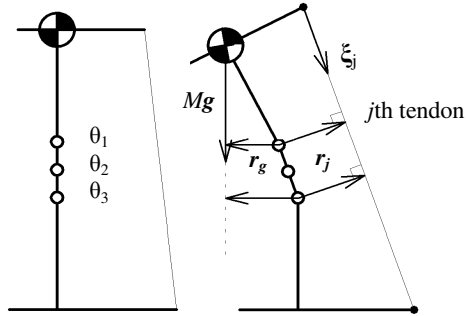


Fig. 7 Model of the three D.O.F. joints driven by  $m$  tendons

The tension force vector for  $j$ th tendon is denoted by  $\xi_j$ , of which modulus is assumed to be a non-linear function with respect to torsion angle of ANLES  $\phi_j$ ,

$$|\xi_j| = |\xi_j|(\phi_j) \tag{3}$$

It is also assumed to be a centered mass located out of the joint. Denoting  $r_{jk}$  for the moment arm vector from the  $k$  th joint axis to the  $j$  th tendon and  $r_{gk}$  for the  $k$  th joint axis to the gravitation force vector, we can derive a statically equilibrium state equation in terms of torque,

$$\left\{ \sum_{i=1}^3 \left( \sum_{j=1}^m (r_{ji} \times \xi_j) + r_{gi} \times Mg \right) \cdot e_k \mid k=1,2,3 \right\} = 0 \tag{4}$$

where,  $e_k, k=1,2,3$  represents the unit vector about the  $k$ th joint axis. Let us consider that infinitesimal torque  $\Delta\tau = \{\Delta\tau_p \mid p=1,2,3\}$  is loaded at the joint and gives rise to infinitesimal rotation of joints  $\Delta\theta_p (p=1,2,3)$ . The relation between them can be derived by differentiating (4) with respect to the joint angles,

$$\Delta\tau = \left[ \sum_{p=1}^3 \left( \sum_{i=1}^3 \left( \sum_{j=1}^m (r_{ji} \times \frac{\partial \xi_j}{\partial \theta_p} + \frac{\partial r_{ji}}{\partial \theta_p} \times \xi_j) + \frac{\partial r_{gi}}{\partial \theta_p} \times Mg \right) \right) \Delta\theta_p \right] \cdot e_k, \tag{5}$$

$(k=1,2,3)$

$\partial \xi_j / \partial \theta_p$  is rewritten by using the length of  $j$ th tendon  $l_j$ ,

$$\frac{\partial \xi_j}{\partial \theta_p} = \frac{\partial \xi_j}{\partial l_j} \frac{\partial l_j}{\partial \theta_p} = \frac{\partial \xi_j}{\partial l_j} \gamma_{jp} \tag{6}$$

where,  $\gamma_{jp} \equiv \partial l_j / \partial \theta_p$  is a moment radius of the  $j$ th tendon with respect to the  $p$ th rotary axis.  $\partial \xi_j / \partial l_j$  is also be rewritten as follows,

$$\frac{\partial \xi_j}{\partial l_j} = \frac{\partial \xi_j}{\partial \phi_j} \frac{\partial \phi_j}{\partial l_j} = \frac{\partial \xi_j}{\partial \phi_j} \frac{1}{\delta_j} \tag{7}$$

with  $\delta_j \equiv \partial l_j / \partial \phi_j$  being a change rate of the tendon length with respect to the torsion angle of the ANLES. The tension force vector  $\xi_j$  is expressed with its unit vector  $\eta_j$ ,

$$\xi_j = |\xi_j| \eta_j \tag{8}$$

Using the notations of (6),(7) and (8), (5) is rewritten as,

$$\Delta \tau = \left\{ \left[ \sum_{p=1}^3 \left( \sum_{i=1}^3 \left( \sum_{j=1}^m \left( \frac{\partial |\xi_j|}{\partial \phi_j} \frac{\gamma_{jp}}{\delta_j} r_{ji} \times \eta_j + \frac{\partial r_{ji}}{\partial \theta_p} \times |\xi_j| \eta_j + \frac{\partial r_{gi}}{\partial \theta_p} \times Mg \right) \right) \Delta \theta_p \right) \cdot e_k \right] \Big|_{k=1,2,3} \right\} = S \Delta \theta \tag{9}$$

where,  $\Delta \theta = [\Delta \theta_1 \ \Delta \theta_2 \ \Delta \theta_3]^T$  and  $S = \{s_{ij} \mid (i, j = 1, 2, 3)\} \in \mathbb{R}^{3 \times 3}$  are identified as a stiffness matrix with respect to the rotation. It is practically difficult to manipulate all elements of the stiffness matrix (6 independent elements). For doing so, it requires more than 9 tendons ,see [11]. The authors consider it is not practical and not necessary to control all of them. Some experiments in the field of the motion physiology [5] show the fact that human can regulate the stiffness-ellipsoid at the end-point of the arm merely in its volume rather than in its shape. So let us consider to manipulate only the diagonal three elements of the stiffness matrix. For the general consideration dealt with off-diagonal elements of the stiffness matrix, please refer the paper [26].

The stiffness vector consisting of the diagonal elements of the stiffness matrix is calculated by the following equation,

$$s \equiv \begin{bmatrix} s_{11} \\ s_{22} \\ s_{33} \end{bmatrix} = M \lambda_0 + \lambda_1 \begin{bmatrix} \partial |\xi_1| / \partial \phi_1 \\ \vdots \\ \partial |\xi_m| / \partial \phi_m \end{bmatrix} + \lambda_2 \begin{bmatrix} |\xi_1| \\ \vdots \\ |\xi_m| \end{bmatrix} \tag{10}$$

with

$$\lambda_1 = \begin{bmatrix} \sum_{i=1}^3 \frac{\gamma_{11}}{\delta_1} (r_{1i} \times \eta_1) \cdot e_1 & \cdots & \sum_{i=1}^3 \frac{\gamma_{m1}}{\delta_m} (r_{mi} \times \eta_m) \cdot e_1 \\ \sum_{i=1}^3 \frac{\gamma_{12}}{\delta_1} (r_{1i} \times \eta_1) \cdot e_2 & \cdots & \sum_{i=1}^3 \frac{\gamma_{m2}}{\delta_m} (r_{mi} \times \eta_m) \cdot e_2 \\ \sum_{i=1}^3 \frac{\gamma_{13}}{\delta_1} (r_{1i} \times \eta_1) \cdot e_3 & \cdots & \sum_{i=1}^3 \frac{\gamma_{m3}}{\delta_m} (r_{mi} \times \eta_m) \cdot e_3 \end{bmatrix},$$

$$\boldsymbol{\lambda}_0 = \begin{bmatrix} \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{gi}}{\partial \theta_1} \times \mathbf{g} \right) \cdot \mathbf{e}_1 \\ \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{gi}}{\partial \theta_2} \times \mathbf{g} \right) \cdot \mathbf{e}_2 \\ \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{gi}}{\partial \theta_3} \times \mathbf{g} \right) \cdot \mathbf{e}_3 \end{bmatrix}, \quad \boldsymbol{\lambda}_2 = \begin{bmatrix} \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{li}}{\partial \theta_1} \times \boldsymbol{\eta}_i \right) \cdot \mathbf{e}_1 & \cdots & \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{mi}}{\partial \theta_1} \times \boldsymbol{\eta}_m \right) \cdot \mathbf{e}_1 \\ \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{li}}{\partial \theta_2} \times \boldsymbol{\eta}_i \right) \cdot \mathbf{e}_2 & \cdots & \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{mi}}{\partial \theta_2} \times \boldsymbol{\eta}_m \right) \cdot \mathbf{e}_2 \\ \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{li}}{\partial \theta_3} \times \boldsymbol{\eta}_i \right) \cdot \mathbf{e}_3 & \cdots & \sum_{i=1}^3 \left( \frac{\partial \mathbf{r}_{mi}}{\partial \theta_3} \times \boldsymbol{\eta}_m \right) \cdot \mathbf{e}_3 \end{bmatrix}$$

Let us consider the infinitesimal rotation of ANLES that give rise to the infinitesimal variation of the stiffness vector under holding a constant joint angle. We derive the following equation from (10),

$$\left\{ \sum_{i=1}^3 \left( \sum_{j=1}^m \left( \mathbf{r}_{ji} \times \Delta \boldsymbol{\xi}_j + \Delta \mathbf{r}_{ji} \times \boldsymbol{\xi}_j \right) + \Delta \mathbf{r}_{gi} \times M \mathbf{g} \right) \cdot \mathbf{e}_k \mid k = 1, 2, 3 \right\} = \mathbf{0} \tag{11}$$

Eq. (11) suggests that the second-order derivatives of tensions with respect to the torsion angle of ANLES play an important role for regulating the stiffness, which implies to require the non-linear elasticity (this fact is also suggested by Yi, et.al.[29][30]). Next, let us consider the infinitesimal variation of the tension vector  $\boldsymbol{\xi}_i$  due to the variation of  $\phi_i$  that give rise to the infinitesimal rotation of the joints. We have the following equation from (4),

$$\Delta \mathbf{s} = \left[ \boldsymbol{\lambda}_1 \text{diag} \left\{ \frac{\partial^2 |\boldsymbol{\xi}|_1}{\partial \phi_1^2}, \dots, \frac{\partial^2 |\boldsymbol{\xi}|_m}{\partial \phi_m^2} \right\} + \boldsymbol{\lambda}_2 \text{diag} \left\{ \frac{\partial |\boldsymbol{\xi}|_1}{\partial \phi_1}, \dots, \frac{\partial |\boldsymbol{\xi}|_m}{\partial \phi_m} \right\} \right] \Delta \boldsymbol{\phi} \equiv \boldsymbol{\Lambda} \Delta \boldsymbol{\phi}$$

This can be rewritten by using (6)(7) and (8),

$$\boldsymbol{\Gamma} \Delta \boldsymbol{\theta} = -\boldsymbol{\Sigma} \Delta \boldsymbol{\phi} \tag{12}$$

where,  $\Delta \boldsymbol{\phi} = [\Delta \phi_1 \cdots \Delta \phi_m]^T$ ,

$$\boldsymbol{\Gamma} = \left\{ \sum_{i=1}^3 \left( \sum_{j=1}^m \left( \mathbf{r}_{ji} \times \frac{\gamma_{jl}}{\delta_j} \frac{\partial \boldsymbol{\xi}_j}{\partial \phi_j} \boldsymbol{\eta}_j + \frac{\partial \mathbf{r}_{ji}}{\partial \theta_i} \times \boldsymbol{\xi}_j \right) + \frac{\partial \mathbf{r}_{gi}}{\partial \theta_i} \times M \mathbf{g} \right) \cdot \mathbf{e}_k \mid (k, l = 1, 2, 3) \right\} \in \Re^{3 \times 3},$$

$$\boldsymbol{\Sigma} = \left\{ \sum_{i=1}^3 \left( \mathbf{r}_{ji} \times \frac{\partial \boldsymbol{\xi}_j}{\partial \phi_j} \boldsymbol{\eta}_j \right) \cdot \mathbf{e}_k \mid (k = 1, 2, 3, j = 1, \dots, m) \right\} \in \Re^{3 \times m}$$

The general solution of (10) with respect to  $\Delta \boldsymbol{\phi}$  is

$$\Delta \boldsymbol{\phi} = -\boldsymbol{\Sigma}^\dagger \boldsymbol{\Gamma} \Delta \boldsymbol{\theta} - \mathbf{P}^\perp(\boldsymbol{\Sigma}) \Delta \boldsymbol{\zeta} \tag{13}$$

where,  $\boldsymbol{\Sigma}^\dagger \in \Re^{m \times 3}$  represents the generalized inverse of  $\boldsymbol{\Sigma}$  and  $\mathbf{P}^\perp(\boldsymbol{\Sigma}) \in \Re^{3 \times 3}$  represents the null projection operator that projects arbitrarily specified vector  $\Delta \boldsymbol{\zeta} \in \Re^m$  on the complementary space of  $\boldsymbol{\Sigma}$ .

### 3.2 *The Formulation for the ANLES to Vary the Joint Stiffness under Constant Joint Angle*

Let us first consider to regulate stiffness of the joint without rotating the joint. This must be carried out by varying the torsion angle vector  $\Delta\phi$  while holding  $\Delta\theta = \mathbf{0}$  in (13),

$$\Delta\phi = -\mathbf{P}^\perp(\Sigma)\Delta\zeta \quad (14)$$

Substitution of (14) into (11) provides,

$$\Delta s = -\mathbf{A}\mathbf{P}^\perp(\Sigma)\Delta\zeta \quad (15)$$

Solving (15) with respect to  $\Delta\zeta$  we have,

$$\Delta\zeta = -(\mathbf{A}\mathbf{P}^\perp(\Sigma))^\dagger \Delta s \quad (16)$$

Substitution of it into (14) gives us,

$$\Delta\phi_s = \mathbf{P}^\perp(\Sigma)(\mathbf{A}\mathbf{P}^\perp(\Sigma))^\dagger \Delta s \quad (17)$$

where,  $\Delta\phi_s$  is the infinitesimal variation of the ANLES torsion angle vector to provide the stiffness variation  $\Delta s$  with no giving rise to the rotation of the joint angles.

### 3.3 *The Formulation for the ANLES to Vary the Joint Angles under Constant Joint Stiffness*

Next it is capable to derive the formula for regulating angles of joints without bringing about the variation of the stiffness of the end-point. The general solution of (11) with respect to  $\Delta\phi$  will be,

$$\Delta\phi = \mathbf{A}^\dagger \Delta s + \mathbf{P}^\perp(\mathbf{A})\Delta\psi \quad (18)$$

where  $\Delta\psi \in \mathfrak{R}^m$  is a vector that is arbitrarily assignable. Since we aim to keep the stiffness being constant,  $\Delta s$  in (18) should be zero,

$$\Delta\phi = \mathbf{P}^\perp(\mathbf{A})\Delta\psi \quad (19)$$

Substituting it into (12), we have

$$\Gamma\Delta\theta = -\Sigma\mathbf{P}^\perp(\mathbf{A})\Delta\psi. \quad (20)$$

Solving it with respect to  $\Delta\psi$  and substituting into (19), we have,

$$\Delta\phi_a = -\mathbf{P}^\perp(\mathbf{A})(\Sigma\mathbf{P}^\perp(\mathbf{A}))^\dagger \Gamma\Delta\theta. \quad (21)$$

where  $\Delta\phi_a$  is the infinitesimal variation of the ANLES torsion angle vector to give rise to the joint rotation with no interference with the joint stiffness.  $\Delta\phi_s$  in (17) is simplified by making use of the nilpotent property of a projection operator,

$$\begin{aligned}
 \Delta \phi_s &= P^\perp(\Sigma) (AP^\perp(\Sigma))^\dagger \Delta s \\
 &= P^\perp(\Sigma) (AP^\perp(\Sigma))^T \left( AP^\perp(\Sigma) (AP^\perp(\Sigma))^T \right)^{-1} \Delta s \\
 &= P^\perp(\Sigma) A^T \left( AP^\perp(\Sigma) A^T \right)^{-1} \Delta s.
 \end{aligned}
 \tag{22}$$

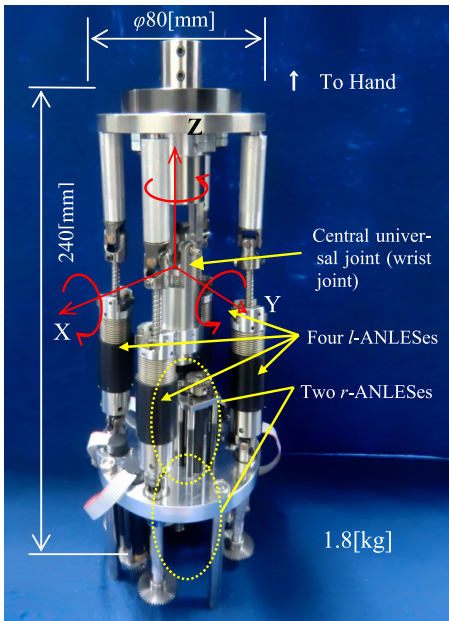
Similar procedure can be applied to (21) as follows,

$$\begin{aligned}
 \Delta \phi_a &= -P^\perp(A) (\Sigma P^\perp(A))^\dagger \Gamma \Delta \theta \\
 &= -P^\perp(A) \Sigma^T \left( \Sigma P^\perp(A) \Sigma^T \right)^{-1} \Gamma \Delta \theta.
 \end{aligned}
 \tag{23}$$

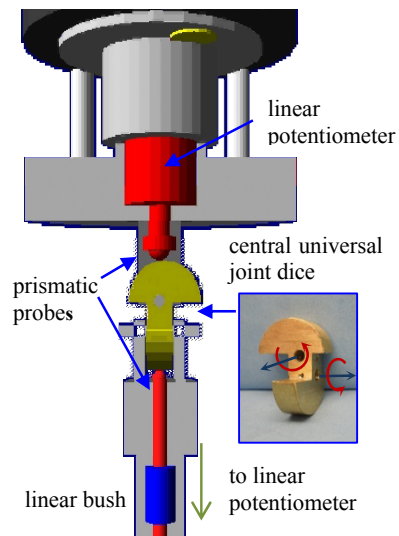
(22) is an equation to calculate the torsion angle of ANLESEs for manipulating the joint stiffness without giving rise to a joint angle variation. (23) is a dual equation of (22) to handle joint angle without giving rise to a stiffness variation. However please notice that the independent control of the stiffness and joint angles by using (22) and (23) may not be perfectly accomplished. It depends on the number of tendons that are individually controlled by ANLES and how they cover three dimensional stiffness space determined by their alignment. Please see [24] for the similar case studies.

### 4 Three DOF Wrist Joint

The previous model was under-actuated in the sense that only 5 motors (4 for *l*-ANLESEs and 1 for rotating the main shaft) are used to control 6 variables (3 joint



**Fig. 8** Three DOF wrist joint controlled by four *l*-ANLESEs and two *r*-ANLESEs



**Fig. 9** The measuring device of two angles: dorsi/plantar flexion and radial/ulnar flexion

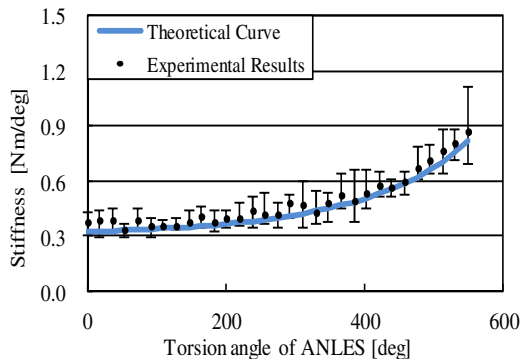
angles and 3 joint stiffness) (see [28]). We approached to develop a fully actuated second model (Fig.8) by introducing 2 *r*-ANLEs for controlling the pronation/supination (Z-axis in Fig.8) in addition to the 4 *l*-ANLEs that control X and Y axes (dorsi/plantar flexion and radial/ulnar flexion), which is the same as the previous model. The assembled view of the two *r*-ANLEs can be seen in Fig.3. Four motors for *l*-ANLEs are allocated under the lower plate. The overall weight is about 1.8 [kg]. The rotation angles of the ext./flex. and radial flex./ulnar flex. are measured by the originally developed central universal joint as shown in Fig.9. The dice of the central universal joint transforms the two axes of rotation into the linear displacements that are measured by linear potentiometers or differential transformers.

## 5 Joint Angle and Stiffness Control

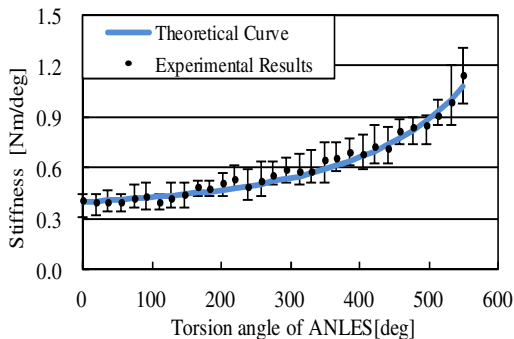
### 5.1 Joint Stiffness Control under Constant Joint Angle

Fig.10 shows the experimental result of measuring the joint stiffness. The experiment was carried out as follows.

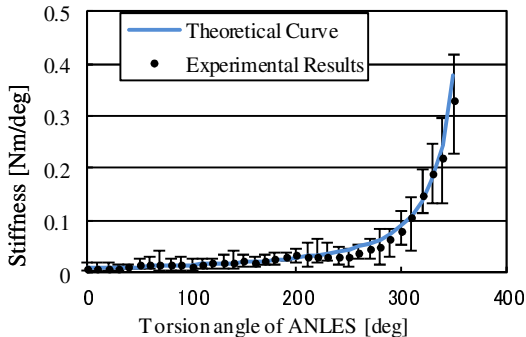
1. All of the ANLEs (*l*-ANLEs for X and Y-axes, *r*-ANLEs for Z-axis in Fig.7) are twisted the same amount of torsion angle according to the Eq. (22).
2. A torque is loaded about an axis by pulling the endpoint with the wire to yield five degree rotation of the joint about each axis. The stiffness is measured according to the loaded torque.
3. 10 trials of the same procedure are carried out to see the measurement variation that is shown as the error bars in Fig.10.



(a) Experimental results of stiffness about X axis

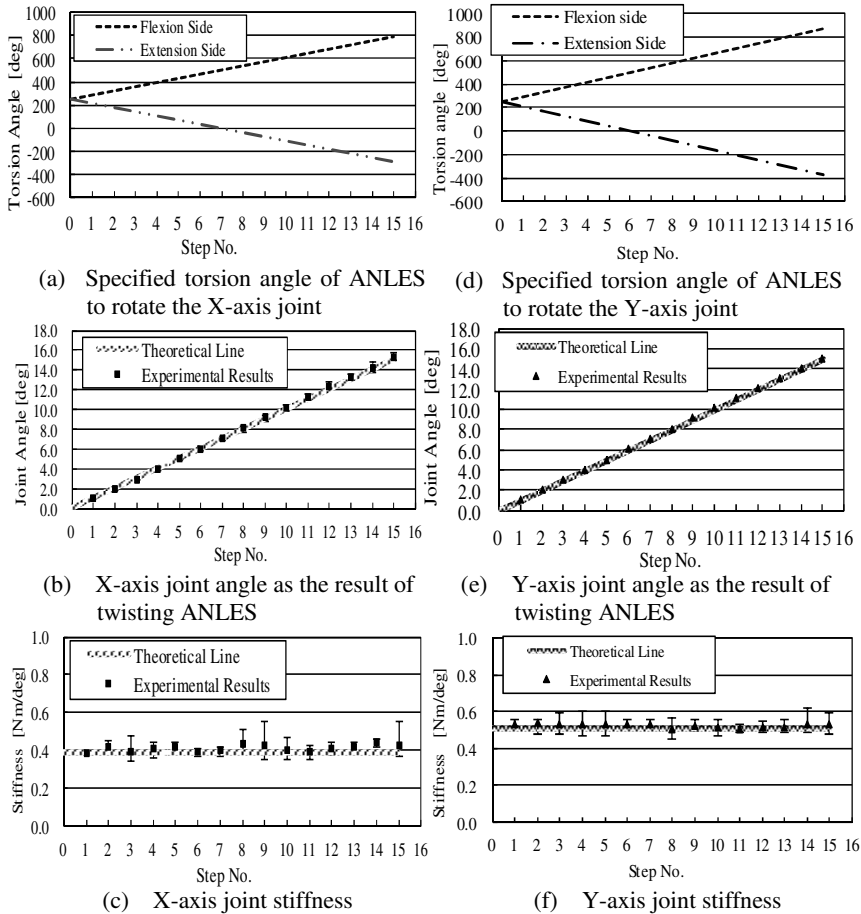


(b) Experimental results of stiffness about Y axis



(c) Experimental results of stiffness about Z axis

**Fig. 10** Theoretical curve and the experimental results of stiffness about 3 DOF wrist joint



**Fig. 11** Experimental results of wrist joint angle control under constant joint stiffness

The results shown in Fig.10 clearly show the validity of the theory and the confidence of the stiffness regulation, although it shows some variation of the data, especially under the high stiffness state, which will be due to the friction torque that arises between the spring and the guide-shaft.

### 5.2 Joint Angle Control under Constant Joint Stiffness

Fig.11 shows the results of joint angle control while sustaining a constant joint stiffness ; 0.40 Nm/deg about X-axis and ; 0.52 Nm/deg about Y-axis. The experimental procedure is as follows.

1. The *l*-ANLESEs are twisted by the theoretically obtained torsion angles according to the Eq.(23) to make the joint rotate about the specified axis under the specified joint stiffness. Two *l*-ANLESEs of the flexion side are twisted by the same torsion angle and another two *l*-ANLESEs of the extension side are twisted by the other same angle as shown in Fig.11 (a)(d).
2. It is verified that the joint rotation angle yielded by the twisting of the *l*-ANLESEs is almost coincident with the theoretically predicted one (Fig.11 (b)(e)). Fig.11 (b)(e) also depict error bars but they are very small.
3. The same procedures to measure the stiffness described in the former section are carried out. (Fig.11 (c)(f))
4. 10 trials of the same procedure are carried out to see the data variation that is shown as error bars in Fig.11(b)(e)(c)(f).

The results of Fig.11 clearly demonstrate that the joint stiffness and the joint angle can be separately controlled. . However Fig.10 and 11 also show some no negligible error bars of the stiffness data. It is mainly due to the rough clearances existing in some universal joints and some ball bearings, and also somewhat due to the friction existing between the torsion springs and the transmission-cylinders (see Fig.1). We are now refining these mechanical defects.

## 6 Conclusions

In this paper the wrist joint that has a musculo-skeletal structure is introduced, in which ANLES has a role of voluntary muscles. Technical advance of the second machine from the first machine resides in its mechanism that provides fully actuated 3 DOF joint in terms of controlling 3 angles and 3 joint stiffness by employing 6 ANLESEs, in which newly proposed architecture of ANLES, the rotary-type ANLES (*r*-ANLES), is introduced. Some remarkable advantages will be pointed out in the proposed mechanism:

1. Stiffness variation of the joint can be easily and rigorously designed.
2. It can be applied for controlling a joint that has multiple DOF like a wrist joint because the *l*-ANLES is able to be used as a linear actuator like muscles.
3. The *r*-ANLES will expand the applicable range of ANLES in anthropomorphic robots that imitate human musculo-skeletal architecture.

**Acknowledgements.** The authors thank the Japan Society for the Promotion of Science, via Grants-in-Aid for Scientific Research: No. 18560258 ‘Stiffness control of an antagonistically driven joint using the actuator with non-linear elasticity’, for their support in the pursuance of this work.

## References

1. Shadmeh, R., Arbib, M.A.: A mathematical analysis of the force-stiffness characteristics of muscles in control of a single joint system. *Biol. Cybern.* 66, 463–477 (1992)
2. Matthews, P.B.C.: The dependence of tension upon extension in the stretch reflex of the soleus muscle of the decerebrated cat. *J. of Physiol.* 147, 521–546 (1959)



3. Fel'dman, A.G.: Functional tuning of the neurons system with control of movement or maintenance of a steady posture. *Biofizika* 11(3), 498–508 (1966)
4. Hoffer, J.A., Andreassen, S.: Regulation of soleus muscle stiffness in premammillary cats. *J. of Neurophysiol.* 45(2), 267–285 (1981)
5. Dolan, J.M., Friedman, M.B., Nagurka, M.L.: Dynamic and Loaded Impedance of Human Arm Posture. *IEEE Trans. on System Man and Cybern.* 23(3), 698–709 (1993)
6. Mussa-Ivaldi, F.A., Hogan, N., Bizzi, E.: Neural, Mechanical and Geometric Factors Subservicing Arm Posture in Humans. *The Journal of Neuroscience* 5(10), 2732–2743 (1985)
7. Jacobsen, S.C., Wood, J.E., Knutti, D.F., Biggers, K.B.: The UTAH/M.I.T. Dextrous Hand: Work in Progress. *The Int. Journal of Robotics Res.* 3(4), 21–51 (1984)
8. Jacobsen, S.C., Ko, H., Inversen, E.K., Davis, C.C.: Antagonistic Control of a Tendon Driven Manipulator. In: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1334–1339 (1989)
9. Laurin-Kovitz, K.F., Colgate, J.E., Carnes, S.D.R.: Design of Components for Programmable Passive Impedance. In: *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, pp. 1476–1481 (1991)
10. Yi, B.J., Freeman, R.A.: Geometric Characteristics of Antagonistic Stiffness in Redundantly Actuated Mechanisms. In: *Proc. of IEEE Int. Conf. on Robotics & Automation*, pp. 654–661 (1993)
11. Kobayasi, H., Hyoudou, K., Ogane, D.: On Tendon-Driven Robotics Mechanisms with Redundant Tendons. *The Int. J. of Robotics Res.* 17(15), 561–571 (1998)
12. Lee, Y.T., Choi, H.R., Chung, W.K., Youm, Y.: Stiffness Control of a Coupled Tendon-Driven Robot Hand. *IEEE Control Systems Magazine*, 10–19 (1994)
13. Yi, B.J., Freeman, R.A.: Synthesis of Actively Adjustable Springs by Antagonistic Redundant Actuation. *Trans. on ASME, Journal of Dynamic Systems, Measurement and Control* 461, 454–461 (1992)
14. Chen, S.F., Kao, I.: Conservative Congruence Trans-formation for Joint and Cartesian Stiffness Matrices of Robotic Hands and Fingers. *The Int. J. of Robotics Res.* 19(9), 835–847 (2000)
15. Tondu, B., Lopez, P.M.: Modelling and Control of McKibben Artificial Muscle Robot Actuators. *IEEE Control Systems Magazine*, 15–28 (2000)
16. Tondu, B., Ippolito, S., Guiochet, J., Daidie, A.: A Seven-degrees-of freedom Robot-arm Driven by Pneumatic Artificial Muscles for Humanoid Robots. *The International J. of Robotics Res.* 24(4), 257–274 (2005)
17. Migliore, S.A., Brown, E.A., DeWeerth, S.P.: Biologically Inspired Joint Stiffness Control. In: *Proc. of the 2005 IEEE Int.l Conf. on Robotics and Automation, Barcelona, Spain*, pp. 4508–4517 (April 2005)
18. Tonietii, G., Schiavi, R., Bicchi, A.: Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction. In: *Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain*, pp. 526–531 (April 2005)
19. Schiavi, R., Grioli, G., Sen, S., Bicchi, A.: VSA-II: A Novel Prototype of Variable Stiffness Actuator for Safe and Performing Robots Interacting with Humans. In: *Proc. of the 2008 IEEE Int. Conf. on Robotics and Automation, Pasadena, CA, USA*, pp. 2171–2176 (May 2008)
20. Haddadin, S., Albu-Schaffer, A., Hirzinger, G.: The Role of the Robot Mass and Velocity in Physical Human-Robot Interaction-Part 2: Constrained Blunt Impacts. In: *IEEE Int. Conf. on Robotics and Automation (ICRA 2008), Pasadena, USA*, p. 1339 (2008)

21. an Ham, R., Sugar, T.G., Vanderborght, B., Hollander, K.W., Lefeber, D.: Compliant Actuator Design. In: *IEEE Robotics & Automation Magazine*, pp. 81–94 (September 2009)
22. Ham, R.V., Vanderborght, B., Damme, M.V., Verrelst, B., Lefeber, D.: MACCEPA, the Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator: Design and Implementation in a Biped Robot. *Robotics and Autonomous Systems* 55, 761–768 (2007)
23. Wolf, S., Hirzinger, G.: A New Variable Stiffness Design: Matching Requirements of the Next Robot Generation. In: *Proc. of 2008 IEEE Int. Conf. on Robotics and Autom.*, Pasadena, CA, USA, pp. 1741–1746 (2008)
24. Byeong-Sang, K., Song, J.: Hybrid Dual Actuator Unit: A Design of a Variable Stiffness Actuator Based on an Adjustable Moment Arm Mechanism. In: *Proc. of 2010 IEEE Int. Conf. on Robotics and Autom.* Anchorage, Alaska, USA, pp. 1655–1660 (2010)
25. Morita, T., Sugano, S.: Design and Development of a new Robot Joint using a Mechanical Impedance Adjuster. In: *Proc. of IEEE Int. Conf. on Robotics and Autom.* pp. 2469–2475 (1995)
26. Koganezawa, K., Shimizu, Y., Inomata, H., Nakazawa, T.: Actuator with Non-linear Elastic System (ANLES) for Controlling Joint Stiffness of Antagonistic Driving. In: *Proc. of IEEE Int. Conf. on robotics and biomimetics, ROBIO* (August 2004)
27. Koganezawa, K.: Mechanical Stiffness Control for Antagonistically Driven Joints. In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots & Systems (IROS)*, Edmonton, Canada (2005)
28. Koganezawa, K., Yamashita, H.: Stiffness Control of Multi-DOF Joint. In: *Proc. of 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, St. Louis, Missouri (2009)
29. Yi, B.J., Freeman, R.A.: Synthesis of Actively Adjustable Springs by Antagonistic Redundant Actuation. *Trans. on ASME, Journal of Dynamic Systems, Measurement and Control* 114, 454–461 (1992)
30. Yi, B.J., Freeman, R.A.: Geometric Characteristics of Antagonistic Stiffness in Redundantly Actuated Mechanisms. In: *Proc. of IEEE Int. Conf. on Robotics & Automation*, pp. 654–661 (1993)

# Lyapunov Based Sampling for Adaptive Tracking Control in Robot Manipulators: An Experimental Comparison

Pavankumar Tallapragada and Nikhil Chopra\*

**Abstract.** In digital computer based controllers, efficient sampling mechanisms for sensors as well as controllers is of great importance. In this paper, we are interested in designing controllers that result in low average frequency of control updates while simultaneously ensuring stability of the robotic system. We experimentally investigate a non-periodic state-triggered control sampling scheme (designed through Lyapunov like analysis) for adaptive tracking controllers in robot manipulators. We implement this scheme on two well known continuous-time adaptive controllers for tracking in robot manipulators and compare their performance heuristically based on the results of experiments performed on a two link planar manipulator.

## 1 Introduction

**Motivation.** The controllers in robotic systems are predominantly digital computer based. These systems inherently involve sampling, which creates a trade-off between sampling/communication cost and achieving the required task at hand, such as stabilization. These issues assume additional significance when the control signal is communicated over a network or when the controller is implemented on an embedded processor with low computational

---

Pavankumar Tallapragada

Department of Mechanical Engineering, University of Maryland, College Park,  
20742 MD, USA

e-mail: pavant@umd.edu

Nikhil Chopra

Department of Mechanical Engineering and The Institute for Systems Research,  
University of Maryland, College Park, 20742 MD, USA

e-mail: nchopra@umd.edu

\* This work was partially supported by the National Science Foundation under grant 0931661.

capabilities. In certain cases, the task of sensing (for example, visual feedback) may itself impose constraints on the update frequency of the control. Thus, there is a clear need to design efficient sampling mechanisms for controllers. In this paper we investigate a state based sampling technique for reducing the average frequency of control updates while ensuring stability of trajectory tracking in robot manipulators.

**Related Work.** Traditional computer based control systems in robotics rely on periodic sampling of the sensors and computation/execution of the control [1–5]. The reason for the popularity of this paradigm is a well developed theory and the ease of analysis of such systems. However, such control algorithms may need to employ high sampling rates. Recently, there has been a growing interest among the control community in what are known as event based control systems [6–9]. In these control systems, timing of control execution is not necessarily periodic and can be state dependent. By encoding the nature of the task (for example stabilization) into an event-triggering condition, it is possible to systematically design sampled-data controllers that make better use of computational and communication resources.

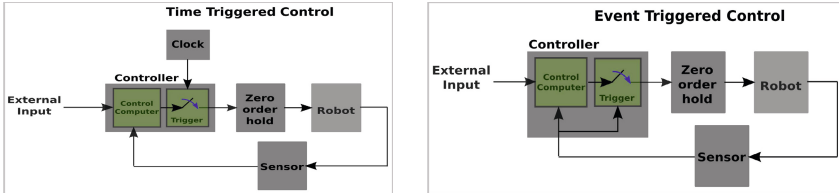
Predominantly, *event-triggered* controllers in the literature are essentially sampled data versions of continuous time controllers, with the sampling instants determined by state based triggering conditions. Additionally, many of these controllers may be called Lyapunov based controllers. This is due to the fact that the task of the controller, whether it is set point or trajectory tracking, can be cast as a stabilization problem and the event-triggering condition can be determined through a Lyapunov like analysis.

While Lyapunov based event-triggered controllers implicitly guarantee stability, they have a major drawback. These controllers rely critically on the knowledge of an accurate model of the system. For example, the results in [6,9] are general enough to hold for robotic manipulators when perfect knowledge of the system is available. However, building a model of high accuracy is a time consuming process and in many cases, it may not even be possible. Therefore, it is important to extend the design of implicitly verified event based controllers to cases where only a poor model of the system is available. This is specially important in the field of robotics, where adaptive and robust controllers are often used.

**Problem Statement.** Motivated by this practical necessity, we have developed [10] an event based adaptive control technique for trajectory tracking that can be used even when the robot system dynamics are unknown. In this paper, this state based Lyapunov sampling/triggering scheme is utilized to realize digital computer implementations of two well known adaptive tracking controllers for robot manipulators - [11] and [12], respectively, and their performance is compared through some experiments on a two-link planar manipulator.

### 1.1 Technical Approach – Lyapunov Based Sampling

The basic idea of time-triggered control and state-triggered control (event-triggered control) is illustrated in Figure 1. The difference between the two is in the manner of determining the sampling instants. While in the time-triggered case the sampling instants are determined by an external clock, in the event-triggered case they are determined implicitly by a state dependent event-triggering condition.



(a) Time-Triggered-Control: The sensor data is required by the controller at predetermined time instances

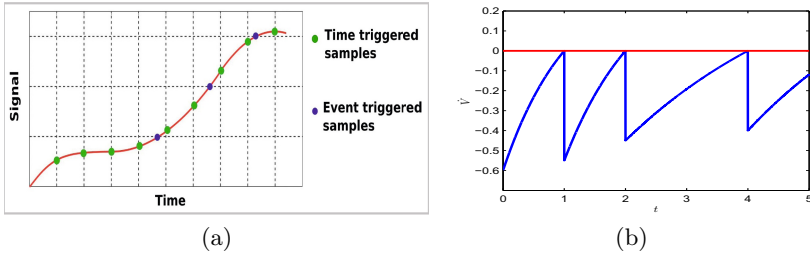
(b) Event-Triggered Control: The sensor data is required by the controller at time instants determined by a triggering function

**Fig. 1** Time-triggered and event-triggered approaches to sampled data control

Figure 2a illustrates a simple time-triggered (periodic sampling) and state-triggered (threshold-triggered) sampling. As can be seen, sampling based on threshold crossings is dependent on the signal, unlike the time-triggered sampling. Thus, the event-triggered approach may be said to sample the signal only when ‘necessary’. This idea of sampling only when necessary can be made more systematic by encoding the control objective in an event-triggerer.

The sampling mechanism utilized in this paper may be called a Lyapunov based sampler as the state based *event-triggering condition* that determines the sampling instants is designed by a Lyapunov like analysis. This approach implicitly guarantees stability of the tracking error. The basic idea behind the Lyapunov based sampling approach is shown in Figure 2b. Ideally, if the control signal is held constant and updated only as the derivative of a Lyapunov function approaches zero from below, the average frequency of sampling may be reduced. Moreover, because this approach ensures that the derivative of a Lyapunov function,  $\dot{V}$ , is strictly negative at all time, it implicitly guarantees stability.

There are two main caveats in this description of the design. The first is that continuously checking the zero-crossing of  $\dot{V}$  is computationally inefficient. In addition, when only inexact information about the robot parameters is available,  $\dot{V}$  cannot be computed precisely. However, it is possible to design an easy to check function (event-triggering condition) that always upper bounds  $\dot{V}$  (even if its precise value is unknown at any given time).



**Fig. 2** (a) Time and Threshold based sampling. (b) Lyapunov based sampling - the control signal is sampled/updated only when the derivative of a Lyapunov function reaches zero.

In the next section, we present a Lyapunov based event-triggered implementation of two well known adaptive controllers for trajectory tracking.

## 2 Event Based Adaptive Control

In this section we present a Lyapunov based event-triggered implementation of two well known adaptive controllers for trajectory tracking in robotic manipulators. In this paper, only the essential steps in the design process are described. Detailed procedure of the design may be found in our previous works. The work in [9] is useful for general nonlinear systems under the perfect knowledge of system. On the other hand, in [10] the design procedure for a specific adaptive tracking controller for robotic manipulators is described.

Now consider a standard n-degree of freedom rigid robot model of the form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u, \quad q \in \mathbb{R}^n, \quad u \in \mathbb{R}^n$$

where  $M : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ ,  $C : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  and  $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . The matrix  $C(q, \dot{q})$  is defined using the Christoffel symbols. Let  $x_d \triangleq [q_d; \dot{q}_d] \in \mathbb{R}^n \times \mathbb{R}^n$  be the state of the desired trajectory that the robot has to track. Here the notation  $[a_1; a_2]$  denotes the column vector formed by concatenating the vectors  $a_1$  and  $a_2$ . This notation is used in this paper to refer to various concatenated vectors. Let  $\tilde{q} \triangleq q - q_d$ , then the tracking error is defined as  $\tilde{x} \triangleq [\tilde{q}; \dot{\tilde{q}}]$ . Let  $u = \gamma(\xi) \in \mathbb{R}^m$  be a known continuous-time control law for trajectory tracking, where  $\xi$  is the data that the controller depends on.

Let us introduce the following notation to denote the sampled data versions of different signals in the system. The sampled data version of any signal  $X$  (which can be a scalar, a vector or a matrix) is denoted by  $X_s$ . In particular, the data sampled by the controller is denoted by  $\xi_s$ , and is defined as

$$\xi_s(t) = \xi(t_i), \quad \text{for all } t \in [t_i, t_{i+1}), \quad \text{for each } i$$

where  $t_i$  are the sampling instants. All the other sampled data signals are similarly defined. In time-triggered or periodic control systems,  $t_{i+1} - t_i = T_s$  for all  $i \in \{0, 1, 2, \dots\}$ , where  $T_s > 0$  is a constant sampling time. On the other hand, in an event-triggered controller the time instants  $t_i$  are defined implicitly by a triggering condition. The sampled nature of the signals can be alternatively viewed as time-continuous signals, albeit with an error in their measurement. Thus, we define the measurement error as

$$e \triangleq \xi_s - \xi = \xi(t_i) - \xi, \quad \text{for } t \in [t_i, t_{i+1}), \quad i \in \{0, 1, 2, \dots\}$$

Note that  $e$  is discontinuous at  $t = t_i$ , for each  $i$ , because  $e(t_i) = \xi(t_i) - \xi(t_i) = 0$  while  $e(t_i^-) \triangleq \lim_{t \uparrow t_i} e(t) = \lim_{t \uparrow t_i} (\xi(t_{i-1}) - \xi(t))$ . The sampled data controller is then given as

$$u_s = \gamma(\xi_s) = \gamma(\xi) + (\gamma(\xi_s) - \gamma(\xi))$$

The second relation is useful in expressing the sampled data control as a perturbation of the continuous-time control law.

It is a well-known fact that the Lagrangian robot dynamics are linearly parametrizable [13]. Specifically, the following is true.

$$M(q)a + C(q, w)v + G(q) = Y(q, w, v, a)\theta$$

where  $\theta$  is the vector of parameters,  $w$ ,  $v$  and  $a$  are arbitrary real vectors of appropriate size. The specific forms of  $w$ ,  $v$  and  $a$  depend on the context.

Now we give the essential elements of the event-triggered implementations of two well known adaptive controllers for tracking in robot manipulators, proposed in [11] and [12], respectively. Detailed design of the event-triggered implementation for the first one can be found in our previous work [10]. The design procedure for the latter case is similar.

**Important Note:** The arguments of matrix  $Y$  differ in the two cases we discuss. This has to be kept in mind because after the initial definitions, the arguments are generally avoided to keep the notation compact.

## 2.1 Case I: Event-Triggered Implementation of *Beghuis et al.* [11]

In this case the controller is an event-triggered sampled-data implementation of the adaptive controller proposed in [11]. The complete system description is then as follows (description of some of the variables is given after the equations).

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u_s, \quad q \in \mathbb{R}^n \tag{1}$$

$$\begin{aligned} u &= \gamma(\xi) = \hat{M}(q)\ddot{q}_d + \hat{C}(q, \rho)\dot{q}_d + \hat{G}(q) - K_d\dot{\tilde{q}} - K_p\tilde{q} \\ &= Y(q, \rho, \dot{q}_d, \ddot{q}_d)\hat{\theta} - K_d\dot{\tilde{q}} - K_p\tilde{q} \end{aligned}$$

$$u_s = \gamma(\xi_s) = Y_s\hat{\theta}_s - K_d\dot{\tilde{q}}_s - K_p\tilde{q}_s \tag{2}$$

$$\xi_s(t) = \xi(t_i), \quad \text{for all } t \in [t_i, t_{i+1}), \quad \text{for each } i$$

$$t_0 = 0$$

$$t_{i+1} = \min\{t \geq t_i : \beta(\tilde{x})L^T|\xi(t_i) - \xi(t)| \geq \sigma\alpha(\tilde{x})\}, \quad 0 < \sigma < 1 \tag{3}$$

$$\dot{\hat{\theta}} = -\Gamma^{-1}Y_s^T\psi \tag{4}$$

where the variables with the subscript  $s$  are sampled versions of the corresponding signals, the variables with a  $\hat{\cdot}$  on top of them are current estimates of the corresponding variables. The desired trajectory is  $[q_d; \dot{q}_d]$ , the tracking error is  $\tilde{x} \triangleq [\tilde{q}; \dot{\tilde{q}}] \triangleq [q - q_d; \dot{q} - \dot{q}_d]$ ,  $\rho \triangleq \dot{q} - \lambda\tilde{q}$  and  $\psi \triangleq \dot{\tilde{q}} + \lambda\tilde{q}$ , where

$$\lambda = \frac{\lambda_0}{1 + \|\tilde{q}\|}, \quad \lambda_0 > 0$$

where  $\|\cdot\|$  denotes the Euclidean norm. The other gains are  $K_d = K_d^T > 0$ ,  $K_p = K_p^T > 0$ ,  $\Gamma = \Gamma^T > 0$  and  $\sigma$  is a design parameter. The functions  $\alpha$  and  $\beta$  are given by

$$\alpha(\tilde{x}) = k_1 \left\| \dot{\tilde{q}} + \frac{\lambda}{2}\tilde{q} \right\|^2 + k_2 \left\| \frac{\lambda}{2}\tilde{q} \right\|^2, \quad \beta(\tilde{x}) = \|\psi\| = \|\dot{\tilde{q}} + \lambda\tilde{q}\|$$

where  $k_1$  and  $k_2$  are given by

$$\begin{aligned} k_1 &= K_{d,m} - 3\lambda_0 M_M - 2\lambda_0 C_M \\ k_2 &= 4\lambda_0^{-1} K_{p,m} - K_{d,M} - 2\lambda_0 M_M - 2\lambda_0 C_M \end{aligned} \tag{5}$$

The constants in (5), which depend on the robot dynamics and the gains of the controller are assumed to satisfy the following assumption.

**(A1)** Assume that the controller gains are chosen such that

$$\lambda_0 < \min \left\{ \frac{K_{d,m}}{3M_M + 2C_M}, \frac{4K_{p,m}}{K_{d,M} + K_{d,m}} \right\}$$

where  $K_{d,m} \equiv \sigma_m(K_d)$ ,  $K_{d,M} \equiv \sigma_M(K_d)$ ,  $K_{p,m} \equiv \sigma_m(K_p)$ , with  $\sigma_m(\cdot)$ ,  $\sigma_M(\cdot)$  the minimum and maximum eigenvalues respectively. The constants  $M_m$ ,  $M_M$  and  $C_M$  satisfy

$$0 < M_m \leq \|M(q)\| \leq M_M, \quad \|C(q, w)\| \leq C_M\|w\|, \quad \text{for all } w \tag{6}$$

where  $w$  denotes an arbitrary vector.



The other assumptions we make are

- (A2)** The norm of the desired trajectory  $[q_d; \dot{q}_d]$ , and that of its first two derivatives are uniformly bounded by known constants. That is,  $q_d$ ,  $\dot{q}_d$ ,  $\ddot{q}_d$  and  $\overset{\cdot\cdot}{\ddot{q}}_d$  exist for all time, and their norms are uniformly bounded by known constants  $d_0$ ,  $d_1$ ,  $d_2$  and  $d_3$ , respectively.
- (A3)** The matrices  $M(\cdot)$ ,  $C(\cdot, \cdot)$  and  $G(\cdot)$  are globally Lipschitz.

Finally, the vector  $L$  is a quantity that depends on the sampled data  $\xi_s$ . The precise form depends on the specific robot. The purpose of this vector is to bound certain quantities resulting from the perturbation due to the sampled data nature of the controller. The vector  $L$  satisfies the essential property described in Lemma 1, which is taken from [10] (and modified to use a more compact notation). In the sequel, the notation  $|\cdot|$  denotes the component-wise absolute value of a vector or matrix and a *Lipschitz vector* is similar to a Lipschitz constant. More specifically, it is a vector of non-negative elements other than the zero vector.

**Lemma 1.** *Suppose that assumptions (A2),(A3) and conditions (6) hold. Also assume that  $K_p > 0$  and  $K_d > 0$ . Then, there exists a Lipschitz vector  $L$  that depends only on the sampled data, and the uniform bound on  $\dot{q}_d$  such that*

$$\|(Y_s - Y)\theta\| + \|Y_s(\hat{\theta}_s - \hat{\theta})\| + \|K_d(\dot{\hat{q}}_s - \dot{\hat{q}}) + K_p(\hat{q}_s - \hat{q})\| \leq L^T |e|$$

Equations (2)-(3) provide a complete description of the event-triggered controller. The condition that implicitly defines the sampling instants, (3), is the triggering condition. Finally, (4) is the adaptation law for estimating the dynamic uncertainty in the system. The following result, taken from [10], says that with the event-triggered controller, the tracking error in the closed loop system globally converges to zero.

**Theorem 1.** *Under assumptions (A1)-(A3) and dynamics (1)-(4), the tracking error,  $\tilde{x} = [\tilde{q}; \dot{\tilde{q}}]$  globally asymptotically converges to zero.*

The proof of this result relies on a Lyapunov like analysis with the candidate Lyapunov function

$$V(\tilde{q}, \dot{\tilde{q}}) = \frac{1}{2}\psi^T M(q)\psi + \frac{1}{2}\tilde{q}^T K_p \tilde{q} + \frac{1}{2}\tilde{\theta}^T \Gamma \tilde{\theta}$$

where  $\tilde{\theta} \triangleq \hat{\theta} - \theta$ . Note that, due to uncertain dynamics of the robot,  $M(q)$ ,  $\tilde{\theta}$  and thus  $V(\tilde{q}, \dot{\tilde{q}})$  are not known precisely. However, it can be shown that the derivative of the Lyapunov function along the flow of the closed loop system (1)-(4) satisfies

$$\begin{aligned} \dot{V} &\leq -\alpha(\tilde{x}) + \psi^T [(Y_s - Y)\theta + Y_s(\hat{\theta}_s - \hat{\theta}) - K_d(\dot{\hat{q}}_s - \dot{\hat{q}}) - K_p(\hat{q}_s - \hat{q})] \\ &= -\alpha(\tilde{x}) + \beta(\tilde{x})L^T |e| \leq -(1 - \sigma)\alpha(\tilde{x}) \end{aligned}$$

which involves only the known quantities. The triggering condition (3) is derived from the second last relation.

### 2.2 Case II: Event-Triggered Implementation of Slotine-Li [12]

In this case, we present an event-triggered sampled-data implementation of the adaptive controller from [12]. This implementation has a structure similar to CASE I, with only minor differences.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u_s, \quad q \in \mathbb{R}^n \tag{7}$$

$$u = \gamma(\xi) = \hat{M}(q)a + \hat{C}(q, \dot{q})v + \hat{G}(q) - Kr = Y(q, \dot{q}, v, a)\hat{\theta} - Kr \tag{8}$$

$$u_s = \gamma(\xi_s) = Y_s\hat{\theta}_s - Kr_s \tag{9}$$

$$\xi_s(t) = \xi(t_i), \quad \text{for all } t \in [t_i, t_{i+1}), \quad \text{for each } i$$

$$t_0 = 0$$

$$t_{i+1} = \min\{t \geq t_i : \beta(\tilde{x})L^T|\xi(t_i) - \xi(t)| \geq \sigma\alpha(\tilde{x})\}, \quad 0 < \sigma < 1 \tag{10}$$

$$\dot{\hat{\theta}} = -\Gamma^{-1}Y_s^T r \tag{11}$$

where  $r \triangleq \dot{\tilde{q}} + \lambda\tilde{q}$ ,  $v \triangleq \dot{q}_d - \lambda\tilde{q}$ ,  $a \triangleq \ddot{q}_d - \lambda\dot{\tilde{q}}$ ,  $\lambda > 0$ ,  $K > 0$ ,  $\Gamma = \Gamma^T > 0$  are constant gains and  $\sigma$  is a design parameter. Note that, unlike in CASE I,  $\alpha$  and  $\beta$  are positive constants, which are given by

$$\alpha(\tilde{x}) = \min(K, K\lambda^2)\|\tilde{x}\|^2, \quad \beta(\tilde{x}) = \sqrt{(\lambda^2 + 1)}\|\tilde{x}\|$$

The Lipschitz vector satisfies a similar property as in CASE I, which is summarized in the following Lemma.

**Lemma 2.** *Suppose that assumptions (A2),(A3) and conditions (6) hold. Also assume that  $K_p > 0$  and  $K_d > 0$ . Then, there exists a Lipschitz vector  $L$  that depends only on the sampled data, and the uniform bound on  $\dot{q}_d$  such that*

$$\|(Y_s - Y)\theta\| + \|Y_s(\hat{\theta}_s - \hat{\theta})\| + \|K(r_s - r)\| \leq L^T|e|$$

Again, it can be shown that with the event-triggered controller the tracking error in the closed loop system globally converges to zero, which is captured in the following result.

**Theorem 2.** *Under assumptions (A2)-(A3) and dynamics (7)-(11), the tracking error,  $\tilde{x} = [\tilde{q}; \dot{\tilde{q}}]$  globally asymptotically converges to zero.*

The proof of this result is again based on a Lyapunov like analysis with the candidate Lyapunov function

$$V(\tilde{q}, \dot{\tilde{q}}) = \frac{1}{2}r^T M(q)r + \lambda K \tilde{q}^T \tilde{q} + \frac{1}{2}\tilde{\theta}^T \Gamma \tilde{\theta} \tag{12}$$

where  $\tilde{\theta} \triangleq \hat{\theta} - \theta$ . Note that, due to uncertain dynamics of the robot,  $M(q)$ ,  $\tilde{\theta}$  and thus  $V(\tilde{q}, \dot{\tilde{q}})$  are not known precisely. However, it can be shown that the derivative of the Lyapunov function along the flow of the closed loop system (7)-(11) satisfies

$$\begin{aligned} \dot{V} &\leq -\alpha(\tilde{x}) + r^T [(Y_s - Y)\theta + Y_s(\hat{\theta}_s - \hat{\theta}) - K(r_s - r)] \\ &= -\alpha(\tilde{x}) + \beta(\tilde{x})L^T |e| \leq -(1 - \sigma)\alpha\|\tilde{x}\|^2 \end{aligned}$$

In the next section, we present the dynamic model of a two link planar manipulator and the corresponding  $L$  vectors.

### 3 Two Link Planar Manipulator

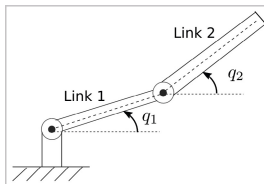
In this section we describe the dynamic model of a planar two-link revolute joint arm, with both the joints driven by motors mounted at the base. We choose this model because of a similar driving mechanism in PHANToM Omni. A schematic of the arm is shown along with the generalized coordinates in Figure 3. The  $M(q)$ ,  $C(q, \rho)$  and  $G(q)$  matrices can be easily derived from the Euler-Lagrange equations or can be found in books on robot modeling (for example, see pages 262-264 [13] ). The regressor matrix,  $Y(q, w, v, a)$ , is given as

$$Y = \begin{bmatrix} a_1 & a_2 \cos(q_2 - q_1) - v_2 \sin(q_2 - q_1)w_2 & 0 & \cos(q_1) & 0 \\ 0 & a_1 \cos(q_2 - q_1) + v_1 \sin(q_2 - q_1)w_1 & a_2 & 0 & \cos(q_2) \end{bmatrix}$$

and the column vector of parameters is given as

$$\begin{aligned} \theta &= [\theta_1; m_2 l_1 l_{c_2}; m_2 l_{c_2}^2 + I_2; (m_1 l_{c_1} + m_2 l_1)g; m_2 l_{c_2} g] \\ \theta_1 &= m_1 l_{c_1}^2 + m_2 l_1^2 + I_1 \end{aligned} \tag{13}$$

Now, the vector  $L$  is presented for the two cases that we considered in Section 2.



**Fig. 3** A schematic of a two link planar revolute manipulator with the second link remotely driven from base of Link 1

**CASE I**

$$L = \begin{bmatrix} \bar{\theta}_2(\phi_{s,1} + d_1\phi_{s,2}) + \bar{\theta}_4 \\ \bar{\theta}_2(\phi_{s,1} + d_1\phi_{s,2}) + \bar{\theta}_5 \\ \theta_2 d_1 \\ \theta_2 d_1 \\ \bar{\theta}_2\phi_{s,1} + \bar{\theta}_4 \\ \bar{\theta}_2\phi_{s,1} + \bar{\theta}_5 \\ \bar{\theta}_2(|\rho_{s,1}| + d_1) \\ \bar{\theta}_2(|\rho_{s,2}| + d_1) \\ \bar{\theta}_1 + \bar{\theta}_2 \\ \bar{\theta}_3 + \bar{\theta}_2 \\ \mathbf{0} \end{bmatrix} + N + D$$

$$\begin{aligned} \phi_{s,1} &= |\rho_{s,1}\dot{q}_{d,s,1}| + |\rho_{s,2}\dot{q}_{d,s,2}| + |\ddot{q}_{d,s,1}| + |\ddot{q}_{d,s,2}| \\ \phi_{s,2} &= \lambda_s(1 + 2|\tilde{q}_{s,1}| + 2|\tilde{q}_{s,2}|) \end{aligned}$$

where  $\bar{\theta}_i$  is a known upper bound on the parameter  $\theta_i$ ,  $\rho_{s_k} = \dot{q}_{s,k} - \lambda_s \tilde{q}_{s,k}$ , for  $k = 1, 2$ ,  $d_1$  is the uniform bound on  $|\dot{q}_{d,1}|$  and  $|\dot{q}_{d,2}|$ . The notation  $\mathbf{0}$  denotes a vector of zeros of appropriate dimension. The vectors  $N$  and  $D$  are given as  $N = [\mathbf{0}^T, \text{Column-wise sum of } |Y_s|]^T$  and  $D = [K_p; K_p; K_d; K_d; \mathbf{0}]$ , respectively. Notice that most of the elements in these vectors are constants or easily computable functions of the sampled data.

**CASE II**

$$L = \begin{bmatrix} \bar{\theta}_2(\phi_s + \lambda(\mu + d_1)) + \bar{\theta}_4 \\ \bar{\theta}_2(\phi_s + \lambda(\mu + d_1)) + \bar{\theta}_5 \\ \bar{\theta}_2(\lambda + |v_{s,1}|) + \bar{\theta}_1\lambda \\ \bar{\theta}_2(\lambda + |v_{s,2}|) + \bar{\theta}_3\lambda \\ \bar{\theta}_2\phi_s + \bar{\theta}_4 \\ \bar{\theta}_2\phi_s + \bar{\theta}_5 \\ \bar{\theta}_2(\mu + d_1 + |v_{s,1}|) \\ \bar{\theta}_2(\mu + d_1 + |v_{s,2}|) \\ \bar{\theta}_1 + \bar{\theta}_2 \\ \bar{\theta}_3 + \bar{\theta}_2 \\ \mathbf{0} \end{bmatrix} + N + D$$

$$\phi_s = |v_{s,1}\dot{q}_{s,1}| + |v_{s,2}\dot{q}_{s,2}| + |a_{s,1}| + |a_{s,2}|, \quad \mu = \sqrt{\frac{V_{est}}{\alpha_1}}$$

where  $\bar{\theta}_i$  is a known upper bound on the parameter  $\theta_i$ ,  $\rho_{s_k} = \dot{q}_{s,k} - \lambda_s \tilde{q}_{s,k}$ , for  $k = 1, 2$ ,  $d_1$  is the uniform bound on  $|\dot{q}_{d,1}|$  and  $|\dot{q}_{d,2}|$ . The notation  $\mathbf{0}$  denotes a vector of zeros of appropriate dimension. The vectors  $N$  and  $D$  are given as  $N = [\mathbf{0}^T, \text{Column-wise sum of } |Y_s|]^T$  and  $D = [\lambda K; \lambda K; K; K; \mathbf{0}]$ , respectively. The quantity  $V_{est}$  is an upper bound on the value of the Lyapunov function (12) at the sampling instant. Even though the precise value

of the Lyapunov function cannot be known, it can still be bounded by known quantities.

$$\alpha_1 \|\tilde{x}_s\|^2 \leq V(\tilde{q}_s, \dot{\tilde{q}}_s) \leq \alpha_2 \|\tilde{x}_s\|^2 + \frac{1}{2} H^T \Gamma H = V_{est}$$

where  $\alpha_1$  and  $\alpha_2$  are positive constants. These constants can be found from either the lower and upper bounds on  $\|M(q)\|$ ,  $M_m$  and  $M_M$ , respectively or from the individual bounds on the parameters,  $\bar{\theta}_i$ .  $H$  is a vector such that  $H_i \geq |\hat{\theta}_{s,i}|$  for each  $i$ . This vector can be estimated as

$$H_i = \begin{cases} \bar{\theta}_i - \hat{\theta}_{s,i}, & \text{if } \hat{\theta}_{s,i} < \bar{\theta}_i \\ \hat{\theta}_{s,i}, & \text{if } \hat{\theta}_{s,i} \geq \bar{\theta}_i \end{cases}$$

Notice that computation of the  $L$  vector is more involved than in CASE I. This is mainly because in CASE I, the arguments  $\rho$ ,  $\dot{q}_d$  and  $\ddot{q}_d$  in the  $Y$  matrix are all uniformly bounded by known constants and do not have to be estimated at each sampling instant.

### 4 Experiments

In the experimental results presented here, the position variables of the desired trajectory were chosen as

$$q_{d,1} = -0.4(\cos(0.8t) - 1.1), \quad q_{d,2} = -0.4(\cos(0.3\pi t) - 1) - (\pi/2)$$

The signals  $\dot{q}_d$ ,  $\ddot{q}_d$  and  $\dddot{q}_d$  were defined simply as the corresponding derivatives of  $q_d$ . The control gains and the parameters were heuristically chosen as

$$\bar{\theta} = [0.0035, 0.0035, 0.002, 0.2, 0.1]^T$$

$$d_1 = 0.5, \quad h_l = 10^{-8}, \quad \sigma \in \{0.95, 0.6, 0.2\}$$

**CASE I:**  $\lambda_0 = 0.7, \quad K_d = 0.03, \quad K_p = 0.7$

$$\Gamma = \text{diag}([30, 40, 50, 10, 10]^T)$$

**CASE II:**  $\lambda = 1.5, \quad K = 0.03, \quad \Gamma = \text{diag}([250, 350, 400, 0.8, 0.9]^T)$

where  $d_1$  is the uniform upper bound on  $|q_{d,1}|$  and  $|\dot{q}_{d,2}|$  and  $h_l$  is a lower bound on  $(\theta_1\theta_3 - \theta_2^2)$ , which can be easily shown to be positive for a two link manipulator. Using these quantities,  $M_M$ ,  $M_m$  and  $C_M$  can be estimated as

$$M_M = \frac{\bar{\theta}_1 + \bar{\theta}_3 + \sqrt{(\bar{\theta}_1 + \bar{\theta}_3)^2 - 4h_l}}{2}$$

$$M_m = \frac{\bar{\theta}_1 + \bar{\theta}_3 - \sqrt{(\bar{\theta}_1 + \bar{\theta}_3)^2 - 4h_l}}{2}, \quad C_M = \bar{\theta}_2$$

which are useful to compute  $\alpha_1$  and  $\alpha_2$  in CASE II. Finally, the initial conditions of the system were chosen as

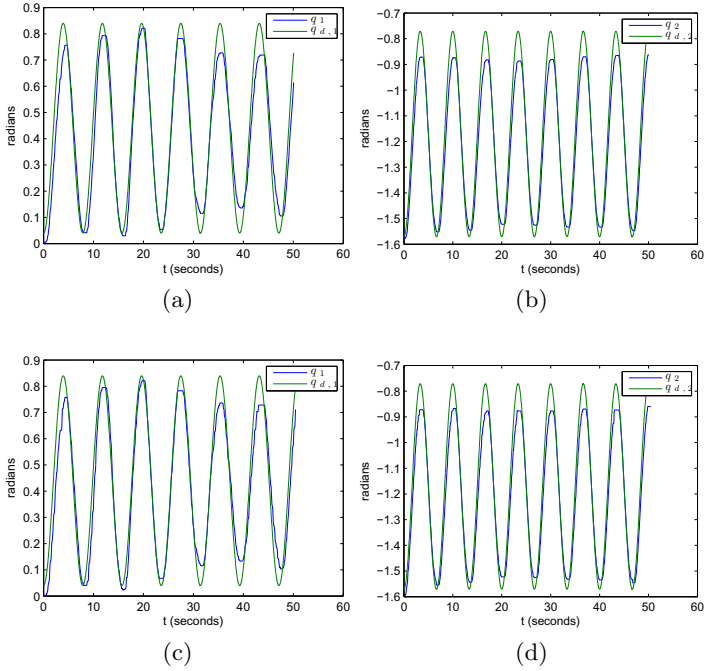
$$\begin{aligned} [q_1, q_2, \dot{q}_1, \dot{q}_2]^T(0) &= [0, -\pi/2, 0, 0]^T \\ \hat{\theta}(0) &= [0.0001, 0.0001, 0.0001, 0.01, 0.001]^T \end{aligned}$$

The choice of such low initial values for  $\hat{\theta}$  is motivated by the fact that initial torques will be lower in the absence of knowledge of the system parameters. Notice from (13) that the first three parameters of the robot involve quadratic terms in the length, while the last two involve only linear terms in the length and also involve acceleration due to gravity,  $g$ . Thus, for the PHANToM Omni, which is a desktop robot with small link lengths and small link masses, the last two parameters are larger than the others by some orders of magnitude. Thus intuitively, the adaptation gain,  $\Gamma^{-1}$ , has to be chosen such that  $\hat{\theta}_4$  and  $\hat{\theta}_5$  evolve at a faster rate. The other control gains were also chosen by heuristic means. For example,  $K_d$  in CASE I and  $K$  in CASE II multiply velocity terms. Since the joint velocity measurements obtained from the device were very noisy, we have chosen relatively smaller values for these gains to limit buzzing, or high frequency and high magnitude torques.

Experiments were performed on a two-link planar manipulator. PHANToM Omni was used as a test bed. For the experiments presented here, only the second and third joint were kept active, to keep the calculations simple. The first joint was never actuated and the remaining joints were either removed or constrained to a fixed position. The OpenHaptics 3.0 [14] was used to program the PHANToM Omni for performing tasks such as reading the sensors and controlling the joint torques. The OpenHaptics 3.0 API does not provide the capability to arbitrarily choose the sampling and control update instants. The sensors are sampled and control torques are updated with a roughly constant period (1ms for the results presented here). Hence, in the experiments the event-triggering condition was checked roughly every 1ms.

Some experimental results for Case I are presented in Figure 4, while the results for Case II are presented in Figure 5. These figures show the desired joint positions and the actual joint positions as the robot is tracking. Notice that for smaller values of  $\sigma$  the tracking performance increases - this is especially clear in joint 2 data of Figure 5. On the other hand, there is a greater overshoot for  $\sigma = 0.2$  in the same figure. The improvement in tracking with decrease in the design parameter  $\sigma$  value is to be expected because it has a clear inverse relationship with the average sampling frequency.

From these figures it appears that the tracking error convergence rate and tracking performance are better in Case I than in Case II, particularly with respect to Joint 2. It must be pointed out that there is considerable friction in the joints, which has not been included in the system model. This friction accounts for some of the persisting tracking error. Similarly, to avoid ‘buzzing’ some of the gains had to be limited to smaller values as the joint velocity measurements are noisy. Another possible factor influencing the bad tracking



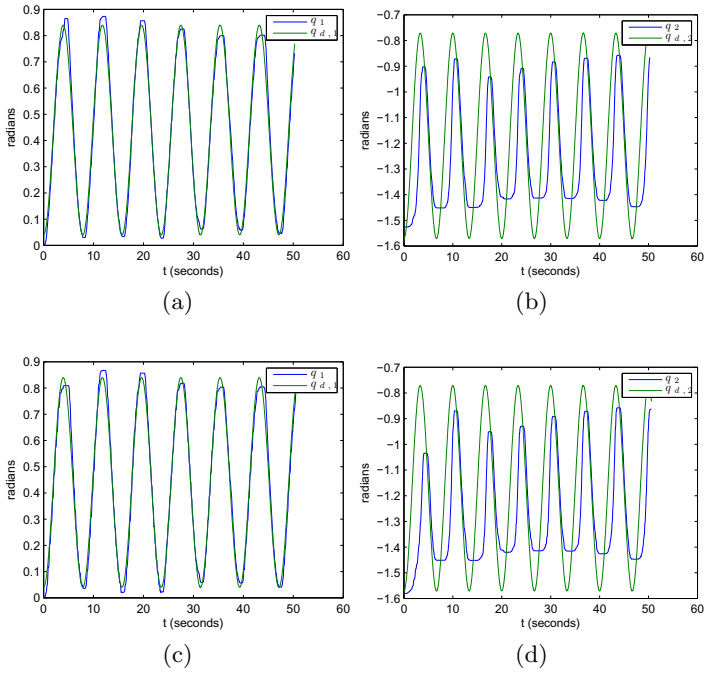
**Fig. 4** The desired joint positions and the actual positions of the robot in Case I. (a), (b)  $\sigma = 0.2$ , (c), (d)  $\sigma = 0.95$ .

of Joint 2 in CASE II is the quadratic dependence on joint velocity in the control law (8).

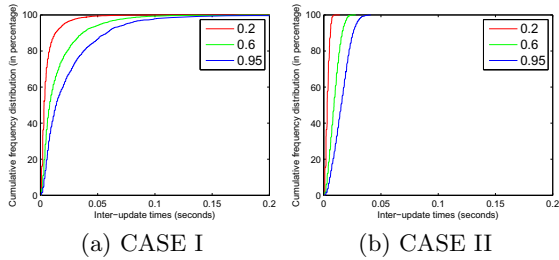
The observed minimum inter-update times and average frequency in experiments are reported in Table 1. In the Phantom Omni system, the control inter-update period is inherently fixed at 1ms, which thus limits the minimum inter-update time of the control value. Figure 6 shows the cumulative frequency distribution of the control inter-update times for different values of  $\sigma$  and for each of the cases. Finally, the irregular state based sampling and control updates is illustrated in Figure 7.

**Table 1** The observed minimum inter-update times and average frequency in experiments

$\sigma$	Case I		Case II	
	Observed min. inter-update time (s)	Observed avg. Frequency (Hz)	Observed min. inter-update time (s)	Observed avg. Frequency (Hz)
0.2	$4.7 \times 10^{-4}$	164	$7 \times 10^{-4}$	298
0.6	$7.4 \times 10^{-4}$	64	$9.7 \times 10^{-4}$	106
0.95	$9.9 \times 10^{-4}$	40	$9.9 \times 10^{-4}$	63



**Fig. 5** The desired joint positions and the actual positions of the robot in Case II. (a), (b)  $\sigma = 0.2$ , (c), (d)  $\sigma = 0.95$ .



**Fig. 6** The cumulative frequency distribution of the control inter-update times in the experiments for different values of  $\sigma$



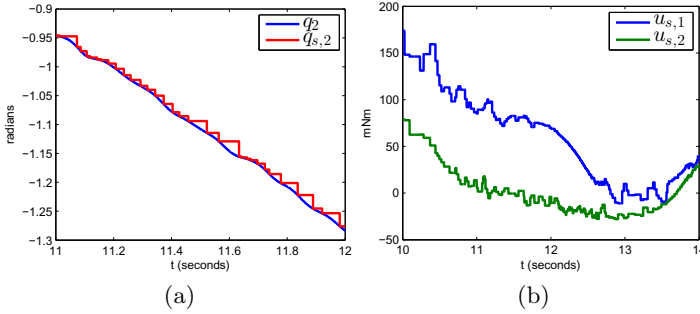


Fig. 7 Irregular state based sampling and control updates

## 5 Discussion and Conclusions

In this paper, a non-periodic state based control sampling technique was experimentally investigated for two well known controllers for tracking in robot manipulators. The experimental results demonstrate the promise that event based algorithms hold in robotic applications.

In the experiments, the parameters and control gains were not optimized. They were chosen by trial-and-error and some heuristic insights to give ‘good’ performance, in terms of low average sampling rates and minimizing tracking error. Similarly, the results presented here are for a single desired trajectory. Yet, these results provide valuable insights about the numerous factors affecting the performance.

The experimental results presented here suggest that the controller in CASE I is better suited for Lyapunov based sampling. It must be pointed out that in CASE II, joint 1 is tracked very well. However, joint 2 tracking is not good. No significant improvement was observed by changing the control gains. By changing the adaptation gains, there was some improvement in tracking performance, though at the cost of increased average sampling frequency. Another possible cause of bad tracking performance and higher average sampling rates in CASE II, is the highly noisy joint velocity data. These factors will be investigated more systematically in future studies.

## References

- [1] Khosla, P.K.: Choosing sampling rates for robot control. In: IEEE International Conference on Robotics and Automation, vol. 4, pp. 169–174 (1987)
- [2] Tarn, T.J., Ganguly, S., Ramadorai, A.K., Marth, G.T., Bejczy, A.K.: Experimental evaluation of the nonlinear feedback robot controller. In: IEEE International Conference on Robotics and Automation, pp. 1638–1644 (1991)

- [3] Simon, D., Castillo Castaneda, E., Freedman, P.: Design and analysis of synchronization for real-time closed-loop control in robotics. *IEEE Transactions on Control Systems Technology* 6(4), 445–461 (1998)
- [4] Leahy Jr., M.B.: Industrial manipulator control with feedforward dynamic compensation. In: *IEEE Conference on Decision and Control*, pp. 598–603 (1988)
- [5] Alici, G., Daniel, R.W.: Experimental comparison of model-based robot position control strategies. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 76–83 (1993)
- [6] Tabuada, P.: Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control* 52(9), 1680–1685 (2007)
- [7] Heemels, W.P.M.H., Sandee, J.H., Van Den Bosch, P.P.J.: Analysis of event-driven controllers for linear systems. *International Journal of Control* 81(4), 571–590 (2008)
- [8] Wang, X., Lemmon, M.D.: Self-triggered feedback control systems with finite-gain  $\mathcal{L}_2$  stability. *IEEE Transactions on Automatic Control* 54, 452–467 (2009)
- [9] Tallapragada, P., Chopra, N.: On event triggered trajectory tracking for control affine nonlinear systems. In: *IEEE Conference on Decision and Control and European Control Conference*, pp. 5377–5382 (2011)
- [10] Tallapragada, P., Chopra, N.: Lyapunov based sampling for adaptive tracking control in robot manipulators. *IEEE Transactions on Robotics* (Submitted)
- [11] Berghuis, H., Ortega, R., Nijmeijer, H.: A robust adaptive controller for robot manipulators. In: *IEEE International Conference on Robotics and Automation*, pp. 1876–1881 (1992)
- [12] Slotine, J., Weiping, L.: Adaptive manipulator control: A case study. *IEEE Transactions on Automatic Control* 33(11), 995–1003 (1988)
- [13] Spong, M.W., Hutchinson, S., Vidyasagar, M.: *Robot Modeling and Control*. John Wiley & Sons, Inc., New York (2006)
- [14] SensAble Technologies OpenHaptics Toolkit 3.0 (January 2009), <http://www.sensable.com/products-openhaptics-toolkit.htm>

# Linguistic Composition of Semantic Maps and Hybrid Controllers<sup>\*</sup>

Neil Dantam, Carlos Nieto-Granda, Henrik I. Christensen, and Mike Stilman

**Abstract.** This work combines semantic maps with hybrid control models, generating a direct link between action and environment models to produce a control policy for mobile manipulation in unstructured environments. First, we generate a semantic map for our environment and design a base model of robot action. Then, we combine this map and action model using the Motion Grammar Calculus to produce a combined robot-environment model. Using this combined model, we apply supervisory control to produce a policy for the manipulation task. We demonstrate this approach on a Segway RMP-200 mobile platform.

## 1 Introduction

This paper provides an approach to generate robot policies by automatically combining Semantic Mapping and Hybrid Control. Semantic mapping and hybrid control are both effective approach within robotics. Semantic mapping produces detailed models of *unstructured environments* [19, 26, 24, 20, 27]; however, this approach provides no direct link to robot action. Hybrid models combine continuous and discrete robot dynamics to efficiently and verifiably represent *robot action* [8, 6, 7, 2, 11, 4]; however, there is no automatic method to produce control models for large, complicated *systems*. While superficially, it appears that semantic mapping and hybrid control are fundamentally different approaches, they are actually closely related. The topological graph of a semantic map and the discrete event system of a hybrid control model are both instances of *formal language*. Thus, we propose

---

Neil Dantam · Carlos Nieto-Granda · Henrik I. Christensen · Mike Stilman  
Center for Robotics and Intelligent Machines, Georgia Institute of Technology,  
Atlanta, GA 30332, USA  
e-mail: {ntd, carlos.nieto}@gatech.edu,  
{hic, mstilman}@cc.gatech.edu

<sup>\*</sup> This work supported by NSF grants CNS1146352 and CNS1059362 and by grants from the ARL by the MAST CTA and Boeing Corporation.

to combine the linguistic representations of semantic maps and robot action models to produce an efficient and verifiable control policy for mobile manipulation in unstructured environments.

This work focuses on the application domain of service robots in human environments. Previously, we developed new techniques for mapping using Semantic SLAM [19, 26] and for hybrid systems using our Motion Grammar [8, 6, 7]. Here, we integrate these approaches to produce a combined robot-environment action model. Then, we apply established methods in supervisory control [5] to derive a robot control policy for a mobile manipulation task. This control design approach formally guarantees that the resultant policy satisfies the task specification. Finally, we demonstrate of this approach on a Segway RMP-200 mobile robot.

## 2 Related Work

Simultaneous Localization and Mapping (SLAM) is the concurrent pose estimation of both the robot and objects in its environment. This is a well studied area with many useful results. Smith and Cheeseman [23] proposed one of the first solutions to the SLAM problem using the Extended Kalman Filter (EKF) to jointly represent the landmark positions along with the robot pose. Folkesson and Christensen developed GraphSLAM [9], an efficient solution to the SLAM problem which preserves landmark independence and is able to find loop closures through nonlinear optimization. Semantic SLAM augments a map with semantically relevant object labels. In this work, we utilize the Semantic SLAM method of Trevor and Nieto [26, 27, 19] to compose a map and hybrid controller.

Hybrid Control is an ongoing research area describing systems with both discrete, event-driven, dynamics and continuous, time-driven, dynamics. Ramadge and Wonham [21] first applied Language and Automata Theory [13] to Discrete Event Systems (DES). Hybrid Automata generally associate differential equations with each state of a Finite Automaton (FA). This method is well studied in control [5, 14, 18, 11, 2]. In this paper, we model hybrid systems using the Motion Grammar which represents continuous dynamics with differential equations and discrete dynamics using a Context-Free Grammar (CFG) [8], and we extend the hybrid approach by automating system modeling using semantic maps. Supervisory control restricts the operation of a DES based on a linguistic specification. This is applied to mobile robot motion planning with FA and Linear Temporal Logic (LTL) by [4, 17]. In our approach, we apply supervisory control to CFGs. Composition of multiple robot behaviors as FA is described in [16]. In this work, we compose a robot model with an automatically generated map while preserving a system representation as a CFG to maintain verifiability and efficiency of execution.

Model checking is the practice of verifying system behavior by modeling the system and verifying that it satisfies a desired specification [3]. This approach is applied to computer software [12] and motion planning for mobile robots [17]. [6] summarizes the classes of models and specifications for which model checking of robotic systems is decidable.

There are several related techniques and alternative approaches for the service robotics domain. Topp and Christensen, [25, 24], provide a separation of regions relating to a user's view on the environment and detection of transitions between them. O'Callaghan [20] developed a new statistical modeling technique for building occupancy maps by providing both a continuous representation of the robot's surroundings and an associated predictive variance employing a Gaussian process and Bayesian learning. In this work we focus on integrating robot mapping with hybrid control methods. The notion of affordances originated in Psychology [10] to describe interaction between agents and environments and has previously provided inspiration for robotics research [22]. We rather focus our approach on direct symbol manipulation techniques with clear algorithmic implementation.

### 3 Background

The method of this paper produces a robot control policy for unstructured environments by combining Simultaneous Localization and Mapping (SLAM) with Hybrid Control. We combine these two approaches through Formal Language. First, we produce a basic grammar for the robot's actions and generate the map of the environment via SLAM. Then we compose the action grammar and environment map using the Motion Grammar Calculus. Finally, we apply a supervisory controller to generate the policy for the robot.

We now explain some background on formal language, define our hybrid systems model, the Motion Grammar, and summarize the SLAM technique.

#### 3.1 Formal Language

Formal language is the underlying representation we use to combine mapping and hybrid control. Language and automata theory provide a rigorous method for reasoning about the discrete dynamics of a robotic system. A *formal language* is a set of strings. Strings are sequences of atomic symbols which we can use to describe discrete events, predicates, locations, or actions within our system. A *grammar* defines a formal language. We first briefly review some relevant points of language theory. For a thorough coverage of formal language and its applicability to robotic systems, please see [13, 5, 6].

**Definition 1 (Context-Free Grammar, CFG).**  $G = (Z, V, P, S)$  where  $Z$  is a finite alphabet of symbols called tokens,  $V$  is a finite set of symbols called nonterminals,  $P$  is a finite set of mappings  $V \mapsto (Z \cup V)^*$  called productions, and  $S \in V$  is the start symbol.

The productions of a CFG are conventionally written in Backus-Naur form. This follows the form  $A \rightarrow X_1 X_2 \dots X_n$ , where  $A$  is some nonterminal and  $X_1 \dots X_n$  is a sequence of tokens and nonterminals. This indicates that  $A$  may *expand* to all strings represented by the right-hand side of the productions. The symbol  $\epsilon$  is used

to denote an empty string. For additional clarity, nonterminals may be represented between angle brackets  $\langle \rangle$  and tokens between square brackets  $[]$ .

Grammars have equivalent representations as *automata* which *recognize* the language of the grammar. This automata form provides a more convenient representation for some tasks, such as defining languages for maps in Sect. 4.1. The equivalence of grammars and automata means that we can freely choose whichever representation is most convenient. In the case of a Regular Grammar – where all productions are of the form  $\langle A \rangle \rightarrow [a] \langle B \rangle$ ,  $\langle A \rangle \rightarrow [a]$ , or  $\langle A \rangle \rightarrow \varepsilon$  – the equivalent automaton is a Finite Automaton (FA), similar to a Transition System with finite state. A CFG is equivalent to a Pushdown Automaton, which is an FA augmented with a stack; the addition of a stack provides the automaton with memory and can be intuitively understood as allowing it to count.

**Definition 2 (Finite Automata, FA).**  $M = (Q, Z, \delta, q_0, F)$ , where  $Q$  is a finite set of states,  $Z$  is a finite alphabet of tokens,  $\delta : Q \times Z \mapsto Q$  is the transition function,  $q_0 \in Q$  is the start state,  $F \subseteq Q$  is the set of accept states.

**Definition 3 (Acceptance and Recognition).** An automaton  $M$  accepts some string  $\sigma$  if  $M$  is in an accept state after reading the final element of  $\sigma$ . The set of all strings that  $M$  accepts is the language of  $M$ ,  $\mathcal{L}_M$ , and  $M$  is said to recognize  $\mathcal{L}_M$ .

Regular Expressions [13] and Linear Temporal Logic (LTL) [3] are two alternative notations for finite state languages. These representations are convenient forms for defining supervisory controllers as in Sect. 4.3. The basic Regular Expression operators are concatenation  $\alpha\beta$ , union  $\alpha|\beta$ , and Kleene-closure  $\alpha^*$ . Some additional common Regular Expression notation includes  $\neg\alpha$  which is the complement of  $\alpha$ , the dot ( $\cdot$ ) which matches any token, and  $\alpha^?$  which is equivalent to  $\alpha|\varepsilon$ . Regular Expressions are equivalent to Finite Automata and Regular Grammars. LTL extends propositional logic with the binary operator *until*  $\cup$  and unary prefix operators *eventually*  $\diamond$  and *always*  $\square$ . LTL formulae are equivalent to Büchi automata, which represent *infinite* length strings, termed  $\omega$ -Regular languages. We can also write  $\omega$ -Regular Expressions by extending classical Regular expressions with infinite repetition for some  $\alpha$  given as  $\alpha^\omega$ . These additional notations are convenient representations for finite state supervisors.

### 3.2 The Motion Grammar

Next, we model robot action using the Motion Grammar (MG), giving an initial set of hybrid control actions the robot can perform. MG represents the operation of a robotic system as a Context-Free language, augmenting a Context-Free Grammar with additional variables to handle the continuous dynamics. We use this combined representation to describe the operation of the full robotic *system* [8, 6].

**Definition 4.** *The Motion Grammar is a tuple  $\mathcal{G}_M = (Z, V, P, S, \mathcal{X}, \mathcal{Z}, \mathcal{U}, \eta, K)$  where,*

$Z$	<i>set of events, or tokens</i>
$V$	<i>set of nonterminals</i>
$P \subseteq V \times (Z \cup V \cup K)^*$	<i>set of productions</i>
$S \in V$	<i>start symbol</i>
$\mathcal{X} \subseteq \mathbb{R}^m$	<i>continuous state space</i>
$\mathcal{Z} \subseteq \mathbb{R}^n$	<i>continuous observation space</i>
$\mathcal{U} \subseteq \mathbb{R}^p$	<i>continuous input space</i>
$\eta : \mathcal{Z} \times P \times \mathbb{N} \times \mapsto Z$	<i>tokenizing function</i>
$K \subseteq \mathcal{X} \times \mathcal{U} \times \mathcal{Z} \mapsto \mathcal{X} \times \mathcal{U} \times \mathcal{Z}$	<i>semantic rules</i>

The Motion Grammar describes the *language* of the robotic system. The terminal symbols of this language are robot events and predicates, representing a discrete abstraction of the system path.

We use two properties to ensure the validity of a system modeled as a Motion Grammar: *completeness* and *correctness*. Completeness ensures that our model  $\mathcal{G}$  is a faithful representation of the physical system  $F$ . We define this property using the *simulation* relation, that all paths in  $F$  are also paths in  $\mathcal{G}$ . Correctness ensures that our model  $\mathcal{G}$  satisfies some desired property  $S$ . We define correctness using the subset relation.

**Definition 5.** Given  $\mathcal{G}_M$  and system  $F$  then complete  $\{\mathcal{G}\} \equiv F \preceq \mathcal{G}_M$

**Definition 6.** A Motion Grammar  $\mathcal{G}$  is correct with respect to some specification  $S$  when all strings in the language of  $\mathcal{G}$  are also in  $S$ : correct  $\{\mathcal{G}, S\} \equiv \mathcal{L}(\mathcal{G}) \subseteq \mathcal{L}(S)$ .

### 3.3 Semantic Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) is the concurrent execution of both *Localization* and *Mapping* on a robot. *Localization* means determining the current position of the robot based on observations. *Mapping* means determining the positions of objects in the environment based on observations. Typical SLAM implementations combine odometry and other geometric measurements such as point clouds or camera features to simultaneously produce an estimate of the position of both the robot and objects. Using this technique, the robot models unstructured environments.

Our mapping system identifies surfaces and connected free spaces in the world [26, 27]. We use the surfaces, such as walls and tables, to localize the robot based on its relative position to these object. We represent free spaces as Gaussian regions in  $\mathbb{R}^3$  with mean at the center of the free space and standard deviation indicating the dimensions of the free space [19]. Topological connections between these Gaussian regions indicate connected free spaces in the environment. For example, a door or hallway between two rooms would connect the Gaussian regions for those rooms.

We then extend the metric and topological information of the map surfaces and connected Gaussians with additional semantic information by labeling each of the Gaussian regions. These *Semantic Maps* provide useful information for navigation

and localization of the robot. In addition, the semantic content of the map permits higher-level reasoning about the spatial regions of the environment. We exploit this semantic information in our composition of the map with a grammar for robot action.

### 4 Composing Maps and Grammars

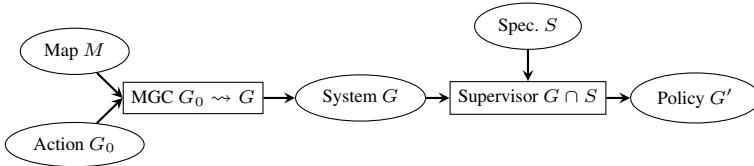


Fig. 1 Sequence of operations to generate policy

We produce the control policy for the robot by composing a semantic map and a base action grammar, following Fig. 1. We will explain this approach using the example map for the Georgia Tech Aware home, Fig. 2(a), and the base grammar for mobile manipulation, Fig. 2(b). First, we convert the map graph into a grammar for the map language. Then, we compose the map grammar and the action grammar using the Motion Grammar Calculus (MGC) to model the robotic system operating within the mapped environment. Finally, we produce a task policy by applying a supervisory controller to this system model.

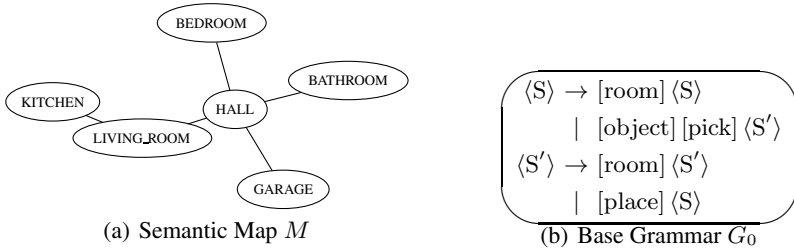


Fig. 2 Example of Semantic Map  $M$  and base manipulation grammar  $G_0$ . This map represents the Georgia Tech Aware Home.

#### 4.1 Map Languages

To better analyze the semantic map, we first represent this map using formal language. The Gaussian free space regions of the map are arranged in a graph, indicating connectivity between these regions. The graph for the Aware Home is Fig. 2(a).



This graph is equivalent to a Regular Language representing the set of all traces through the map.

**Definition 7.** Let Map  $M = (N, V)$ , where  $N$  is a finite set of location symbols, and  $V \subset N \times N$  is the set of adjacent symbols  $n_i \rightarrow n_j$ .

We can transform any Map  $M$  into a regular grammar. We note that when analyzing Finite Automata, the language symbols are typically given along transitions [13, 1] whereas in a map, location symbols mark a state. For regular languages, these two conventions – terminal language symbols on states and terminal language symbols on edges – are equivalent. Algorithm 1 transforms the state terminal map to an edge terminal automaton. Then, we can directly convert this automaton to a Regular Grammar.

We demonstrate the conversion for the map in Fig. 2(a). First, we apply Algorithm 1 to produce a FSM from the map graph. Since the output of this algorithm is a Nondeterministic Finite Automaton with more than the minimum necessary number of states, we convert the NFA to a DFA [1, p152] and minimize the number of DFA states with Hopcroft’s Algorithm [1, p180]. This result is Fig. 3(a). Note that in this example, we save two states over the original map in Fig. 2(a). Finally, we convert the FSM to the Regular grammar in Fig. 3(b).

---

**Algorithm 1.** State to Edge Symbols

---

```

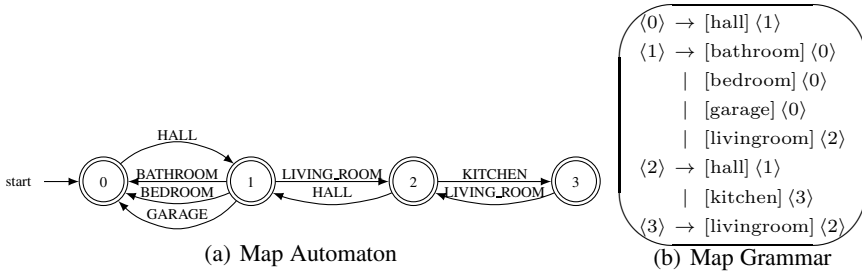
Input:  $Q$  ; // Initial States
Input:  $E : Q \times Q$  ; // Initial Edges
Output:  $Q'$  ; // Final States
Output:  $Z'$  ; // Edge Symbols
Output:  $E' : Q' \times Z' \times Q'$  ; // Final Edges
1  $Z' = Q$ ;
2  $Q' = E$  ;
3  $E' = \emptyset$ ;
4 forall  $q \in Q$  do
5   forall  $(e_i = Q \rightarrow q) \in E$  do
6     forall  $(e_j = q \rightarrow Q) \in E$  do
7        $E' = E' \cup e_i \xrightarrow{q} e_j$ 

```

---

### 4.2 Composition Using the Motion Grammar Calculus

In order to semantically merge the robot and environment models, we apply our Motion Grammar Calculus (MGC). MGC is a set of rewrite rules for hybrid systems modeled in the Motion Grammar [7]. According to these rules, we extend our action grammar with each map symbol while maintaining only those transitions allowed by the map. While supervisory control can only operate to restrict system  $G$  using



**Fig. 3** Representing maps with formal language

existing symbols, the MGC crucially describes how to introduce *new symbols* into  $G$ . There are two relevant rewrite rules from the MGC that we use here.

**Transform 1 (Symbol Splitting).** *Given some  $\zeta_0 = [x \in \mathcal{R}_0] \in Z$ , create tokens  $\zeta_1 = [x \in \mathcal{R}_1]$  and  $\zeta_2 = [x \in \mathcal{R}_2]$  such that  $\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}_0 \wedge \mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$  and update token set  $Z' = Z - \zeta_0 \cup \{\zeta_1, \zeta_2\}$ . The new nonterminal set is  $V' = V \cup \{A_0, A_1, A_2, A_3, A_4\}$ . The new production set is  $P' = P - \{(A \rightarrow \alpha_1 \zeta_0 \kappa \alpha_2) \in P\} \cup \{(A \rightarrow \alpha_1 A_0), (A_0 \rightarrow A_1 | A_2) : (A \rightarrow \alpha_1 \zeta_0 \kappa \alpha_2) \in P\} \cup \{(A_1 \rightarrow \zeta_1 \kappa A_3), (A_2 \rightarrow \zeta_2 \kappa A_4) : (A \rightarrow \alpha_1 \zeta_0 \kappa \alpha_2) \in P\} \cup \{(A_3 \rightarrow A_2 | \alpha_2), (A_4 \rightarrow A_1 | \alpha_2) : (A \rightarrow \alpha_1 \zeta_0 \kappa \alpha_2) \in P\}$ .*

**Transform 2 (Adjacency Pruning).** *For  $p_1 = A \rightarrow r_A \kappa_A B, B \rightarrow \beta_1 | \dots | \beta_n$ , if  $r_A$  is not adjacent to  $\mathcal{R}_0(\beta_n)$  WLOG, then  $P' = P - p_1 \cup \{A \rightarrow r_A \kappa_A B'\} \cup \{B' \rightarrow \beta_1 | \dots | \beta_{n-1}\}$*

By applying these transforms, we can introduce the map symbols into the action grammar while preserving the validity of the model. Each derivation step maintains the *completeness* of the model according to the path of the hybrid system. By assuming that the initial model is complete, this ensures that all derived models are also complete. For the remainder of the MGC and proofs of its correctness, please see [7].

In addition to these two transforms, we also use the `first()` and `follow()` sets [1] to define initial and adjacent symbols. The `first()` set defines all terminals which may begin some derivation of a grammar symbol. The `follow()` set defines all terminals which may appear immediately to the right of some symbol in a grammatical derivation [1][p221].

**Definition 8 (First Set).** *Define  $\text{first}(X)$  for some grammar symbol  $X$  to be the set of terminals which may begin strings derived from  $X$ .*

**Definition 9 (Follow Set).** *Define  $\text{follow}(X)$  for grammar symbol  $X$  to be the set of terminals  $a$  that can appear immediately to the right of  $X$  in some sentential form.*

Note that for map grammars such as Fig. 3(b), the follow set for each terminal symbol is equivalent to the adjacent nodes in the map graph Fig. 2(a).

**Proposition 1.** *Given a grammar  $G$  representing some map  $M$ ,  $\text{follow}(z)$  of some terminal symbol  $z$  of  $G$  represents the set of all map locations adjacent to  $z$ .*

Algorithm 2 describes how we apply these transforms to compose the Map and Action grammars. First, we introduce all map symbols into the action grammar by repeatedly splitting the initial terminal symbol of the action grammar by direct application of Transform 1. Next, we prune out productions indicating transitions between non-adjacent map locations. To prune these productions, we apply Transform 2 by intersecting the grammar with sets of allowable transitions. The disallowed transitions are indicated by the regular expression  $L = (.^*z_1Z_A^*z_2.^*)$  in line 2 of Algorithm 2. The complement of this regular expression defines all paths which do not move directly from  $z_1$  to  $z_2$ . Since  $z_1$  and  $z_2$  are non-adjacent, intersecting with  $\bar{L}$  will preserve only paths which do not contain the disallowed transition. The result is a grammar which contains the original action model and all permissible transitions from the semantic map.

---

**Algorithm 2.** Composing Map and Action Grammars

---

```

Input:  $(Z_M, V_M, P_M, S_M)$ ; // Map Grammar
Input:  $(Z_A, V_A, P_A, S_A)$ ; // Action Grammar
Output:  $(Z, V, P, S)$ ; // Combined Grammar
1  $(Z, V, P, S) \leftarrow (Z_A, V_A, P_A, S_A)$ ;
   /* Add map symbols by splitting  $\text{first}(S_A)$  */
2  $z_0 = \text{first}(S_A)$ ;
3 forall  $z \in Z_M$  do
4    $\lfloor (Z, V, P, S) \leftarrow \text{Transform 1 to split } z_0 \text{ into } z \text{ and } z_0$ 
   /* Prune non-adjacent map symbols */
5 forall  $z_1 \in Z_M$  do
6   forall  $z_2 \in Z_M$  do
7     if  $z_2 \notin \text{follow}(z_1)$  then
8        $\lfloor (Z, V, P, S) \leftarrow (Z, V, P, S) \cap \overline{\mathfrak{L}\{.^*z_1Z_A^*z_2.^*\}}$ ;

```

---

We apply Algorithm 2 to combine the map grammar, Fig. 3(b), with the base grammar for mobile manipulation, Fig. 2(b). In this process, the initial nonterminal of the base grammar, [room], is repeatedly split into all the symbols of the semantic map. Then all transitions between non-adjacent map symbols are pruned away. This produces the combined grammar of Fig. 4(a).

### 4.3 Supervisory Control

Finally, we use supervisory control to produce the policy  $G'$  from our system model  $G$  and task specification  $S$ , [5, p133]. This application of supervisory control will

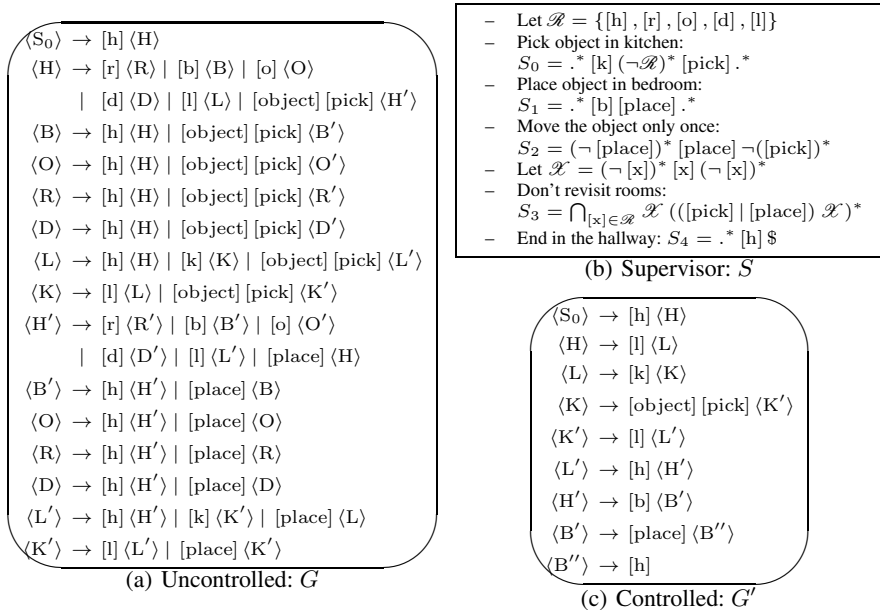
permit only those transitions of the model  $G$  which are also contained in specification  $S$ . We represent this as the intersection,

$$G' = G \cap S \quad (1)$$

Given that  $G$  is Context-Free and  $S$  is Regular, we use the algorithm defined in [13, p135] to produce Context-Free  $G'$ , ensuring that we can efficiently execute the policy given by  $G'$ . This algorithm operates on a Context-Free language model for system  $G$  and a Regular language specification for correct operation  $S$  with the assumption that we can *block* any undesirable transitions in  $G$ . The corrected system language, then, is  $G' = G \cap S$ , where  $G'$  is also Context-Free. We note in addition that to prune non-adjacent regions permitted by Transform 2 in Algorithm 2, we apply this same language intersection operation.

We use supervisory control of the grammar in Fig. 4(a) to perform the desired mobile manipulation task. To instruct the robot to bring an object from the kitchen to the human in the bedroom, we construct our supervisor according to the regular expressions in Fig. 4(b). Thus, our controlled system is,

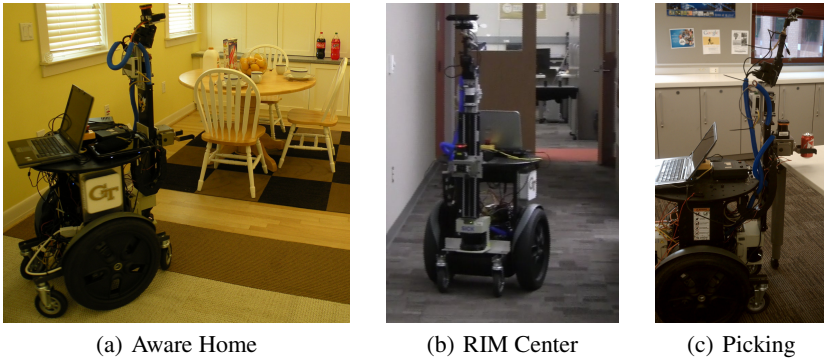
$$G' = G \cap \bigcap_{i=0}^4 S_i = [h] [l] [k] [object] [pick] [l] [h] [b] [place] [h] \quad (2)$$



**Fig. 4** Grammars for the Uncontrolled and Controlled mobile manipulator in the Aware Home

## 5 Experiments

We implemented this approach on a Segway RMP-200 mobile platform as shown in Fig. 5. This platform is equipped with an ASUS Xtion PRO LIVE camera, providing RGBD information for plane and surface extraction and with a UTM-30LX Hokuyo laser used to label the spatial regions as Gaussian models. It includes a Schunk parallel jaw gripper to manipulate objects. We conducted the experiments in the Georgia Tech Aware Home [15] and RIM center.

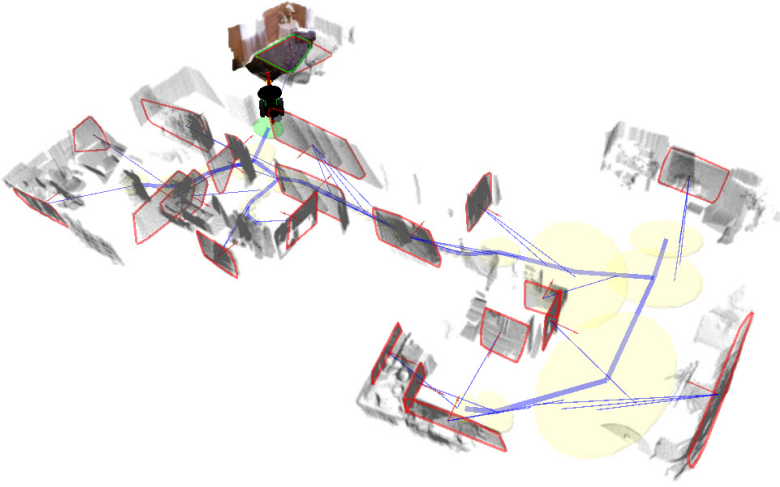


**Fig. 5** Segway RMP-200 mobile platform in the Georgia Tech Aware, the RIM Center, and picking a soda can

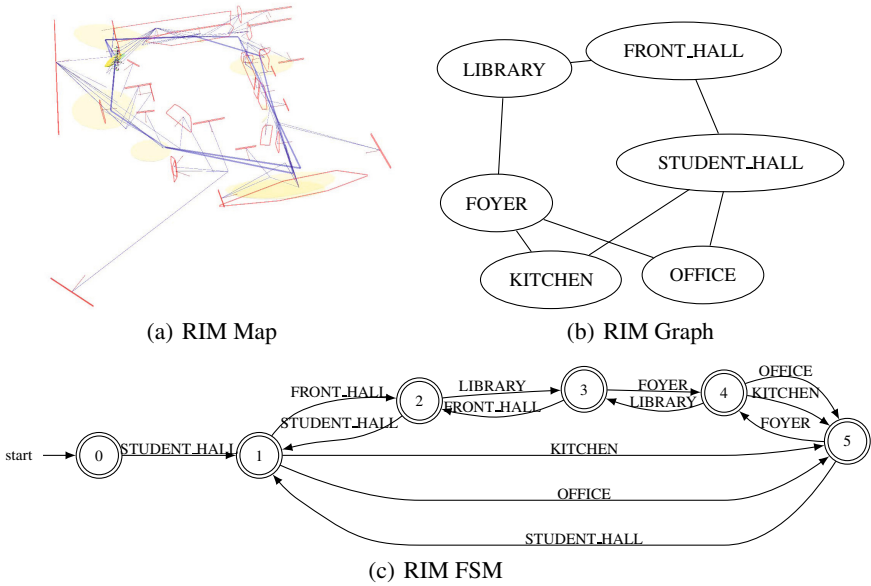
For both of the home and office environments, we first drove the robot through each area collecting 3D point clouds, laser, and odometry. Our mapper extracts planes and surfaces in the environment, building the map and localizing the robot. During the navigation, the robot partitions the environment into Gaussian regions. This produces the Gaussian map in Fig. 6. Then, we annotate the Gaussian regions of the map with semantic labels. The result is a graph, shown previously for the Aware Home in Fig. 2(a) and also for the RIM center in Fig. 7. This resulting map is suitable for both human interpretation and automatic symbol manipulation.

Next, we apply the method described in Sect. 4 to generate the symbolic model for the robot in each of the environments. For the Aware home, this model is given in Fig. 4(a), and for the RIM center in Fig. 7. For the Aware Home, we asked the robot to perform the following task, *Collect a soda from the kitchen and bring it to the bedroom*, expressed as the specification in Fig. 4(b). For the RIM Center, we apply a similar supervisor in Fig. 8(b) to collect a soda from kitchen and bring it to library.

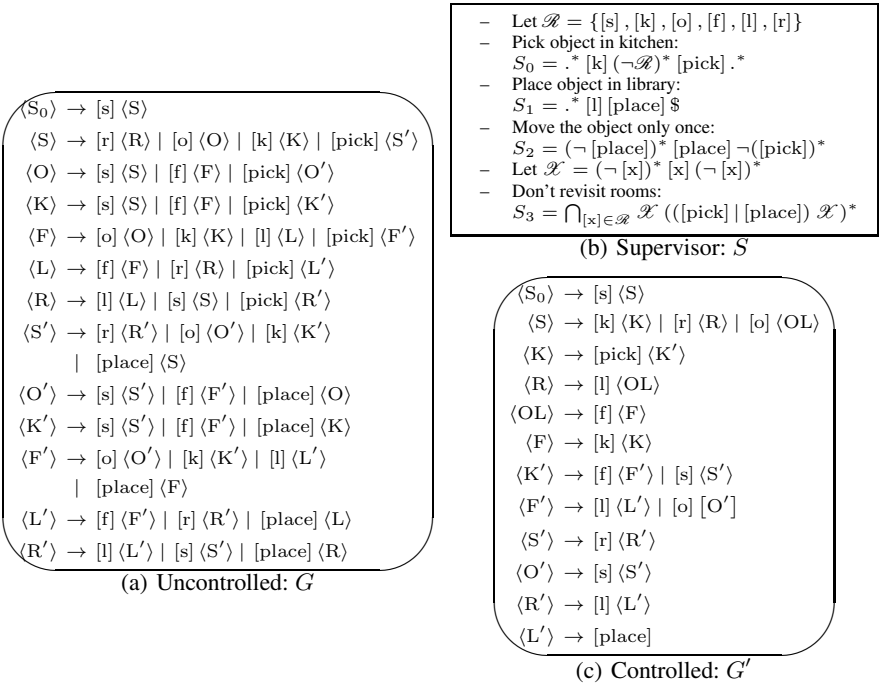
The policy for the task in the RIM environment, Fig. 8(c), is more complicated than for the Aware Home, Fig. 4(c). This is because the RIM map contains multiple



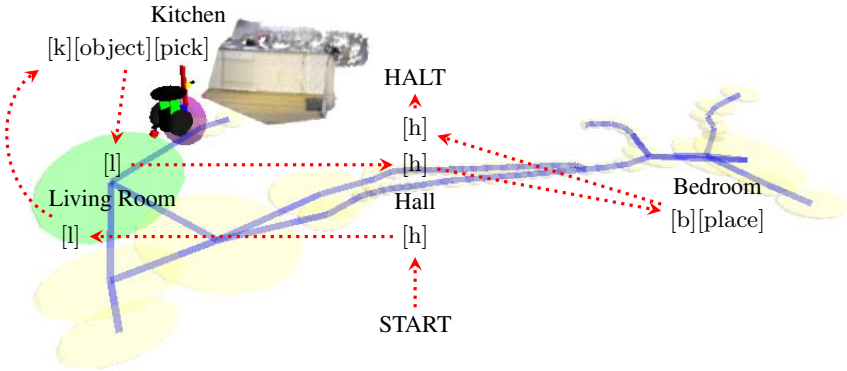
**Fig. 6** Generated Semantic Maps for the Aware Home. In the map, black shows 3D robot model, gray shows point clouds, yellow shows connected Gaussian regions (blue edges), and red shows the surfaces.



**Fig. 7** Generated Semantic map of Georgia Tech RIM Center and the equivalent graph and Finite Automata forms



**Fig. 8** Grammars for the Uncontrolled and Controlled mobile manipulator in the RIM Center. Notice how the policy captures all possible paths through the environment that satisfy the specification.



**Fig. 9** Path of the robot following controller in Fig. 4(c) and (2), shown as robot enters the living (green oval). Solid blue lines show the map connections between rooms, and dotted red lines show the robot path.

paths between all rooms. Thus, all these possible paths are captured in the control policy grammar. The result is the nine strings represented by the following regular expression,

$$G' = (k|rlfk|olfk) [\text{pick}] (fl|fosrl|srl) [\text{place}] \quad (3)$$

These generated policies direct the robot along the path to complete the specified task. For the Aware Home, the robot fetches the object from the kitchen and delivers it to the bedroom, illustrated in Fig. 9. This figure shows the path of the robot, both as a trajectory through the map and as the sequence of language symbols.

## 6 Discussion

In this approach, we combine a Semantic Map and a Motion Grammar using the Motion Grammar Calculus (MGC). This ensures the validity of our final system model because each transform of the MGC preserves *completeness* of the model. Then, applying a supervisory controller guarantees that the final policy is *correct* with regard to the specification. Thus, the overall approach is *correct-by-construction* in the sense that the final system model is guaranteed by the MGC to simulate our initial system, and the resultant policy satisfies the supervisory control specification.

The defining characteristic of this method is the uniform representation of the set of all robot paths as a language with an explicit grammar. This representation allows iterative development of the grammatical control policy by the progressive application of MGC transformations and supervisory control specifications. At each step of this derivation, the mechanical application of the MGC transforms and supervisory control ensures that we maintain a valid model of the system. Furthermore, because the policy for each task is itself a grammar, we can compose multiple individual task policies to produce a system to perform each of those tasks, all within the same grammatical framework. We expect these capabilities for incremental design and policy composition to be useful as we extend our work to multiple tasks and more complicated systems with larger grammars.

While search-based motion planning could perform some of the tasks in this paper, there are certain advantages given by our linguistic formulation and use of supervisory control for policy generation. Random-sampling planners such as RRTs and PRMs assume a continuous search space, while our application domain includes discrete features for detecting and manipulating objects. General search based planning assumes an explicit goal state and produces a *plan* to reach that state. In contrast, the linguistic approach considers the set of acceptable *paths* and produces a *policy* to stay within that set of paths.

## 7 Conclusions and Future Work

In this work, we address two significant challenges faced by robot mapping and hybrid controls. Robot mapping produces precise models of the environment, but gives no direct link to robot action. Formal hybrid control models are precise,



verifiable, and efficient representations of robot action, but developing these models for large and complicated systems is a tedious task. The linguistic composition demonstrated in this paper eases the challenges posed by each of these separate approaches. Through the automatic, symbolic composition of a map and base hybrid model, we produce a verifiable and executable model of the whole *robotic system*.

We will continue this work with in several ways. First, we will extend our implementation of this method to a variety of mobile manipulation tasks. Next, to provide a natural human interface for the mobile manipulation, we will compose multiple task policies with a grammar for simple human utterances. Finally, to increase the flexibility of this approach, we will extend the offline composition of maps and grammars to online composition as the semantic map is acquired.

## References

- [1] Aho, A., Lam, M., Sethi, R., Ullman, J.: *Compilers: Principles, Techniques, & Tools*, 2nd edn. Pearson (2007)
- [2] Alur, R., Courcoubetis, C., Henzinger, T., Ho, P.: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In: Grossman, R.L., Ravn, A.P., Rischel, H., Nerode, A. (eds.) *HS 1991 and HS 1992. LNCS*, vol. 736, pp. 209–229. Springer, Heidelberg (1993)
- [3] Baier, C., Katoen, J.P., et al.: *Principles of Model Checking*. MIT Press, Cambridge (2008)
- [4] Belta, C., Kumar, V.: Abstraction and control for groups of robots. *IEEE Transactions on Robotics* 20(5), 865–875 (2004)
- [5] Cassandras, C.G., Lafortune, S.: *Introduction to Discrete-Event Systems*, 2nd edn. Springer (2008)
- [6] Dantam, N., Stilman, M.: The motion grammar: Linguistic planning and control. In: *Robotics: Science and Systems* (2011)
- [7] Dantam, N., Stilman, M.: Deriving models of context-free hybrid systems. In: *American Controls Conference*. IEEE (2012)
- [8] Dantam, N., Kolhe, P., Stilman, M.: The motion grammar for physical human-robot games. In: *IEEE Intl. Conf. on Robotics and Automation*. IEEE (2011)
- [9] Folkesson, J., Christensen, H.: Graphical SLAM - a self-correcting map. In: *Intl. Conf. on Robotics and Automation* (2004)
- [10] Gibson, J.J.: *The senses considered as perceptual systems*. Houghton Mifflin, Boston (1966)
- [11] Henzinger, T.A.: The theory of hybrid automata. In: *Logic in Computer Science*, pp. 278–292. IEEE (1996)
- [12] Holzmann, G.J.: *The Spin Model Checker*. Addison-Wesley, Boston (2004)
- [13] Hopcroft, J.E., Ullman, J.D.: *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading (1979)
- [14] Hristu-Varsakelis, D., Levine, W.S.: *Handbook of Networked and Embedded Control Systems*. Birkhäuser, Basel (2005) ISBN 0817632395
- [15] Kientz, J.A., Patel, S.N., Jones, B., Price, E., Mynatt, E.D., Abowd, G.D.: The Georgia Tech aware home. In: *CHI 2008 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2008*. ACM, New York (2008)

- [16] Koščeká, J., Christensen, H.I., Bajcsy, R.: Experiments in behavior composition. *Robotics and Autonomous Systems* 19(3), 287–298 (1997)
- [17] Kress-Gazit, H., Fainekos, G.E., Pappas, G.J.: Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics* 25(6), 1370–1381 (2009)
- [18] Lygeros, J., Johansson, K.H., Simic, S.N., Zhang, J., Sastry, S.S.: Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control* 48(1), 2–17 (2003)
- [19] Nieto-Granda, C., Rogers III, J.G., Trevor, A.J.B., Christensen, H.I.: Semantic map partitioning in indoor environments using regional analysis. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Taiwan*, pp. 1451–1456. IEEE (October 2010)
- [20] O’Callaghan, S., Ramos, F.T., Durrant-Whyte, H.F.: Contextual occupancy maps using gaussian processes. In: *IEEE Intl. Conf. on Robotics and Automation*, pp. 1054–1060. IEEE (2009)
- [21] Ramadge, P.J., Wonham, W.M.: Supervisory control of a class of discrete event processes. *Analysis and Optimization of Systems* 25(1), 206–230 (1987)
- [22] Şahin, E., Çakmak, M., Doğar, M.R., Uğur, E., Üçoluk, G.: To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior* 15(4), 447–472 (2007)
- [23] Smith, R., Cheeseman, P.: On the representation and estimation of spatial uncertainty. *Intl. Journal of Robotics Research* 5(4), 56–68 (1987)
- [24] Topp, E.A., Christensen, H.I.: Detecting region transitions for human-augmented mapping. *IEEE Transactions on Robotics*, 1–5 (2010) ISSN 1552-3098
- [25] Topp, E.A., Christensen, H.I.: Topological modelling for human augmented mapping. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pp. 2257–2263 (October 2006)
- [26] Trevor, A.J.B., Rogers III, J.G., Nieto-Granda, C., Christensen, H.I.: Tables, counters, and shelves: Semantic mapping of surfaces in 3d. In: *IROS Workshop on Semantic Mapping and Autonomous Knowledge Acquisition* (2010)
- [27] Trevor, A.J.B., Rogers III, J.G., Christensen, H.I.: Planar surface slam with 3d and 2d sensors. In: *IEEE Intl. Conf. on Robotics and Automation* (2012)

# Part XI: ISER Session Summary on “Field Robotics”

Stephen Nuske

Carnegie Mellon University

## Session Summary

Piecing together a common thread between the papers in the field robotics session lead to one conclusion: information gathering. Four out of the five papers were focused specifically on collecting information with outdoor mobile sensors and the fifth paper was to do with using an underused information source: acoustic sensors. It goes without saying that there is immense value in gathering information and in field robotics domains the value is magnified because of the difficulty, expense and potential danger of collecting information manually.

Patrick Plonski et al. first up showed a mobile robot collecting information on the spatial distribution of solar energy in an environment for the purpose of planning efficient routes for solar-powered robots. Routes may be longer in distance but more optimal for an overall energy budget because they lead through direct sunlight.

Aravindhnan Krishnan et al. then presented work processing vast aerial datasets of tectonic fault lines for detecting and understanding activity resulting from earthquakes.

Qi Wang et al. demonstrated how to collect harvest yield predictions in apple orchards. Yield measurements that are accurate and dense collected in advance of harvest give rise to intelligent, efficient and precise orchard management.

Young-Ho Kim et al. then focused on the general problem of information gathering and studied the use of Gaussian processes with different approaches to measure the variance demonstrated more accuracy whilst maintaining consistency.

Tews and Dunbabin in the final paper of the session showed that acoustic information has great utility in certain situations. Here, an approach is shown to conceal movement of a robot by waiting for periods where ambient noise exceeds a level to conceal the robot’s own motion.

# Energy-Efficient Path Planning for Solar-Powered Mobile Robots

Patrick A. Plonski, Pratap Tokekar, and Volkan Isler

**Abstract.** We explore the problem of energy-efficient, time-constrained path planning of a solar powered robot embedded in a terrestrial environment. Because of the effects of changing weather conditions, as well as sensing concerns in complex environments, a new method for solar power prediction is desired. We present a method that uses Gaussian Process regression to build a solar map in a data-driven fashion. With this map, we perform energy-optimal path planning using a dynamic programming algorithm. We validate our map construction and path planning algorithms with outdoor experiments, and perform simulations on our solar maps to determine under which conditions the weight of added solar panels is worthwhile for a mobile robot.

## 1 Introduction

Mobile robots have the potential to perform many critical outdoor tasks but their potential for long-term deployment is limited due to energy concerns. A possible method to increase the battery life of robots is by harvesting energy from the environment, e.g. with photovoltaic solar panels. Solar harvesting has proven to be useful in marine and extra-terrestrial robotics applications [11, 1] which take place in open space. However, in applications where the robot must operate in complex environments, such as urban search and environmental monitoring, the utility of solar harvesting is not obvious. In this work we focus on extending the battery life of mobile robots using solar panels in such settings.

We study techniques for energy-minimizing path planning for a mobile robot with a photovoltaic panel that uses recent measurements of solar intensity as its only source of information about future solar power. This is an interesting problem

---

Patrick A. Plonski · Pratap Tokekar · Volkan Isler  
Department of Computer Science and Engineering, University of Minnesota,  
200 Union Street SE, Minneapolis MN 55455, USA  
e-mail: {plonski, tokekar, isler}@cs.umn.edu

because there are many applications where mobile robots do not necessarily have the sensors or computing power to estimate solar maps using sophisticated techniques such as raytracing on 3d models of the environment. However, energy-efficient paths are still desired. Intuitively, it seems feasible for a good solar map of the environment to be built if the robot is in the field long enough. We provide experimental evidence to support this intuition.

To accomplish energy-efficient path planning, we first build a map of how much solar power the robot is likely to get in its operating environment (Section 2). Next we show how the robot's energy consumption can be modeled and how we can compute energy efficient paths given a solar map (Section 3). We present results from experiments that demonstrate the utility of our techniques (Section 4). We also present simulation results on our solar maps to demonstrate the utility of added solar panels on a robotic platform (Section 5).

## ***1.1 Related Work***

Energy efficient planning for mobile robots has received increased attention recently. Mei [8] studied the problem of modeling the power consumption of motion, sensing, communication and embedded hardware for commercially available robots. These power models are then used to compare various strategies for high-level tasks such as coverage, exploration and networking between robots, and increase the lifetime of the system.

Motion is a major source of power consumption for typical robots. Tokekar et al. [15], Wang et al. [17], and Kim and Kim [6] have studied the problem of minimizing the energy consumption by optimizing the velocity profiles for a given path. Sun and Reif [13] studied the problem of finding energy optimal paths between two points on terrains where the cost depends on friction and gravity and is thus direction dependent. They present an approximation algorithm for finding the minimum energy path, but do not optimize the velocity profile along the path. Liu and Sun [7] recently studied the problem of computing energy-efficient paths and trajectory profiles by optimizing the parameters of Bezier curves using an energy-based heuristic. However, the presented method is not guaranteed to minimize energy and the general problem of simultaneously optimizing the path and velocity for given start and goal pose remains unsolved.

Energy efficient motion planning in the context of applications such as coverage and data muling is a subject of recent study. Derenick et al. [2] studied the problem of maintaining persistent coverage using a network of robots by deriving control laws that allow robots with depleted batteries to reach corresponding access points. Similarly, Jensen et al. [5] presented strategies for reconfiguring robot formations for patrolling application.

Sugihara and Gupta [12] presented path planning algorithms for a data muling system for optimizing the trade-off between the energy consumption of the sensors and latency of the data carried by the robot. Tekdas et al. [14] studied the problem of finding time-efficient trajectories for a mobile robot downloading data from a set

of wireless nodes, and by setting the parameters proportional to energy cost their approximation algorithm can minimize energy instead of time. In these works, the energy consumption of the robot is not considered. Here we present energy harvesting and path planning techniques that can potentially be useful for such applications.

The aforementioned works have not considered energy harvesting from the environment, and solar-aware path planning has received limited attention. In extraterrestrial applications and some environments on earth (e.g. in Antarctica [10]) collected solar energy can be treated as mostly independent of the path chosen. The TEMPEST mission-level path planner [16] uses ephemeris software to determine the position of the sun and then performs raytracing on known nearby terrain to build a solar map that is used to estimate the energy cost of paths. This is feasible when nearby terrain is known or when it can be accurately detected, but many otherwise feasible platforms for long-term environmental monitoring lack the necessary sensors to do this. In this paper we focus on predicting solar power in complex environments using only the robot's previously recorded position estimates and solar power measurements.

## 1.2 Problem Statement

Our problem statement is as follows: Suppose we have a mobile solar-powered robot that has been performing a task while also logging the power received from an on-board solar array. Each solar measurement is associated with an estimated robot position. Suppose the robot is required to perform a new task that requires it to reach a goal position within some time limit. How can the robot plan the path that minimizes its net energy consumption?

## 2 Solar Modeling

In this section we introduce the method we use to predict how much solar power the robot will receive at a given position. Before we present the details of our Gaussian Process (GP) regression, we first cover the basics of predicting electrical output from a photovoltaic panel.

### 2.1 Basics of Solar Power Prediction

The amount of current  $I$  a solar cell will output when it is fixed to a particular voltage  $V$  is the solution to the equation

$$I = I_L - I_s(e^{(V+IR_s)/V_T} - 1) - \frac{V + IR_s}{R_{SH}}$$

where  $I_s$  is the reverse saturation current of the diode and  $V_T = \frac{k*T}{q}$  which is known as the thermal voltage.  $I_L$  is proportional to the number of photons that impact the solar cell, and therefore so is  $I$ .  $I$  decreases with higher voltage, but the effect isn't

pronounced until the diode knee voltage is reached at around 0.5 volts for a silicon cell. The knee voltage increases with decreased temperature, but in general the voltage limit varies much less than the current.

Because the voltage of an individual cell is low, cells are usually connected in one or more strings such that each string is electrically in series. These strings have the property that the amount of current output is limited by the *weakest* cell in the string (ignoring the effect of bypass diodes). The weakest cell could be the cell with the smallest dot product between its normal vector and the sun angle vector, or it could be a cell which happens to be in a shadow. This response to partial shading of the array causes the correct solar map to have sharp edges between sun and shade.

Sunlight reaches a solar panel in three different ways[4]: If it comes directly from exactly the part of the sky that contains the sun, it is called direct insolation. If it comes from any other part of the sky, it is called diffuse insolation. Finally, if it comes from anywhere else (i.e. from terrain or objects), it is called reflected insolation. Reflected insolation is most relevant when a solar panel is tilted towards a reflective surface (such as snow), or near a reflective building. On a sunny day direct insolation is high and diffuse insolation is low whereas on a cloudy day direct insolation is low and diffuse insolation is high (and total insolation is much lower than on a sunny day). If a cell has no line of sight to the sun it is in a shadow, and direct insolation drops to zero. However, for diffuse insolation to drop to zero the entire sky must be blocked. Therefore we can expect shadows and therefore the correct solar map to be much sharper on a sunny day than on a cloudy day.

It is challenging to detect the environment and perform raytracing for these three types of insolation so we sidestep and instead construct our solar map using regression from prior measurements of solar power associated with positions.

## 2.2 *Gaussian Process Regression*

A Gaussian Process (GP) is defined as a set of random variables such that any subset of the random variables has a joint Gaussian distribution [9]. GP regression is a general regression technique used to predict the most likely value of a function at any point given measured values of the function at some other points, without assuming an explicit parametric model for the function. GP regression, however, requires a suitable covariance function to model the joint Gaussian distribution for points. For more details on GP regression in general see [9].

In our application we associate each measurement of solar power with a position and use GP regression to predict the distribution of solar power at any desired position. When all of the solar cells are horizontal, or if they are otherwise suitably symmetric, the rotation of the robot can be ignored in these position measurements. This makes the solar map easier to learn by eliminating a dimension along which solar power can vary. In this paper we neglect the solar map's time dependence from the changing position of the sun. This is justified when the robot stays in the same environment each day, and can therefore build a separate solar map for various discrete time segments.

In Section 4.4 we present more details of our particular implementation of GP regression, and we empirically compare the performance of different covariance functions.

### 3 Path Planning

In this section we show how we use a solar map to plan the path that will reach the goal within the time limit while consuming the least amount of energy overall.

Our robot is differential-driven, so it can turn in place, and turning is a relatively expensive operation. We empirically determine in Section 4.3 that for our robot the energy consumption of a path with a certain top speed is well represented as a short initial spike during acceleration, and then a steady cost per meter traveled. The planned path as time-stamped waypoints with straight line segments connecting them, each line segment traversed at a constant speed with instantaneous speed changes between line segments. We model the energy sent to the motors as the following: At any particular speed, there is a constant cost per meter traveled  $C_s$ , a constant cost per radian rotated  $C_r$ , and an initial acceleration cost  $C_a$ . When transitioning from a non-zero speed, the acceleration cost is the  $C_a$  for the new speed minus the  $C_a$  for the old speed, but with a minimum cost of 0. This makes sense if we assume that acceleration cost is proportional to kinetic energy. We can mathematically state the cost of traversing line segment  $l_i$ :

$$cost_i = C_s(speed_i)|l_i| + C_r(speed_i)|\theta_i - \theta_{i-1}| + \max(C_a(speed_i) - C_a(speed_{i-1}), 0)$$

The cost constants as functions of speed are specific to the robot and the terrain. The terrain where our experiments were conducted was flat and uniform, so in this work we do not consider changes in elevation, friction, or rolling resistance.

The total cost of a path is given by the sum over the path  $\sum_{i=0}^{n-1} cost_i$  minus the expected amount of solar energy collected while traversing the path. An idle power draw (constant) can be subtracted from the solar power; we do not consider idle power draw because our focus is on path planning and idle power does not affect the optimal path to reach the target in the time scales we consider.

#### The Algorithm

The expected value for any particular point in our solar map can be determined in closed form, however there is no convenient closed form model for the entire map as a whole; that is, there is no general geometric model we can use to represent our environment. Therefore some amount of discretization of the solar map is necessary for us to do planning. It is possible in this domain to plan on a set of sampled actions or path shapes (e.g. with an sampling based planner) but since the state space is relatively small we use a complete grid. We then perform dynamic programming to compute the optimal solution for a given resolution. We discretize both space and time, and we also have a dimension in the dynamic programming table for heading and a dimension for whether the robot is moving or the robot is waiting, to account for the cost to rotate and the cost for initial acceleration. In this way we ensure that



the output path is always optimal in its resolution, according to our power to drive model. The trajectories generated by our algorithm move at a constant speed when they are on Manhattan edges and a faster constant speed when they are on diagonal edges; traversing to any neighboring state takes the same amount of time.

We observe from the output of this algorithm that optimal trajectories consist of either continuous movement, continuous movement with a wait at the beginning or the end, or continuous movement broken up by a wait in the middle. As more time is allowed the optimal path transitions between those three types: at first there is no time to wait anywhere, then there is time to wait but not enough to compensate for the energy loss from having to re-accelerate, and then finally there is enough time to wait somewhere in the middle for long enough to recoup the extra acceleration cost and possibly enough time to allow deviation from a shortest path. See Figure 3a for examples of planned paths output by our algorithm.

## 4 Field Experiments

We performed three sets of experiments in the environment shown in Figure 1b: we calibrated our power to drive parameters, we measured solar panel current along paths and used this to construct solar maps using different covariance functions, and we executed energy-minimal paths that were planned on these maps.

### 4.1 System Description

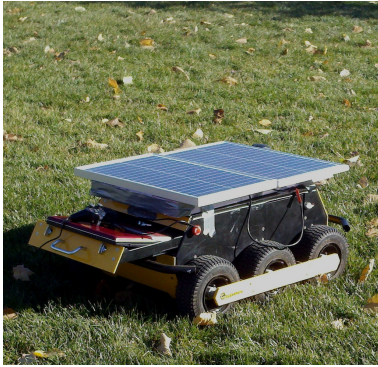
The chassis of our system was a Husky A100, built by Clearpath Robotics<sup>1</sup>. The A100 is a six wheel, two motor, differential drive machine. The datasheet mass is 35 kg, the maximum payload is 40 kg, and the dimensions are 0.860 meters long by 0.605 meters wide by 0.350 meters tall. In its experimental configuration the A100 was powered by a single lead-acid battery that was nominally 12v and 21 amp hours. See Figure 1a for a photo of the A100 during one of our experiments.

The solar panels used by our system were two SPM020Ps from Solartech Power<sup>2</sup>. The SPM020P supplies 20w at the optimal voltage of 17.2v under standard test conditions of 1000 w/m<sup>2</sup> insolation and a temperature of 25°C. The panel is wired as a single series string with 36 cells in it. The dimensions are 560x360x18(mm), and each panel nominally weighs 2.5kg.

We placed the panels horizontally on the robot for ease of mounting, for quality in overcast conditions, and to eliminate the dimension of panel rotation in the solar map built. Both panels were connected in parallel with the battery; therefore solar panel current was proportional to solar power. Battery voltage and motor current measurements were provided by the A100, and current from the panel to the battery was measured with a hall-effect current sensor.

<sup>1</sup> <http://www.clearpathrobotics.com/>

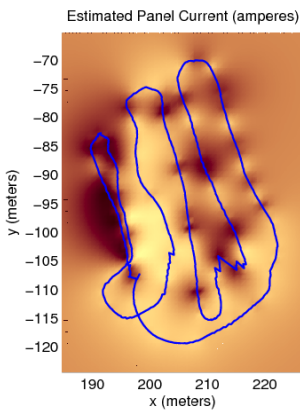
<sup>2</sup> <http://www.solartechpower.com/>



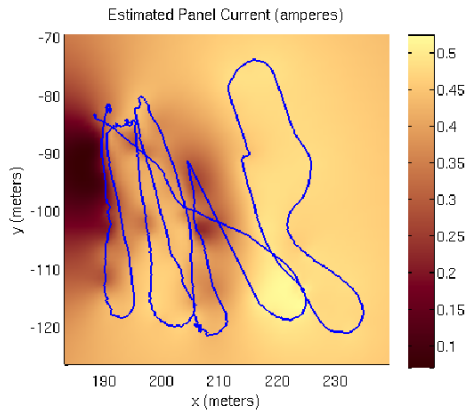
(a) Clearpath Husky A100



(b) Test Site



(c) Sunny Solar Map



(d) Cloudy Solar Map

**Fig. 1** Our configuration of the A100 (a), top-down view of the test site (b), solar map constructed for 13:42 on November 18, 2011 (c) (this was a sunny day), and solar map constructed for 11:22 on September 16, 2011 (d) (this was a cloudy day). Both solar maps are overlaid with their source paths. The cloudy map was built by sampling with only a single solar panel.

Localization of the robot was achieved by using an EKF to fuse GPS measurements with wheel-encoder propagation.

## 4.2 Terrain Description

We performed our experiments in the field next to the McNamara Alumni Center, on the Minneapolis campus of the University of Minnesota (see Figure 1b). The field is roughly 40 meters by 30 meters and it is relatively flat, with uniform short grass. Other than a few poles the only objects that occlude the sun are scattered

trees. While our calculated power to drive parameters and solar map parameters are likely to change in other environments, the methodology we present here to obtain those parameters remains the same.

We performed our experiments on dry days when there was no snow on the ground. We would expect power to drive to significantly change in wet weather or if there is accumulated snow. All solar parameters except the chosen covariance function were re-learned for each new solar map; this was necessary to account for short term changes from the varying position of the sun, medium term changes from varying weather, and long term changes. One of these long term changes was a seasonal change in solar power that occurred as the leaves fell off the trees as summer turned to winter.

### 4.3 Power to Drive Experiments

We controlled the forward movement of the A100 by directly setting the motor voltage. We found that this method required less energy than using a closed loop PID speed controller. For a particular motor voltage and on particular terrain, the A100 travels at a particular steady-state speed and consumes a steady amount of energy per unit distance traveled, after a brief acceleration period. To characterize the steady-state cost and acceleration cost we drove straight at a variety of commanded motor voltages and fit a line to the plot of cumulative cost vs. distance for each voltage. The slope of the line determined the steady state cost and the intercept determined the acceleration cost. Then we performed linear regression on the steady state costs as functions of speed and quadratic regression on the acceleration costs as functions of top speed, and ended up with the following equations for our parameters  $C_s$  and  $C_a$  (see Figure 2):

$$C_s = (-17.6624 * speed + 139.4576) \text{ Joules per meter}$$

$$C_a = (321.0671 * speed^2 - 285.3912 * speed + 154.9553) \text{ Joules to accelerate}$$

Then to characterize turning cost we commanded a tight left turn and tight right turn, and examined the steady state energy per radian.

$$C_r = 406.5963 \text{ Joules per radian}$$

### 4.4 Solar Map Construction

The input for the solar map is a long path with noisy measurements of solar current taken at 20 Hz, each measurement associated with a position on the path. This accumulates to a very large number of measurements if the robot is embedded in the environment for a long time. As GP regression relies on matrix multiplication of all training points, using all measurements as individual training points becomes infeasible. Fortunately, since we only care about associating solar current to  $x - y$  position we can discard information about rotation and time and combine measurements with similar  $x - y$  position. In this way the number of measurements considered by the GP regression is bounded by the size of the environment rather than the

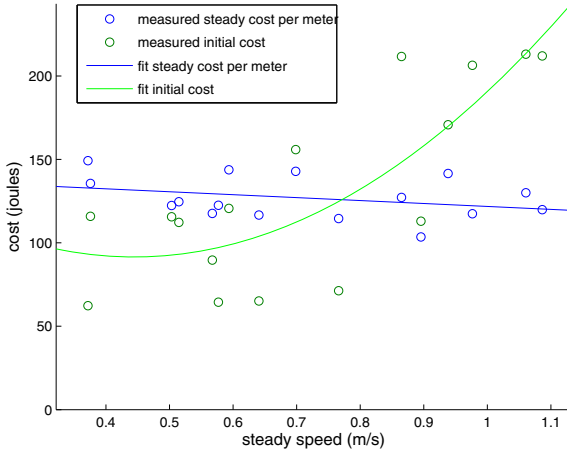


Fig. 2 Power To Drive Test Results

length of time the robot is collecting data. Also it is valuable for optimizing the GP hyperparameters for measured positions to be weighted equally instead of weighted in proportion to the amount of time the robot has spent there.

In our implementation we placed in a bucket all measurements that were within 0.3 meters of the first measurement and then removed them from the list, and repeated this process until every measurement was in a bucket. The bucket’s position was set as the centroid of the positions of the measurements in it, and its value was set as the mean of the values of the measurements in it. We calculated the variance of each bucket from the variance of the measurements in the bucket, treating the bucket solar current as an average of uncorrelated random variables. Then for the regression we treated the noise variance as equal to the average of the variances of the buckets. This was again to induce balanced weighting of different areas; if the robot had waited 20 minutes at the same position we did not want the bucket containing that position to be significantly more valuable than nearby buckets because still only a small portion of the possible points that could go into that bucket would have been explored. The prior mean and prior variance were computed from the mean and variance of the set of buckets.

To perform GP regression we need a covariance function. For this we considered different versions of the Matérn covariance function (detailed in [9]). The Matérn class of covariance functions is given by:

$$k(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu}r}{\ell} \right)^\nu K_\nu \left( \frac{\sqrt{2\nu}r}{\ell} \right)$$

where  $\nu$  is a positive parameter that affects the smoothness of the process,  $\ell$  is the positive length parameter, and  $K_\nu$  is a modified Bessel function. If  $\nu$  is 1/2 the function becomes the exponential covariance function, and as  $\nu \rightarrow \infty$  the function

becomes the squared exponential covariance function. Other than the exponential and the squared exponential, the most commonly used Matérn covariance functions are where  $\nu = 3/2$  and  $\nu = 5/2$ , so those are the covariance functions we tested in addition to the exponential and squared exponential.

To optimize the Matérn function's length hyperparameter we performed numerical gradient-descent searches maximizing the likelihood of the observed values given the covariance function. We compared the likelihoods of the different Matérn functions on various data sets we collected and we found that  $\nu = 1/2$  was the most likely on two out of three of the cloudy days tested, and five out of eleven of the sunny days tested, with  $\nu = 3/2$  the most likely on the other days. However, even on the days where it was the most likely, the solar maps constructed using  $\nu = 3/2$  had overshooting at the sharp boundaries between sun and shade. This overshooting made it so that the positions with the most predicted solar power were close to boundaries, and therefore planned energy-minimal paths were pulled towards boundaries. As a real system always has some localization error, a better strategy is to stay away from shadows if possible. This is the behavior that results when we use  $\nu = 1/2$  in our regression, so that is what we did even though it was often less likely given the data and our GP assumption.

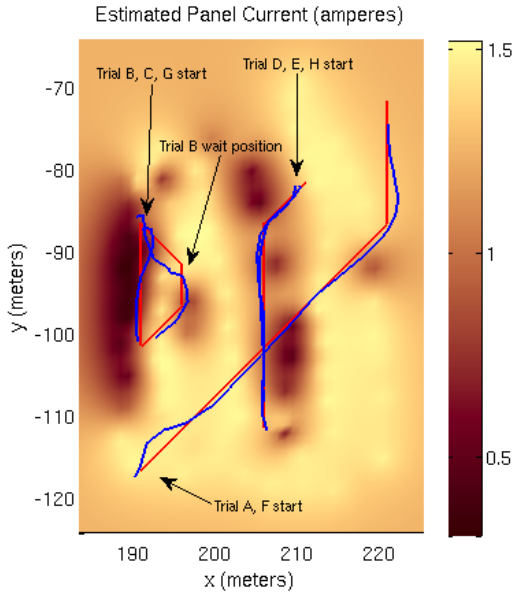
Holding  $\nu = 1/2$ , the most likely length varied between 2.05 meters and 12.65 meters on sunny days, and between 3.68 meters and 18.67 meters on cloudy days. This difference is because diffuse insolation dominates over direct insolation on cloudy days, and diffuse insolation varies slower than direct with changing position.

## 4.5 Path Planning and Execution

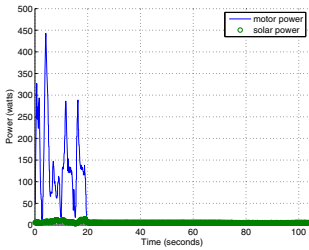
At 13:10 on February 18, 2012 we drove the A100 around the field in Figure 1b, optimized the length hyperparameter for that dataset with an exponential covariance function, used GP regression to build a solar map, planned paths with our planner detailed in Section 3, and then executed the paths. The A100 had some localization error even when GPS worked well, so a fairly low spatial resolution of 5m was used. Temporal resolution was set to 8 seconds. To calculate the expected solar current in a grid square the expected solar current was calculated on a higher resolution 1m grid and then downsampled. In addition to the planned solar-aware paths the A100 also executed shortest paths after we removed the solar panels (slightly decreasing the power to drive due to decreased weight) from the same start position to the same end position. These paths provide a comparison, allowing us to directly demonstrate the utility of the added panels. See Table 1 for summaries of the executed paths.

**Table 1** Path Execution Results

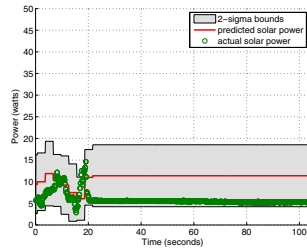
Solar Trial	Duration	Expected Solar	Actual Solar	Expected Cost	Actual Cost	Control Trial	Duration	Cost
A	401 s	7,025.5 J	6,974.1 J	577.16 J	744.6 J	F	45 s	6,295.4 J
B	400 s	6,606.6 J	6,828.6 J	-3,265.9 J	-3,256.7 J	G	19.1 s	2,888.1 J
C	104 s	1,148.3 J	611.26 J	879.99 J	2,253.4 J			
D	104 s	1,600.9 J	1,297.6 J	2,907.3 J	3,480.3 J	H	30.4 s	3,530.4 J
E	104 s	1,600.9 J	1,156.2 J	2,907.3 J	2,822.5 J			



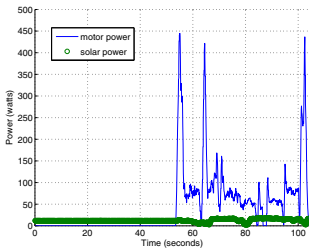
(a) Solar map at 13:10 on February 18, 2012, with planned paths in red and executed paths in blue



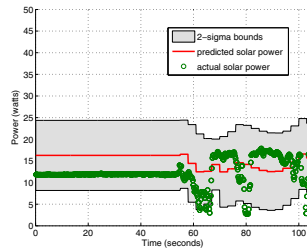
(b) Trial C Power



(c) Trial C Solar



(d) Trial D Power



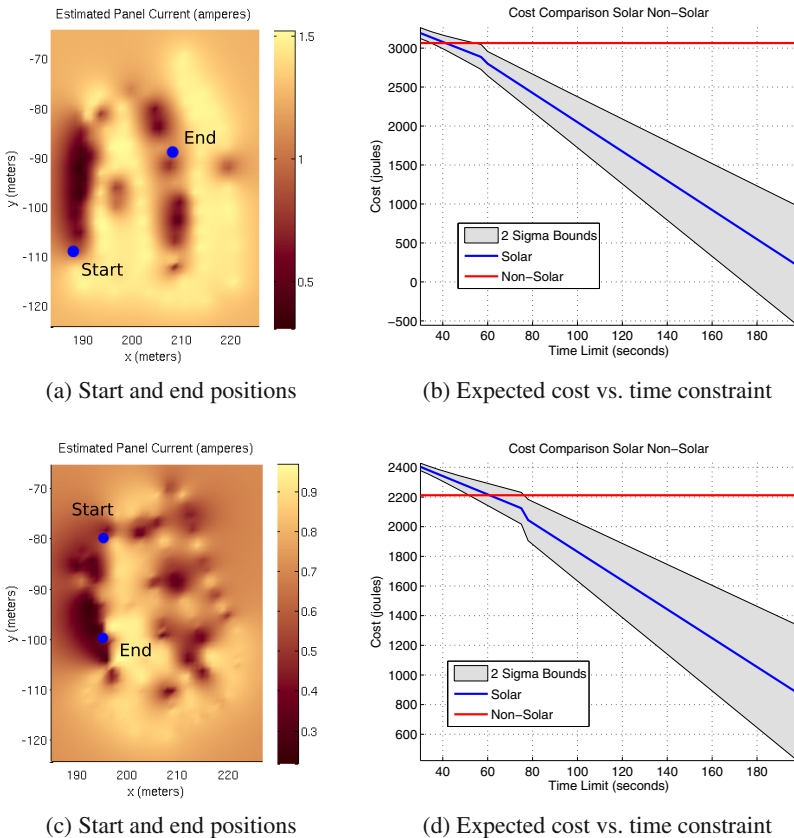
(e) Trial D Solar

**Fig. 3** Planned solar-aware paths and example trials. Note that in trial D the planner chose to wait at the beginning given the information it had but it turned out the position at the end of the path received more solar power.

### 5 Power Comparison

To further investigate the benefits gained from solar panels, we ran simulated comparisons between our solar powered robot using our path planner and our robot, stripped of its panels, driving straight towards the destination. We picked a start position and end position, planned the optimal solar-aware path for a range of time limits, and compared the cost to drive straight without a panel with the distribution of likely solar robot costs. For these simulations we did not consider localization errors, so we increased the resolution of our planning grid to 2 meters per square, 3 seconds per square. We intentionally chose start and end positions in the shade, to see how the system would perform under somewhat adverse conditions.

First we considered a robot traveling from the southwest part of the trees to the northeast part of the trees, at 13:10 on February 18 (the same day as our path



**Fig. 4** Simulations for 13:10 on 02-18-2012 (a and b) and 13:42 on 11-28-2011 (c and d). When not much time is allowed the weight of the solar panels ensures that the cost of carrying them is greater than the benefit of solar power, however when the robot is allowed to wait a while in the sun the benefit of panels can be large.



execution trials). For details of this simulation see Figures 4a and 4b. The start position was (188, -109) and the end position was (208, -89). At a speed of 1 m/s we expect the baseline path to consume 3,065.0 Joules. For the solar robot to be on average more energy efficient than the baseline it requires at least 42 seconds to execute its path. This is an overall speed of 0.6734 m/s. For the solar robot to be more energy efficient with 95% confidence, it requires at least 57 seconds which is an overall speed of 0.4962 m/s.

Second we considered a robot traveling south through the shade of the west line of trees, at 13:42 on November 28. For details of this simulation see Figures 4c and 4d. The start position was (195, -80) and the end position was (195, -100). At a speed of 1 m/s we expect the baseline path to consume 2,211.9 J. For the solar robot to be on average more efficient than the baseline it requires at least 63 seconds which is an overall speed of 0.3175 m/s. For the solar robot to be more efficient with 95% confidence it requires at least 78 seconds which is an overall speed of 0.2667 m/s.

## 6 Experimental Insights and Concluding Remarks

In our experiments, we observed that true solar energy collected during a trial was close to the expected solar energy obtained from GP regression. However, the predicted probability distributions did not necessarily resemble the true distributions. This is because the probability distribution of sunlight at a point is poorly modeled by a Gaussian distribution: on a sunny day the correct probability distribution of expected solar power at any given point is bimodal, with separate peaks of expected power for the case where the panel is in the sun and the case where it is in the shade. Since Gaussian models cannot capture this behavior well, it may not be best to optimize the covariance function hyperparameters for maximum likelihood. This is an issue we plan to investigate further.

On February 18 the system did not lose much accuracy by neglecting to consider the sun's movement, though the solar map was constructed for 13:10 and the last solar trial (trial E) began at 14:19. The impact of moving shadows may have been mitigated by the fact that shadows were sparse due to bare branches on the trees.

Our power to drive model was reasonably accurate. It tended to underestimate power to drive but not by much: on average it missed by 396.5 J, which was on average 11.2% off from the true value. It underestimated four times and overestimated once. This indicates that our learned parameters were correct and that the A100 waypoint navigation software was not performing too many corrective turns. To get the waypoint navigation software to this state we disallowed backtracking and instead counted the waypoint as reached whenever the plane perpendicular to the path was crossed. This had the effect of slightly decreasing solar prediction accuracy, but also significantly decreasing average power to drive for a trial.

Our path planner worked well at its resolution. If we move to higher resolution there is a danger of the following: the path planner chooses to wait in a position that has sun but due to localization error the A100 ends up waiting in the shade, and an expected good path becomes very bad. With our path planning there was very high



cost to deviate from a straight path: the cost of four  $45^\circ$  turns and at least 10 meters increased distance. Therefore if there is not much time the optimal path will choose to wait at the sunniest spot on the shortest path instead of deviating to a sunnier spot that is slightly off the path. It might be feasible to use something such as Field D\* [3] to plan smoother paths that vary only slightly from the shortest path.

Our simulation results show that with our platform and in the environment we tested, the addition of heavy commercial solar panels decreases cost on sunny days in November and February only if the average speed is not required to be greater than 0.6734 m/s for the trial in February or greater than 0.3175 m/s for the trial in November. These were both sunny days, but they were particularly challenging for sunny days: it was the dark part of the year, and the trials both started and ended in the shade. We would therefore expect the addition of solar panels to be feasible in many situations requiring higher average speeds.

In our future work, we will investigate the effect of the varying sun angle on our solar maps, as well as methods to use the known sun angle to improve our predictions. We also plan to further investigate methods of optimizing the hyperparameters, and methods to plan smoother paths on our solar map.

**Acknowledgements.** This material is based upon work supported by the National Science Foundation under grant numbers 1111638, 0916209, 0936710, and 0934327.

## References

1. Carsten, J., Rankin, A., Ferguson, D., Stentz, A.: Global path planning on board the mars exploration rovers. In: 2007 IEEE Aerospace Conference, pp. 1–11 (March 2007)
2. Derenick, J., Michael, N., Kumar, V.: Energy-aware coverage control with docking for robot teams. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3667–3672 (September 2011)
3. Ferguson, D., Stentz, A.: Using Interpolation to Improve Path Planning: The Field D \* Algorithm. *Journal of Field Robotics* 23(2), 79–101 (2006)
4. Goswami, D.Y., Kreith, F., Kreider, J.F.: *Principles of Solar Engineering*, 2nd edn. Taylor & Francis (1999)
5. Jensen, E., Franklin, M., Lahr, S., Gini, M.: Sustainable multi-robot patrol of an open polyline. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 4792–4797. IEEE (2011)
6. Kim, C., Kim, B.: Minimum-energy translational trajectory generation for differential-driven wheeled mobile robots. *Journal of Intelligent & Robotic Systems* 49(4), 367–383 (2007)
7. Liu, S., Sun, D.: Optimal motion planning of a mobile robot with minimum energy consumption. In: 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 43–48 (July 2011)
8. Mei, Y.: *Energy-Efficient Mobile Robots*. PhD thesis, Purdue University (2006)
9. Rasmussen, C., Williams, C.: *Gaussian processes in machine learning*. MIT Press (2006)
10. Ray, L., Lever, J., Streeter, A., Price, A.: Design and Power Management of a Solar-Powered Cool Robot for Polar Instrument Networks. *Journal of Field Robotics* 24(7), 581–599 (2007)

11. Sauze, C., Neal, M.: Long term power management in sailing robots. In: 2011 IEEE - OCEANS, Spain, pp. 1–8 (June 2011)
12. Sugihara, R., Gupta, R.: Optimizing energy-latency trade-off in sensor networks with controlled mobility. In: IEEE INFOCOM 2009, pp. 2566–2570. IEEE (2009)
13. Sun, Z., Reif, J.: On finding energy-minimizing paths on terrains. *IEEE Transactions on Robotics* 21(1), 102–114 (2005)
14. Tekdas, O., Bhadauria, D., Isler, V.: Efficient Data Collection from Wireless Nodes under the Two-Ring Communication Model. *The International Journal of Robotics Research* 31(6), 774–784 (2012)
15. Tokekar, P., Karnad, N., Isler, V.: Energy-optimal velocity profiles for car-like robots. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1457–1462. IEEE (2011)
16. Tompkins, P., Stentz, A., Wettergreen, D.: Mission-level path planning and re-planning for rover exploration. *Robotics and Autonomous Systems* 54(2), 174–183 (2006)
17. Wang, G., Irwin, M., Fu, H., Berman, P., Zhang, W., Porta, T.: Optimizing sensor movement planning for energy efficiency. *ACM Transactions on Sensor Networks (TOSN)* 7(4), 33 (2011)

# Change Detection Using Airborne LiDAR: Applications to Earthquakes

Aravindhnan K. Krishnan, Edwin Nissen, Srikanth Saripalli,  
Ramon Arrowsmith, and Alejandro Hinojosa-Corona

**Abstract.** We present a method for determining 3-dimensional, local ground displacements caused by an earthquake. The technique requires pre- and post-earthquake point cloud datasets, such as those collected using airborne Light Detection and Ranging (Lidar). This problem is formulated as a point cloud registration problem in which the full point cloud is divided into smaller windows, for which the local displacement that best restores the post-earthquake point cloud onto its pre-earthquake equivalent must be found. We investigate how to identify the size of window to be considered for registration. We then present an information theoretic approach that classifies whether a region contains an earthquake fault. These methods are first validated on simulated earthquake datasets, for which the input displacement field is known, and then tested on a real earthquake. We show results and error analyses for a variety of different window sizes, as well as results for our fault detection algorithm.

**Keywords:** Change detection, Airborne systems, Registration, ICP.

## 1 Introduction

Continental earthquakes occur within wide networks of faults which pose a serious hazard to local populations, yet most of these faults remain unmapped

---

Aravindhnan K. Krishnan · Edwin Nissen · Srikanth Saripalli · Ramon Arrowsmith  
Autonomous Systems Technologies Research and Integration Laboratory  
Arizona State University, USA

e-mail: [aravindhnan.krishnan@asu.edu](mailto:aravindhnan.krishnan@asu.edu)

<http://robotics.asu.edu>

Alejandro Hinojosa-Corona  
Division de Ciencias de la Tierra, CICESE, Mexico

e-mail: [alhinc@cicese.mx](mailto:alhinc@cicese.mx)

<http://www.cicese.mx>

or poorly documented [4]. To better understand the tectonics of these regions and to help constrain the likely timing and magnitude of future seismicity, it is crucial to map earthquake-related surface deformation, and from this, calculate the distribution and sense of slip on the causative faulting. Satellite radar interferometry (InSAR) has proved a powerful method for measuring far-field earthquake displacements, but the technique often breaks down close to the fault rupture (due to ground disruption) and is insensitive to North-South motions (because of its viewing geometry). Sub-pixel correlation of optical images helps solve these problems, but can only determine lateral displacements, leaving the important vertical component unresolved.

Sub-meter resolution topographic data derived from airborne Lidar offer huge potential for complementing these existing techniques by providing 3-dimensional, near-fault surface displacements and fault slip. Such datasets are rapidly becoming widespread; in California, for instance, Lidar data have been collected along most of the key active faults over the past decade, including the full length of the onshore San Andreas Fault [2, 7]. Were a future earthquake to occur on one of these faults, a repeat Lidar scan of the fault would enable differential analysis of dense, pre- and post-earthquake topographic data. The 4 April 2010 El Mayor-Cucapah earthquake ( $M_w$  7.2) in northern Mexico is currently the only earthquake rupture with both pre- and post-event Lidar coverage. A simple differencing of gridded Digital Elevation Models (DEMs) generated from these point clouds revealed spectacular images of surface faulting and complex off-fault deformation [6]. However, these maps do not account for lateral displacements and so cannot be directly equated to any single component of the 3-D displacement field.

Computing the full 3-D surface displacements following an earthquake could potentially revolutionize our understanding of rupturing processes and would greatly aid research on faulting and tectonics in earthquake-prone regions. The objectives of this work are to devise a method to compute full 3-D displacements from pre- and post-earthquake Lidar datasets, and in doing so identify the causative faulting and its sense and magnitude of slip.

## 2 Problem Statement

The problem can be formulated as follows. Given pre- and post-earthquake Lidar point clouds (each containing a scattered distribution of points), find the 3-dimensional displacement (with rotation and translation components) that has best shifted the post-earthquake point cloud from its pre-earthquake equivalent. These shifts will vary spatially, depending on the distance to the fault, the sense and magnitude of slip and secondary effects such as landsliding. For this reason, the area must be divided into separate windows and the best local transformation identified for each one. To complicate matters, post-event windows which contain surface faulting will not be related by a rigid body transformation to their pre-event equivalents.

A few things must be considered in this problem statement. Firstly, how do we decide upon an appropriate window size for splitting the data? Secondly, without any prior knowledge, how do we identify whether a particular window contains the fault, or lies away from the fault and has been shifted?

### 3 Data Description

We began our experiments using a synthetic earthquake dataset, before moving on to real earthquake displacements. The synthetic post-earthquake dataset was generated by adding displacements of known magnitude and sense to a real point cloud (the ‘target cloud’), to be tested against another, unaltered point cloud representing the pre-earthquake ground surface (the ‘source cloud’). This way, we were able to identify an approach which best reproduced the known input displacements. We used publicly available “B4” Lidar data [2] covering a  $\sim 2 \times 2$  km section of the San Andreas Fault (SAF) near Coachella, CA, collected on five separate, parallel flight lines with  $\sim 50\%$  overlap between adjacent swaths. In the realistic case, pre- and post-earthquake datasets would utilize different Lidar scan lines, so we split the original dataset by flight line, using the 1st, 3rd and 5th swaths for the source cloud and adding synthetic earthquake displacements to the 2nd and 4th flight lines for the target cloud. Both datasets have average point cloud densities of  $\sim 2$  points/m<sup>2</sup>. Our synthetic fault strikes North-West through the center of the target cloud, close to the real surface trace of the SAF. To simulate a vertical, right-lateral rupture, we displaced points North-East of the fault 2 m towards the South-East, and displaced points South-West of the fault 2 m towards the North-West. To evaluate our ability to detect vertical motions, we also raised points on the North-East of the fault by 1 m. After investigating the synthetic case, we go on to test the method using real pre- and post-earthquake data from part of El Mayor-Cucupah earthquake rupture in Mexico [6]. Here, the pre- and post-earthquake point cloud densities are on average 0.013 points/m<sup>2</sup> and  $\sim 9$  points/m<sup>2</sup>, respectively.

### 4 Algorithm

We use the Iterative Closest Point (ICP) algorithm [1] with a point to plane metric [8] for point cloud alignment. ICP operates by finding the corresponding point  $q_i$  in the target cloud for every point  $p_i$  in the source cloud, and determines the rigid body transformation that minimizes the distances between these points. It is an iterative process where the correspondences and the errors are computed at every iteration and the rigid body transformation is applied to the source cloud repeatedly until it aligns with the target cloud. With the point to plane error metric, the objective is to minimize the distance between the source point ( $p_i$ ) and the tangent plane at the corresponding target point ( $q_i$ ). The error metric can be written as follows

$$E = \sum_i \| (\phi p_i - q_i) \cdot n_i \|^2 \quad (1)$$

where  $\phi$  is the rigid body transformation that minimizes the error metric and  $n_i$  is the normal to the tangent plane at  $q_i$ . The transformation matrix consists of a translation component and a rotation component.  $\phi = T(t_x, t_y, t_z) \cdot R(\alpha, \beta, \gamma)$ . A linear approximation [5] can be made to the rotation matrix where  $\theta \approx 0$  and the new transformation matrix is of the form below.

$$\phi = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

We explain how the computed transformation is validated in section 6.

## 5 Approach

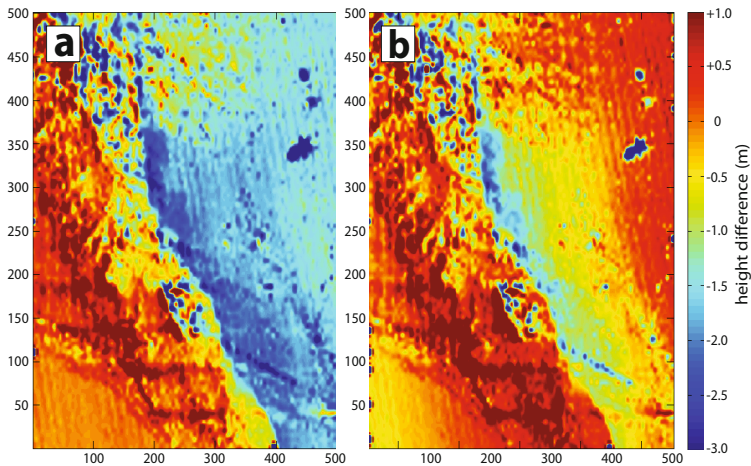
We began by choosing an arbitrary window size in the source cloud (e.g. 200 m  $\times$  200 m). For each of these windows, the corresponding window in the target data is identified based on  $x$  and  $y$  coordinates. This target window is then enlarged (e.g. by 10%) such that the displacements that we are trying to quantify are fully accommodated. Next, we computed the rigid body transformation between the source and target windows using the ICP algorithm. This window is split into four smaller windows of equal size and the rigid body transformation is computed on every child window. The transformation is validated after each split (explained in section 6) and the associated error computed. Based on the differences in error after consecutive splits, we decide whether further splitting is necessary. We verified experimentally that we cannot have small errors for very small window sizes ( $\sim 10$  m) given the point cloud densities and input displacements. An analysis of this error indicates when to stop splitting.

After running ICP using a good window size, each window is then considered for a fault analysis. The curvature of the local surface is computed at every point in the transformed source windows (obtained by applying the computed transformation on the source window i.e.  $\phi p_i$ ) and target windows ( $q_i$ ) and the curvature distribution is estimated by assigning the curvature computed at each point to different bins of an histogram (ranging from *max-curvature* to *min-curvature*) and then computing the probability mass function from this histogram. If there is no rigid body transformation (in case of windows containing the fault) the source and target curvature distributions will not be the same. An information theoretic measure is used to detect this inconsistency in the curvature distributions. The information gain between the transformed source cloud ( $X$ ) and the target cloud ( $Y$ ) is given by

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \quad (3)$$

where  $H$  is the entropy of the curvature distribution.  $H(X, Y)$  is computed on the curvature distribution of the merged clouds  $X$  and  $Y$ . When the right window size is used on regions related by a rigid body transformation, the information gain should be maximum. If the estimated transformation is sub-optimal (i.e. if ICP converges to a local minima) or if the considered region is not related by a rigid body transformation (in the case of windows containing faulting) the information gain should be minimal. Hence thresholding based on information gain highlights which windows contain the fault, along with a few false positives where ICP results may be different from the ground truth. It is important to choose the right window size. If a window containing the fault is too large, then points lying away from the fault will dominate the curvature distribution and the fault detection mechanism will be affected.

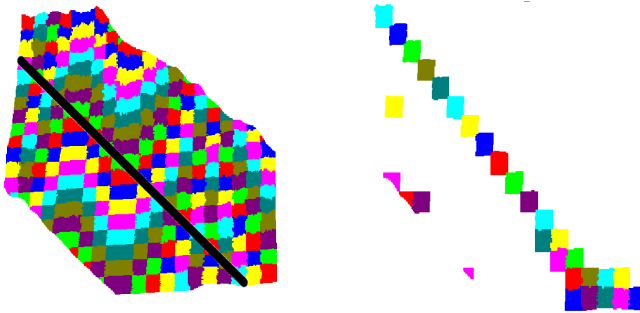
Various non-rigid body transformation methods are available in medical imaging literature [3] and can be considered for this problem. Our goal is not only to get the best alignment possible, but also to identify regions containing the fault. A rigid body transformation estimation followed by a transformation validation achieves both the objectives, whereas the second objective is not met by non-rigid body transformation estimation methods.



**Fig. 1** (a) Height difference map of the Mexico earthquake, before global ICP, with  $x$  and  $y$  coordinates in meters. Height changes across the fault are clear. (b) Height difference map after global ICP, with height differences reduced.

## 6 Transformation Validation

We validate the transformations by randomly choosing  $N$  points per iteration in the transformed source window ( $\phi p_i$ ) and finding the closest point in the target window ( $q_i$ ). The error for the  $k^{th}$  iteration is computed as  $E_k = \sum_i \|\phi p_i - q_i\|^2$  and the standard deviation of this error is calculated over  $k$  iterations. For a good alignment the standard deviation should be minimal. Figure 7 shows the standard deviation of errors for different window sizes. It can be seen that the standard deviation increases gradually as the windows become smaller (part *a* of the figure shows plots for window sizes of 75 m, 50 m and 25 m). However, at a particular point (for our data, a window size of 10 m) the standard deviation jumps markedly, as shown in part *b* (note the difference in  $y$ -axis scales between *a* and *b*). If this happens, it is because the computed transformation for that window is wrong. To discard these invalid transformations, we use a thresholding based on the change in standard deviation as a stopping criteria for window splitting (whereby the standard deviation should not exceed  $1/m$  times that of the previous step).



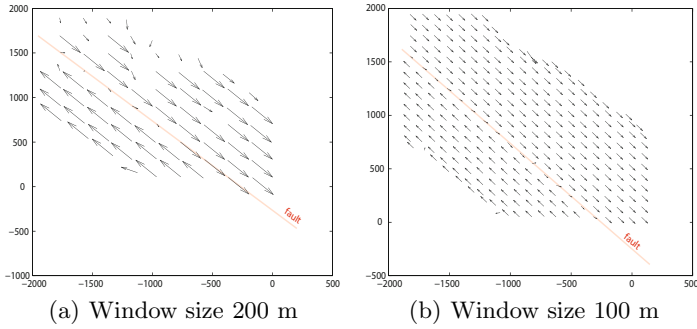
(a) Top view of data split into multiple windows, the thick line shows the line along which the fault was defined  
 (b) Top view of windows containing the fault, there are a few false positives - these are places where ICP converges to a locally optimal solution

**Fig. 2** Window split and fault detection

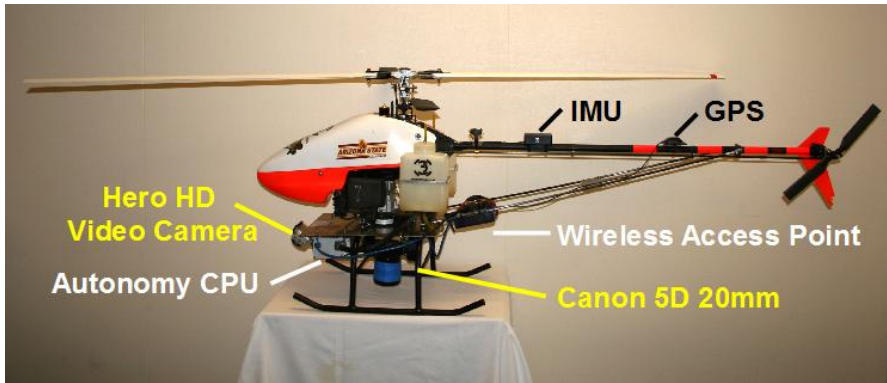
## 7 Results

Figure 1a shows a simple height differencing of the raw Mexico earthquake data, with clear positive height changes West of the fault and negative changes East of the fault. After a global registration, these height differences are reduced with similar height changes on both sides, as seen by the red shading in Figure 1b. This is because ICP has minimized the least square error over the entire point cloud, including both those regions that contain





**Fig. 3** Displacement vectors for different window sizes. The approximate length of the displacement vectors is 2 m. Notice the change in vector directions on both sides of the fault.  $x$  and  $y$  values are in meters.



**Fig. 4** Our Autonomous Helicopter platform

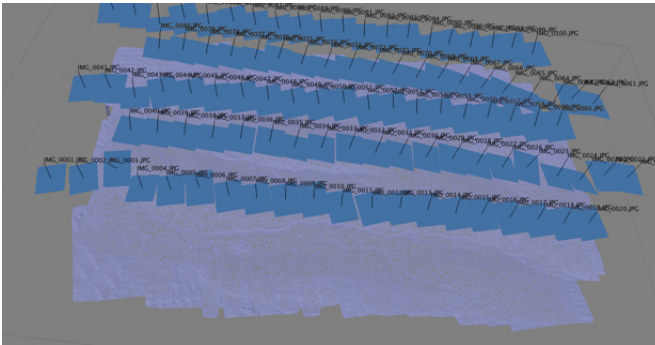
the fault and those that are displaced. The alignment occurring as a result of this least square minimization is not sensitive to the local displacements that we are trying to quantify, and hence a global registration is not suitable for this problem.

Figure 2(a) shows the data split up into multiple, randomly coloured windows with the thick black line showing the synthetic fault line, either side of which artificial displacements were added (as described in section 3).

Figures 3(a) and 3(b) show the displacement vectors ( $\sim 2\text{m}$  in length) obtained for different window sizes for the synthetic earthquake dataset. The change in the direction of the displacement vectors either side of the fault (shown by the red line) are obvious. However, the displacement vectors for windows along the fault are inconsistent. These are windows that are not related by a rigid body transformation and ICP finds the transformation



**Fig. 5** 3D model of Las Cruces test site (400 x 100 m) generated from UAV flights at approximately 50m AGL

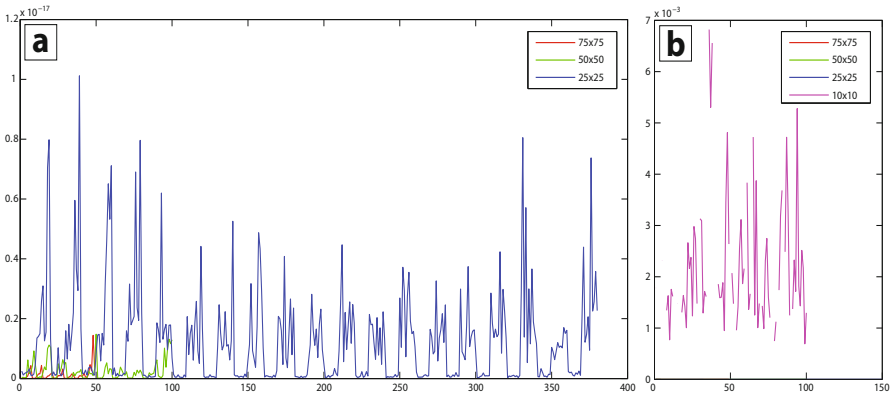


**Fig. 6** 3D model of Las Cruces test site with UAV position and attitude inferred from photogrammetric process

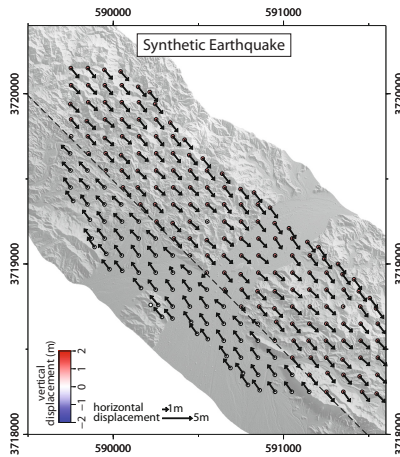
that minimizes the least squares error. Reducing window sizes beyond this point did not satisfy our transformation validation criteria and hence further splitting of windows was stopped.

Figure 2(b) shows the results of our fault detection method, which filters out windows based on information gain as described in section 5. Compared to figure 2(a), only those windows which fall below the information gain threshold are now shown, including a North-West trending sequence of windows along the fault. In addition, there are a few false positives, mostly along the edges where window splitting has left few data points in one of the datasets. We hypothesise that ICP converges to a local minima in these windows.

Figure 7 shows the standard deviation plots discussed in the previous section. Figure 8 shows the displacements calculated for the synthetic earthquake overlaid on the actual topography (we used a DEM derived from publicly available “B4” Lidar data). Black arrows are horizontal displacements and coloured circles denote vertical displacements. The differences in these displacements are clear on either side of the fault. Finally results on a real

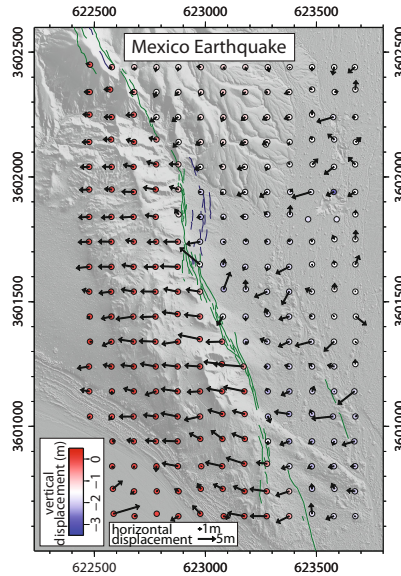


**Fig. 7** Standard deviations by window number, for different window sizes. (a) shows the plot for window sizes of 75 m (red), 50 m (green) and 25 m (blue). In (b), we also plot standard deviations for window size of 10 m (purple), with an enlarged  $y$ -axis scale such that the standard deviations for the 75 m, 50 m and 25 m window sizes are barely visible. There is a huge increase in standard deviation when the window size is reduced from 25 m to 10 m, suggesting that window splitting should be stopped at 25 m.



**Fig. 8** Results for the simulated earthquake. Horizontal displacements (black arrows) and vertical displacements (coloured circles) can clearly be seen to change markedly either side of the fault.  $x$  and  $y$  axes show UTM Zone 11 coordinates, in meters.

earthquake dataset (for the 2010 Mexico earthquake) can be seen in Figure 9. Again, differences in the horizontal and vertical displacements on opposing sides of the fault are clear.



**Fig. 9** Results for the real earthquake. The thin lines show the earthquake surface faulting, as observed by geologists, with E-facing scarps in green and W-facing scarps in blue. Again, the horizontal and vertical displacements clearly change markedly across the fault.  $x$  and  $y$  axes show UTM Zone 11 coordinates, in meters.

## 8 Conclusions and Future Work

We have demonstrated a technique for determining local displacements caused by an earthquake. Our technique uses a windowing approach to determine the correct displacements and an information theoretic approach for determining the regions where these local displacements are present. We have demonstrated the efficacy and accuracy of our technique on datasets collected using airborne lidar. We are able to discern displacements of 1.4 m over an area of  $2 \times 2$  km in our synthetic earthquake experiments and around 1 m over  $2 \times 2$  km in the real earthquake experiment. Currently our technique depends on the pre and post LIDAR data obtained using expensive airborne LIDAR. We plan on using the pre data obtained from airborne LIDAR but post data obtained using Structure from Motion techniques. We propose to use an autonomous helicopter equipped with a downward looking Canon 5D as our platform for obtaining these post point clouds. Our autonomous helicopter is shown in Figure 4. This platform has been outfitted with a vibration isolating camera mount to which the main SFM camera (a Canon 5D) is attached. Figure 5 shows a typical 3D terrain model obtained from our UAV. Figure 6 shows the attitude and position of the UAV calculated using SFM, as the images were taken. This was generated in Las Cruces,

New Mexico, for an area approximately 400 x 100 m. The model has a resolution of 10cm/pixel and an accuracy of 20 cm. This has been determined using pre-surveyed points using a Total Station. In the future we plan on using such dense 3D models created from SFM techniques as our point point clouds. Using such models combined with registration techniques will enable us to determine local displacements accurately. We plan on demonstrating this in the near future.

**Acknowledgements.** The B4 project was lead by Ohio State University, the US Geological Survey and the National Center for Airborne Laser Mapping (NCALM), with support from the National Science Foundation (NSF). We thank all these groups — as well as others involved in supporting the project, including the many landowners along the fault — for the exceptional quality of their dataset and for making it publicly available through the OpenTopography portal. We are very grateful to the Instituto Nacional de Estadística y Geografía (INEGI) for giving us access to the Mexico pre-earthquake dataset. The Mexico post-earthquake data were collected by NCALM through an NSF RAPID award to Mike Oksin (UCD) and Ramon Arrowsmith. We would like to further extend our thanks to all these (and other) collaborators for many fruitful discussions on this exciting new topic of research. OpenTopography and NCALM are supported by the Earth Sciences Instrumentation and Facilities Program at the US National Science Foundation.

## References

1. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 239–256 (1992)
2. Bevis, M., Hudnut, K., Sanchez, R., Toth, C., Grejner-Brzezinska, D., Kendrick, E., Caccamise, D., Raleigh, D., Zhou, H., Shan, S., Shindle, W., Yong, A., Harvey, J., Borsa, A., Ayoub, F., Shrestha, R., Carter, B., Sartori, M., Phillips, D., Coloma, F.: The B4 Project: Scanning the San Andreas and San Jacinto Fault Zones. In: *AGU Fall Meeting Abstracts*, pp. H34–B01 (December 2005)
3. Crum, T., Hartkens, W.R., Hill, D.L.G.: Non rigid Image Registration: theory and practice. *British Journal of Radiology* (2004)
4. England, P., Jackson, J.: Uncharted seismic risk. *Nature Geoscience* 4, 348–349 (2011)
5. Low, K.-L.: Linear least squares optimization for point-to-plane ICP surface registration. Technical Report TR04-004
6. Oskin, M.E., Arrowsmith, J.R., Hinojosa-Corona, A., Elliott, A.J., Fletcher, J.M., Fielding, E.J., Gold, P.O., Garcia, J.J.G., Hudnut, K.W., Liu-Zeng, J., Teran, O.J.: Near-Field Deformation from the El Mayor-Cucapah Earthquake Revealed by Differential LIDAR. *Science* 335, 702–705 (2012)
7. Prentice, C.S., Crosby, C.J., Whitehill, C.S., Arrowsmith, J.R., Furlong, K.P., Phillips, D.A.: Illuminating Northern California's Active Faults. *Eos Trans. AGU* 90, 55–55 (2009)
8. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: *International Conference on 3-D Digital Imaging and Modeling* (2001)

# Automated Crop Yield Estimation for Apple Orchards

Qi Wang, Stephen Nuske, Marcel Bergerman, and Sanjiv Singh

**Abstract.** Crop yield estimation is an important task in apple orchard management. The current manual sampling-based yield estimation is time-consuming, labor-intensive and inaccurate. To deal with this challenge, we developed a computer vision-based system for automated, rapid and accurate yield estimation. The system uses a two-camera stereo rig for image acquisition. It works at nighttime with controlled artificial lighting to reduce the variance of natural illumination. An autonomous orchard vehicle is used as the support platform for automated data collection. The system scans both sides of each tree row in orchards. A computer vision algorithm detects and registers apples from acquired sequential images, and then generates apple counts as crop yield estimation. We deployed the yield estimation system in Washington state in September, 2011. The results show that the system works well with both red and green apples in the tall-spindle planting system. The crop yield estimation errors are -3.2% for a red apple block with about 480 trees, and 1.2% for a green apple block with about 670 trees.

## 1 Introduction

Crop yield estimation is an important task in apple orchard management. Accurate yield prediction helps growers improve fruit quality and reduce operating cost by making better decisions on intensity of fruit thinning and size of the harvest labor force. It benefits the packing industry as well, because managers can use estimation results to optimize packing and storage capacity. Typical yield estimation is performed based on historical data, weather conditions, and workers manually counting apples in multiple sampling locations. This process is time-consuming and labor-intensive, and the limited sample size is usually not enough to reflect the yield distribution across the orchard, especially in those with high spatial variability. Therefore, the current yield estimation practice is inaccurate and inefficient, and improving it would be a significant result to the industry.

Apple growers desire an automated system to conduct accurate crop yield estimation; however, there are no off-the-shelf tools serving this need. Researchers

---

Qi Wang · Stephen Nuske · Marcel Bergerman · Sanjiv Singh  
The Robotics Institute, Carnegie Mellon University,  
Pittsburgh, PA 15213, USA  
e-mail: {nuske,marcel,ssingh}@cmu.edu

have been working on the development of related technologies for a few decades [1]. A wildly adopted solution to automated fruit yield estimation is to use computer vision to detect and count fruit on trees. Swanson et al. [2] and Nuske et al. [3] developed computer vision systems to estimate the crop yield of citrus and grape, respectively. However, there is no reported research leading to satisfactory yield estimation for apples.

Current efforts on apple yield estimation using computer vision can be classified in two categories: (1) estimation by counting apples and (2) estimation by detecting flower density. A few researchers have worked on the first category using color images [4-6], hyperspectral images [7], and thermal images [8]. Their common point is that they only deal with apple detection from a single or multiple orchard scenes; however, no further research is reported about yield estimation, which requires continuous detection and counting. Aggelopoulou et al. [9] worked on the second category. They sampled images of blooming trees from an apple orchard, and found a correlation between flower density and crop yield. However, this flower density-based method is not accurate because multiple unpredictable factors (such as weather conditions) during the long period between bloom and harvest could make the correlation vary year by year.

When conducting the apple counting-based yield estimation, computer vision systems face three challenges due to the characteristics of orchard environments:

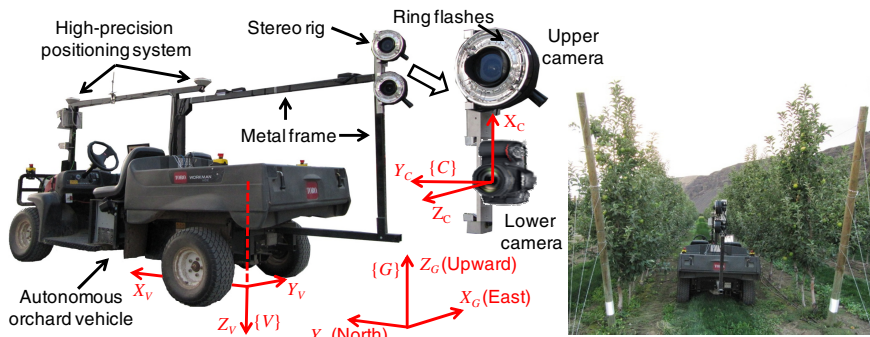
- Challenge 1: variance in natural illumination. It prevents from developing a reliable vision-based method to detect apples from an orchard scene.
- Challenge 2: fruit occlusion caused by foliage, branches, and other fruit.
- Challenge 3: multiple detections of same apple in sequential images. Unsuccessful registration of these detections will cause miscounting.

Our overall research goal is to design, develop, and deploy an automated system for rapid and accurate apple yield estimation. The system reduces labor intensity, and increases work efficiency by applying computer vision-based, fast data acquisition. Meanwhile, it improves prediction accuracy by relying on a large-scale data acquisition. At this stage of the research, we focus on two specific objectives: (1) develop system hardware and major algorithm modules for data acquisition and yield estimation; (2) conduct preliminary performance tests in an orchard.

## 2 System Overview

The hardware of the yield estimation system consists of three major parts (Fig. 1):

1. A stereo rig composed of two high-resolution monocular Nikon D300s cameras (Nikon Inc., Melville, NY, USA) with wide-angle lenses (focal length: 11 mm). The D300s is a consumer product with a low cost comparing to industrial or scientific imaging systems. The two cameras are mounted on an aluminum bar with a distance of about 0.28 m to form a stereo pair. The two cameras are triggered synchronously at 1 Hz.



**Fig. 1** The crop yield estimation system hardware and coordinate frames: camera  $\{C\}$ , vehicle  $\{V\}$ , and ground  $\{G\}$ .  $\{C\}$  originates at the focal point of the lower camera;  $\{V\}$  originates at the projection of the central point of the rear axle of the vehicle on the ground;  $\{G\}$  is a combination of the Universal Transverse Mercator coordinate system and elevation. The geometric relationship between  $\{C\}$  and  $\{V\}$  is calibrated. The geometric relationship between  $\{V\}$  and  $\{G\}$  is determined by the on-board positioning system.

2. Controlled illumination. The system is designed for night use to avoid interference from unpredictable natural illumination, thus addressing the Challenge 1 in Section 1. Ring flashes (model: AlienBees ABR800, manufactured by Paul C. Buff, Inc. Nashville, TN, USA) around the two lenses are used as active lighting during image acquisition. The energy release of each flash is set at 20 Ws. The two cameras are both set with aperture  $f/6.3$ , shutter speed  $1/250$  s, and ISO 400 for an optimal exposure of apple trees (about 2 m away from the cameras) under this controlled illumination.

3. A support vehicle. An autonomous orchard vehicle [10] developed at Carnegie Mellon University is used as the carrying platform for automated data acquisition. The platform is able to travel through orchard aisles at a preset constant speed by following fruit tree rows. The speed is set at 0.25 m/s for our data acquisition. The stereo rig is attached to a frame at the rear of the vehicle (Fig. 1). Each tree row is scanned from both sides. The acquired sequential images provide multiple views of every tree from different perspectives to reduce fruit occlusion, which addresses the Challenge 2 in Section 1. The on-board high-precision positioning system, POS LV manufactured by Applanix (Richmond Hill, Ontario, Canada), provides the geographic coordinates of the vehicle. The position and pose of the vehicle is used by the system software to calculate the geographic position of every detected apple. We use the global coordinates of apples to register the multiple detections of same apple to reduce over counting and address the Challenge 3 in Section 1.

The software of the crop yield estimation system has two major parts: (1) online processing, and (2) post-processing. The online processing controls the start and the stop of data acquisition. It is written in Python (Python Software



Foundation). The software processes the acquired data off-line for apple detection, apple registration, and final apple count. Matlab 2010a (The MathWorks, Inc., Natick, MA, USA) is the programming language for post processing.

### 3 Apple Detection

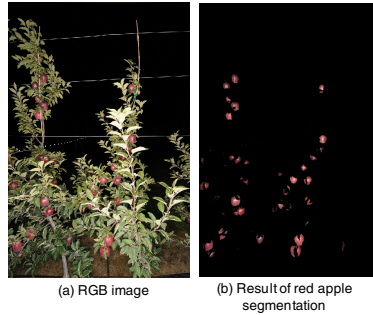
The algorithm uses the following procedure to detect apples from an image. Firstly, it reads a color image ( $1072 \times 712$  pixels) acquired by the system and removes distortion. Then, it uses visual cues to detect regions of red or green apples in the image. Finally, it uses morphological methods to convert apple regions into apple counts in the image.

#### 3.1 Detection of Red Apple Pixels

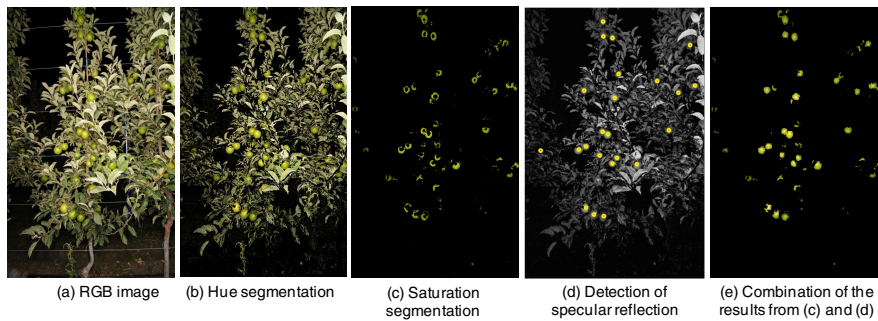
Under the controlled illumination, the red color of apples can be distinguished from the colors of other objects in the orchard, such as the ground, wires, trunks, branches, and foliage (Fig. 2a). The algorithm uses hue, saturation, and value in the HSV color space as visual cues for red apple detection. The hue values of red apple pixels are mainly in the ranges from  $0^\circ$  to  $9^\circ$  and from  $349^\circ$  to  $360^\circ$ . The hue values of other objects are out of the two ranges. It is necessary to exclude the background pixels during hue segmentation of red apple pixels because hue is undefined for pixels without any color saturation (white, grey, black colors), and for dark pixels close to black the saturation is low and the hue channel is unreliable. Therefore, the procedure for red apple segmentation is: (1) segment pixels with  $0^\circ \leq \text{hue} \leq 9^\circ$  or  $349^\circ \leq \text{hue} \leq 360^\circ$ ; (2) remove (background) pixels with saturation  $\leq 0.1$  or value  $\leq 0.1$ . After the processing, the regions of red apple are segmented from the image (Fig. 2b).

#### 3.2 Detection of Green Apple Pixels

We use three visual cues: hue, saturation, and intensity profile to detect green apple pixels from an image. Analysis of 10 sample images shows that the hues of green apples and foliage are mainly in the range from  $49^\circ$  to  $75^\circ$ . We use this hue range to segment green apples and foliage from an image. The dark background and most non-green objects are removed after the hue segmentation (Fig. 3b). Although apples and foliage are both green, the apple pixels have a stronger green color which can be separated from leaves using the saturation channel. The algorithm uses a saturation threshold ( $\geq 0.8$ ) to segment green apple pixels. Most foliage pixels are removed after the saturation segmentation (Fig. 3c); however, the central parts of most apples are removed as well because the camera flashes generate specular reflections at the central part of a green apple. In HSV color space, specular reflection has high brightness and low saturation. They cannot be detected as apple pixels by the saturation segmentation.



**Fig. 2** Red apple segmentation using hue, saturation and value (brightness) as visual cues



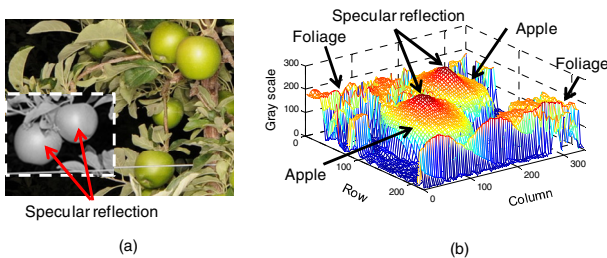
**Fig. 3** (a) An image acquired by the crop yield estimation system in a green apple orchard. (b) The result of hue segmentation for green colors. (c) The result of saturation segmentation for the green color of apples. (d) The result of specular reflection detection (marked by circles). (e) The result of apple region detection using color and specular reflection.

The next step is to detect the apple regions with specular reflections. The rectangular area marked by dash lines in Fig. 4a is the minimum bounding rectangle of the apple area detected by the hue and saturation thresholding procedure, and extended by 25% to guarantee some apple pixels undetected by the saturation segmentation are included in the region. Fig. 4b is the light intensity (grayscale value) map of the rectangular region in Fig. 4a. The two apples have conical shape in their intensity profiles, which represents gradually descending light intensity from the peak to different directions. Compared to the apple regions, the foliage regions have more irregular intensity profiles. Based on these features, we use the shape of intensity profile to detect specular reflections.

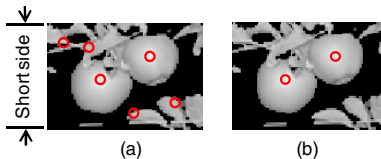
The algorithm detects specular reflections by searching for local maxima in the grayscale map (Fig. 4b). During the search, the size of the support neighborhood for the local maxima is 30% of the length of the short side of the rectangular region (Fig. 5a). The result of local maxima detection (Fig. 5a) includes the points with specular reflections, and also some points without specular reflections. To distinguish them, the algorithm checks the intensity profiles on four lines passing

through each local maximum. As shown in Fig. 6, four lines go through a local maximum with slopes of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ , respectively. Each line is 21 pixels in length. The intensity profiles around the specular reflection of an apple descend gradually from the local maximum to different directions (Fig. 6a). However, the intensity profiles around a local maximum, which is not a specular reflection, have irregular changes of grayscale values. Based on this difference, the algorithm uses the following procedure to decide whether a local maximum is a specular reflection. (1) It calculates the gradients of grayscale values between every two adjacent pixels on the four line segments. (2) It splits each line segment into two parts in the middle (at the local maximum). (3) It calculates a roundness score ( $R_i$ ) by checking the signs of gradient in the eight parts. If the gradients in one part have the same sign, the roundness score of the part is one; otherwise, it is zero. (4) It calculates the total roundness score:  $R = \sum_{i=1}^8 R_i$ . If  $R \geq 4$ , the local maximum is a specular reflection; otherwise, it is a false positive. The reason for using four rather than eight as a threshold is to make sure that the specular reflections on some partially-occluded apples can also be recognized. (5) It keeps specular reflection and removes false positives from the search region (Fig. 5b).

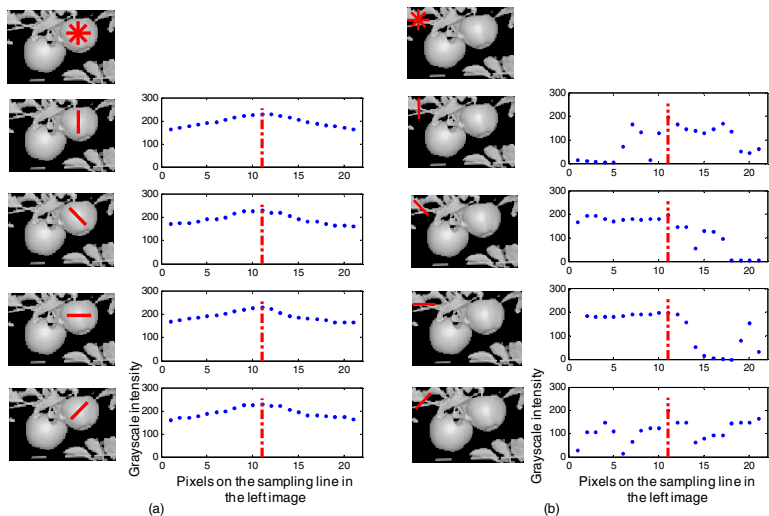
The algorithm overlies the results of saturation segmentation (Fig. 3c) with the pixels of specular reflection (Fig. 3d) and their surrounding neighborhoods ( $18 \times 18$  pixels). This combination yields a more complete detection of apple pixels in the image (Fig. 3e).



**Fig. 4** (a) An example of specular reflection on the surface of apples. The rectangular region is transformed to grayscale image. (b) The mesh plot of the grayscale values of the rectangular region in (a).



**Fig. 5** (a) Local maxima (marked by circles) in a search region. (b) The results of detecting and removing local maxima that are not specular reflections.



**Fig. 6** (a) The intensity profiles around a local maximum, which is the specular reflection of an apple, in four directions. (b) The intensity profiles around a local maximum, which is not a specular reflection, in four directions.

### 3.3 Segmenting Individual Apples

The previous sections described how to detect the apple pixels in an image for both red and green apples. Here we describe morphological operations to convert these pixel regions into distinct individual apples.

Firstly, the software loads a binary image of apple regions. To realize apple counting, the software needs to determine the average diameter ( $\bar{D}$ ) of apples in the loaded image. It calculates the eccentricity ( $E$ ) of each apple region, and uses a threshold  $0 < E < 0.6$  to find regions that are relatively round. These relatively round regions are usually the apples that have less occlusion and do not touch other apples, which is convenient for determining apple diameter. A few small round regions are also detected. They are the visible parts of some partially occluded apples and happen to be round in shape. Usually, they only account for a small portion of all the relatively round regions. To remove the noise, the software calculates the area ( $S$ ) of the relatively round regions and their average area ( $\bar{S}$ ). It uses a threshold  $S > \bar{S}$  to remove the noise with small area. Then, it calculates the length (in pixel) of the minor axis of the ellipse that has the same normalized second central moments as a remaining round apple region. The mean of the minor axis length of all the remaining regions is used as the average diameter of apples in the image.

Some apple regions contain two or more touching apples. The algorithm is able to detect them and split them into two apples. It calculates the length ( $L_{major}$ , in pixel) of the major axis of the ellipse that has the same normalized second central moments as an apple region. Any region with  $L_{major} > 2\bar{D}$  is treated as a region

with touching apples. It splits the major axis into two segments in the middle (at Point A in Fig. 7), and then uses the central points of each segment (Points B and C in Fig. 7) as the center of the two apples. It should be mentioned that the current version of software is designed to split this kind of region into only two apples. The design is based on the fact that apple clusters are usually thinned down to two apples in commercial orchards to keep the quality of individual apples in clusters. For orchards with larger clusters of apples we need to make improvements to deal with more apples per cluster.

Some apples are partially occluded by foliage or branches, and may be detected as multiple apple regions. The software calculates the distance between the centers (with the same definition as Point A in Fig. 7) of any two apple regions. Any pair with a distance less than  $\bar{D}$  is treated as one apple. The midpoint of the two original centers is the new center of the apple.

In the end, the software records the locations of the centers of the remaining apple regions as the final detection of apples in the image.

## 4 Apple Registration from Multiple Images

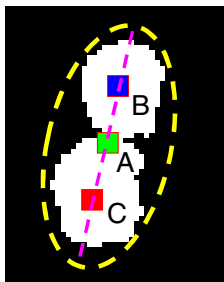
Apple registration is required to merge apples that are detected multiple times. One apple can be seen up to seven times from one side of a tree row in the sequential images taken by the system. Some apples can be seen from both sides of a tree row.

During continuous counting, the software detects apples in every frame of the image sequences taken by the two cameras of the stereo rig. The software applies the following procedure to calculate the global locations of detected apples, and use the locations to register apples.

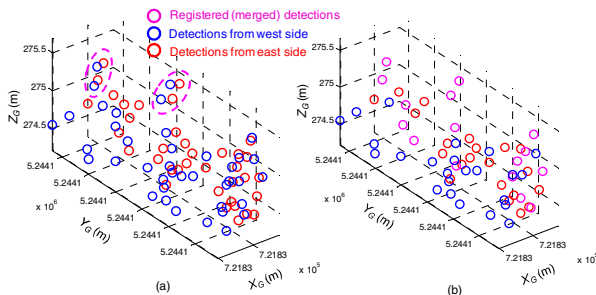
Firstly, the software uses block matching to triangulate the 3D positions (in  $\{C\}$ ) of all apples detected in the image sequences. The block matching is conducted in both directions between one pair of images taken by the binocular stereo rig. When an apple detected in the lower image and an apple detected in the upper image are matches reciprocally, the software triangulates and records the 3D position (in  $\{C\}$ ) of the apple center (in 2D image coordinates); otherwise, it discards the detection.

The software transforms apple locations from  $\{C\}$  to  $\{G\}$ . The transformation provides the global coordinates for every detected apple.

The software merges the apples that are detected multiple times from one side of a tree row. It calculates the distance between every two apples in  $\{G\}$ , and then merges the apples with a distance less than 0.05 m from each other. The new location of the merged apple is obtained by averaging the locations of the multiple appearances of this apple. The software discards two kinds of detected apples as noise from the results: (1) apples that are detected only once in the sequential images; and, (2) dropped apples on the ground. Since the height of an apple on a tree is usually more than 0.3 m above the ground, apples with a height less than 0.3 m are treated as dropped apples.



**Fig. 7** The result of splitting a region with two touching apples



**Fig. 8** (a) The apple detection results of three trees from two opposite sides of a row. (b) The results of registering apples detected from both sides.

The software uses global coordinates to register apples detected from both sides of a tree row, which requires precise positioning. However, we have noticed two issues in our apple positioning approach. First, when the vehicle returns on the opposite side of the row, the GPS system has noticeably drifted in its reported height above ground. We have also noticed a bias in the stereo triangulation algorithm causing the apple location to be estimated closer to the camera. To solve these positioning problems, we calculate the GPS drift and stereo triangulation bias by locating objects on the orchard infrastructure, triangulating their position in world coordinates and repeating from the other side of the row. The error between the two reported locations of an object from each side of the row gives us a position correction term that we apply to the apple locations. The landmarks can be any stationary location such as the ends of posts, stakes, and wires. We used flagging tape that was placed every three trees and at present we manually record the landmark positions in the images, however this will be replaced in future iterations of the system by an algorithm that can automatically detect the orchard infrastructure.

After correcting the apple locations, we merge the apples detected from both sides of the row. Fig. 8a shows an example in which some apples (marked by ovals) can be detected from both east and west sides of a row. To avoid double counting, the software calculates the distances between apples detected from one side and those detected from the other side. It merges apples within a distance of 0.16 m from each other. We use such a large threshold (0.16 m, about twice the average apple diameter) to tolerate errors in stereo triangulation. After this operation, the apples detected from both sides of a row are registered (Fig. 8b), and the software obtains a final apple count for the orchard.

## 5 Experiments and Results

The crop yield estimation system was deployed at the Sunrise Orchard of Washington State University, Rock Island, WA in September, 2011. The goals of the deployment are: (1) to evaluate the estimation accuracy of the current system, and (2) to discover issues that need to be improved for future practical applications.

## 5.1 Experimental Design

The experimental design includes four critical issues: apple variety, orchard planting system, the area of orchard for data collection, and ground truth.

We selected two blocks of typical apple trees – Red Delicious (red color) and Granny Smith (green color). These two popular commercial products are typical varieties in either red or green apples. Based on the suggestions of horticulturists, we selected the “tall spindle” planting system (as shown in Fig. 1) for the field tests. This system features high tree density, a thin canopy, and well-aligned, straight tree rows. It maximizes profitability through early yield, improved fruit quality, reduced spraying, pruning, and training costs. “Tall spindle” is being adopted by more and more apple growers, and is believed to be one of the major planting systems of apple orchard in the future, because of its ability to rapidly turn over apple varieties from those less profitable to those more profitable.

The area of each block is about half acre. Specifically, there are 15 rows of red apple trees and 14 rows of green apple trees. Each row has about 48 trees. The ground truth of yield estimation is the human count conducted by professional orchard workers. Every tree row is split into 16 sections with three trees per section. The sections are marked by flagging tape that is used by the workers to count the number of apples in each section and likewise we force our algorithm to report the count for each section by manually marking the flags in the images.

## 5.2 Results

The software processes the sequential images obtained from the two blocks, and generates apple counts for every section (three trees). The results and analysis are presented as follows.

Fig. 9 shows the crop yield estimation and ground truth of the red apple block. In rows 1-10 that received regular fruit thinning, the computer count is close to the ground truth. The estimation errors of each row have a mean of -2.9% with a standard deviation of 7.1%. If we treat the 10 rows together, the estimation error is -3.2%. The numbers show that the crop estimation from the system is fairly accurate and consistent for rows 1-10. However, we undercount rows 11-15 by 41.3% because these trees were not fruit thinned (which would normally be conducted in most commercial orchards) leaving large clusters of apples. The larger clusters of apples cause two problems: (1) some apples are invisible and cannot be detected due to the occlusion caused by other apples nearby; (2) some fruit clusters consist of more than two apples, and the current version of software can only split a cluster into two apples. Although the estimations per row are significantly below the ground truth, the standard deviation is small at 3.2%, which shows that the system performs consistently. Therefore, it is possible to calibrate the system, which will be discussed later.

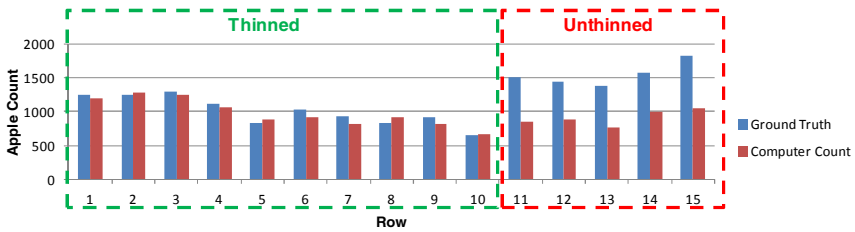


Fig. 9 Crop yield estimation and ground truth of the red apple block

Similarly, the errors of the raw counts from the green apple block rows have a mean of -29.8% with a standard deviation of 8.1%. The trees in the green apple block have more foliage than those in the red apple block. The occlusion caused by foliage is thought to be the main reason for the undercount of green apples. Despite the large level of undercounting, the error is relatively consistent, making calibration for the raw counts possible. We perform calibration by selecting 10 random sections from the 224 sections in the green apple block, and conduct linear regression (with intercept = 0) between the computer count and the ground truth (as shown in Fig. 10). The slope of the linear equation is the calibration factor. We run the method for 100 times, the average calibration factor is 1.4 with a standard deviation of 0.1. The small standard deviation shows that the sample size of 10 sections is big enough to obtain a steady calibration. Using 1.4 as a calibration factor to correct the yield estimation in the green apple block, the average yield estimation error at row level falls to 1.8% with a standard deviation of 11.7% (Fig. 11). The compensated yield estimation error for the whole green apple block is 1.2%. We apply the same method to rows 11-15 in the red apple block, and find a calibration factor of 1.7. After the compensation, the average estimation error at row level falls to 0.4% with a standard deviation of 5.5%.

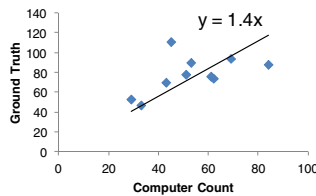


Fig. 10 A linear regression between computer counts for ten random green apple sections and the ground truth

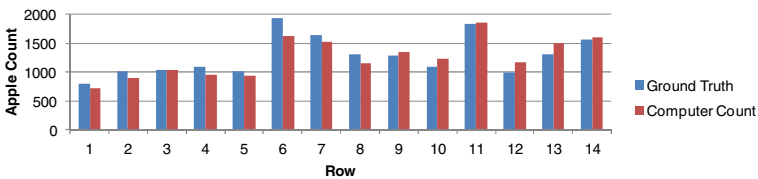


Fig. 11 Calibrated crop yield estimation and ground truth of the green apple block

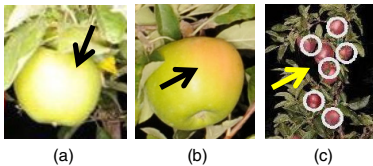


## 6 Discussion

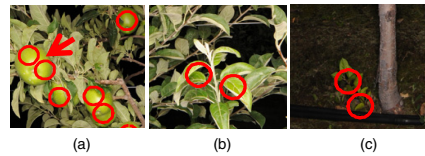
The accuracy of the crop yield estimation system is subject to two major aspects: (1) how accurately it detects visible apples; (2) how accurately it estimates invisible apples. They are discussed in this section.

### 6.1 Detection Error of Visible Apples

In certain situations, the current software makes errors in detecting visible apples. For example, in our data set, the images of a few green apples are overexposed because these apples are much closer to the fleshes than the majority of the apples. As shown in Fig. 12a, a green apple in an overexposed image loses its original color. A small amount of green apples have sunburn (red blemish) on their skins (Fig. 12b). Green apples in an overexposed image or with sunburn cannot pass the (green) hue segmentation, and are undetectable in the image processing. New visual cues other than color should be considered in the future to deal with this problem. As mentioned earlier, the software has a limitation in dealing with fruit clusters comprised of more than two apples (Fig. 12c) and should be corrected in future iterations. False positive detections happen infrequently, but do occur in some situations as seen in Fig. 13. Future version of the software will look to reduce these false detections with a more strict set of image processing filters.



**Fig. 12** Examples of visible apples that are not detected by the software. (a) An overexposed image of a green apple. (b) A green apple with sunburn. (c) Apples missed in a large fruit cluster.



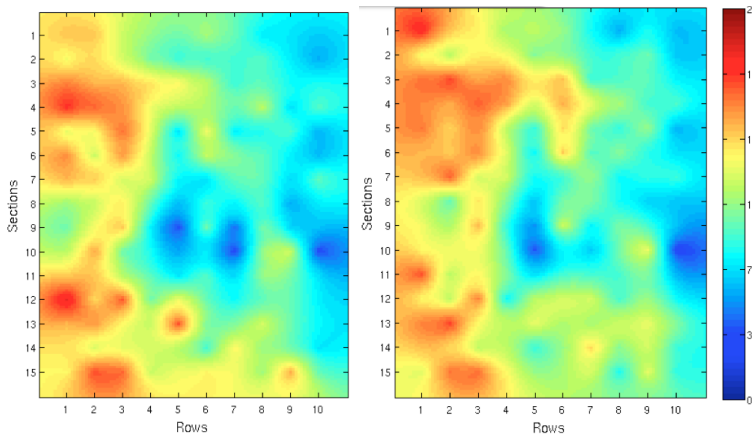
**Fig. 13** Examples of false positive detections. (a) An apple with a short distance from the cameras. (b) Some new leaves with a color similar to green apples. (c) Weed with a color similar to green apples.

### 6.2 Calibration for Occluded Apples

The computer vision-based system cannot detect invisible apples that are occluded by foliage or other apples. As mentioned earlier, our solution is to calculate a calibration factor based on human sampling, and use the factor to predict the crop yield including invisible apples. The results show that the calibration method works well. In future work we will study the calibration procedure and evaluate accuracy versus sample size on larger orchard blocks. Too much sampling increases the cost of yield estimation; while, too less may harm the accuracy of estimation. We also will study if calibration factors can be used from prior years or from other orchards of similar varieties, similar ages and grown in similar styles.

### 6.3 Yield Maps

In addition to providing the grower with the total number of apples in an orchard, the system is also able to generate a yield map (Fig. 14) that provides information of the spatial distribution of the apples across the orchard. A yield map could be used for precision orchard management, enabling the grower to plan distribution of fertilizer and irrigation, to perform variable crop thinning, and to improve operations by increasing efficiency, reducing inputs and increasing yield over time in underperforming sections.



**Fig. 14** High-resolution yield map representing spatial distribution of apples across the Red Delicious block. Color-coded legend uses the units of apples per tree. Left: Apple counts generated by our automated algorithm. Right: Ground truth apple counts. Our system provides a map that is a very close resemblance to the true state of the orchard.

## 7 Conclusions

Field tests show that the system performs crop yield estimation in an apple orchard with relatively high accuracy. In a red apple block with good fruit visibility, the crop yield estimation error is -3.2% for about 480 trees. In a green apple block with significant fruit occlusion caused by foliage, we calibrate the system using a small sample of hand measurements and achieve an error of 1.2% for about 670 trees.

In future work we will improve the system to deal with orchards with larger clusters of apples, which will require more precise and advanced methods to segment the apple regions within the images. We will also improve the registration and counting algorithm to better merge apples detected from two sides of a row.

**Acknowledgements.** This work is supported by the National Institute of Food and Agriculture of the U.S. Department of Agriculture under award no. 2008-51180-04876. The authors wish to thank Kyle Wilshusen for the support with processing the yield maps.

## References

1. Jimenez, A.R., Ceres, R., Pons, J.L.: A survey of computer vision methods for locating fruit on trees. *Transactions of the American Society of Agricultural Engineers* 43(6), 1911–1920 (2000)
2. Swanson, M., Dima, C., Stentz, A.: A multi-modal system for yield prediction in citrus trees. In: *ASABE Annual International Meeting*, Pittsburgh, PA, United States, pp. 20–23, Paper number: 1009474. ASABE (June 2010)
3. Nuske, S., Achar, S., Bates, T., Narasimhan, S., Singh, S.: Yield estimation in vineyards by visual grape detection. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, San Francisco, CA, USA, September 25–30 (2011), doi:10.1109/iros.2011.6048830
4. Linker, R., Cohen, O., Naor, A.: Determination of the number of green apples in RGB images recorded in orchards. *Computers and Electronics in Agriculture* 81, 45–57 (2012), doi:10.1016/j.compag.2011.11.007
5. Rakun, J., Stajniko, D., Zazula, D.: Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry. *Computers and Electronics in Agriculture* 76(1), 80–88 (2011), doi:10.1016/j.compag.2011.01.007
6. Tabb, A.L., Peterson, D.L., Park, J.: Segmentation of apple fruit from video via background modeling. In: *ASABE Annual International Meeting*, Portland, OR, United States, July 9–12, Paper number: 063060. ASABE (2006)
7. Safren, O., Alchanatis, V., Ostrovsky, V., Levi, O.: Detection of green apples in hyper-spectral images of apple-tree foliage using machine vision. *Transactions of the ASABE* 50(6), 2303–2313 (2007)
8. Stajniko, D., Lakota, M., Hočevár, M.: Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. *Computers and Electronics in Agriculture* 42(1), 31 (2004), doi:10.1016/s0168-1699(03)00086-3
9. Aggelopoulou, A., Bochtis, D., Fountas, S., Swain, K., Gemtos, T., Nanos, G.: Yield prediction in apple orchards based on image processing. *Precision Agriculture* 12(3), 448–456 (2011), doi:10.1007/s11119-010-9187-0
10. Hamner, B., Bergerman, M., Singh, S.: Autonomous orchard vehicles for specialty crops production. In: *ASABE Annual International Meeting*, Louisville, KY, United States, August 7–10, Paper number: 1111071. ASABE (2011)
11. Bouguet, J.Y.: Camera calibration toolbox for Matlab (2008), [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/) (accessed May 14, 2012)

# Spatial Interpolation for Robotic Sampling: Uncertainty with Two Models of Variance

Young-Ho Kim, Dylan A. Shell, Colin Ho, and Srikanth Saripalli

**Abstract.** Several important forms of robotic environmental monitoring involve estimating a spatial field from comparatively few measurements. A number of researchers use linear least squares estimation techniques, frequently either the geostatistical Kriging framework or a Gaussian Process regression formulation, that provide estimates of quantities of interest at unmeasured locations. These methods enable selection of sample locations (e.g., for adaptive sampling) by quantifying uncertainty across the scalar field. This paper assesses the role of pose uncertainty and measurement error on variance of the estimated spatial field. We do this through a systematic empirical comparison of scalar fields reconstructed from measurements taken with our robot using multiple imperfect sensors and actively estimating its pose. We implement and compare two models of variance: Kriging Variance (KV) and Interpolation Variance (IV), illustrating that the latter—which has not been used in a robotics context before—has several advantages when used for online planning of sampling tasks. Using two separate experimental scenarios, we assess the estimated variance in scalar fields constructed from measurements taken by robots. Physical robots sampling within our office building suggest that using IV to select sampling sites gathers more data for a given time window (45% more than KV), travels a shorter distance to collect the same number of samples (25% less than KV), and has a promising speed-up with multiple robots. Water quality data from an Autonomous Underwater Vehicle survey of Lake Pleasant, AZ. also show that IV produces better qualities for given a distance and time.

**Keywords:** Environmental monitoring, robotic sampling, Measurement and position error.

---

Young-Ho Kim · Dylan A. Shell  
Texas A&M University, College Station, TX  
e-mail: {yhkim, dshell}@cse.tamu.edu

Colin Ho · Srikanth Saripalli  
Arizona State University, Tempe, AZ  
e-mail: {colinho, srikanth.saripalli}@asu.edu

## 1 Introduction

Large-scale environmental monitoring is a particularly promising application for robots [8]. Robots have already begun to collect oceanographic data sets of unprecedented scale and resolution (*e.g.*, [4]). The underlying challenge addressed by such systems stems from the fact that the measured data are sparse compared to the large spatial areas/volumes of interest. Mobility makes adaptive strategies for *in situ* sampling possible but leads naturally to the question: “*given the data already captured, where should the robots go in order to sample further?*” This is an important basic problem in robotic monitoring and data collection, and is one for which a variety of solutions have been proposed *e.g.*, [7, 5, 14].

Linear least-squares estimation methods have been among the most successfully used in robotics for spatial interpolation and region sampling. By way of example, we include the recent work of Kempainen *et al.* [7] in the Gaussian Process regression framework, and Elston *et al.* [5] via Kriging interpolation. Both employ equivalent minimum error-variance estimation techniques [11] that permit measured data to be interpolated in a way that takes into account a statistical description of spatial covariance [3]. Along with an estimate of values of interest at particular locations in the field, these methods also associate a measure of estimate uncertainty. Within the Kriging framework, the standard error measure is called the Kriging Variance (KV) or Kriging Error. It plays an important role for robotic adaptive sampling due to the fact that researchers (including, but not limited to [7] and [5]) have used it to select future sampling locations and to plan informative paths.

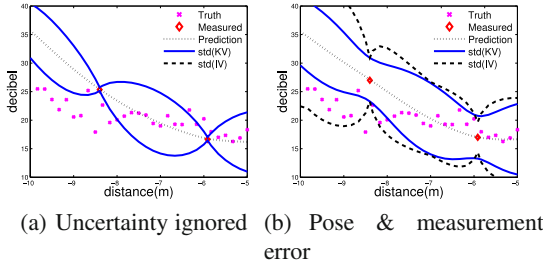
This paper assesses the role of pose uncertainty and measurement error on variance of such an estimated spatial field. We conducted a systematic empirical comparison of scalar fields reconstructed from measurements taken by robots using imperfect sensors.

### 1.1 Problem Statement

Two complementary aspects are investigated experimentally in this work:

*Issue 1: Pose and measurement uncertainty*— Classical formulations for spatial field estimation involve idealizations that may be ill-suited for robotic sampling. Specifically, standard formulations lack explicit consideration of measurement noise and position uncertainty, flying in the face of practical experience with real sensors. This paper includes a formulation which addresses these two aspects. Because significant prior robotic work employs formulations without modeling these forms of uncertainty explicitly (*cf.* [7, 5, 14]), we conducted an empirical evaluation of how estimates of the spatial field differ depending on whether the method employed considers these sources of uncertainty or not. Scale and severity of the uncertainty are important considerations too.

*Issue 2: An alternative to KV*— Yamamoto [12] argued that the traditional KV measure fails to measure local data dispersion appropriately because it is computed from a global description of spatial variance (the variogram) averaged over the whole



**Fig. 1 Richer treatment of site uncertainty is justified.** The standard model (a) underestimates field variance compared to explicit treatment of uncertainty (b). Note also Interpolation Variance (IV) shown in (b).

estimate. To address this shortcoming he introduced a new measure called the Interpolation Variance (IV). It is computed as the weighted average of the squared differences between measured data and the interpolation estimate, which is intuitively analogous to the traditional expression for statistical variance. Several geostatistical papers have evaluated IV and compared it to KV [10, 13], but IV appears to be unknown in the robotics literature.

Figure 1 illustrates the difference in estimated field uncertainty when sample position uncertainty and measurement noise are factored into estimates of field uncertainty measured with KV. Without considering these aspects, field variance is underestimated. The figure on the right also shows the IV measure for comparison; IV offers a distinct and in some ways more informed estimate of the interpolation uncertainty.

The following are the paper’s contributions:

- A unified Ordinary Kriging (OK) formulation with both pose uncertainty and measurement noise which explicitly separates sensing error from the variogram.
- A particle filter realization and implementation of the formulation.
- An empirical investigation of the impact of sample site uncertainty with respect to KV and IV. Most importantly, IV is an informative measure for selection of future measurement sites, which suggest a new set of adaptive sampling approaches.
- The demonstration of autonomous sampling site selection with single and multiple robots and a comparison KV and IV in terms of resulting performance.

*Notation:* Since no single definitive formalism has yet emerged for robotic sampling, we have elected to describe the results with terminology from the Kriging framework. The following motivated this choice:

1. Interpolation Variance has only been proposed within the Kriging framework.
2. Our work requires that we communicate our findings, including measured and estimated data, with geostatisticians.
3. Our applications involve robots estimating two or three dimensional fields. The family of variogram models have been informed by experience with the underlying physical processes (*cf.* covariance models in [9]).

Equivalency implies that the results hold for the Gaussian processes model as well.

## 1.2 Related Work

Measurement error is a common problem in robotics and it is widely recognized that most sensors are imperfect. In contrast, Cressie's classic geostatistics text [2] mentions that measurement error is usually implicitly included the nugget variance and that the practitioner often ignores the measurement error because it is typically considered smaller than the spatial variogram. When neither identified nor directly treated, micro-level variation of the spatial process is conflated with error introduced by imperfect sensing. The distinction is particular important when the variogram is intended to describe the intrinsic spatial variability of the statistical process, rather than its observation through a particular sensor. This is significant when multiple sensors are available for a particular phenomenon and when sensor noise characteristics are estimated online.

Also, perfect robot pose information is not normally available. Chilés [1] was the first to consider position uncertainty in the Ordinary Kriging (OK) interpolator. He assumed a probability distribution over poses as the only form of error, naming this "attribute position error." Cressie and Kornak [3] take both position and measurement uncertainty into account in their Universal Kriging (UK) model. The UK variant is more general than the OK model, since it includes parameters to estimate strong trend in the underlying spatial process. The UK's generality comes at a price, however: interpolation error is not (numerically) simple to estimate. In fact, most existing robotics experiments employ OK [7, 14] over UK for this reason. Additionally, evidence presented in by Zimmerman *et al.* [15] shows that OK actually outperforms UK for realistic data since the latter is prone to over-fit. The OK formulation presented here is suited for robotics as it incorporates both pose uncertainty and measurement noise, is tractable when realized with a particle filter, and explicitly represents the sensor noise model.

In robotics, Kempainen *et al.* [7] used Kriging without treatment of pose or measurement uncertainty in considering well-localized robots in an experimental environment. Zhu *et al.* [14] applied OK to find a optimal sampling strategy based on KV. Elston *et al.* [5] evaluated trajectory quality on the basis of a variogram model; the simulation did not consider measurement error.

As a measurement of interpolation error, KV does not depend on data values but only on the semi-variogram model and data sample sites. Yamamoto [12] suggested that the Interpolation Variance (IV) would be well-founded because it measures the reliability of estimates by considering both the Kriging weights and data values, *i.e.*, it is a heteroscedastic measure. We are not aware of any evaluation of IV that considers measurement or position uncertainty.

## 2 Technical Approach

The purpose of Kriging is to estimate the values of a spatial random variable,  $Z$ , from sparse sample data. Ordinary Kriging, the most common type of Kriging in practice, assumes that the spatial statistical process that generates the random values can be

characterized by an unknown mean [11]. An estimate  $Z^*(x_0)$  is obtained from  $\lambda_i$  and samples  $Z(x_i)$  via (1):

$$Z^*(x_0) = \sum_{i=1}^n \lambda_i Z(x_i), \quad \sum_{i=1}^n \lambda_i = 1. \tag{1}$$

The coefficients  $\lambda_i$ , called Kriging weights, show that the interpolated points are obtained from a distance weighted average of nearby measured points. The weights are computed in matrix form via a pseudo-inverse of (2):

$$\mathbf{A} = \begin{pmatrix} \gamma(x_1, x_1) & \cdots & \gamma(x_1, x_n) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma(x_n, x_1) & \cdots & \gamma(x_n, x_n) & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}, \tag{2}$$

$$\mathbf{b} = \begin{pmatrix} \gamma(x_1, x_0) \\ \vdots \\ \gamma(x_n, x_0) \\ 1 \end{pmatrix}, \quad \mathbf{A}\boldsymbol{\lambda} = \mathbf{b}. \tag{3}$$

The value depends on the semi-variogram function  $\gamma$  which represents the strength of spatial relationships in the random field. It is defined as the square of the expected difference between values at different locations, *i.e.*,  $\gamma(x, y) = [E(Z(x)) - E(Z(y))]^2$ .

KV is given by  $\sigma^2(x_0) = \sum_{i=1}^n \lambda_i \gamma(x_i, x_0) + \psi(x_0)$ . Taken together, these yield the optimal OK prediction of unobserved values of the process when neither measurement noise nor position uncertainty are included.

### 2.1 Measurement Error

**Kriging Variance:** The traditional Kriging approach usually treats the measurement noise as zero, *i.e.*, that the measured data represent true values. It is common for geostatistical texts (*e.g.*, [2, 11]) to add that if there is reason for a practitioner to suspect measurement error (described with variance,  $c_m$ ) then the remedy is to subtract this value from the field Kriging variance, as so

$$\sigma_m^2(x_0) = \sum_{i=1}^n \lambda_m^i \gamma_m(x_i, x_0) + \psi(x_0) - c_m, \tag{4}$$

where  $\gamma_m$  is the variogram and  $\psi$  is a Lagrange multiplier. This is both counter-intuitive and incorrect for imperfect measurement as typically considered in the robotic sampling context. Understanding why depends on two observations. Firstly, despite moving from idealized to noisy sensing, it has tacitly been assumed that the variogram is constructed using the same noisy sensor as employed for the sampling itself. Thus, the variogram no longer merely encodes the intrinsic spatial



dependency of the underlying statistical process. The result is that  $\gamma_m$  is a variogram that includes measurement error and has been shifted by  $c_m$  automatically. Secondly, applying (4) removes the implicitly captured sensing variation in order to describe only the interpolation variance. It does not, therefore, yield a variance that estimates what would result if you were to place the sensor at the given location.

Instead, we define  $\gamma_m$  to be  $\gamma_{ruth} + c_m$ , where  $\gamma_{ruth}$  is calculated using a high-fidelity sensor or from a theoretical understanding of the spatial process. The resultant expression for the variance is

$$\sigma_{\epsilon}^2(x_o) = \sum_{i=1}^n \lambda_i^j \gamma_{ruth}(x_i, x_o) + \psi(x_o) + c_m. \tag{5}$$

One is now no longer subtracting the measurement error from the prediction, which decreases the variance so it no longer matches measured data, but instead increasing the degree of uncertainty, as one might expect.

**Interpolation Variance**

In contrast, interpolation variance estimates are affected by the variogram only indirectly. The IV computation is analogous to a form of variance where the probability density is replaced by OK weight  $\lambda$ . The choice of  $\gamma_m$  alters the computed  $\lambda$ , but the sensitivity to measurement noise is dominated by the local data variation. Clearly this latter effect has no impact on KV. Equation (6) shows IV in terms of  $\lambda$  and the variance of the input consistent with the other notation:

$$s_o^2 = \sum_{i=1}^n \lambda_i [z(x_i) - z^*(x_o)]^2. \tag{6}$$

**2.2 Position Uncertainty**

To take position uncertainty into account within the Kriging framework, Chilés [1] proposed the following model. The measurement believed to be made at a point  $\mathbf{x}_\alpha$  is actually performed at another position  $\mathbf{x}_\alpha + \mathbf{u}_\alpha$ , where  $\mathbf{u}_\alpha$  is a random vector which contributes to the position error. Given probability density  $p(\mathbf{u}_\alpha)$  and also the joint density  $p(\mathbf{u}_\alpha, \mathbf{u}_\beta)$  for any pair of points, one may extend the OK equations (2)–(3) by randomizing vectors  $u_\alpha$  over poses. This produces equations (7)–(8):

$$\begin{aligned} \tilde{\mathbf{x}}_\alpha &= \mathbf{x}_\alpha + \mathbf{u}_\alpha, \\ \tilde{\mathbf{A}}_{\alpha\beta} &= \iint \gamma_m(\tilde{x}_\alpha, \tilde{x}_\beta) p(u_\alpha, u_\beta) du_\alpha du_\beta. \end{aligned} \tag{7}$$

$$\tilde{\mathbf{b}}_{o\alpha} = \int \gamma_m(\tilde{x}_\alpha, \tilde{x}_o) p(u_\alpha) du_\alpha. \tag{8}$$

The expressions for  $\tilde{\mathbf{A}}_{\alpha\beta}$  and  $\tilde{\mathbf{b}}_{\mathbf{o}\alpha}$  can be directly applied in (2)–(3) in place of  $\mathbf{A}$  and  $\mathbf{b}$ . Employing this strategy propagates pose uncertainty (position variation) through to the field estimate and produces an increase in the interpolation variance. It is important to note that even if the distributions are unimodal and isotropic, the observed values may differ from the interpolator predictions at the associated maximum likelihood positions.

In practice, incorporating pose uncertainty in Kriging for real systems involves application of (7)–(8) in a representation which ensures the operations involved can be carried out tractably. We describe an efficient realization of Kriging via a particle filter. Additionally, this has the advantage of already being a practical representation of pose uncertainty, *e.g.*, see [6]. Particle filters approximate a probability density through a set of weighted samples,  $\omega_{(n)}$ , drawn from the distribution being represented.

The following equations illustrate a Kriging computation, analogous to that described above, using the particle filter. Essentially, integrals have been replaced with summations, and appropriate weights used:

$$\tilde{\mathbf{A}}_{\alpha\beta} = \sum_{i=0}^n \sum_{j=0}^m \gamma_m(x_i, x_j) \omega_i \omega_j, \tag{9}$$

$$\tilde{\mathbf{b}}_{\mathbf{o}\alpha} = \sum_{i=0}^n \gamma_m(x_i, x_o) \omega_i, \tag{10}$$

where  $n$  and  $m$  correspond to the number of samples drawn from the distribution. Similar to the equations described above, once the relevant matrices are computed in this way, interpolation proceeds as before.

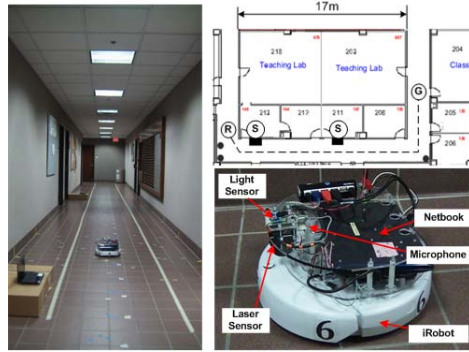
Our particular robots localize themselves using a particle filter based approach, so this treatment is particularly straightforward: Kriging inputs are computed via (9)–(10) on the localization particles and, once the relevant matrices are computed in this way, interpolation proceeds directly.

### 3 Experiments

#### 3.1 Equipment and Experimental Scenarios

Two separate experimental scenarios were used. Both assess the estimated variance in scalar fields constructed from measurements taken by robots:

**1. Sound and light, indoors:** An iRobot Create robot, equipped with a Hokuyo URG-04LX-UG01 laser sensor and an Asus Eee PC 1005HA netbook, was given a map of our building; it moved around the environment using an adaptive particle filter to localize itself. The localized robot was given several target measurement positions along a linear corridor. A total of 44 measurement positions, each approximately 50cm apart, were provided as navigation goals. After the robot arrived at each goal position, it saved its current localization estimates by writing the



**Fig. 2** Environment in which measurements were recorded: the robot navigates along the corridor from R to G, taking denote measurements. The two S symbols denote the positions of sound sources.

particles representing the pose probability distribution to disk. The robot remained in place for 2 seconds, collecting data from three sensors: (1.) ambient sound volume in decibels via the onboard microphone on the netbook; (2.) light intensity measured in lux with a Phidgets “Precision Light Sensor” and (3.) ambient temperature via a Phidgets “Precision Temperature Sensor”. We did not report values from the third sensor, as the building climate control made them uninteresting.

Figure 2 shows the second floor of our building. For simplicity the measurements we analyze further all come from a single corridor about 17m in length. Two sound sources emit the same continuous beeping sound at locations shown in the figure (the sound itself is a flute C7 note at 2000Hz.) The measurements were recorded at night so that the florescent lights in the ceiling were the sole light source.

The light and sound fields represent opposite extremes. Light intensity was essentially constant, except with a spatial period representing light spacing, the light sensor showed low variation, and the empirical variogram shows a short range spatial covariance. On the other hand, the measured sound volume had large fluctuations and a variogram illustrating longer range spatial structure.

The sensor noise variance  $c_m$  was estimated for each sensor, by computing the mean of variances computed from 200 values taken over 2 seconds at the same position. Variogram models were fitted passing through the origin from samples averaged over the 200 values in order to minimize sensor noise. We adopted the spherical variogram model (11), where  $n$  is the height of the jump of the semivariogram at the origin (known as the *nugget*),  $s^*$  is the limit of the variogram at infinite lag (the *sill*), and  $r$  is the distance at which this is first reached (*range*). Light within our environment is characterized as follows:  $r = 2m$ ,  $s^* = 580lux$ , and the sensor has variance  $c_m = 5$ . Similarly sound has  $r = 5m$ ,

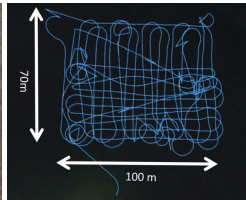
$s^* = 80$  decibels, and the sensor variance  $c_m = 30$ .

$$\gamma_{sph}(x_1, x_2) = s^* \cdot \left[ \frac{3 \|x_1 - x_2\|}{2r} - \frac{\|x_1 - x_2\|^3}{2r^3} \right] + n. \quad (11)$$

**2. Interpolated H<sub>2</sub>O quality indicators, from Lake Pleasant:** An AUV equipped with a YSI 6600vs Sonde (see Fig. 3) took dense measurements within a 70m × 100m window. These readings along with GPS positions were used as “ground truth” data on which we simulated sampling experiments as follows: readings at any point were generated from an OK interpolation with Gaussian noise added ( $\sigma^2$  selected per YSI’s listed measurement accuracy per quality indicator).



(a) Ocean Server Iver2 AUV.



(b) Sampling pattern.

**Fig. 3 Real world data were collected in Lake Pleasant, AZ.** (−33° 51′ 55.66″N, 112° 17′ 45.01″W) by directing a AUV in (a) to trace the 3215m long “lawnmower” pattern in (b). Measurements of five different quality indicators were taken 3448 times at 0.5Hz over the 1hr 55min traversal.

Samples, collected in the way described above, were then processed four different ways in order to consider the different treatments of sample site uncertainty:

1. The single best pose hypothesis with field measurement assumed to be error free (PBMB).
2. The single best pose hypothesis with field measurement treated as noisy (PBMU).
3. The full pose distribution with measurement assumed to be error free (PUMB).
4. The full pose distribution with measurement treated as noisy (PUMU).

Instances involving PB used the single sample from the particle filter with maximum weight. The PU instances employed the whole distribution in order to pass the uncertainty onto the spatial field representation. The measurement treatment was analogous: variance of the sensor only being considered in MU cases.

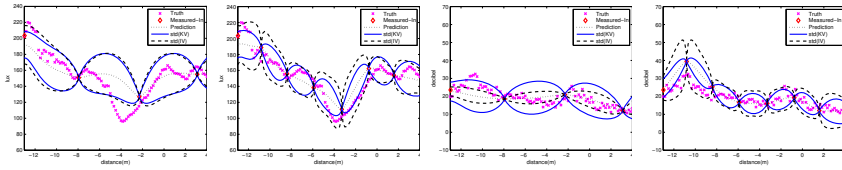
### 3.2 Planners: Greedy KV- and IV-Based Selection

In the data reported in Section 4, we provide the distance travelled as a robot attempts to reduce field uncertainty. In these cases, we considered a naïve planner in which the robot selects new sample locations by greedily picking the position in the field estimate with greatest uncertainty, sampling there, incorporating the data, and repeating the process. This uncertainty is measured using either KV or IV, as will be indicated.

## 4 Results

### 4.1 Comparison of KV and IV: Density Dependence

Figure 4 illustrates that KV depends on sample position but is independent of the observed data.<sup>1</sup> KV has the same shape between measurement sites no matter what the measured data actually are. However, IV considers different local data values and exhibits some degree of anisotropy in shape.

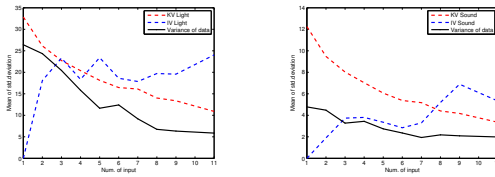


(a) Light, 4 samples (b) Light, 8 samples (c) Sound 4 samples (d) Sound 8 samples

**Fig. 4** The two variance estimates have different behavior with regard to sample density

Figure 4 also shows that KV and IV have different characteristics and that these depend on the two different sensors. As the number of measured data increases, KV decreases smoothly. In contrast, IV can increase due to an implicit dependency on data variation. Thus, IV has the smaller value of the two estimates for sparse data, but with the increasing density, the tendency of both measures causes them to cross at a certain point. Figure 5 quantifies this effect for the sensors we considered: the light sensor results in an intersection for between 3 and 5 samples. For the sound sensor this occurs between 7 to 8 samples. Moreover, the data variance decreases as additional measurements are added. The evaluation of KV and IV for the two sensors in subsequent sections uses 4 inputs for light and 8 inputs for sound.

**Fig. 5 Differing behavior as a function of data density.** Analysis of variance (measured and estimated) as a function of sample density. (KV in red, IV in blue.)



(a) Light sensor: Variances. (b) Sound sensor: Variances.

### 4.2 Measurement Error

As the reported  $c_m$  values illustrate, the light sensor is considerably more sensitive than the sound sensor. To make clear the impact of measurement error on variance estimates, we show only the sound case.

<sup>1</sup> Since the variogram is isotropic, the KV inherits this property too.

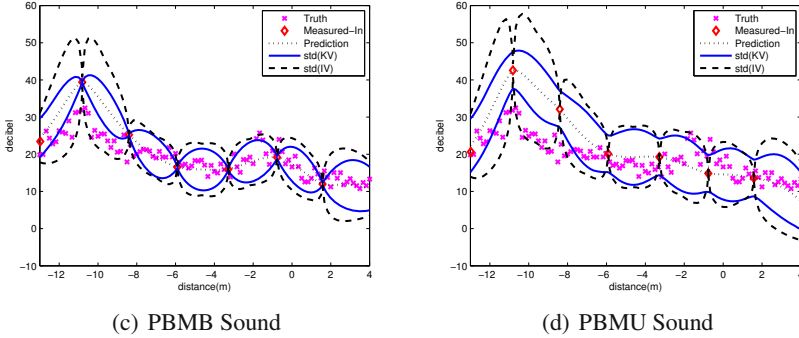


Fig. 6 Comparison of the measurement error at 8 points

Figure 6 illustrates several properties: IV equals zero at the measured points, while KV is increased by  $c_m$  at the measured points. Moving beyond the measured points, both increase approximately linearly with increasing noise. Figure 6 shows clearly how IV underestimates at measured points. The rate that IV increases reflects unexpected uncertain data unlike the KV which merely considers distance between measurements.

Although IV vanishes at measurement points, it compensates when multiple sensed values are close to one another since the sensor noise inherently limits the quality of prediction from the local neighborhood. The measurement error is implicitly represented by IV, which relies on the increasing numbers of measurements, to truly reflect the data spread. We suggest that IV will capture  $c_m$  when multiple measurements are taken from the same location. However, IV must be taken as the *envelope* computed from all measurements and one should not compute IV from the mean.

### 4.3 Position Uncertainty

Figure 7 shows the effect of position uncertainty for each of the sensed fields. Neither, KV nor IV are zero at the measured points but both increase linearly with IV increasing more than KV. Severe position uncertainty degrades interpolation because  $\tilde{A}_{\alpha\beta}$  represents a convolution weight. It is worth observing that KV increases most near measurements. On the other hand, IV increases variance everywhere else. Unlike the previous case, the interpolated Kriging may no longer be a punctual interpolator, *i.e.*, the estimate may not coincide with the measurement.

### 4.4 Effects of Both Forms of Sample Uncertainty

It is worth considering both measurement error and position uncertainty together, as it best reflects the reality of most robotic sampling scenarios. Figure 8(a) shows the PBMB case (no uncertainty modelled) for the light sensor. The KV, however,

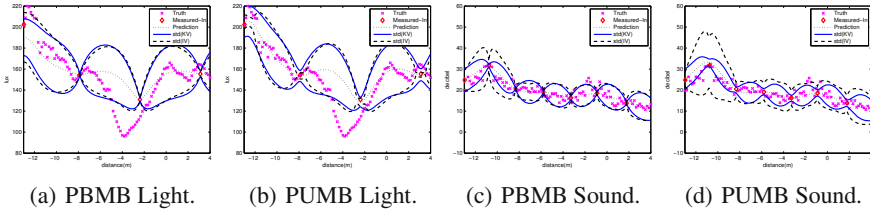


Fig. 7 Position uncertainty: light with 4 inputs, sound with 8 inputs

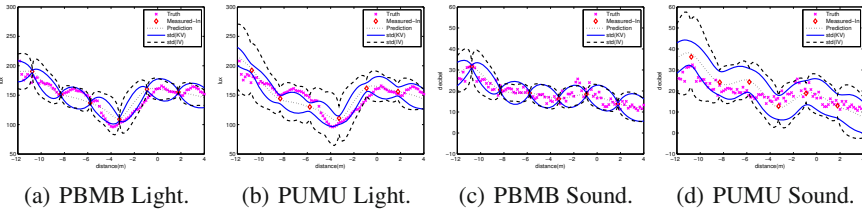


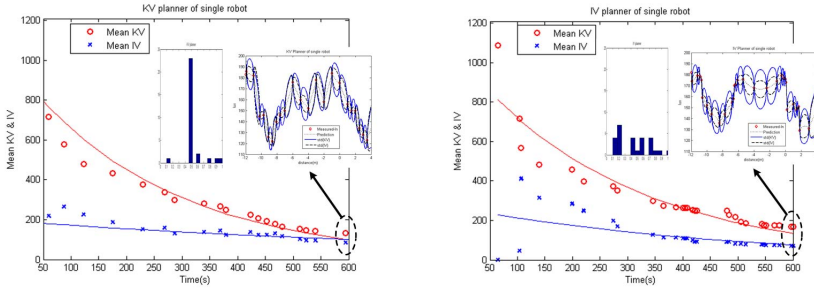
Fig. 8 Both variance measures for 8 light and sound readings

underestimates the variance in some locations, partly because the light sensor has a small variance. Figure 8(c) shows the underestimation of variance (observable by comparison to the ground-truth data). By considering uncertainty, Figure 8(b) shows that KV does not fully resolve the underestimation problem, while IV appears to be a better estimate. However, it may overestimate in cases with small scale sensor variance like the light sensor. In Figure 8(d), the KV does not show the variation of sensor values. However, IV shows good estimates for the noisy sound sensor, and the PUMU IV case shows the uncertainty near the measurements.

In many cases, the absolute variance estimate is used to select navigation goals. While both forms of variance can be used, it is important to note that the sensor variance can influence the appropriate choice of goal.

### 4.5 Planner Effectiveness of for Adaptive Sampling

Greedy planners based on KV and IV can both be effectively used for autonomous sampling since either decreases (both) measures of variance. Fig. 9 summarizes the experiments conducted with a single robot indoors for estimating the light intensity field. Selecting targets with KV results in an approximately even spacing between samples, as shown in inter-sample distance histogram inset in Fig. 9(a). In contrast, the insets in Fig. 9(b) show how the heteroscedastic property causes different spacing. The right inset of Fig. 9(b) also illustrates the importance of employing the envelope scheme when there are multiple measurements at a particular site or even very nearby sample sites.



(a) KV-based planner (first 600 seconds). (b) IV-based planner (first 600 seconds).

**Fig. 9 Autonomous sampling drives variance measures down.** The sample position selected as either (a) maximal KV, or (b) maximum IV position. The vertical axis represents the mean KV and IV, the horizontal axis denotes time. The inset histogram shows the distribution of inter-sample distances. The right inset is the resultant light field (in lux) along with variance estimates.

Figure 10 shows data from multi-robot tests with KV or IV for sample site planning. The plots show that the two robots are able to sample with efficiency that is almost twice that of the individual robots, if we consider the rate of decrease in total field variance.

### 4.6 Data-Dependence on Planner Performance

While KV is widely known and a frequently used measure, IV has complementary aspects as a candidate for robotic sampling applications. Fig. 11 shows that this observation is true in both scenarios. For light indoors, the robot covers a distance  $\pm 25\%$  shorter to collect the same number of samples as a KV planning robot. In a given time interval, the IV method collects  $\pm 45\%$  more samples than KV. The table below shows Lake Pleasant data.

**Table 1** Application of IV and KV planner in Lake Pleasant, AZ. The results compare 10 cross-validation trials, showing the mean and standard deviation values. The estimated field uses 30 samples at point A in Fig. 11(b).

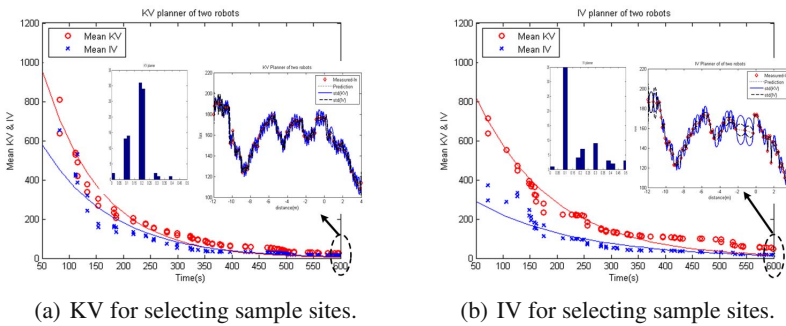
	IV planner	KV planner
B/G Algae	(3467.7, 37104)	(3186.8, 26543)
Chlorophyll	(0.4117, 0.0015)	(0.4321, 0.0013)
pH	(0.1509, 0.0008)	(0.1687, 0.0004)
Temperature	(0.1939, 0.0001)	(0.2071, 0.0002)

(Left) Reported statistics are  $(\mu, \sigma)$  of MSE between estimated field and ground-truth values, for  $n = 10$  independent trials.

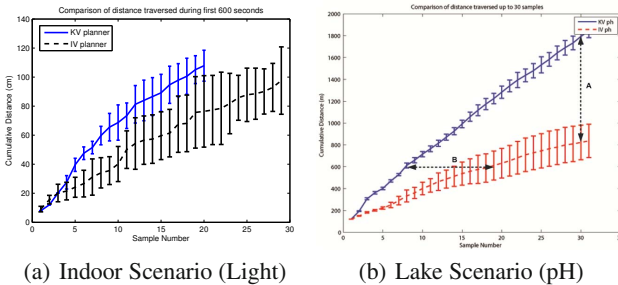


**Table 2** Application of IV and KV planner in Lake Pleasant, AZ. The results are mean and standard deviation of MSE measured cross-validation between estimated field and ground-truth values for 10 trials. The estimated field are for samples drawn within the first 600m in B of Fig. 11(b).

(Right) Cross-validation analogous to Table 1, but with an upper bound on distance traversed rather than a fixed number of samples. The $s$ value denotes number of samples used.	IV planner	KV planner
B/G Algae	(3467.7, 37104) $s = 30$	(3236.7, 29308) $s = 9$
Chlorophyll	(0.4084, 0.0014) $s = 25$	(0.5212, 0.0048) $s = 8$
pH	(0.1786, 0.0024) $s = 18$	(0.1796, 0.0008) $s = 8$
Temperature	(0.1939, 0.0001) $s = 30$	(0.2007, 0.0003) $s = 5$



**Fig. 10** A multi-robot demonstration of KV and IV for sampling. Two robots sample light and temperature concurrently.



**Fig. 11** Minimizing IV collects more data for a given duration than KV, and travels shorter distances to collect the same number of samples. Data are means and variances from ten separate trials of each greedy planner.

### 5 Conclusion

In robotics, Kriging spatial interpolation is generally used without considering measurement error and position uncertainty. However, our data demonstrate that doing so may result in underestimation of interpolation error. This paper shows how both of these forms of uncertainty can be easily incorporated into the standard OK interpolator, allowing the uncertainty to reflect itself as an increase in variance of the

underlying field. Our physical robot data show that measurement error and position uncertainty do affect the error estimates for both KV and IV. Our empirical assessment resulted in several observations about the models variance and their relationship. For example, data density affects both KV and IV, but in opposite ways; we first postulated and then determined that a cross-over point exists. Secondly, even though IV equals zero at measured points, employing the envelope of the variance function is natural near multiply measured points. Finally, sample site uncertainty not only changes the variance, but can also result in a non-punctual interpolator.

While KV is widely known and frequently used traditional measure, IV is a new candidate for robotic sampling applications which has several aspects that are complementary to KV. We have shown the feasibility of IV as a measure of uncertainty by incorporating it in a demonstration of an autonomous robot system which adaptively samples light and temperature in our building. When compared to KV, selecting points to greedily minimize IV collects more data for a given duration and travels shorter distances to collect the same number of samples. Additionally, IV appears no less suited to multi-robot applications than KV in terms of potential sampling speed-up.

## References

1. Chilés, J.P.: How to adapt Kriging to non-classical problems: three case studies. In: Guarascio, M., David, M., Huijbregts, C. (eds.) *Advanced Geostatistics in the Mining Industry*, pp. 69–90. D. Reidel Publishing Co., Holland (1993)
2. Cressie, N.: *Statistics for Spatial Data*. John Wiley and Sons, Ltd., New York (1993)
3. Cressie, N., Kornak, J.: Spatial Statistics in the Presence of Location Error with an Application to Remote Sensing of the Environment. *Statistical Science* 18(4), 436–456 (2003)
4. Dolan, J.M., Podnar, G.W., Stancliff, S., Low, K.H., Elfes, A., Higinbotham, J., Hosler, J., Moisan, T., Moisan, J.: Cooperative Aquatic Sensing Using the Telesupervised Adaptive Ocean Sensor Fleet. In: *Proceedings of the SPIE Conference on Remote Sensing of the Ocean, Sea Ice, and Large Water Regions*, Berlin, Germany (2009)
5. Elston, J., Stachura, M., Frew, E., Herzfeld, U.: Toward Model Free Atmospheric Sensing by Aerial Robot Networks in Strong Wind Fields. In: *Proceedings of the International Conference on Robotics and Automation (ICRA 2009)*, Kobe, Japan (2009)
6. Fox, D.: KLD-sampling: Adaptive particle filters. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) *Advances in Neural Information Processing Systems (NIPS-14)*, pp. 713–720. MIT Press, Cambridge (2001)
7. Kemppainen, A., Makela, T., Haverinen, J., Roning, J.: An Experimental Environment for Optimal Spatial Sampling in a Multi-Robot System. In: *Proceedings of the International Conference on Intelligent Autonomous Systems*, Baden-Baden, Germany (2008)
8. Kumar, V., Rus, D., Sukhatme, G.S.: Networked Robots. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, ch. 41. Springer, Heidelberg (2008)
9. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge (2006)
10. da Rocha, M.M., Yamamoto, J.K.: Comparison Between Kriging Variance and Interpolation Variance as Uncertainty Measurements in the Capanema Iron Mine, State of Minas Gerais-Brazil. *Natural Resources Research* 9(3), 223–235 (2000)

11. Webster, R., Oliver, M.: *Geostatistics for Environmental Scientists*. John Wiley and Sons, Ltd., New York (2007)
12. Yamamoto, J.: An Alternative Measure of the Reliability of Ordinary Kriging Estimates. *Mathematical Geology* 32(4), 489–509 (2000)
13. Yamamoto, J.K., da Rocha, M.M.: Properties and Applications of the Interpolation Variance Associated with Ordinary Kriging Estimates. In: *Proceedings of the International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, Shanghai, China, pp. 70–77 (2008)
14. Zhu, X., Yu, J., Ren, S., Wang, X.: Near-optimal Collecting Data Strategy Based on Ordinary Kriging Variance. In: *Proceedings of OCEANS*, Sydney, NSW, Australia, pp. 1–6 (2010)
15. Zimmerman, D., Pavlik, C., Ruggles, A., Armstrong, M.P.: An Experimental Comparison of Ordinary and Universal Kriging and Inverse Distance Weighting. *Mathematical Geology* 31(4), 375–390 (1999)

# Acoustic Masking of a Stealthy Outdoor Robot Tracking a Dynamic Target

Ashley Tews and Matthew Dunbabin

**Abstract.** This work is motivated by the desire to covertly track mobile targets, either animal or human, in previously unmapped outdoor natural environments using off-road robotic platforms with a non-negligible acoustic signature. The use of robots for stealthy surveillance is not new. Many studies exist but only consider the navigation problem to maintain visual covertness. However, robotic systems also have a significant acoustic footprint from the onboard sensors, motors, computers and cooling systems, and also from the wheels interacting with the terrain during motion. All these can jeopardise any visual covertness. In this work, we experimentally explore the concepts of opportunistically utilizing naturally occurring sounds within outdoor environments to mask the motion of a robot, and being visually covert whilst maintaining constant observation of the target. Our experiments in a constrained outdoor built environment demonstrate the effectiveness of the concept by showing a reduced acoustic signature as perceived by a mobile target allowing the robot to covertly navigate to opportunistic vantage points for observation.

**Keywords:** acoustic, covert, stealth, robot, tracking.

## 1 Problem Statement

Our research problem is focused towards addressing the question: can you maintain constant observation of a moving natural object of interest whilst being visually and acoustically covert? A solution is achieved by addressing covert movements to pseudo-optimal monitoring locations, and on reducing the conspicuousness of the robot's acoustics. In the first case, the robot examines its surroundings whilst monitoring the target object and identifies potential locations that offer a substantial

---

Ashley Tews · Matthew Dunbabin  
Autonomous Systems Laboratory, CSIRO ICT Centre,  
P.O. Box 883, Kenmore QLD, Australia  
e-mail: Ashley.Tews@csiro.au

opportunity for camouflage and observation. In the acoustic case, the robot monitors the periodicity of noise sources of significant amplitude and determines those that offer a high probability of covering any ego-noise, and are cyclic enough to be predictable.

Examples of significant distracting sounds for: built environments includes forklifts, trucks, compressors, and machinery; urban environments include cars, mobile phones and dogs; natural environments include birds, insects and wind. The prediction of upcoming noise events can act as a trigger to allow the robot to move to its next monitoring location. This concept significantly extends current state-of-the-art by considering; (1) maintaining stealth in an outdoor environment, (2) a dynamic target, (3) variable background and distractive sounds, and (4) the tracker must maintain constant visual observation of the target. In this paper, we present our preliminary research into visual and acoustic covertness as a step towards a fully predictive system.

## 2 Related Work

Stealthy navigation for robots has been typically focused towards minimizing the risk of exposure of the robot to known or unknown observers using either potential fields or cost functions. In unknown outdoor environments, Birgersson [1] and Tews [8] have used either approach for navigating a robot around detected objects in the presence of a known observer location. Marzouqui [6] has demonstrated a similar approach using a cost function to plan a path through an unknown environment that may contain sentries. The function chooses a path that offers the least exposure to open areas where the possibility of sentries was higher. In known environments, Masoud [7] has demonstrated a potential field approach for evasion in the presence of multiple pursuers.

The goal of these approaches was to reduce visibility to one or more target objects (typically referred to as “observers”). Our approach requires constant observation of a mobile target object. Cook [3] have developed a method to evaluate the information gain of strategic observation locations versus the cost of being detected. Their domain was more directed towards the scenario of military scouting from long ranges but has similarities to ours. Our approach is more dynamic, shorter range and considers the acoustic profile of the tracker and its surroundings.

There is a paucity of literature relating to the acoustic masking of a robot for stealthy navigation. The most significant work to date has been by Martinson [5], who describes an approach to use known noise sources within the environment to determine the locations where a robot’s own acoustic signature is minimized to a potential “listener” (known in this paper as the observer or target). Whilst the approach takes into account the listener’s audio direction sensitivity, the small scale of the experiment and constant noise source limits its practicality in the natural and built environments, particularly when the noise sources are non-constant and the listener is moving along an unknown trajectory. This paper significantly extends these concepts for allowing operation in previously unmapped large-scale natural or built

environments. Principally, the approach considers the combined visual and acoustic covertness of the robot, whilst maintaining constant observation of the target. The acoustic covertness is achieved by opportunistically utilizing natural sounds from within the environment as distractions for masking its own acoustic signature to manoeuvre between goal locations. These sounds are characterized to determine the level of masking in both time and space. Sound characterization can be performed in a number of ways such as matching point features [2] or Markov model based clustering [4] for example. The temporal duration of masking of a particular sound can be estimated by studying the soundscape and developing a likelihood table. This soundscape characterization is beyond the scope of this paper and it assumed in this experimental campaign that the masking distractions have known minimum duration, although can occur randomly.

### 3 Technical Approach

Our system consists of two major parts; stealthy navigation and acoustic masking in an unknown outdoor environment. The key assumptions in this analysis are; (1) the robot knows its own location relative to a reference frame, (2) the robot knows the location of the target/s relative to its own coordinate system, (3) the robot is self aware of its acoustic output (not necessarily constant), and (4) the robot has sensing capabilities to measure the local sound field and directional components.

**Stealthy Navigation:** The navigation part of the system is motivated by the requirement that the tracking robot needs to be able to monitor the target at all times, whilst remaining as inconspicuous as possible. It is reasonable to consider that a naive target en route to a goal location will not be distracted by moving objects it cannot see or hear. In our system where the environment is not known a priori, the tracking robot will determine opportunistic observation locations in response to the environment state. Visible detection of the robot occurs through it being conspicuous against its background by *movement* and *appearance*. For this preliminary study, we assume the robot is camouflaged against background objects when stationary. Any movement should be masked acoustically and undertaken when the robot is not in visible angular range of the target. Hence, if the target is moving forward, the robot is clear to move behind it, but should not undertake in prolonged continuous movements which increase its chances of being detected acoustically and visibly. It should effectively “hop” between vantage locations as appropriate to maintain continuous observation and acoustic cloaking.

The vantage locations are determined using an occupancy grid and assigning a cost to each occupied cell that is visible to the target. All other cells are ignored since only locations between an occupied cell (object) and the target are considered as viable candidates. Cells within 4m of the target are also excluded as these are deemed too close to allow stealthy traverses. The cost function is based on the grid location of the cell being evaluated and those containing the target and the tracker:

$$cost_{xy} = 0.5 * |cell_{xy} - tracker_{xy}| + |\alpha - |cell_{xy} - target_{xy}|| \quad (1)$$

where  $\alpha$  is set to 6m.

The distance from the cell to the tracker is used to bias locations close to the tracker to reduce distant goals being selected. In future iterations of the equation, target heading and time-to-goal will be included. The bias of 6m to the target in the second part of the equation is selected as the ideal viewing range for the sensor monitoring the target.

The resulting cost grid is analysed and the lowest valued cell chosen as the candidate goal location. A small amount of hysteresis is used so if the goal is within 3m of the previous, it will not change. This is to prevent the robot from thrashing between goals and creating more visual and acoustic disturbance.

Once a new goal location is selected, the robot waits for the signal that it is acoustically masked (described next) before it moves. For the current instantiation of the research, the robot will continue onto the goal irrespective of any decrease in masking noise level.

**Acoustic Masking:** The approach for determining when the robot should move is to predict the expected sound pressure level (SPL) increase at the target. We assume the background SPL is reverberant and locally the same at both the robot and target, and adopt a spherical spreading model to estimate the direct sound of the tracker to the target. Using these sound field approximations, two estimation steps are conducted onboard the robot either offline or when the robot is not in tracking mode; (1) adaptive estimation of the background SPL, and (2) estimation of the robot's own ego-motion noise.

Although we assume the background SPL is reverberant and the same at the target and tracker, it can vary temporally to account for time of day and general background noise sources. The background SPL estimate ( $\hat{L}_{BG}$ ) is determined whilst the tracker is stationary by the recursive moving average approximation:

$$\hat{L}_{BG_k} = \min \left\{ \frac{L_k + (n-1)\hat{L}_{BG_{k-1}}}{n}, L_k \right\} \quad (2)$$

where  $L_k$  is the instantaneous measured SPL at time-step  $k$  and  $n$  is the number of time-steps for the average. As any intermittent high-level distractions can bias the background estimate, we force  $\hat{L}_{BG}$  to the lower of the estimate and actual level.

The second estimation is conducted whilst the tracker is moving within the environment. Here we assume that the local sound field only consists of the background and that produced by the tracker. Therefore, we map the tracker's own SPL as a function of its velocity ( $v$ ) and turn rate ( $\dot{\psi}$ ) such that:

$$\hat{L}_R(v, \dot{\psi}) = 10 \log \left( 10^{L_P/10} - 10^{\hat{L}_{BG}/10} \right) \quad (3)$$

where  $L_P$  is the peak sound pressure recorded during ( $t_1 \leq k \leq t_2$ ) given by:

$$L_P = \max [L(k)]_{k=t_1}^{k=t_2} \tag{4}$$

The second phase of the acoustic tracking is prediction of the tracker’s contribution to the target’s observed SPL. For this analysis, we assume we know the approximate relative direction of the distraction sound from the robot to the target ( $\theta$ ). The estimated total SPL at the target is then the logarithmic sum of the direct and reverberant SPL’s. Therefore, the estimate of the robot’s SPL at the target is given by:

$$\hat{L}_{R/T} = 10 \log \left( 10^{(\hat{L}_R(v, \psi) - 20 \log(r/r_o))} + 10^{\hat{L}_{BG}/10} \right) \tag{5}$$

where  $r$  is the range from the robot to the target, and  $r_o$  is the reference distance from the robot’s sound source to its onboard sound pressure level meter.

In this study, we consider the masking provided by the observed increase in SPL above background at the target. To be a valid distraction, we need to estimate its increase in the target’s SPL. The robot observes the instantaneous increase in SPL and determines the strength of the distraction at the robot to be:

$$L_{D/R} = 10 \log \left( 10^{L_k/10} - 10^{\hat{L}_{BG}/10} \right) \tag{6}$$

We assume a disturbance is a temporally distinct point source and is modelled via spherical spreading model. We also assume that it originates beyond the target (i.e. not between the tracker and target). Therefore, the estimate of the distraction SPL at the target is:

$$\hat{L}_{D/T} = \begin{cases} 10 \log \left( 10^{(L_{D/R} + 20 \log(r/r_o))} + 10^{\hat{L}_{BG}/10} \right) & , \text{ if } \cos(\theta) > 0 \\ 10 \log \left( 10^{(L_{D/R} - 20 \log(r/r_o))} + 10^{\hat{L}_{BG}/10} \right) & , \text{ otherwise} \end{cases} \tag{7}$$

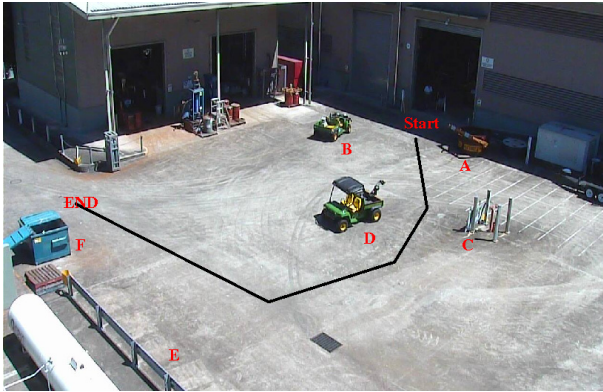
In practice, the robot does not move unless  $\cos(\theta) > 0$ , as the directionality of the distraction and robot sound could direct attention towards the tracker/robot.

The final step is to predict when the robot considers itself to have sufficient masking to avoid acoustic detection. If the expected SPL of the robot at the tracker including background level is less than 1 dB less than the expected total SPL of the robot, background and distraction at the target, then we assume the robot will be acoustically masked and flags the robot’s controller that waypoint navigation is possible. The expected duration of acoustic masking is dependent on the type of distraction, its persistence and can be pre-specified or predicted. In this preliminary experimental evaluation, we consider a weighted temporal difference of ( $\hat{L}_{D/T} - \hat{L}_{R/T}$ ). This simplification assigns the maximum expected level and duration of distraction. This is a gross assumption and it is a current topic of research to predict the distraction parameters more accurately based on the frequency content and likelihood from learnt soundscape parameters.



### 4 Experiments

To demonstrate the effectiveness of acoustic masking for a mobile tracking robot, we conducted experiments with a human target moving across a built environment with a robot tracker. In this preliminary approach, the human walks between and near objects around an open area, as shown by the black path in Figure 1. The robot will pursue the human by going to goal locations near objects as they are detected and offer substantial vantage points for observation. Fig. 1 shows the 30x30 m experimental area showing the human target’s path from Start to End, and potential objects (labeled A-F) for the robot to use whilst tracking the person.



**Fig. 1** The outdoor experimental scenario. The target roughly follows the black path around obstacles A-F.



**Fig. 2** The robot used in the experimental campaign showing the sound pressure level meter, Hokuyo laser scanner and computing hardware

The robotic platform used was a Jaguar 4x4 developed by Dr Robot ([www. dr-robot.com](http://www.dr-robot.com)) fitted with a top-mounted Hokuyo 30LX laser, Microstrain 3DMG GX3 IMU and externally mounted laptop running the software for sensing and control (Figure 2). Two sound recording devices were installed on the robot at different phases of the experimental campaign. The first was a calibrated Sound Pressure Level Meter with an analog output proportional to the measured SPL in dBA. The second was a MP3 recorder used to measure and record the sound level, and was carried at a constant orientation by the target. As this is not a calibrated sound level meter, the results shown in Figures 4 and 7 are relative to an arbitrarily chosen reference. The MP3 player was adopted to allow frequency content analysis of the soundscape for characterisation and prediction of distraction sounds (not considered in this paper). A series of day and night-time experiments with and without acoustic distractions were conducted in the area shown in Figure 1. During the day, the area has a large number of sound source distractions from vehicles, compressors, and personnel in the vicinity. It is in a semi rural location and there are also many sources of significant environmental noises such as wind, birds, insects and frogs, particularly at night after the domestic noises have reduced.

Due to the acoustic variability of the experimental environment and the fact we used a non-standard audio recording device, we conducted a number of tracking experiments at night with low background noise levels and no distractions. This allowed the robot's acoustic signature to be estimated at different velocities, orientations and wheel slip conditions.

## 5 Results

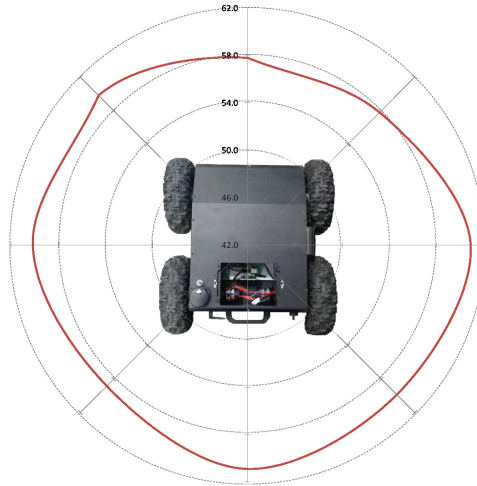
A number of experiments were conducted to acoustically characterise the robot and validate the modelling assumptions. Figure 3 shows the measured SPL at 1m around the robot (held off the ground with motors running). As seen, the robot's SPL is relatively consistent with direction.

Figure 4 shows the estimated SPL at 1m from the robot with varying velocities. For estimating the expected SPL at the target, a conservative acoustic signature of the robot was adopted as shown by the black line.

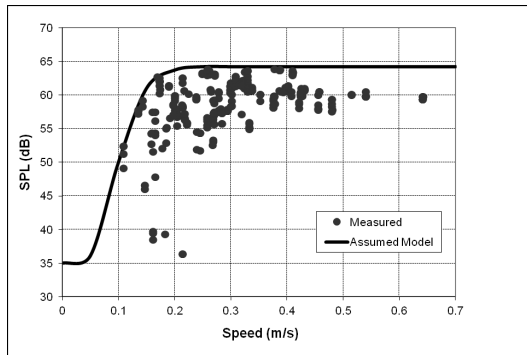
To validate the modelling assumptions, the robot was commanded to undertake turns and linear velocity runs at set distances from the target, with the SPL level recorded at the robot and target in a built environment. Figure 5 shows the results of the SPL recorded at the target and that predicted by the robot.

As seen in Figure 5, the robot predicts the SPL at the target reasonably well for both the turning and linear velocity cases. The main error results from the differences in background SPLs at the robot and target showing that in more cluttered environments, the less predictable the local sound fields.

Figure 6 shows the results of one of the daytime experiments with the robot tracking a moving target. The paths of the target and tracker are similar between the objects in the environment. The blue vantage points were generated



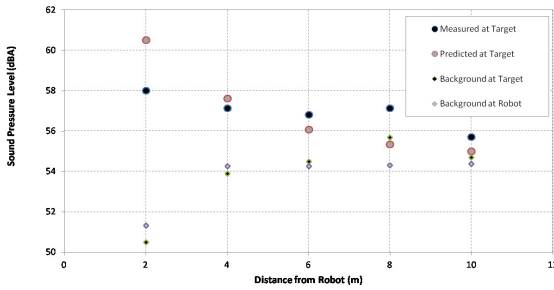
**Fig. 3** Measured SPL at 1m around the robot platform with motors running (no ground contact)



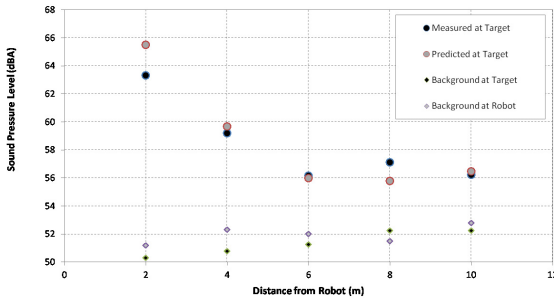
**Fig. 4** Variability of estimated robot SPL at 1m with robot speed, taken from the night time experiments. The black line shows the assumed conservative acoustic SPL model adopted in this experimental campaign.

opportunistically as the robot uncovered new information about the environment whilst tracking the target. Each was generated behind the target’s location.

To evaluate the acoustic masking of the robot, a distracting sound source was introduced into the environment from object D’s location in Figure 1 for random lengths of time. When this sound is detected, the robot moves to the next vantage point. The acoustic covertness of a daytime scenario shown in Figure 6 is illustrated in Figure 7. The top trace shows the measured total SPL (background, robot and distraction) at the target (with a 2 second running average), the second trace shows the range from the target to the tracker, the third trace shows the robot’s



(a) Straight

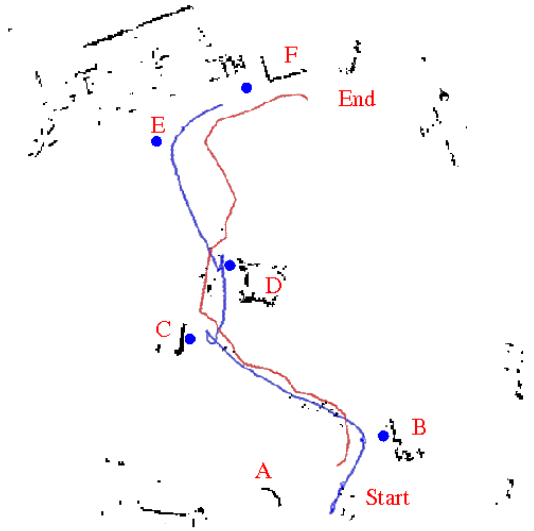


(b) Turning

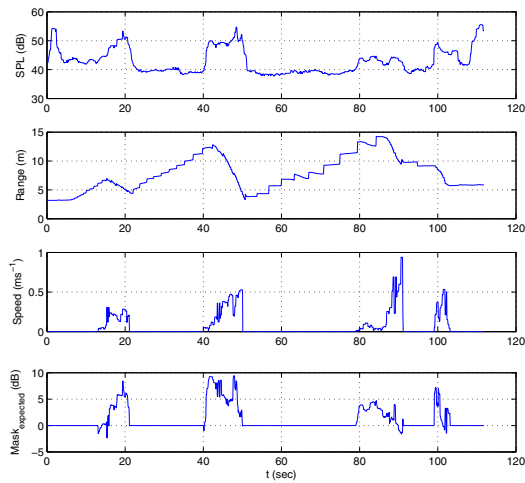
**Fig. 5** Measured and predicted sound pressure levels at the target from the robot turning and moving linearly. Also shown is the background SPL estimate measured at the robot and target.

speed indicating a manoeuvre between waypoints, with the lower trace showing the acoustic difference from the maximum expected masking potential. An expected masking potential greater than -1 dB is the threshold that indicates that the robot is acoustically masked at the target.

From three daytime experiments, it was determined that the distractions were effective at conservatively masking the robot from the target a minimum of 97% of the time whilst moving between goal points. Further night time experiments show that without acoustic distractions, the robot is 100% detectable (0% masked) even at distances up to 30 m from the target. Even with a moderate level of acoustic distraction (a whistling target) with the same scenario as above, the robot was only masked for 56% of the time whilst moving.



**Fig. 6** A map generated from the tracking robot with paths and locations overlaid. The target's path is shown in red and went from Start to End. The tracking robot's path is shown in blue. The blue circles represent the vantage points generated during the traverse and the letters correlate to the objects' locations shown in Fig. 1.



**Fig. 7** Experimentally evaluated acoustic covertness from the experimental scenario. (Upper) the measured SPL at the target, (second) the range from the tracker to the target, (third) the robot's speed, and (lower) the acoustic difference for masking of the robot at the target.

## 6 Discussion

This experimental evaluation of visual and acoustic covertness in a built environment has demonstrated that it is possible to navigate and track a naive mobile target of interest whilst being undetected. Furthermore, opportunistically utilizing the acoustic soundscape, however random, can greatly enhance the ability of a robot to stealthily navigate, as illustrated by the experimental results. The reactive cost function and its associated algorithm provide rational vantage points for tracking and monitoring the target. During all experiments, the selected opportunistic locations were in locations out of view of the target's gaze, but close enough to provide a strategic monitoring location for the next phase of target observation.

During the experimental campaign, it became evident that in order to accurately predict the expected SPL of the robot at the target, it was necessary to (1) reasonably understand the robot's acoustic signature as a function of speed and terrain, and (2) monitor the background SPL and variability in real time. Fortunately, the background level can be established whilst the robot is stationary, waiting for the next opportunity to go to the next vantage location. Secondly, although not considered in this paper, it is important to be able to predict the location of the distracting sound, as well as model the expected "attention focus" of the target to this sound. This is important in situations where the robot is required to traverse large distances between waypoints whilst being potentially visible to the target. Additionally, we believe higher level heuristics are required to modify the stealthy navigation cost function based on the time of day, weather conditions, and dynamic soundscapes.

Anecdotally, the performance of the acoustic and visual stealthy behavior presented was very effective. The targets noted that if they were not watching the robot, they could not tell when it had moved to a new location and hence was tracking them. We have made several assumptions regarding the target and environment to undertake this preliminary research. In particular, experiments were only conducted in one environment, and the target had a relatively simple trajectory with a fixed gaze. Different environments, different targets, gaze detection and prediction are elements of future work.

## References

1. Birgersson, E., Howard, A., Sukhatme, G.S.: Towards stealthy behaviors. In: Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 2, pp. 1703–1708 (October 2003)
2. Chu, S., Narayanan, S., Jay Kuo, C.-C.: Environmental sound recognition using mp-based features. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008, pp. 1–4 (April 2008)
3. Cook, D.J., Gmytrasiewicz, P., Holder, L.B.: Decision-theoretic cooperative sensor planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(10), 1013–1023 (1996)
4. Lee, K., Ellis, D.P.W., Loui, A.C.: Detecting local semantic concepts in environmental sounds using markov model based clustering. In: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), March 2010, pp. 2278–2281 (2010)

5. Martinson, E.: Hiding the acoustic signature of a mobile robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007, pp. 985–990 (November 2007)
6. Marzouqi, M., Jarvis, R.: Covert robotics: Covert path planning in unknown environments. In: Proceedings of Australasian Conference on Robotics and Automation (ACRA 2004) (December 2004)
7. Masoud, A.A.: Evasion of multiple, intelligent pursuers in a stationary, cluttered environment using a poisson potential field. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2003, vol. 3, pp. 4234–4239 (September 2003)
8. Tews, A., Mataric, M.J., Sukhatme, G.S.: Avoiding detection in a dynamic environment. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2004, vol. 4, pp. 3773–3778 (September-October 2004)

# Part XII: ISER Session Summary on “Marine Robotics”

Daniela Rus

Massachusetts Institute of Technology

## Session Summary

The Marine session introduced five state-of-the-art results in underwater robotics, with a focused emphasis on adaptation and learning to enable improved operation of robots under challenging situations.

This first paper, "Autonomous Adaptive Underwater Exploration using Online Topic Modeling", describes an autonomous robotic system that can be used to assist in exploring dangerous underwater environments. The key contribution is an online spatio-temporal approach to topic modeling which captures the presence of high-level patterns in the scene to capture the objects commonly found in the sea. Given a fixed trajectory, the robot traversed it with a non-uniform speed, stopping at locations containing surprising observations, and moving at high speeds over seemingly boring or previously observed regions. Experiments show that the resulting summaries capture the visual diversity of the underwater environment.

The second paper, "Active and Adaptive Dive Planning for Autonomous Seafloor Reconstruction", examines how to plan dives for an Autonomous Underwater Vehicle (AUV) to generate a dense bathymetric map using sidescan sonar, where the main goals are (1) proper modeling of the local uncertainty of the 3D reconstruction, (2) efficient dive planning to reduce this uncertainty, and (3) determination of when to re-plan adaptively based on new information. The proposed solution uses non-parametric Bayesian regression to model the expected accuracy of the map, which provides principled cost functions for planning subsequent dives. Experiments on the propeller-driven YSI EcoMapper AUV equipped with a sidescan sonar in an inland lake demonstrate that efficient dive planning has significant performance gain over standard lawn-mower patterns.

The third paper, "Exploring Space-Time Tradeoffs in Autonomous Sampling for Marine Robotics", aims to quantify the idea that following a drifting device in the real ocean and using a pseudo-Lagrangian context for field reconstruction may improve the quality of location estimation. The hypothesis is that a field which was initially not separable in space and time could be made approximately separable by a complete or partial transformation into drifter-relative coordinates. Simulation studies under idealized conditions support this idea. However, the results from field experiments in coastal waters, were less clear and point to several hypothesis for further investigation.

The fourth paper, "Autonomous, Localization-Free Underwater Data Muling using Acoustic and Optical Communication", describes a novel apparatus and



scheme for data muling between a robot and a sensor node whose underwater location is fixed. Acoustic communication is used by the robot to approximately locate the sensor node. Optical communication is used to precisely locate the node and establish a wireless communication link capable of high data rates. The algorithms have been demonstrated in practice in a wide range of natural environments.

The fifth paper, "Local-Search Strategy for Multi-Modal, Multi-Target, Active Localization of Invasive Fish" considers the localization of radio-tagged fish aggregations with robots. The paper proposed a local search-based method to estimate the location of a single target based on observations from field experiments, followed by extensions to multiple targets. Precisely localizing a radio-tagged target given an initial estimate is challenging to limited and unknown range, large measurement time, and ambiguity in bearing measurements. The solution uses an intermediate initialization phase to transition from search to localization. The resulting algorithm was validated during field experiments where a robot successfully detects, initializes and then localizes nearby targets.

These five papers enable increased perception, planning, control, and communication capabilities for underwater robots. The controllers involve novel theory validated by hardware. The results are exciting and bring us closer to the dream of autonomous underwater observatories.

# Autonomous Adaptive Underwater Exploration using Online Topic Modeling

Yogesh Girdhar, Philippe Giguère, and Gregory Dudek

## 1 Introduction

Exploration of underwater environments, such as coral reefs and ship wrecks, is a difficult and potentially dangerous tasks for humans, which naturally makes the use of an autonomous robotic system very appealing. This paper presents such an autonomous system, and shows its use in a series of experiments to collect image data in an underwater marine environment. We presents novel contributions on three fronts. First, we present an online topic-modeling based technique to describe what is being observed using a low dimensional semantic descriptor. This descriptor attempts to be invariant to observations of different corals belonging to the same species, or observations of similar types rocks observed from different viewpoints. Second, we use the topic descriptor to compute the *surprise* score of the current observation. This is done by maintaining an online summary of observations thus far, and then computing the surprise score as the distance of the current observation to the summary, in the topic space. Finally, we present a novel control strategy for an underwater robot that allows for intelligent traversal; hovering over surprising observations, and swimming quickly over previously seen corals and rocks.

Exploration, in the context of robotics, has been studied before. Work has been done on autonomous mapping of challenging environments [19], [12], frontier expansion [22], minimizing uncertainty [21], and utility based exploration[8]. Our focus is instead on traversing an environment similar to how a tourist might do so in a new city; stopping and recording any surprising sights, while moving fast when

---

Yogesh Girdhar · Gregory Dudek

Centre for Intelligent Machines, McGill University

318-3480 University Street, Montreal, QC, Canada, H3A 0E9.

e-mail: {yogesh, dudek}@cim.mcgill.ca

Philippe Giguère

Département d'informatique et génie logiciel, Université Laval

1065, Avenue de la Médecine, Quebec, QC, Canada, G1V 0A6.

e-mail: philippe.giguere@ift.ulaval.ca

nothing new is in sight. This is similar to the vacation snapshot problem described in [2].

We used an untethered amphibious robot (Aqua[15]) with an in-house designed autopilot to carry the exploration task. Images were taken with a downward-looking camera, with all computations performed onboard. Its propulsion is based on six flippers that can provide motion in five degrees of freedom. By using a novel combination of gaits, the robot was able to move at various speeds while maintaining its orientation, despite external disturbances. This was necessary in order to complete this exploration task.

## 2 Approach

The abstract task of controlling the motion of a robot, based on surprise, can be broken down into three sub-problems: computing surprise of a new observation, having a meaningful descriptor for observations over which surprise could be computed, and finally, controlling the robot given the surprise score of the current observation. We describe these sub-problems in the following sections.

### 2.1 Summaries and Surprises

Summarizing observations made by a robot has recently gained popularity in robotics[13, 6]. Our goal, however, is to compute a summary which assists in evaluating the novelty of a new observation. We do this by maintaining a summary that is representative of all of observations made thus far, and then compute the surprise score as the distance to this summary.

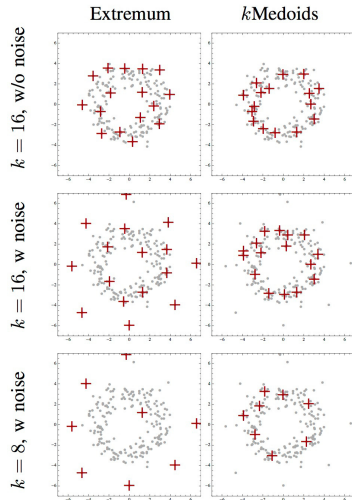
Let  $\mathbf{M}^t = \{M_1, \dots, M_t\}$  be the set of all observations till time  $t$ . We maintain a subset of  $k$  observations as the summary  $\mathbf{S} = \{S_1, \dots, S_k\}$ ,  $\mathbf{S} \subseteq \mathbf{M}^t$ , such that the maximum distance of an observation to its closest summary sample is minimized. The cost function is thus defined as:

$$\text{Cost}(\mathbf{S}|\mathbf{M}^t) = \max_i \min_j d(M_i, S_j), \quad (1)$$

where  $d$  is the distance function, which measures distance as the symmetric KL divergence between the corresponding topic distributions, which we describe in Sec 2.2. Such a summary is sometimes called as an *Extremum Summary* [6], because minimizing the above cost function is essentially minimizing the distance of the worst outlier to the summary. This is different from a more typical  $k$ -medoids clustering based summary, which tries to minimize the mean distance of an observation to the closest summary.

The novelty or surprise of a new observation  $\xi(Z_t|\mathbf{S})$  can then be defined as its Hausdorff distance to the summary [5].

$$\xi(M_t|\mathbf{S}) = \min_j d(M_t, S_j). \quad (2)$$



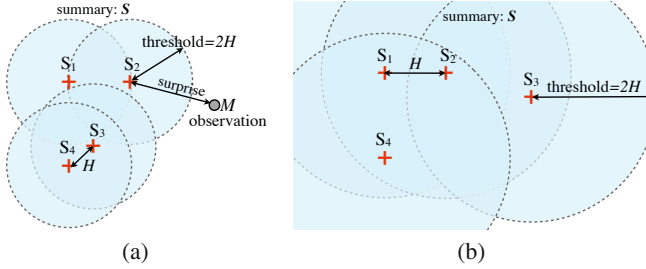
**Fig. 1** Extremum vs  $k$ -Medoids Summaries. The dataset consists of 200 points generated randomly around a circle in  $\mathbb{R}^2$ . The summaries generated by the two algorithms are shown in the first row. Since there are no outliers in the dataset, the summaries seem similar. In the second row, we add 8 extra samples from a different distribution, which are all outliers in the context of the other points. Adding these outliers highlights the differences between the two strategies. We see that extremum summary favors picking the outliers, whereas the  $k$ -medoids summary ignores these outliers completely. In the last row, we reduce the summary size and see the differences exaggerated even more. The extremum summary is almost entirely made up of the outliers, whereas the  $k$ -medoids summary is only representative of the mean.

If the distance function obeys the triangle inequality, which is true in our case, then not only is this problem NP-hard, but Huse and Nemhauser [11] showed that any  $\alpha$ -approximation of this problem is also NP-hard for  $\alpha < 2$ . Gonzalez [7] proved that the simple greedy solution of recursively picking the farthest samples, has an approximation ratio of 2, which is likely the best we can do unless P=NP.

In the online case, Charikar et al. [4] have proposed a simple strategy where after each pick, the picking threshold is doubled. This leads to a summary which is guaranteed to have a cost less than  $8 \times$  'optimal'. However, since the topic assignment of samples in the summary are continuously being refined, we instead set the threshold dynamically to  $2 \times$  'minimum inter-sample distance in  $\mathbf{S}$ ', as illustrated in Fig. 2.

To control the summary size, we simply use the greedy offline summarization algorithm on the summary to keep the summary of desired size. In our prior work[6], we have studied the rate of growth of the summary, when threshold is set to the mean distance of a summary sample, to the remaining summary. This is useful in the case when we want the summary size to grow with the data.

Figure 1 highlights the characteristic difference in summaries generated by the extremum summary algorithm, and the  $k$ -Medoids algorithm. The summaries



**Fig. 2** (a) Given a summary, represented here by the ‘+’ sign, we define the threshold score for updating the summary as twice the smallest inter-sample distance  $H$ . When a new observation  $M$  arrives, we compute its surprise: the distance to the closest summary sample. If the surprise exceeds the threshold  $2H$ , then the summary is updated to include the new observation. The updated summary and the threshold are shown in (b).

generated by the two algorithms are shown in the first row. Since there are no outliers in the dataset, the summaries seem similar. In the second row of Figure 1, we add 8 extra samples from a different distribution, which are all outliers in the context of the other points. Adding these outliers highlights the differences between the two strategies. We see that extremum summary favors picking the outliers, whereas the  $k$ -medoids summary ignores these outliers completely. In the last row of Figure 1, we reduce the summary size and see the differences exaggerated even more. The extremum summary is almost entirely made up of the outliers, whereas the  $k$ -medoids summary is still only representative of the mean.

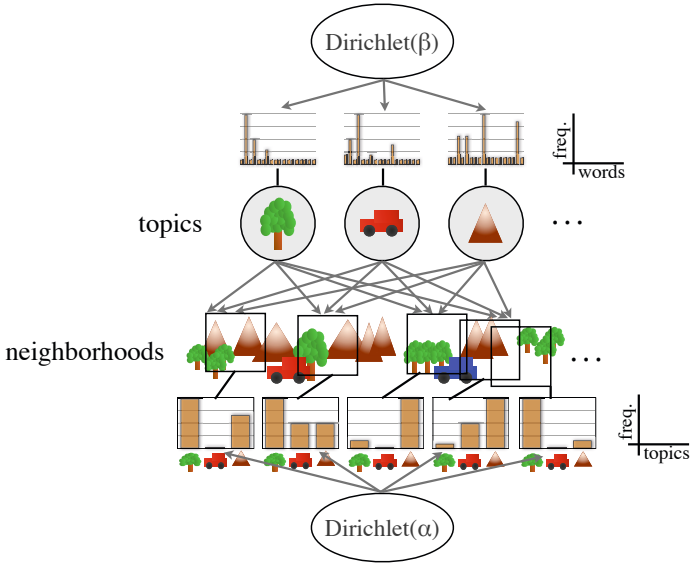
Although a  $k$ -medoids summary might be useful when we want to model the mean properties of an environment, if however we are interested in identifying the range of what was observed, then an extremum summary is more useful, since its objective function ensures that each observations is close to at least one of the summary samples.

## 2.2 Online Spatiotemporal Topics

To have meaningful summaries, and thus a meaningful surprise score, we must use an image descriptor that is sensitive to thematic changes in the scene, while being immune to low level image changes. We do this via the use of a topic modeling framework, using which we describe an incoming observation with a low dimensional distribution over topics.

### 2.2.1 Topics

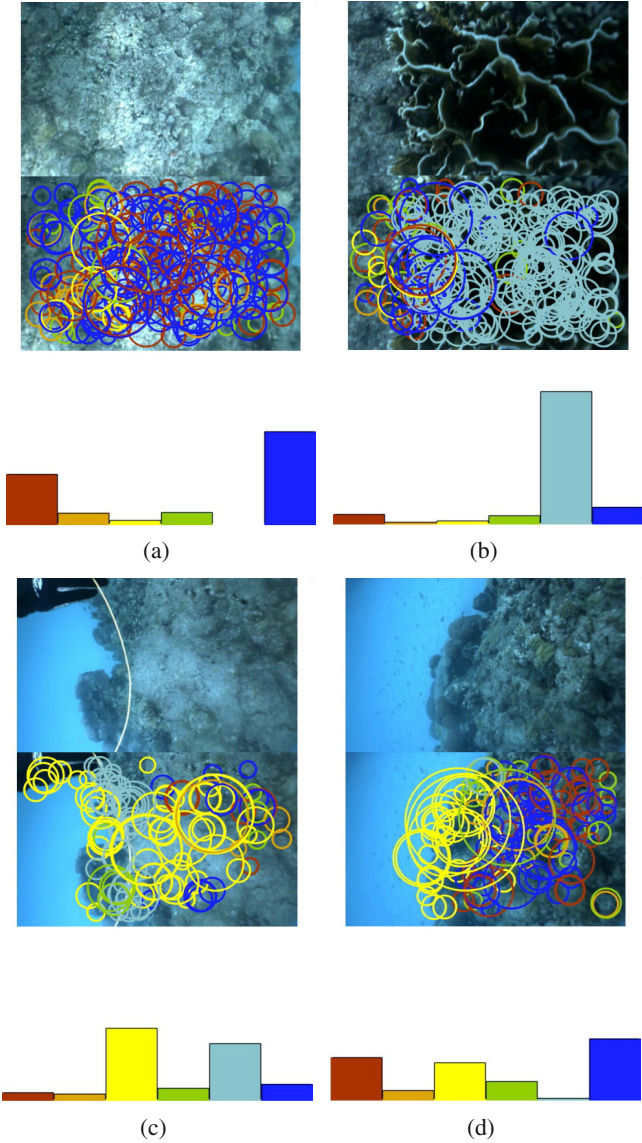
Topic modeling methods were originally developed for text analysis. Hoffman [10] introduced the idea of probabilistic Latent Semantic Analysis (pLSA) for text documents, which modeled the probability of observing a word  $w_i$  in a given document  $m$  as:



**Fig. 3** Spatiotemporal Topics: As a robot observes the world, we would like its observations to be expressed as a mixture of topics with perceptual meaning. We model the topic distribution of all possible overlapping spatiotemporal regions or neighborhoods in the environment, and place a Dirichlet prior on their topic distribution. The topic distribution of the current observation can then be inferred given the topic labels for the neighborhoods in the view. Modeling neighborhoods allows us to use the context in which the current observation is being made to learn its topic labels. To guarantee realtime performance, we only refine a constant number of neighborhoods in each time step, giving higher priority to recently observed neighborhoods.

$$\mathbb{P}(w_i|m) = \sum_{k=1}^K \mathbb{P}(w_i|z_i = k)\mathbb{P}(z_i = k|m). \tag{3}$$

The central idea being the introduction of a latent variable  $z$ , which models the underlying topic, or the context responsible for generating the word. The topic variable  $z$  takes a value from  $1 \cdots K$ , where  $K$  is a small number compared to the size of the vocabulary. The distribution of these topics in a document gives us a low dimensional semantic description of the document. The words in our context corresponds to visual words[17], and documents correspond to observed images. Latent Dirichlet Allocations[1] improve upon pLSA by setting Dirichlet priors on the distribution of words in a topic, and the distribution of topics in a document.



**Fig. 4** Example of topics learned on images of the ocean floor taken by the Aqua robot, for a single trajectory. Each visual word is marked by a circle, the size of which corresponds to the size of the visual feature. Histograms depicting the content of each color-coded topic are shown below.

### 2.2.2 Spatiotemporal Topics

In this work, we use a novel online spatiotemporal LDA for computing topic labels, which not only is able to efficiently converge the topic labels for a new observation,

but also update topic labels for previous observations in the light on new incoming data. Moreover, we take into account the location of the observed visual words during the refinement, using them to compute topic priors which are sensitive to changes in time and location of the robot. Our algorithm's iteration runs in constant time, and hence is suitable for realtime use on autonomous vehicles with limited computing capabilities. Figure 4 shows examples of topics which were learnt by running the above topic model on an underwater image sequence containing 2000 images.

Each observation is a tuple  $(w_i, \mathbf{x}_i, t_i)$  consisting of an observed visual word  $w_i = 1 \dots V$  from a fixed vocabulary of size  $V$ , associated coordinates  $\mathbf{x}_i$ , and timestamp  $t_i$ . In the presence of location information,  $\mathbf{x}_i$  is the location of the word  $w_i$  in world coordinates. In absence of location information, we can simply use the pixel coordinates of the visual words. The neighborhood of an observation at  $(\mathbf{x}, t)$ , denoted by  $G(\mathbf{x}, t)$ , is the set of observations in its spatiotemporal neighborhood. This could either be defined using  $k$  nearest neighbors, or using a radius search. Instead of computing topic distributions over documents in a traditional LDA [1], or image windows in Spatial-LDA [20], we compute topic distributions over these spatiotemporal neighborhoods (Figure 3). Modeling topic distribution over neighborhoods allows us to use spatiotemporal context in which an observation is being made, which in turn results in much faster convergence as is shown later in our results.

Given a location and time  $(\mathbf{x}_i, t_i)$ , we use the following generative model for the corresponding observed word  $w_i$ :

1. Word distribution for each topic  $k$ :  $\phi_k \sim \text{Dirichlet}(\beta)$ ,
2. Neighborhood for an observation at  $(\mathbf{x}_i, t_i)$ :  $G(\mathbf{x}, t) \sim$  uniformly from all neighborhoods which contain  $(\mathbf{x}_i, t_i)$ ,
3. Topic distribution the neighborhood  $G(\mathbf{x}, t)$ :  $\theta_{G(\mathbf{x}, t)} \sim \text{Dirichlet}(\alpha)$ ,
4. Topic label for location  $(\mathbf{x}_i, t_i)$ :  $z_i \sim \text{Discrete}(\theta_{G(\mathbf{x}, t)})$ ,
5. Word observed at location  $(\mathbf{x}_i, t_i)$ :  $w_i \sim \text{Discrete}(\phi_{z_i})$ ,

where  $x \sim Y$  implies that random variable  $x$  is sampled from distribution  $Y$ .

### 2.2.3 Gibbs Sampling

Similar to the Gibbs sampler proposed by Griffiths et al.[9], we can define the posterior topic distribution of a neighborhood  $G(\mathbf{x}, t)$ :

$$\mathbb{P}(z_i = k | \mathbf{z}_{-i}, w_i = w, \mathbf{w}_{-i}, G(\mathbf{x}, t)) \propto \frac{n_{k,-i}^w + \beta}{\sum_{w=1}^V (n_{k,-i}^w + \beta)} \cdot \frac{n_{G(\mathbf{x}, t), -i}^k + \alpha}{\sum_{k=1}^K (n_{G(\mathbf{x}, t), -i}^k + \alpha)}, \quad (4)$$

where  $n_{k,-i}^w$  counts the number of words of type  $w$  in topic  $k$ , excluding the current word  $w_i$ , and  $n_{G(\mathbf{x}, t), -i}^k$  is the number of words with topic label  $k$  in neighborhood  $G(\mathbf{x}, t)$ , excluding the current word  $w_i$ .

Several different strategies exist in the literature to do online refinement of the topic assignment in a given streaming dataset [18, 3]. The general idea is to initialize



the topic label of the current observation with random labels, and then do a batch refinement of the entire dataset. This allows for previous topic assignments to be updated in the light of new observed data. Convergence is guaranteed because in the limit of time going to infinity, the algorithm behaves like a batch Gibbs sampler.

Since the number of neighborhoods is extremely large, we cannot refine word labels individually. In offline mode, sampling a neighborhood uniformly from all neighborhoods, computing its topic distribution, and refining all the word labels (using Eq 4) in it will approximate the proposed generative model. Online, we can do the same, however, in a robotics context, the number of refinements between two observation needs to be constant. We propose to sample the neighborhoods from a Beta( $a, 1$ ) distribution, with  $a > 1$ , giving higher picking probability to recently observed regions. This ensures that new observations quickly converge, while older observations are less likely to change their labels. In this work, we set  $a = 2$  for all experiments, however, increasing the value of  $a$  with time might lead to better results for long experiments. Algorithm 1 shows an iteration of the topic refinement strategy.

```

while no new observation do
   $W \leftarrow |\mathbf{w}|$  (* total number of observed words *)
  Randomly sample  $r \sim \text{Beta}(a, 1)$ 
   $i \leftarrow \lfloor W * r \rfloor$  (* index of the observation in the center of the neighborhood *)
  foreach  $j$  in  $G(\mathbf{x}_i, t_i)$  do
    (*update the topic label for each observation in the neighborhood *)
     $z_j \sim \mathbb{P}(z_j = k | \mathbf{z}_{-j}, w_j = w, \mathbf{w}_{-j}, G(\mathbf{x}_i, t_i))$ 
  end
end

```

**Algorithm 1.** REFINETOPICS ( $\mathbf{z}, \mathbf{X}, \mathbf{w}, \mathbf{t}$ ). Refine topic labels, given the current assignment of topics ( $\mathbf{z}$ ) for the set of all observed words ( $\mathbf{w}$ ), their locations ( $\mathbf{X}$ ), and observation times ( $\mathbf{t}$ ).

### 2.3 Robot Control

Let  $q(t) = \xi(M_t | \mathbf{S}) / 2H$  be the normalized surprise score of an incoming observation at time  $t$ . We then set the speed ( $v$ ) of the robot by mapping the surprise score through a sigmoid function:

$$v(t) = \frac{1}{1 + e^{-\gamma(q(t) - 0.5)}}, \quad (5)$$

where  $\gamma$  controls the *responsiveness* of the robot. A higher  $\gamma$  made the scheduling of the forward velocity  $v$  more aggressive. We calculated  $\gamma$  empirically, and found  $\gamma = 10$  to perform well during our sea trials.

The attitude (pitch, roll and heading) of the vehicle were kept stable during the experiments via simple PD controllers. In order to maintain depth, two different strategies were employed. Previously [14], we employed pitch corrections to maintain constant depths. This strategy was viable at high forward constant velocities. For the experiments in this paper however, the robot had to be able to control its depth, even when the speed  $v$  was 0. Early trials confirmed that at lower speeds ( $v < 0.2$ ), controlling depth by changing the pitch angle was not possible. This is due to the fact that depth change via pitching is only possible for significant forward velocity  $v$ . Otherwise, the robot simply pitches up. Consequently, a heaving motion from a previously developed hovering gait [16] was used to maintain depth at lower speeds, or when the robot was standing nearly still over 'interesting' areas.

### 3 Results

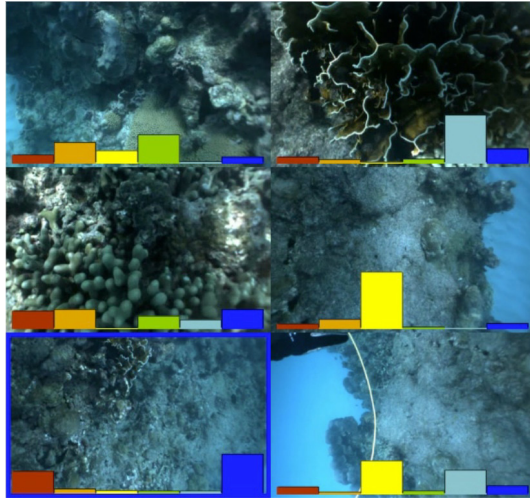
We tested the proposed underwater system above a coral reef, in open waters. This demonstrated the applicability of our system to a real-life scenario that included unpredicted water currents, image noise due to floating particles (sediment and plankton) as well as illumination changes due to sunlight variations.

#### 3.1 Spatiotemporal Topics Based Surprise

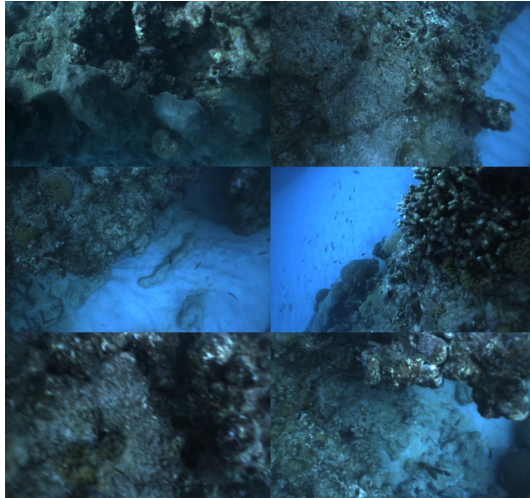
We set both summary size and topic size to 6 for our experiments. The hyperparameters for the LDA were determined empirically. A video demonstration of the robot as it traverses a path and reacts to surprising observations is available online<sup>1</sup>. There we can see the robot stopping over different, previously unobserved visual features, and then moving on at higher speeds when there is nothing of surprise. Some examples of the learned topic labels are shown in Fig 4. We see that the topics are representative of underlying physical phenomenon, and do well in describing scenes where a mixture of these exist. Red and blue topics are being used to represent rocks in the dataset, yellow for the sand-rock boundary, and cyan for the fire coral and the white rope.

Fig. 5(a) shows an example of the final summary generated by our online topics based summarizer from a sample trajectory. The corresponding histograms show the distribution of topics in the image. Fig. 5(b) shows uniformly sampled images over the same trajectory, presented here for comparison. We see that the proposed algorithm is able to recognize different species of corals (images 2 and 3), and the accidental inclusion of a diver's hand with a rope (image 6). When these images were observed, the robot evaluated them as surprising and as a result slowed down to a halt. Once these images are added to the summary, the surprise score falls instantly, and the robot continues forward in search for new surprises.

<sup>1</sup> <http://cim.mcgill.ca/~yogesh/iser2012>



(a) Topics

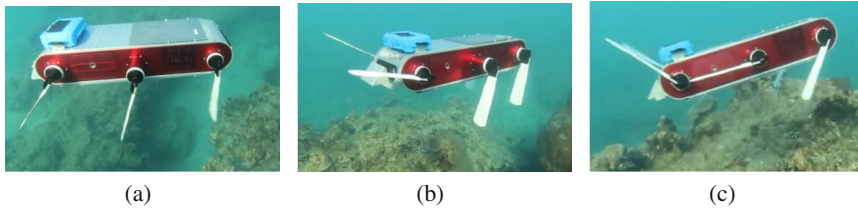


(b) Uniform

**Fig. 5** (a) A summary of six images generated online by the system. The histogram shows the distribution of visual topics in the image, each color corresponding to a different topic. (b) For comparison we show images sampled uniformly over the robot trajectory.

### 3.2 *Hovering Autopilot*

Fig. 6 shows the flipper placement configurations employed in the hovering gait and used to maintain attitude and depth, at different forward speeds  $v$ . As mentioned earlier, we employed two different strategies to maintain depth. At higher forward velocities ( $v > 0.2$ ), depth was maintained via pitch angle changes, as

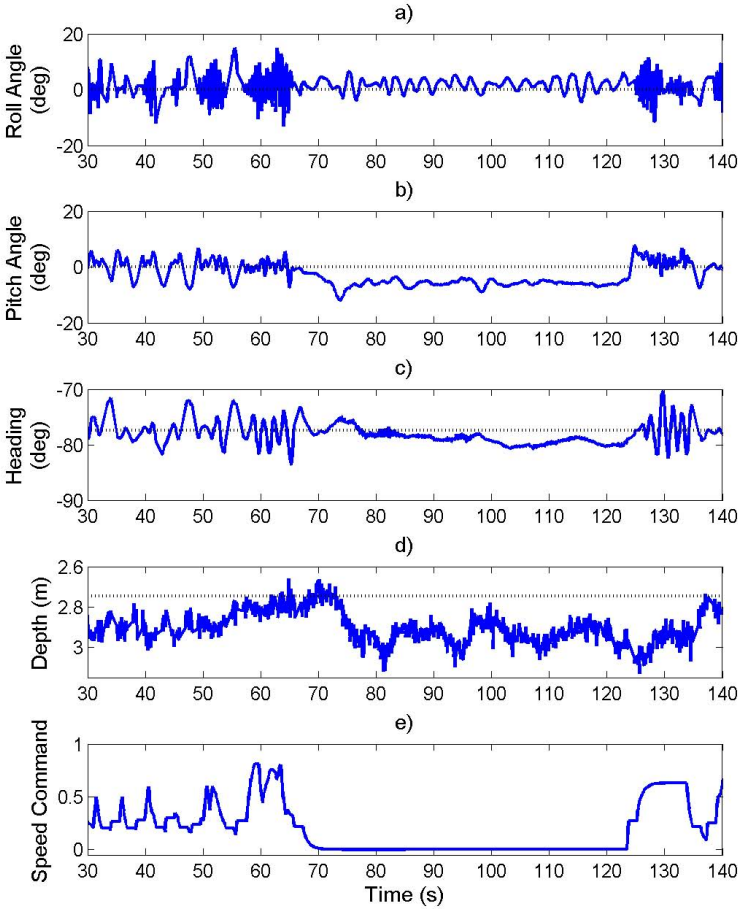


**Fig. 6** Pictures showing the flippers' angle due to the action of the autopilot system, during one of the sea trials. (a) the robot is performing a heave-up maneuver to maintain depth and attitude at zero forward speed. (b) the robot is executing a combined heave up, pitch up and slow forward speed maneuver. (c) the robot is performing a pitch-up maneuver at high speed.

depicted in Fig. 6 (c). By pitching up or down, the forward travel of the robot induced a change in depth, thus allowing depth control. When the robot had no forward velocity ( $v = 0$ ), maintaining depth required the use of a heaving motion. This motion is accomplished by having the 6 flippers pointing upward or downward, as illustrated in Fig. 6 (a). This way, the net thrust produced by the oscillating flippers does not induce forward motion. Attitude stabilization is still possible with this leg configuration, by means of a forward/aft differential thrust for pitch corrections and left/right differential thrust for roll corrections. For low velocities ( $v < 0.2$ ), the robot flippers were placed so as to generate both heaving and forward motion (Fig. 6 (b)). All of these pictures were taken from a single trial, demonstrating the need to adapt the locomotion strategy in order to satisfy motion requirements.

During the trials, we noted a decreased performance of depth and attitude control at lower velocities. This is a common phenomenon, as controllability of a vehicle operating in a fluid decreases with its velocity, due to the reduced lift forces on the control surfaces (flippers). This reduction in controllability can be seen for depth in Fig. 7 d). At commanded speed  $v > 0.5$  (corresponding roughly to forward velocities above  $20 \text{ cm/s}$ ), the depth error is reduced. For example, at  $t = 63\text{s}$  where the speed command  $v$  is above 0.5, the depth static error is less than  $0.1 \text{ m}$ . At low velocities  $v$ , this static error in depth increased and was generally positive (i.e., deeper than commanded), due to the fact that the robot was negatively buoyant, i.e., had a tendency to sink when standing still.

One issue plaguing the autopilot was unwanted roll oscillations of  $\pm 10^\circ$  at low velocities  $v$ . These oscillations had never been observed with the autopilot system before, when the robot was operated at medium to high velocities (above  $20 \text{ cm/s}$ ). We suspect that these were caused by an unforeseen change in the hydrodynamic behavior of the vehicle and its flippers at low velocity. As a consequence, controller improvements at low velocities are under investigation.



**Fig. 7** Performance of the autopilot system in hovering mode, with robot attitude a)-c) and depth d) over time, during one of the sea trials. The dotted lines in a)-d) correspond to the target values sent to the autopilot system. The open-loop speed command  $v$  in e) provided by the surprise module is unit-less: a value of 1.0 corresponded to maximum forward velocity of the vehicle.

## 4 Conclusion

We have demonstrated a novel autonomous robotic system that can be used to assist in exploring dangerous underwater environments. Our use of an online spatiotemporal topic modeling was an attempt to model semantic surprise, which is sensitive to presence of high-level patterns in the scene, such as different coral species, rocks, and sand. Given a fixed trajectory, the robot traversed it with a non-uniform speed, stopping at locations containing surprising observations, and moving at high speeds over seemingly boring or previously observed regions. From a locomotion point of

view, these experiments helped validate the use of heaving motion for depth control at low velocities, as well as highlighted certain deficiencies of our roll control system in that regime. The resulting summaries produced by our system is able to capture the visual diversity of the underwater environment. Our ongoing future work is focused on developing better realtime online topic modeling techniques, such as the use of nonparametric hierarchical Dirichlet processes, and their use in control of different robotic platforms for exploration tasks.

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *The Journal of Machine Learning Research* 3, 993–1022 (2003)
2. Bourque, E., Dudek, G.: Automated Image-Based Mapping. In: *IEEE Computer Vision and Pattern Recognition (CVPR)–Workshop on Perception of Mobile Agents*, pp. 61–70 (1998)
3. Canini, K.R., Shi, L., Griffiths, T.L.: Online Inference of Topics with Latent Dirichlet Allocation. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 5(1999), pp. 65–72 (2009)
4. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. In: *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, STOC 1997*, pp. 626–635. ACM Press, New York (1997), <http://dl.acm.org/citation.cfm?id=258533.258657>, doi:10.1145/258533.258657
5. Girdhar, Y., Dudek, G.: ONSUM: A System for Generating Online Navigation Summaries. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 746–751 (2010), <http://cim.mcgill.ca/~yogesh/publications/iros2010.pdf>, doi:10.1109/IROS.2010.5650315
6. Girdhar, Y., Dudek, G.: Efficient on-line data summarization using extremum summaries. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3490–3496 (2012), <http://cim.mcgill.ca/~yogesh/publications/icra2012.pdf>, doi:10.1109/ICRA.2012.6224657
7. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* 38, 293–306 (1985), doi:10.1016/0304-3975(85)90224-5
8. Grabowski, R., Khosla, P., Choset, H.: Autonomous exploration via regions of interest. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 2, pp. 1691–1696. IEEE (2003), <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1248887>, doi:10.1109/IROS.2003.1248887
9. Griffiths, T.L., Steyvers, M.: Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* 101(suppl. 1), 5228–5235 (2004), <http://www.pnas.org/content/101/suppl.1/5228.abstract>, doi:10.1073/pnas.0307752101
10. Hofmann, T.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning* 42(1), 177–196 (2001), doi:10.1023/A:1007617005950
11. Hsu, W.L., Nemhauser, G.L.: Easy and hard bottleneck location problems. *Discrete Applied Mathematics* 1(3), 209–215 (1979), <http://www.sciencedirect.com/science/article/B6TYW-46HW0N4-5/2/1a9ef82be9ca95988c9d75e76c216024>, doi:10.1016/0166-218X(79)90044-1



12. Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F.: iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research* 31(2), 216–235 (2011), <http://ijr.sagepub.com/cgi/content/abstract/31/2/216>, doi:10.1177/0278364911430419
13. Paul, R., Rus, D., Newman, P.: How was your day? Online Visual Workspace Summaries using Incremental Clustering in Topic Space. In: *IEEE International Conference on Robotics and Automation, ICRA (2012)*
14. Plamondon, N., Nahon, M.: A trajectory tracking controller for an underwater hexapod vehicle. *Bioinspiration and Biomimetics* 4(3), 036,005 (2009)
15. Sattar, J., Dudek, G., Chiu, O., Rekleitis, I., Giguère, P., Mills, A., Plamondon, N., Prahacs, C., Girdhar, Y., Nahon, M., Lobos, J.P.: Enabling Autonomous Capabilities in Underwater Robotics. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, pp. 3628–3634 (2008)*, <http://cim.mcgill.ca/~yogesh/publications/iros2008.pdf>, doi:10.1109/IROS.2008.4651158
16. Sattar, J., Giguere, P., Dudek, G.: Sensor-based behavior control for an autonomous underwater vehicle. *I. J. Robotic Res.* 28(6), 701–713 (2009)
17. Sivic, J., Zisserman, A.: Video Google: Efficient Visual Search of Videos. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) *Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 127–144. Springer, Heidelberg (2006)*, [http://dx.doi.org/10.1007/11957959\\_7](http://dx.doi.org/10.1007/11957959_7)
18. Song, X., Lin, C.Y., Tseng, B.L., Sun, M.T.: Modeling and predicting personal information dissemination behavior. In: *Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD 2005, p. 479. ACM Press, New York (2005)*, <http://dl.acm.org/citation.cfm?id=1081870.1081925>, doi:10.1145/1081870.1081925
19. Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., Hanel, D., Montemerlo, M., Morris, A., Omohundro, Z., Reverte, C.: Autonomous exploration and mapping of abandoned mines. *IEEE Robotics & Automation Magazine* 11(4), 79–91 (2004), [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1371614](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1371614), doi:10.1109/MRA.2004.1371614
20. Wang, X., Grimson, E.: Spatial Latent Dirichlet Allocation. In: *Advances in Neural Information Processing Systems, vol. 20, pp. 1577–1584 (2007)*, <http://people.csail.mit.edu/xgwang/papers/STLDA.pdf>
21. Whaite, P., Ferrie, F.: Autonomous exploration: driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(3), 193–205 (1997), <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=584097>, doi:10.1109/34.584097
22. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA 1997, Towards New Computational Principles for Robotics and Automation, pp. 146–151. IEEE Comput. Soc. Press (1997)*, [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=613851](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=613851), doi:10.1109/CIRA.1997.613851

# Active and Adaptive Dive Planning for Dense Bathymetric Mapping<sup>\*</sup>

Geoffrey A. Hollinger, Urbashi Mitra, and Gaurav S. Sukhatme

**Abstract.** We examine the problem of planning dives for an Autonomous Underwater Vehicle (AUV) to generate a dense bathymetric map using sidescan sonar. The three key challenges in this scenario are (1) proper modeling of the local uncertainty of the 3D reconstruction, (2) efficient dive planning to reduce this uncertainty, and (3) determination of when to re-plan adaptively based on new information. To address these challenges, we propose using non-parametric Bayesian regression to model the expected accuracy of the map, which provides principled cost functions for planning subsequent dives. In addition, we propose an efficient greedy method to reduce this uncertainty, and we show that it achieves theoretically bounded performance given assumptions on the sensor model and the form of the uncertainty function. We present experiments on the propeller-driven YSI EcoMapper AUV equipped with a sidescan sonar in an inland lake. The experiments demonstrate the benefit of efficient dive planning, with our results providing performance gains of up to 83% versus standard lawnmower patterns.

## 1 Introduction

The increasing capabilities of Autonomous Underwater Vehicles (AUVs) have enabled their deployment in oceans, coastal environments, and inland waterways.

---

Geoffrey A. Hollinger · Gaurav S. Sukhatme  
Computer Science Department, Viterbi School of Engineering,  
University of Southern California, Los Angeles, CA 90089  
e-mail: {gahollin, gaurav}@usc.edu

Urbashi Mitra  
Ming Hsieh Department of Electrical Engineering, Viterbi School of Engineering,  
University of Southern California, Los Angeles, CA 90089  
e-mail: ubli@usc.edu

<sup>\*</sup> This research has been funded in part by the following grants: ONR N00014-09-1-0700, ONR N00014-07-1-00738, ONR N00014-06-10043, NSF CCR-0120778, and NSF CNS-1035866.





**Fig. 1** YSI EcoMapper propeller-driven AUV used to perform bathymetric mapping experiments in Puddingstone Lake. The vehicle is equipped with an Imaginex Sportscan sidescan sonar for underwater imaging as well as a Doppler Velocity Log (DVL) for navigation.

There are a number of potential applications for such vehicles, including ecological monitoring [26], mine sweeping [20, 34], inspection of submerged structures [15], and underwater habitat mapping [24, 27]. In such scenarios, deployment time for the AUV represents a considerable resource cost, and there is significant motivation to improve the efficiency of the deployment relative to the desired mission goal. The problem of planning the AUV's mission to improve the quality of the sensor data fits into the broad framework of *active perception* that dates back to early seminal research in robotics and computer vision [3]. The key idea behind active perception is that we can plan the path and sensor views of robotic vehicles to maximize information gained while minimizing time and/or energy consumption.

In addition to actively planning the vehicle's dives, we also explore the benefit of *adaptive* dive planning for AUVs. When calculating efficient dive patterns, it may be beneficial to adaptively re-plan as new information is acquired. In our recent work, we have begun to quantify the potential benefit of adaptivity in underwater inspection scenarios [15, 16], which moves towards planning algorithms that selectively adapt to new information when it is most beneficial. In the current paper, we explore these ideas experimentally on an AUV performing underwater mapping tasks.

Our experiments validate the principles of active and adaptive perception in the domain of underwater bathymetric mapping using an AUV. The two hypotheses to be experimentally tested are (1) active dive planning improves the efficiency of bathymetric mapping and (2) selective use of adaptivity further improves this efficiency. These hypotheses are suggested by our earlier theoretical analysis [16] and by experiments in related underwater inspection domains [15]. The experiments presented here serve to advance our understanding of the principles behind these related active sensing problems.

The remainder of this paper is organized as follows. We first discuss related work in active perception and underwater mapping (Section 2). We then describe the active dive planning problem in more detail and propose solutions for measuring the uncertainty of the reconstruction and planning informative dives (Section 3). Next, we discuss experimental results from the AUV performing active and

adaptive strategies for dense bathymetric mapping (Section 4). Finally, we conclude and discuss avenues for future work (Section 5).

## 2 Related Work

The study of general active perception problems dates back to early work in active vision [1, 3] and next-best view planning [9]. Early techniques were often concerned with the “next-best” action to maximize information with relatively less focus on long-term planning methods. While this early work made limited use of information theoretic methods, later work in this area has increasingly focused on incorporating information theory and probabilistic techniques into active perception systems [25]. Two recent surveys describe the breadth of work in active perception and its development over the past two decades [7, 22]. Our experiments are complementary to this prior work, in that we provide experimental validation of the benefit of active dive planning in the underwater mapping domain. In addition, our work examines the use of long-term planning and the benefit of adaptivity, which are often overlooked in prior work.

Our formulation connects the active perception problem with sequential hypothesis testing, another classical problem where an observer must select a sequence of noisy experiments to determine the nature of an unknown [30]. Hypothesis testing has applications in information theory, machine learning, and wireless communications. Of particular interest is the connection between sequential hypothesis testing and feature selection in Bayesian learning, which has been applied to sensor placement problems in the context of identifying objects in 2D images [6]. This early work in robotic perception provides formal justification for the formulation of active perception as a Bayesian learning problem.

Many active perception problems can also be seen as instances of submodular orienteering and informative path planning [23], where the goal is to optimize the path of a robot to gain the maximal amount of information. Many of these problems allow for efficient algorithms with performance guarantees due to the property of submodularity, a formal characterization of the property of diminishing returns. Recent advances in submodular optimization have extended informative sensor placement problems to domains that require adaptive re-planning [14]. In complementary work, the stochastic optimization community has begun examining the potential benefit of adaptive re-planning for such problems as stochastic covering and stochastic knapsack [11]. To our knowledge, these ideas have not previously been applied to underwater robotics applications.

Acoustic range sensing, essential in the inspection of turbid-water environments, has been used to produce 3D point clouds throughout various underwater domains. Early work in 3D underwater mapping dates back more than a decade to the use of diver-held sonar devices that utilized an inertial measurement unit to help align subsequent images [2]. Such techniques inevitably suffer from drift over time, which can be mitigated through Simultaneous Localization and Mapping (SLAM) techniques [12]. It is also possible to generate underwater maps without motion

estimation through sophisticated techniques for accurate registration [5]. In the large body of prior work in underwater mapping, the vehicle is remotely controlled by a human operator, and the potential for autonomous operation is not explored.

The underwater robotics community has recently begun examining problems of AUV dive planning for such applications as mine sweeping [33] and seabed classification [32]. These works focus on the dive planning problem, but they do not integrate methods for measuring 3D reconstruction accuracy. One early approach for measuring uncertainty in 3D reconstructions uses a linear approximation of the surface, where the resulting covariance provides a measure of accuracy [31]. Our approach using Gaussian Processes provides a more general uncertainty measure that utilizes both data sparsity and surface complexity when estimating uncertainty. Similar non-parametric regression approaches have been used to define reconstruction accuracy metrics for 2.5D surface estimation [29]; though these techniques do not actively plan to reduce uncertainty. In addition, our approach utilizes the augmented input vector method [28] to provide non-stationary kernels that improve the richness of the uncertainty representation.

### 3 Algorithm Design

The goal is to plan the dives of the autonomous underwater vehicle to provide an accurate reconstruction of the bottom of a body of water. The first step in solving this problem is to define a reconstruction accuracy metric that is computable across the area of interest. To do so, we propose using non-parametric Bayesian Regression in the form of Gaussian Processes (GPs) [21].<sup>1</sup> If we view the 3D reconstruction as a function approximation of surface height over a 2D bottom plane, the GP representation has the advantage of providing a principled measure of variance, which we use as a measure of uncertainty, as well as a mean value at each point. The resulting uncertainty measure can then be used in a greedy planning framework to plan subsequent dives that we expect to be highly informative for generating a dense and accurate mapping.

#### 3.1 Gaussian Process Modeling

A GP models a noisy process  $z_i = f(\mathbf{x}_i) + \varepsilon$ , where  $z_i \in \mathbb{R}$ ,  $\mathbf{x}_i \in \mathbb{R}^d$ , and  $\varepsilon$  is Gaussian noise. We are given some data of the form  $D = [(\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), \dots, (\mathbf{x}_n, z_n)]$ . For 2.5D surfaces that do not loop on themselves, as is typically the case for bathymetric mapping,  $\mathbf{x}_i$  is a point in the 2D plane ( $d = 2$ ), and  $z_i$  represents the height at that point.<sup>2</sup>

<sup>1</sup> We note that Gaussian Process regression is closely related to Kriging interpolation as described in the geostatistics literature [4].

<sup>2</sup> For more complex mapping tasks with arbitrary 3D geometry, the parametric Gaussian Process Implicit Surface extension can be used [15].

We refer to the  $d \times n$  matrix of  $\mathbf{x}_i$  vectors as  $\mathbf{X}$  and the vector of  $z_i$  values as  $\mathbf{z}$ . To define a GP, it is necessary to choose a covariance function that relates points in  $\mathbf{X}$ . We employ the commonly used squared exponential, which produces a smooth kernel that drops off with distance:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left( - \sum_{k=1}^d w_k (\mathbf{x}_{ik} - \mathbf{x}_{jk})^2 \right). \quad (1)$$

The hyperparameter  $\sigma_f$  represents the process noise, and each hyperparameter  $w_k$  represents a weighting for the dimension  $k$ . Once the kernel has been defined, combining the covariance values for all points into an  $n \times n$  matrix  $\mathbf{K}$  and adding a Gaussian observation noise hyperparameter  $\sigma_n$  yields  $\mathbf{K}_z = \mathbf{K} + \sigma_n^2 \mathbf{I}$ . We can now estimate the kernel hyperparameters  $\theta = (\sigma_f, \sigma_n, w_{1:d})$  using the standard method of maximizing the likelihood of the measurements given the data and the hyperparameters [21]:

$$\log p(\mathbf{z}|\mathbf{X}, \theta) = -\frac{1}{2} \mathbf{z}^T \mathbf{K}_z^{-1} \mathbf{z} - \frac{1}{2} \log |\mathbf{K}_z| - \frac{n}{2} \log 2\pi. \quad (2)$$

This likelihood is maximized using conjugate gradient optimization. We note that for larger data sets, downsampling or block learning approaches may be required to make hyperparameter learning tractable.

We now wish to predict the mean function value (height)  $\bar{f}_*$  and variance  $\mathbb{V}[f_*]$  at a selected point  $\mathbf{x}_*$  given the measured data:

$$\bar{f}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{z}, \quad (3)$$

$$\mathbb{V}[f_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*, \quad (4)$$

where  $\mathbf{k}_*$  is the covariance vector between the selected point  $\mathbf{x}_*$  and the training inputs  $\mathbf{X}$ . This model provides a mean height and variance at all points of interest in  $\mathbb{R}^2$ . In this model, the variance gives a measure of uncertainty based on the sparsity of the data and the hyperparameters. It should be noted that this variance is not modeled as a random variable, and it is only dependent on data density. Thus, we can estimate the variance exactly before we have taken measurements assuming that we know exactly which points we will observe. We note that this assumption may not hold, and we provide a more detailed discussion of this issue later.

The formulation above determines correlation between points solely based on their proximity in space. Another important consideration when determining uncertainty is the amount of variability in an area. For instance, an underwater ridge would require a large number of points to reconstruct efficiently. To model uncertainty caused by surface variability, we utilize the augmented input vector approach [28]. The idea is to modify the input vector by adding additional terms that affect the correlations between the data. We propose using an initial estimate of surface height (found by running the GP with a standard kernel or another interpolation method) to

modify the input vector to be  $\mathbf{x}' = (x, y, \bar{z})$ , where  $\bar{z}$  is the initial estimate of the surface height at that point. The weighting hyperparameters can be adjusted to modify the effect of spatial correlation versus surface height correlation.

To address the problem of scalability, we utilize a local approximation method using KD-trees [29]. Since correlated points will typically be near each other, we can pre-calculate a KD-tree and then retrieve the  $k$  closest points when calculating the estimate and variance at a given point. Since the retrieval from the KD-tree is on average  $O(\log n)$ , the resulting local approximation of the GP requires  $O(k^3 \log n)$  computation for each point, or  $O(nk^3 \log n)$  total. Comparing this to  $O(n^3)$  computation of a typical GP, we see that if  $k \ll n$ , this approximation provides a significant reduction in computation. The value of  $k$  can be selected based on the necessary computing power, which allows the approximation to improve in accuracy with increasing computation.

### 3.2 Variance Reduction Algorithm

For a given GP representation, we define a total variance by integrating over the space of interest  $\mathcal{X}$ . The goal of the dive planning is to generate a dive pattern that maximally reduces this variance. More formally, we define a policy  $\pi$  that executes a given dive based on a given uncertainty representation. We can now define the following metric for dive planning:

$$J_{var}(\pi) = \int_{\mathcal{X}} \mathbb{V}_0(x) - \mathbb{V}_{\pi}(x) dx, \quad (5)$$

where  $\mathbb{V}_0(x)$  is the initial variance at point  $x$ , and  $\mathbb{V}_{\pi}(x)$  is the variance at point  $x$  after executing policy  $\pi$ .

For a policy  $\pi$ , we set the measure of information quality to  $J_{var}(\pi)$ , and we define  $c(\pi)$  as the cost of executing the policy. In the application of interest, this cost will be determined by the number of planned dives and their length. We will choose dives that optimize the following maximization problem:

$$\pi^* = \operatorname{argmax}_{\pi} J_{var}(\pi) \text{ s.t. } c(\pi) \leq B, \quad (6)$$

where  $B$  is a budget constraint on time or energy.<sup>3</sup> Setting the cost as a budget constraint is natural in underwater applications where the battery life or deployment time are limiting factors. For this paper, we define the budget as a pre-specified number of dives multiplied by a fixed dive length (i.e., we only consider dives of equal length). The extension of this algorithm to dive patterns with dives of unequal length is a topic for future work.

To provide informative dive planning, we propose greedily selecting dives that maximize variance reduction until the budget is reached and then running a gradient

<sup>3</sup> We note that we can alternatively maximize the weighted sum  $\alpha J(\pi) - \beta c(\pi)$  with appropriate weighting constants using a Lagrangian relaxation [13].

optimization that perturbs each dive and locks the pattern into a local optimum. The gradient optimization is not strictly necessary, though we expect it will provide some limited improvement over the greedy policy. We note that even with the gradient optimization, we do not expect the algorithm to yield the optimal dive pattern in all cases due to local maxima. A summary of the proposed algorithm is given in Algorithm 1.

The summary in Algorithm 1 does not specify how we calculate the expected variance reduction for a given dive  $d$ . Since we do not know which observations we will receive before making them, and re-running the GP is computationally expensive, the calculation of this quantity is not trivial. We provide an approximation to the variance reduction by first calculating the sum of variance in the area viewed by the dive. For overlapping dive patterns, we approximate the reduction in variance caused by each subsequent view using an exponential drop off. The resulting variance at a point  $x$  is found by  $\mathbb{V}_n(x) = \mathbb{V}_0(x) \exp(-n/\alpha)$ , where  $\mathbb{V}_0(x)$  is the initial variance at the point,  $n$  is the number of times the point has been viewed in the dive plan, and  $\alpha$  is a length scale parameter. The length scale parameter was set to  $\alpha = 1$  based on fitting to test runs where the exact reduction was calculated.

A more computationally intensive way of calculating the variance reduction would be to simulate a dive by generating a point cloud of the expected dive result. The GP could then be re-run with the updated point cloud. This approach would require running the GP once for each candidate dive during each planning iteration. We found that the exponential drop off assumption provided sufficient accuracy while remaining computationally tractable.

### 3.3 Guarantees and Need for Adaptivity

It has been shown in prior work that greedy placement of static sensors to reduce variance in Gaussian Processes provides a constant-factor performance guarantee relative to optimal of  $1 - 1/e \approx 63\%$  [18]. Thus, even in the worst-case, the greedy deployment will still achieve 63% of the variance reduction as the optimal policy and can be expected to perform significantly better in practice [17, 23]. When the dives are of equal length, a pre-planned series of greedily planned dives provides the same bounded approximation of the optimal dive pattern if the following assumptions hold:

1. The variance reduction objective function is monotone and submodular. Monotonicity implies that additional measurements always improve the objective. Submodularity implies that the objective follows the law of diminishing returns (i.e., the more measurements observed, the less incremental benefit of receiving a new measurement). This assumption holds in many cases for variance reduction in Gaussian Processes with fixed hyperparameters [10, 18].
2. The 2D locations where measurements will be received are known in advance (i.e., we can predict exactly which points in 2D we will receive data from before executing a dive). We note that for realistic sensor models, this assumption may not be valid.

**Algorithm 1.** Active Dive Planning Algorithm

---

```

1: Input: Uncertainty map  $\mathbb{V}_0$  and dive length budget  $B$ 
2: Select a set of possible dives  $\mathcal{N}$  of equal length  $L$ 
3: Initialize remaining budget  $B_r \leftarrow B$ , uncertainty map  $\mathbb{V}_r \leftarrow \mathbb{V}_0$ , set of selected dives
    $\mathcal{M} \leftarrow \emptyset$ 
4: while  $B_r \geq L$  do
5:   for each dive  $d \in \mathcal{N}$  do
6:     Calculate expected uncertainty reduction  $J(d) \leftarrow \mathbb{V}_r - \mathbb{V}_d$ 
7:   end for
8:   Select dive  $d^* = \operatorname{argmax}_d J(d)$ 
9:   Update selected dives  $\mathcal{M} \leftarrow \mathcal{M} \cup d^*$ , uncertainty map  $\mathbb{V}_r \leftarrow \mathbb{V}_d$ , budget  $B_r \leftarrow B_r - L$ 
10: end while
11: while not converged and planning time remains do
12:   for each dive  $d \in \mathcal{M}$  do
13:     for selected directions  $a$  do
14:       Perturb dive by  $\varepsilon$  in direction  $a$  to get  $d^p$ 
15:       Calculate new uncertainty  $J(d^p) \leftarrow \mathbb{V}_r - \mathbb{V}_{d^p}$ 
16:       if  $J(d^p) > 0$  then
17:         Update dive pattern  $\mathcal{M} \leftarrow d^p \cup \mathcal{M} \setminus d$  and uncertainty  $\mathbb{V}_r \leftarrow \mathbb{V}_{d^p}$ 
18:       end if
19:     end for
20:   end for
21: end while
22: Sort dives in  $\mathcal{M}$  to minimize execution time
23: Execute dive pattern  $\mathcal{M}$ 

```

---

3. The hyperparameters do not change during the dive planning. For the case of augmented input vectors (see above), this implies that the surface height does not change significantly (i.e., the surface heights act as implicit hyperparameters). This assumption will be violated whenever significant changes occur in the reconstruction.

Since the last two assumptions can be violated in real-world scenarios, there is motivation to re-plan subsequent dives based on new information. However, it is not obvious how much they will be violated, which depends both on the rate of change of the 3D reconstruction and the sensor model of the sidescan sonar. Thus, we examine the benefit of adaptivity in active dive planning through experimental trials.

## 4 Experiments and Results

We now explore the benefit of active and adaptive planning for 3D reconstruction by mapping a portion of the bottom of Puddingstone Lake in Southern California (Lat. 34.088854°, Lon. -117.810667°). The region of interest is approximately 10 m deep and covers an area of 100 m  $\times$  50 m of lakebed. We utilize a YSI EcoMapper propeller-driven AUV (shown in Figure 1) capable of moving at speeds of 5 knots and diving down to 100 m. A downward-looking Sportscan Imaginex sidescan sonar



is mounted on the vehicle. The vehicle is also equipped with a Doppler Velocity Log (DVL) and GPS unit, which provide navigation capabilities. A photograph of the vehicle is shown in Figure 1.

#### 4.1 3D Reconstruction from Sidescan Sonar

The Imaginex sidescan sonar returns 2D intensity images that do not include 3D depth information. Example images are shown in Figure 2. It is important to note that the AUV has a significant “blind spot” directly under it, which necessitates overlap in images to provide a complete reconstruction. In order to generate a 3D reconstruction from a 2D sidescan image, we use standard shape-from-shading techniques that have been successfully applied in prior work [8]. The central idea is to make assumptions on the reflectance properties of the sonar and then use the geometry of the sidescan position to develop a fully 3D reconstruction from a collection of 2D images.

The sensor provides an intensity return from each point viewed on the 2D bottom plane. We use the traditional Lambertian model [19], which relates intensity to the angle of reflection. This model assumes that the scattering is diffuse and that the returned intensity is not dependent on the angle of observation or the frequency of the sonar pulse. For a viewed point  $p$ , the intensity can be computed as:

$$I(p) = K\Phi(p)R(p)\cos(\theta(p)), \quad (7)$$

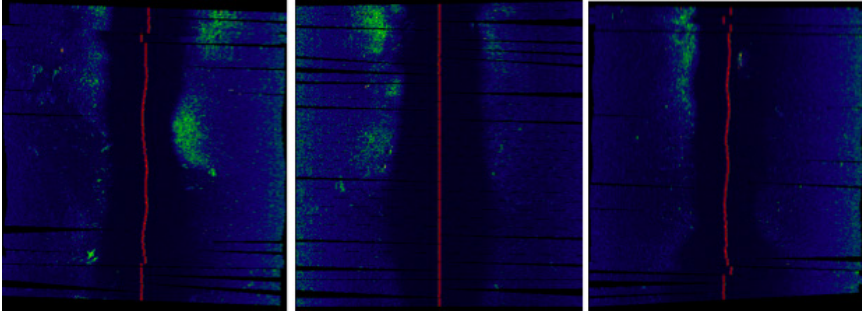
where  $\Phi$  denotes the intensity of the illuminated sonar pulse,  $R$  denotes the reflectivity of the seabed,  $\theta$  is the angle of incidence, and  $K$  is a constant. Values for  $\Phi$ ,  $R$ , and  $K$  were determined empirically in this work based on accepted values from prior work [8, 19]. Improving the estimates of these values would serve to improve the 3D reconstruction, and our active planning techniques can easily be applied to such improved reconstructions.

From the above equation, we can determine an angle of incidence for each viewed point from known quantities. Given the angles of incidence, we reconstruct a 3D representation of the bottom through numerical integration. More detail on shape-from-shading approaches can be found in prior work [8]. Since the focus of this paper is on validating the benefit of active dive planning, we do not implement post-processing of the reconstruction through constrained optimization, though we expect such techniques would improve the quality of the 3D reconstruction.

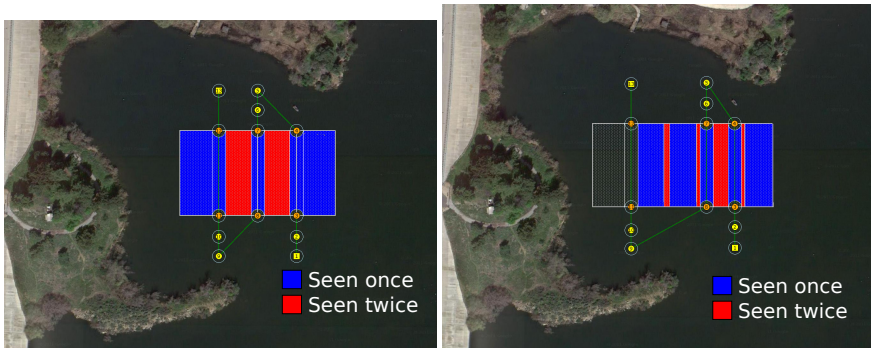
#### 4.2 Active Dive Planning

We now test the benefit of active planning for informative dives to improve the 3D reconstruction. Planning and learning was performed on a desktop machine with a 3.2 GHz Intel i7 processor and 9 GB of RAM. A variance map could be calculated in approximately 30 seconds using a discrete grid approximation with 1 m resolution and a KD-tree local approximation of the GP using 100 points. Once a variance





**Fig. 2** Example sidescan sonar images providing bathymetry information in Puddingstone Lake. Variations in the lakebed topography are apparent as bright returns in the images. A full 3D reconstruction can be found using shape-from-shading methods that make assumptions on the reflectance properties of the sonar.



**Fig. 3** Dive patterns optimized to maximize coverage overlap (left) and uncertainty reduction (right). The region of interest is the 100 m  $\times$  50 m area (Lat. 34.088854°, Lon. -117.810667°) surrounded by the rectangular sensor footprint and highlighted in blue and red. Points of interest covered once by the sidescan sonar are labeled blue (darker), and areas of interest covered twice are labeled red (lighter).

map was available, the dive patterns were calculated in less than a second using the exponential drop off approximation (see Section 3). The plan was then transmitted to the vehicle on the surface using a standard wireless network.

The experiment progresses as follows: (1) The AUV executes three evenly spaced dives to generate an initial 3D reconstruction of the area of interest, (2) the initial reconstruction is used to plan a subsequent three dive pattern using Algorithm 1, and (3) the vehicle executes the resulting dive pattern. The final uncertainty after six dives is then compared to an alternative dive pattern that maximizes overlap in the sidescan images without using the uncertainty representation. The difference between the uncertainties represents the benefit of active planning in this domain.

Figure 4 shows a comparison of 3D reconstructions for the two dive patterns described above: one maximizing coverage and one optimizing variance reduction as in Algorithm 1. The reconstruction resulting from the minimum variance dive pattern does not contain the clearly erroneous valleys that occur in the maximum coverage dive pattern. Figure 5 shows a quantitative comparison of the total uncertainty of the two dive patterns, as well as total uncertainty after the initial 3 dives and after an exhaustive 9 dive pattern. The variance dive pattern using 6 dives provides nearly the same uncertainty as the 9 dive pattern. The 9 dive pattern represents a target uncertainty at which sensor noise and errors from the shape from shading make further reduction difficult.

Comparing the reduction in uncertainty of the variance dive pattern and the coverage dive pattern, we see that the variance dive pattern improves the uncertainty reduction by 83%, showing a significant benefit from using active planning. In addition, the 6 dive pattern that reduces variance achieves a smaller percent error for the reconstruction when the percent error is computed relative to the 9 dive pattern. However, we note that the reconstruction from the 9 dive pattern does not provide perfect ground truth.

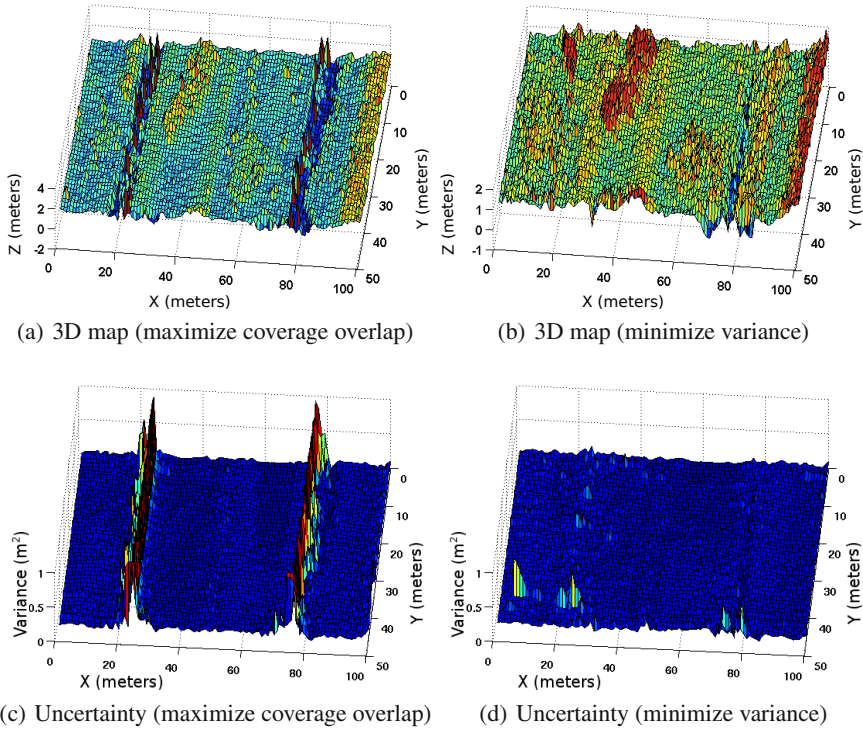
### 4.3 *Benefit of Adaptivity*

The experiment described above utilizes adaptive behavior only when re-planning the last three dives (i.e., it learns the uncertainty model and chooses the resulting dives). Alternatively, the uncertainty model could be re-learned more often during the mission. We next examine the benefit of such re-planning by running additional AUV data collections. Figure 6 shows a long 20 dive run that took approximately two hours to execute. The 20 dives were pre-run, which allowed us to test different selection methods that choose a subset of the dives. We note that the gradient optimization step (see Algorithm 1) was not used in this experiment since there are a limited number of dives to select at pre-set locations.

Figure 6 shows the quantitative results with a different number of re-planning cycles. The first additional re-planning cycle provides approximately an 8% reduction in uncertainty, and the second re-planning cycle does not provide any improvement. We note that the benefit from the first re-planning cycle is significant; however, the pre-planned path still provides an accurate 3D reconstruction. Thus, adaptivity is beneficial in this scenario, but the essential component of the algorithm is pre-planning to reduce the uncertainty of the reconstruction. Based on these results, it is clear that we can predict the returns that we will receive from the sidescan sonar to some extent, which motivates the use of long-term planning in this scenario.

## 5 Discussion and Conclusions

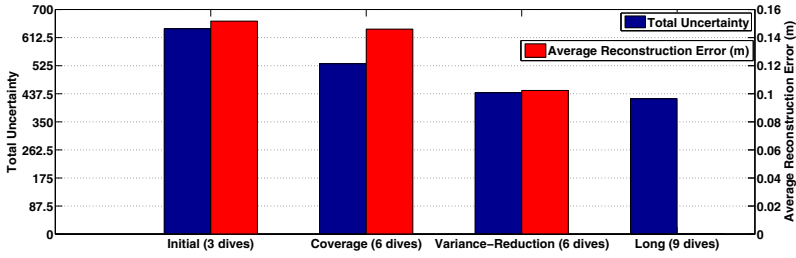
Our experiments have examined the benefit of active dive planning in the real-world application domain of 3D bathymetric mapping. We have shown that planning a dive pattern that greedily maximizes variance reduction provides a significant



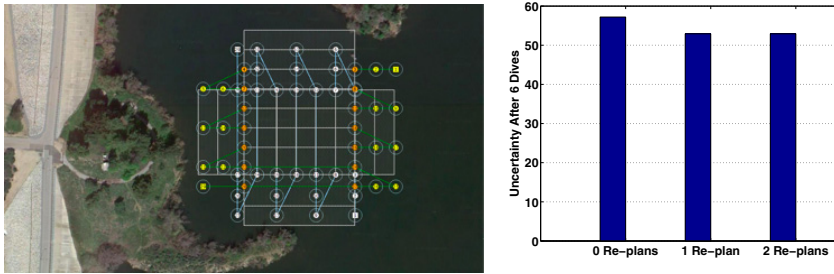
**Fig. 4** 3D bathymetric reconstructions found using sidescan sonar (top) and accompanying variance maps (bottom). Left: reconstruction resulting from dive patterns that maximize overlap in sensor coverage. Right: reconstruction from dive patterns that minimize the expected variance in a Gaussian Process representation of the reconstruction. The right image does not suffer from erroneous valleys that result from data sparsity, and the variance map matches this qualitative observation.

quantitative and qualitative improvement over standard lawnmower approaches. In addition, we have shown that adaptive planning that re-plans provides some additional improvement. We note that the re-planning occurs once the vehicle has surfaced, which allows for the necessary computation to be placed offboard and communicated to the vehicle. In addition, the planned paths can be checked by a human operator before execution to ensure safety. These experimental findings provide a baseline for future research in active mapping and inspection. In addition, our experiments provide insight into the validation of recent theoretical advances in submodular optimization and active sensing.

In underwater applications, perhaps more so than in other robotics domains, there is significant interest in simplifying the behaviors that occur underwater. Complex behaviors, such as adaptive planning, can potentially lead to safety concerns and increase the necessary computation carried by the AUV. Our experiments have helped



**Fig. 5** Comparison of the resulting bathymetric reconstructions for different dive patterns. The initial dive pattern uses 3 evenly spaced dives. The coverage dive pattern adds an additional 3 dives (total of 6 dives) that maximize coverage overlap. The variance dive pattern instead uses the additional three dives (total of 6 dives) to maximize reduction of uncertainty in the map. The variance dive pattern provides improved uncertainty reduction, nearly that of an exhaustive dive pattern that uses 9 dives. The graph shows both total uncertainty remaining (dark blue) and average reconstruction error computed relative to the 9 dive pattern (light red).



**Fig. 6** Crisscross pattern used to test the benefit of adaptive dive planning (left) and the uncertainty remaining in the 40 m × 40 m area after 6 dives for different numbers of re-planning cycles (right). The vehicle may choose to execute any 6 of 20 dives (10 North/South and 10 East/West) to maximize the accuracy of the 3D bathymetric reconstruction. The dives are pre-run, which allowed post-processing the data to determine the benefit of different dive selection methods. Re-planning consists of re-calculating the uncertainty representation before determining the next dive. The first additional re-planning cycle provides approximately an 8% reduction in uncertainty versus a pre-planned dive pattern, but the second re-planning cycle does not provide any improvement. Note that there is less remaining uncertainty in these results than those in Figure 5 due to the smaller area being surveyed.

determine the level of adaptivity necessary to provide high performance bathymetric mapping as well as the granularity with which this adaptivity must be made. Our results demonstrate that there is limited benefit to adaptive planning, which avoids some of the necessity for onboard computation on the AUV. The ability to provide good 3D reconstructions without in situ adaptation significantly reduces the computational burden on the vehicle and also simplifies the testing requirements. As a

result, our algorithm can easily be applied to any propeller-driven AUV capable of executing a series of waypoints.

Future work includes analysis of alternative uncertainty representations that utilize the specifics of the shape-from-shading method to provide better predictions of areas that need further inspection. For instance, alternative kernels in the GP may allow for more accurate uncertainty modeling. Additional experimental validation includes testing of the proposed method on 3D reconstructions generated from other sensors (e.g., monocular vision, stereo, and infrared imaging). The techniques proposed here and the experimental methods employed are applicable throughout underwater inspection domains as well as on ground vehicles and aerial platforms. Ultimately, the experimental validation of active and adaptive planning moves towards more efficient use of robotic perception for autonomous vehicles.

**Acknowledgements.** The authors gratefully acknowledge Jonathan Binney, Jnaneshwar Das, Arvind Pereira, and Hordur Heidarsson at the University of Southern California for their insightful comments and assistance with experiments.

## References

1. Aloimonos, Y., Weiss, I., Bandopadhyay, A.: Active vision. *International Journal of Computer Vision* 1(4), 333–356 (1988)
2. am Ende, B.A.: 3D mapping of underwater caves. *IEEE Computer Graphics and Applications* 21(2), 14–20 (2001)
3. Bajcsy, R.: Active perception. *Proc. IEEE, Special Issue on Computer Vision* 76(8), 966–1005 (1988)
4. Banerjee, S., Carlin, B.P., Gelfand, A.E.: *Hierarchical Modeling and Analysis for Spatial Data*. Chapman and Hall/CRC Press (2004)
5. Buelow, H., Birk, A.: Spectral registration of noisy sonar data for underwater 3D mapping. *Autonomous Robots* 30(3), 307–331 (2011)
6. Cameron, A., Durrant-Whyte, H.: A Bayesian approach to optimal sensor placement. *Int. J. Robotics Research* 9(5), 70–88 (1990)
7. Chen, S., Li, Y., Kwok, N.M.: Active vision in robotic systems: A survey of recent developments. *Int. J. Robotics Research* 30(11), 1343–1377 (2011)
8. Coiras, E., Petillot, Y., Lane, D.M.: Multiresolution 3-D reconstruction from side-scan sonar images. *IEEE Trans. Image Processing* 16(2), 382–390 (2007)
9. Connolly, C.: The determination of next best views. In: *Proc. IEEE Conf. Robotics and Automation*, pp. 432–435 (March 1985)
10. Das, A., Kempe, D.: Algorithms for subset selection in linear regression. In: *Proc. ACM Symp. Theory of Computing*, pp. 45–54 (May 2008)
11. Dean, B., Goemans, M., Vondrak, J.: Approximating the stochastic knapsack: the benefit of adaptivity. *Mathematics of Operations Research* 33(4), 945–964 (2008)
12. Fairfield, N., Kantor, G.A., Wettergreen, D.: Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *J. Field Robotics* 24(1-2), 3–21 (2007)
13. Fisher, M.L.: The lagrangian relaxation method for solving integer programming problems. *Management Science* 27(1), 1–18 (1981)
14. Golovin, D., Krause, A.: Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artificial Intelligence Research* 42, 427–486 (2011)

15. Hollinger, G., Englot, B., Hover, F., Mitra, U., Sukhatme, G.: Uncertainty-driven view planning for underwater inspection. In: Proc. IEEE Int. Conf. Robotics and Automation, pp. 4884–4891 (May 2012)
16. Hollinger, G., Mitra, U., Sukhatme, G.S.: Active classification: Theory and application to underwater inspection. In: Int. Symp. Robotics Research (August 2011)
17. Hollinger, G., Singh, S., Djughash, J., Kehagias, A.: Efficient multi-robot search for a moving target. *Int. J. Robotics Research* 28(2), 201–219 (2009)
18. Krause, A., Singh, A., Guestrin, C.: Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Machine Learning Research* 9, 235–284 (2008)
19. Langer, D., Hebert, M.: Building qualitative elevation maps from side scan sonar data for autonomous underwater navigation. In: Proc. IEEE Int. Conf. Robotics and Automation, pp. 2478–2483 (April 1991)
20. Myers, V., Williams, D.: A POMDP for multi-view target classification with an autonomous underwater vehicle. In: Proc. IEEE OCEANS Conf., pp. 1–5 (September 2010)
21. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. The MIT Press (2006)
22. Roy, S.D., Chaudhury, S., Banarjee, S.: Active recognition through next view planning: A survey. *J. Pattern Recognition* 37(3), 429–446 (2004)
23. Singh, A., Krause, A., Guestrin, C., Kaiser, W.: Efficient informative sensing using multiple robots. *J. Artificial Intelligence Research* 34, 707–755 (2009)
24. Singh, H., Armstrong, R., Gilbes, F., Eustice, R., Roman, C., Pizarro, O., Torres, J.: Imaging coral I: Imaging coral habitats with the SeaBED AUV. *J. Subsurface Sensing Technologies and Applications* 5(1), 25–42 (2004)
25. Sipe, M.A., Casasent, D.: Feature space trajectory methods for active computer vision. *IEEE Trans. Pattern Analysis and Machine Learning* 24(12), 1634–1643 (2002)
26. Smith, R., Schwager, M., Smith, S., Jones, B., Rus, D., Sukhatme, G.: Persistent ocean monitoring with underwater gliders: Adapting sampling resolution. *J. Field Robotics* 28(5), 714–741 (2011)
27. Steinberg, D., Williams, S.B., Pizarro, O., Jakuba, M.: Towards autonomous habitat classification using Gaussian mixture models. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 4424–4431 (October 2010)
28. Thompson, D., Wettergreen, D.: Intelligent maps for autonomous kilometer-scale science survey. In: Proc. Int. Symp. Artificial Intelligence, Robotics and Automation in Space (February 2008)
29. Vasudevan, S., Ramos, F.T., Nettleton, E.W., Durrant-Whyte, H.F.: Gaussian process modeling of large scale terrain. *J. Field Robotics* 26(10), 812–840 (2009)
30. Wald, A.: Sequential tests of statistical hypotheses. *Ann. Mathematical Statistics* 16(2), 117–186 (1945)
31. Whaite, P., Ferrie, F.P.: Autonomous exploration: Driven by uncertainty. *IEEE Trans. Pattern Analysis and Machine Intelligence* 19, 193–205 (1997)
32. Williams, D.: AUV-enabled adaptive underwater surveying for optimal data collection. *Intelligent Service Robotics* 5(1), 33–54 (2012)
33. Williams, D.P.: Bayesian data fusion of multiview synthetic aperture sonar imagery for seabed classification. *IEEE Trans. Image Processing* 18(6), 1239–1254 (2009)
34. Williams, D.P.: On optimal AUV track-spacing for underwater mine detection. In: Proc. IEEE Int. Conf. Robotics and Automation, pp. 4755–4762 (May 2010)



# Exploring Space-Time Tradeoffs in Autonomous Sampling for Marine Robotics

Rishi Graham, Frédéric Py, Jnaneshwar Das, Drew Lucas,  
Thom Maughan, and Kanna Rajan

## 1 Introduction

In the coastal ocean, biological and physical dynamics vary on spatiotemporal scales spanning many orders of magnitude. At large spatial ( $\mathcal{O}(100\text{km})$ ) and temporal ( $\mathcal{O}(\text{weeks to months})$ ) scales, traditional shipboard and moored measurements are very effective at quantifying mean and varying oceanic properties. At scales smaller than the internal Rossby radius ( $\mathcal{O}(10\text{km})$  for typical coastal stratification at mid-latitude), horizontal, vertical and temporal inhomogeneity is the rule rather than the exception. The in-situ measurement of such submesoscale variability has proven difficult using traditional techniques. However, numerical modeling and limited field studies have shown the importance of submesoscale dynamics to physical processes such as front formation [1], momentum transfer between the atmosphere and the ocean [2], and turbulence, mixing, and upwelling [3]. Furthermore, there is good evidence for the importance of submesoscale dynamics to biological primary productivity [4, 5, 6], carbon sequestration [7], and patch formation in phytoplankton [8]. The development of techniques to test in situ the prediction of numerical models is a vital step towards a fundamental understanding of coastal physical and biological processes. A combination of Lagrangian profiling (capable of diagnosing vertical variability on the scale of centimeters and temporal variability on the scale of minutes) and directed Eulerian sampling (capable of diagnosing horizontal

---

Rishi Graham · Frédéric Py · Thom Maughan · Kanna Rajan  
Monterey Bay Aquarium Research Institute, California  
e-mail: {rishi, fpy, tm, Kanna.Rajan}@mbari.org

Jnaneshwar Das  
Dept. of Computer Science, Univ. of Southern California  
e-mail: jnaneshd@usc.edu

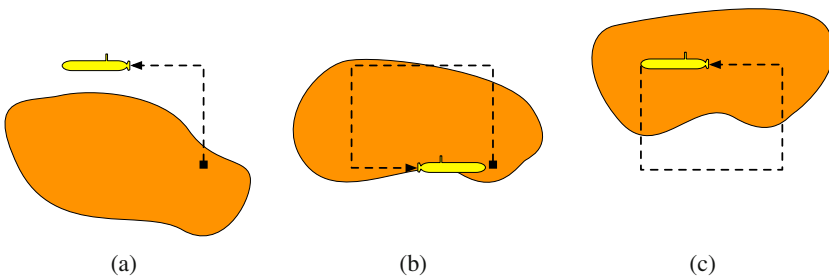
Drew Lucas  
Dept. of Marine Sciences, Univ. of California, Santa Cruz  
e-mail: anjlucas@ucsc.edu

variability on the order of tens of meters) is an especially appealing approach as it maximizes the strengths of available platforms.

Robotic methods in sampling are cost-effective and allow for fine targeting, while traditional methods tend to be hit-or-miss, leading to substantial under sampling of the coastal ocean. It is for this reason that we are engaged in a long-term interdisciplinary science program of observing biological lifecycles at MBARI using underwater and surface-based robotic platforms as part of CANON [9]. Our initial focus, in this context, has been in tracking oceanographic features using powered Autonomous Underwater Vehicles (AUVs). One such experiment is described in this paper.

Methodologically this drives a requirement that an AUV “stay” with an identified advecting patch of water while continuously monitoring key variables (e.g., temperature, salinity, nitrate, chlorophyll among others) in the upper water-column using onboard sensors. An adaptive software controller (the Teleo-Reactive EXecutive or T-REX ) which dynamically synthesizes temporal plans in-situ and is responsive to environmental change, controls the vehicle as it performs repeated surveys within the identified patch [10, 11, 12, 13, 14]. Scientists identify the relevant patch using in-situ sensors from moorings and manned ships in addition to remote sensing data when available, and tag this patch with a passive drifter which flows with the water mass.

Consider a single AUV trying to map a dynamic water feature under uncertain conditions. If the vehicle moves slower than the advecting field, then it cannot make use of spatial correlations and the problem becomes one of time series analysis with field reconstruction further afield becoming challenging. If the AUV moves substantially faster, then it could capture spatial snapshots of the field and the reconstruction problem can be treated spatially and the spatial response evolved over time, in which case sophisticated dynamical models may be employed. The problem we consider in this paper lies somewhere in the middle. The vehicle is fast enough to discover and make use of some of the spatial structure, but not so fast as to capture the full spatial field at any time. Fig. 1 shows a simplified two dimensional depiction of this scenario.



**Fig. 1** Simplified 2D depiction of an AUV mapping a “feature” which might be defined by a contour of the scalar field being measured. As the AUV progresses, the feature changes shape and to some degree size, and also shows an overall northward movement. The dotted lines show the AUV track lines.



There are multiple ways to reconstruct a dynamic field from sparse measurements, all of which require some degree of statistical inference. We consider a simple form of optimal interpolation known as kriging (see, e.g. [15]) and leave the examination of more sophisticated models to future work. Fig. 1 illustrates a common difficulty among all space-time reconstruction methods, which is the distinction between overall drift and local dynamics. This is analogous in spatial statistics to the distinction between changes in the mean field and local variation. The more accurately one can represent the mean value, usually as a function of spatial location, the more accurate the reconstruction and smaller the predictive variance.

Our earlier work [16, 17] focused on the experimental methodology of following such a moving water mass with an AUV moving around the drifter centroid. This method was selected in order to provide the environmental context of the water mass and to map the interior of the volume to understand the biological dynamics occurring within the patch. In addition to providing an appropriate reference frame of observation, the work provides an upper bound on tracking water mass speed that a powered robotic platform can follow within an *enclosure criterion* [17]. More recently we have focused on using an AUV to follow a sensing device mounted on a drifter [18, 19] which have provided a unique opportunity to compare space-time estimates from the AUV data against ground truth provided by the drifting sensor. In these experiments, we use a combination of Eulerian and Lagrangian frames of reference to improve the the mean squared error of such predictions by parameterizing the assumption that the surrounding water mass moves with the drifting platform.

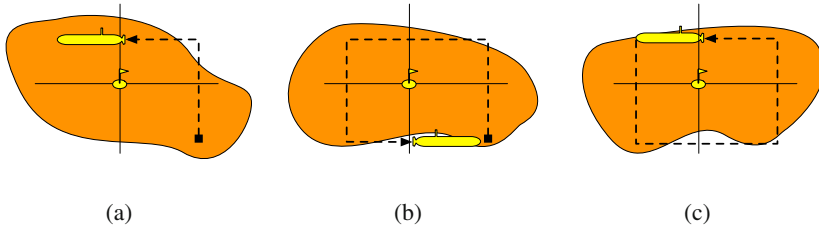
Drifters approximate some sense of “Lagrangian” dynamics. Usually they follow the current at a particular depth, subject to some wind forcing at the surface, and are equipped with minimal instrumentation to convey current location. We examine the use of a drifter as a proxy for the overall drift of the random field within some neighborhood. In particular, we consider correlations between samples as a function of both the traditional space-time locations and relative locations in a pseudo-Lagrangian frame of reference<sup>1</sup> centered around the drifter. The example in Fig. 1 is depicted in the pseudo-Lagrangian framework in Fig. 2, in which the AUV stays with the feature by tracking the drifter.

Sampling the Real ocean is substantially more complex; the drifter for instance, is an imperfect mechanism which only approximates the current at a single fixed depth and it is often impacted by wind. Further, there is no guarantee of continuity of the water mass in the neighborhood of the drifter. In this paper, we examine data from a number of missions involving AUV’s and drifting assets with particular emphasis on comparing the quality of drifter data as a proxy for mean transport. This is done by comparing estimated values at specific space-time locations against “ground truth” data gathered at those locations by other vehicles.

The structure of this paper is as follows. Section 2 places this work in the context in ocean sciences and engineering. Section 3 is the core of the paper where we

---

<sup>1</sup> We call this a “pseudo-Lagrangian” frame of reference for two reasons: first, the drifter is restricted to horizontal motion and does not capture the full three-dimensional dynamics of the coastal ocean; second, the frame of reference does not rotate depending on the direction of motion of the drifter, as a truly Lagrangian frame of reference would.



**Fig. 2** Example of the feature from Fig. 1 being tracked with the use of a drifter. Our correlation function uses a combination of fixed space-time coordinates and a pseudo-Lagrangian coordinate system centered on the drifter.

describe the family of Gaussian Process models used to estimate the value of the measured field at unmeasured locations and the methods used to compare the results. Section 4 presents some illustrative results when the theory is applied to an idealized simulated field. In Section 5 we outline a series of at-sea experiments focusing in particular on an experiment in which the drifter is equipped with a suite of sensors which we then use to provide ground truth for model comparison. This is followed with analysis of the at-sea experimental data in Section 6, with conclusions and future work in Section 7.

## 2 Related Work

Drifters have traditionally been used for Lagrangian studies for measurement of biological processes in-situ at appropriate temporal scales with ship-support [20] as well as for physical oceanographic measurements of current flow and turbulence related to ocean modeling [21, 22]. The end result is often to make measurements of dynamic ocean fields for field reconstruction.

Our work extends these applications by using the drifter to provide a pseudo-Lagrangian frame of reference in which overall advection can be considered negligible. We use the word “Lagrangian” approximately, referring to a coordinate system which is centered, at any given time, at the current location of the drifter but follows the definition in [23]. Further, we use a robotic AUV platform with in-situ plan synthesis to provide overall environmental context of the water mass around the drifter [17]. Our technique is novel and provides a substantially more accurate rendering of the drifting field; traditional ship-based drifter methods cannot be qualified as truly Lagrangian.

[24] focuses on using Empirical Orthogonal Functions (EOFs) to determine the structure of dynamic random fields. Such spectral methods are common practice in the oceanographic and atmospheric communities [25], as well as the statistical modeling literature [26], and are especially useful due to their compact representation. Spectral methods, however, suffer from two major limiting features when it comes to tracking a moving water mass with an AUV. First, they are generally predicated on an assumption that overall advection is negligible. We on the other hand are

explicitly interested in cases where this may not be true. Second, they do not easily handle sparse data points in a continuous domain. Methods for estimating correlation spectra generally require data on a regular grid. One counter-example is the mingled spectrum principle of [27], which involves first approximating the spectral components of the signal purely as a function of distance along the AUV path and then determining to what extent the underlying space-time coordinate axes can be separated within that signal.

We chose a hierarchical Bayesian approach to Gaussian Process (GP) modeling [28, 29, 30] because GPs lend themselves to sparse data points in a continuous domain and illustrate a clear and direct connection between measurements and predictions. Outside the ocean sciences, related work has been mostly in the context of using spatial statistics for terrestrial measurements. [31] uses fixed monitoring cites to derive an estimate of advection for rainfall measurement; our efforts come closest to this work. They define a Lagrangian coordinate system in which the temporal distance from a sample to a predictive location is based on the lag from the start of rainfall at that particular monitor. They use kriging to produce estimates of the field in both Eulerian and Lagrangian frameworks and compare using an adjusted mean squared error criterion. [32, 33] talk about advection modeling also in the domain of rainfall forecasting. They too deal with multiscale and Lagrangian modeling where “advection” is gleaned from sequential radar snapshots, thereby ensuring commingling of space/time effects is at a minimum. Interestingly [34] uses a method for registration brain scan images while comparing pixels in one image to pixels in another. We on the other hand deal with continuous space and effectively look for spatio-temporal change of that single “pixel” from each survey to measure water-column change.

A common assumption in space-time statistics is one of separability (e.g. [35]), in which the covariance between two points in space-time can be written as the product of spatial and temporal covariance functions. Separability breaks down when the field in question is under the influence of a dominant advection term, which is what we face in water-column measurements. We conjecture, however, that if the field reconstruction were done in a coordinate system centered on a drifting element, the resulting correlation structure would be approximately separable, at least within the neighborhood of the drifter. This notion (that there is an overall advection component which may be removed and the remaining field exhibits separability) has been put forth in a more general form as Taylor’s Hypothesis [36]; [35] has a good description of Taylor’s hypothesis as pertains to functional correlation estimation.

Several recent efforts have focused on determining whether or not a random field is separable [37, 38, 39, 40], however these, and all others that we are aware of have assumed a network of fixed monitoring stations with repeated measurements in time in order to formally separate space and time in the correlation structure. One promising avenue of inquiry is introduced in [41] and further expanded in [35], in which the authors use a functional covariance built as a parameterized mixture of a separable space-time covariance and a purely spatial one in the Lagrangian frame of reference. While the authors use this method on data sets generated by networks of fixed monitoring stations, there is nothing about the method which precludes a

single moving agent sampling a dynamic field. We employ a version of this model in this paper.

### 3 Space-Time Model

We consider a hierarchical Bayesian approach to GP estimation inspired by space-time Kriging [35] (also Optimal Interpolation, Best Linear Unbiased Estimation, and Objective Analysis, to name a few variants of the same basic concept). One of the advantages of such an approach is the propagation of uncertainty from prior to posterior. Here, we are primarily interested in the posterior mean, and how it compares against two measures of ground truth (measurements taken from another vehicle, and leave-one-out self comparison). As such, we use a fully Bayesian approach to estimate the parameters of the covariance, but only consider the conditional distribution given point estimates of the parameters for validation. While this falls short of a full posterior or marginal analysis in terms of uncertainty quantification, it allows for simpler results and shorter computation times.

Specification of the covariance is important in this approach to stochastic modeling. For simplicity and computational efficiency, spatial reconstructions often include assumptions of stationarity and/or isotropy [42]. In the coastal ocean, the interplay between large scale and small scale dynamics, as well as external factors such as biological entities, may make such assumptions unrealistic. For example, an upwelling front is a boundary region where two dramatically different water types intersect [43]. Along this frontal region, spatial correlation scales dramatically differ from those within the two large scale features, making a small scale spatial reconstruction of the boundary region difficult. We however, will use these simplifying assumptions of stationarity and isotropy for three reasons. First, stationary models form the basis of many more realistic models, so this is a logical first step in the process. Second, we approach the problem as one of approximation. We know that we are not modeling the process exactly and we have ground truth for comparison, we compare one approximation against another in terms of a fit to the ground truth. Finally, we are more concerned here with the interplay between space and time than with accuracy of the spatial model. Integrating more complex spatial models into the Lagrangian framework is a topic for future work.

Specification of covariance is challenging when trying to reconstruct dynamic fields from sparse measurements. How does one estimate space-time covariance structure? Previous methods of space-time inference and dynamic field reconstruction have generally fallen into one (or both) of two categories. Either the measurements are assumed to be taken on a space-time grid (meaning that for any spatial location, a time series of values is available), or the grid assumption is relaxed and replaced with an assumption that overall advection of the field is negligible. In [41] and [35], the authors assume that the field satisfies Taylor's hypothesis in the sense that the covariance between two points can be written as a linear combination of a separable covariance in space-time coordinates and a purely spatial covariance in a moving, pseudo-Lagrangian coordinate system. The resulting hierarchical Bayesian

model is then fitted using Markov Chain MonteCarlo (MCMC) simulation. We take a similar approach, but consider a family of hierarchical correlation models which differ in separability and in the coordinate systems for distance calculations.

### 3.1 Basic Setup

Assume that a moving vehicle has taken  $n \in \mathbb{N}$  point measurements of a dynamic scalar process  $Z$  over a region of interest  $\mathcal{D} \subset \mathbb{R}^2$ . The domain is of arbitrary dimension, but we will deal in  $\mathbb{R}^2$  for simplicity. Associated with the  $i$ th measurement taken at a spatial location,  $s_i$  at time  $t_i$ . We will denote by  $z_i$ , the value of the  $i$ th measurement, with  $\mathbf{z} = (z_1, \dots, z_n)^T$ , and by  $z(s, t)$  the value of  $Z$  at an arbitrary location,  $s \in \mathcal{D}$  and time  $t \in \mathbb{R}$ . Now assume that we would like to reconstruct the evolution of  $Z$  over the unsampled locations of  $\mathcal{D}$  within some time interval. Assume that the joint distribution of the samples is given by:

$$\mathbf{z} \sim N_n(\mu, \Sigma(\theta)),$$

an  $n - variate$  normal distribution with constant mean,  $\mu$ , and covariance matrix  $\Sigma(\theta)$ , which depends on a vector of unknown parameters,  $\theta$ . The covariance is modeled as a scalar variance term, a functional kernel, and a small diagonal measurement error:

$$\Sigma = \sigma^2(\mathbf{K} + \tau^2\mathbf{I}),$$

where  $\tau^2$  is a known constant, and  $\sigma^2$  is inferred from the data. We use a family of functional correlation forms for the kernel matrix  $\mathbf{K}$ , described in more detail below. In general, for a given  $\theta$ , we will write the correlation between  $Z(s, t)$  and  $Z(s', t')$  as:

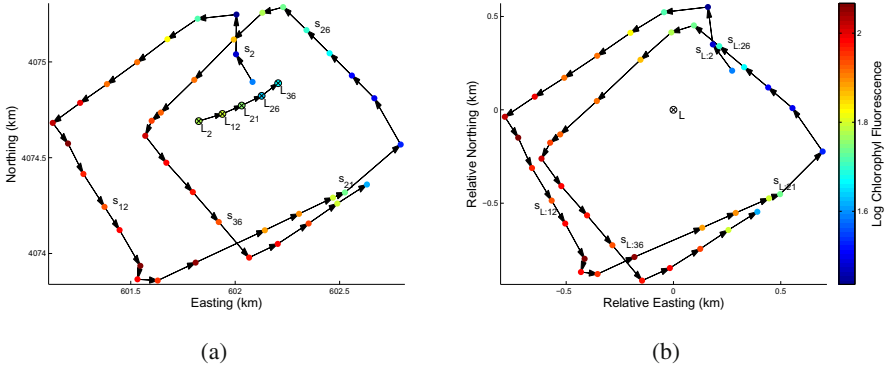
$$\text{Cor}[Z(s, t), Z(s', t')|\theta] = C(s, t, s', t'; \theta).$$

Where the dependence on  $\theta$  is not important we will suppress the explicit notation. A prediction at arbitrary space-time location  $(s_p, t_p)$  may be formulated from the conditional expectation,

$$\mathbf{E}(Z(s_p, t_p)|\theta, \mathbf{z}) = \mu + \mathbf{C}^T(\mathbf{K} + \tau^2\mathbf{I})^{-1}\mathbf{z}, \tag{1}$$

where  $\mathbf{C} = \mathbf{C}(\theta)$  is the vector of covariances between samples and predictive location, i.e.,  $[\mathbf{C}]_i = C(s_i, t_i, s_p, t_p)$ .

For the purpose of pseudo-Lagrangian distance calculations, we introduce the notion of a drifter, which identifies a frame of reference that moves along with the water mass. Let  $L_i \in \mathcal{D}$  denote the location of the drifter at time  $t_i$ . Any space-time location,  $(s_i, t_i)$  then has a corresponding location in the pseudo-Lagrangian drifter-relative space which we denote by  $s_{L:i} = s_i - L_i$ . Fig. 3 illustrates a sequence of measurements made in Eulerian (i.e., geospatial) coordinates, and the corresponding locations in pseudo-Lagrangian coordinates.



**Fig. 3** Chlorophyll Fluorescence data collected by the AUV at a depth of 6m during a part of the *WireWalker* following experiment. In 3(a) we see the trajectory in geospatial (Eulerian) coordinates. The colored circles around the square trajectory are measurements made by the AUV, with arrows showing the path of the AUV. Select points in the *WireWalker* path ( $L_i$ ) have been plotted to show the advection component. For each  $L_i$  there is an associated time  $t_i$ , and the location of the AUV at time  $t_i$  is marked with  $s_i$ . In 3(b) the same AUV trajectory in the *WireWalker* -relative (Lagrangian) coordinates,  $s_{L,i} = s_i - L_i$  is shown. The *WireWalker* measurements are not depicted here since they all occur at (0, 0) in this coordinate system (marked with circle-x).

### 3.2 Correlation Models

We now describe the specific correlation models used in this work. The idea is to explore the utility of defining correlation in Eulerian coordinates, pseudo-Lagrangian coordinates, or a combination of the two, particularly with an eye towards separability. As such, we consider a family of correlation functions built from a common basic kernel and compare the results of using one versus another. We use as our basic correlation building block the common powered exponential kernel, in the spatial, temporal, and space-time domains:

$$\begin{aligned}
 C_S(s_i, s_j; \nu) &= e^{-(\|s_i - s_j\|)^\nu}, \\
 C_T(t_i, t_j; \nu) &= e^{-(|t_i - t_j|)^\nu}, \\
 C_{ST}(s_i, t_i, s_j, t_j; \nu) &= e^{-\left(\sqrt{\|s_i - s_j\|^2 + (t_i - t_j)^2}\right)^\nu}.
 \end{aligned}$$

$\nu \in (0, 2]$  is a smoothness parameter (with  $\nu = 2$ , the function and thereby the process [42] is infinitely differentiable). In order to simplify notation in the sequel, we introduce the vector of all possible parameters:

$$\theta = (\sigma^2, \lambda, \nu_{es}, \nu_{et}, \nu_{est}, \nu_{ls}, \nu_{lt}, \nu_{lst}, r_{es}, r_{et}, r_{ls}, r_{lt}),$$

where  $\sigma^2$  is the scalar variance term described above,  $\lambda$  is a mixing coefficient for Eulerian and Lagrangian components,  $\nu_*$  are smoothness parameters, and  $r_*$  are

coordinate specific range parameters designed to allow for geometric anisotropy between space and time. Simple correlation kernels in the Eulerian and Lagrangian spatial coordinates may be written as:

$$C_{\text{ESP}}(s_i, s_j, \theta) = C_S\left(\frac{s_i}{r_{es}}, \frac{s_j}{r_{es}}; \nu_{es}\right),$$

$$C_{\text{LSP}}(s_i, s_j, \theta) = C_S\left(\frac{s_{L:i}}{r_{ls}}, \frac{s_{L:j}}{r_{ls}}; \nu_{ls}\right).$$

Extending this to the space-time domain, we consider first the non-separable versions:

$$C_{\text{ENSEP}}(s_i, t_i, s_j, t_j; \theta) = C_{ST}\left(\frac{s_i}{r_{es}}, \frac{t_i}{r_{et}}, \frac{s_j}{r_{es}}, \frac{t_j}{r_{et}}; \nu_{est}\right),$$

$$C_{\text{LNSEP}}(s_i, t_i, s_j, t_j; \theta) = C_{ST}\left(\frac{s_{L:i}}{r_{ls}}, \frac{t_i}{r_{lt}}, \frac{s_{L:j}}{r_{ls}}, \frac{t_j}{r_{lt}}; \nu_{lst}\right),$$

then the separable versions:

$$C_{\text{ESEP}}(s_i, t_i, s_j, t_j; \theta) = C_S\left(\frac{s_i}{r_{es}}, \frac{s_j}{r_{es}}; \nu_{es}\right)C_T\left(\frac{t_i}{r_{et}}, \frac{t_j}{r_{et}}; \nu_{et}\right), \text{ and}$$

$$C_{\text{LSEP}}(s_i, t_i, s_j, t_j; \theta) = C_S\left(\frac{s_{L:i}}{r_{ls}}, \frac{s_{L:j}}{r_{ls}}; \nu_{ls}\right)C_T\left(\frac{t_i}{r_{lt}}, \frac{t_j}{r_{lt}}; \nu_{lt}\right).$$

Following [41], we next consider a correlation function which combines a separable kernel in the Eulerian space with a purely spatial kernel in the Lagrangian:

$$C_{\text{ESEPLSP}}(s_i, t_i, s_j, t_j; \theta) = (1 - \lambda)C_{\text{ESEP}}(s_i, t_i, s_j, t_j; \theta) + \lambda C_{\text{LSP}}(s_i, s_j; \theta).$$

Table 1 summarizes the correlation functions introduced here. We consider simulated and sea-going experiments in which some element of advection can be identified, and in which predictions can be compared against ground truth. In each case, we compare the models in terms of how well they predict the ground truth as well as using a leave-one-out error metric to ensure measurement fidelity. For each of the correlation functions, we complete the following steps:

1. Run MCMC to estimate the range, smoothness, and mixing parameters of the correlation model.
2. Calculate the predictions at unsampled locations and examine the distribution of errors.
3. For each measurement, calculate the hypothetical prediction at that (space-time) location using only the other measurements, then examine the distribution of these “leave-one-out” errors.

We then compare the performance of the different correlation functions on the two error metrics for various slices of data. In the following section, we describe a set of idealized simulated data designed to illustrate the extreme Lagrangian case.

**Table 1** Summary of correlation kernels

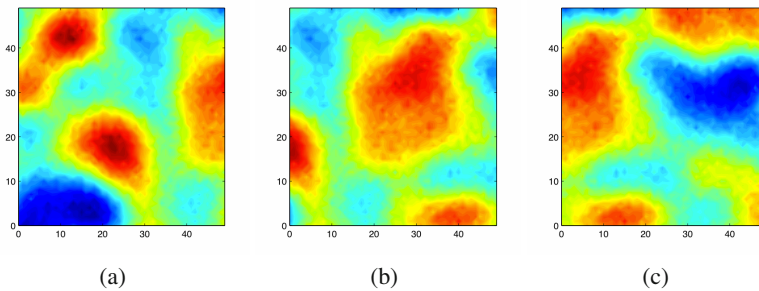
Name	Description
$C_{ESP}$	Spatial only correlation in Eulerian space
$C_{LSP}$	Spatial only correlation in Lagrangian space
$C_{ENSEP}$	Non-separable correlation in Eulerian space
$C_{LNSEP}$	Non-separable correlation in Lagrangian space
$C_{ESEP}$	Separable correlation in Eulerian space
$C_{LSEP}$	Separable correlation in Lagrangian space
$C_{ESEPLSP}$	Linear combination of $C_{ESEP}$ and $C_{LSP}$

## 4 Simulated Field

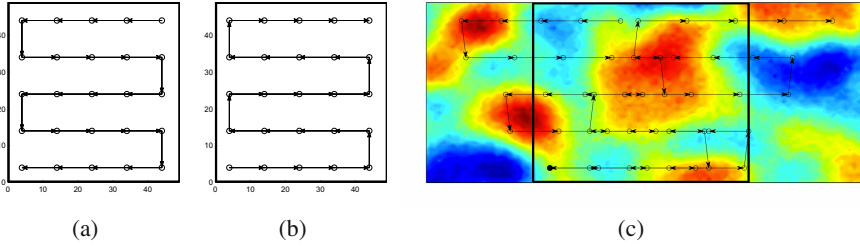
Using simulated data, we generate an idealized example for illustrative purposes. Consider the extreme case in which an unchanging scalar field,  $Z$ , advects through a region of interest  $\mathcal{D}$  with constant, uniform velocity over 50 time steps. For simplicity, advection occurs strictly from east to west. This allows us a very simple representation of the Eulerian and Lagrangian frames of reference: in the Eulerian frame of reference,  $\mathcal{D}$  is static, and  $Z$  moves westward with constant speed; in the Lagrangian frame of reference,  $Z$  is static and  $\mathcal{D}$  moves eastward with constant speed. Fig. 4 depicts the evolution of such a simulated “ground truth” field, in the Eulerian (fixed) frame of reference.

At the same time a simulated vehicle is executing a fixed “lawnmower” pattern over  $\mathcal{D}$ , taking one sample each time step. The resulting vector of measurements,  $\mathbf{z}$ , are simply taken from the simulated ground truth, with no measurement error. We are interested in reconstructing the field over  $\mathcal{D}$  mid-way thru the mission at time  $T = 25$ . Since we want to get as much information as possible out of the space-time measurements, we consider the hind-casting case, where all measurements are taken into account post-facto. Fig. 5 shows the lawnmower pattern in both Eulerian and Lagrangian frames of reference.

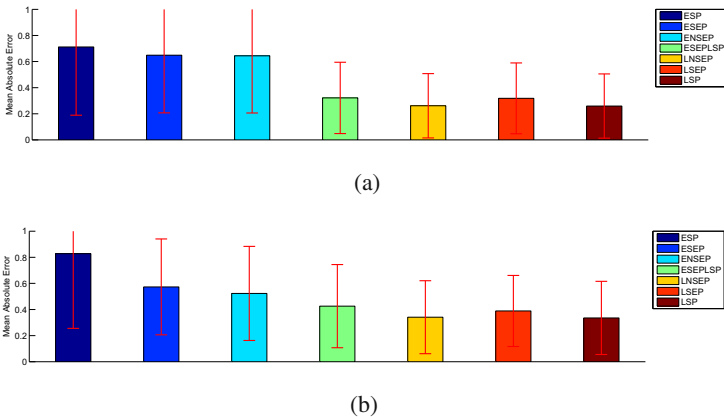
Since the simulated data are defined on a grid, we use those grid locations as our ground truth. Let  $\mathbf{s}_p$  denote the vector of grid locations over  $\mathcal{D}$  in the Eulerian frame

**Fig. 4** The simulated field over  $\mathcal{D}$  at time 1 in 4(a), time 25 in 4(b), and time 50 in 4(c)





**Fig. 5** In the fixed, Eulerian frame of reference, the AUV executes a lawnmower from top right to bottom left 5(a), and then back again 5(b). In 5(c) we see the trajectory of the AUV in the Lagrangian frame of reference, which moves with the simulated current. Here the black window shows the region  $\mathcal{D}$  at time  $T = 25$  (compare against figure 4(b)). Because of the special nature of these simulated data, in this frame of reference we can see the values measured by the AUV in a purely spatial context.

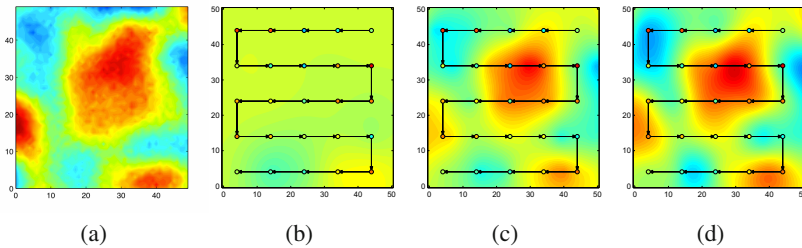


**Fig. 6** Comparison of predictive error distributions based on the different correlation models. In 6(a) we compare predictions against ground truth from the simulated values over  $\mathcal{D}$  at time  $T = 25$ . In 6(b) the leave-one-out errors are generated by estimating the value of the  $i$ th measurement based on all other measurements and repeating for all measurements. In each case, the box height shows the mean of the absolute errors, while the error bars show one standard deviation. The Eulerian kernels are on the left, Lagrangian on the right, strictly spatial to the outside with more freedom in the space-time interactions as we move inwards on each side.

of reference, and let  $\mathbf{z}_p$  denote the values of the simulated ground truth at spatial locations  $\mathbf{s}_p$  and time  $T = 25$ . As described in Section 3, for each correlation function in Table 1, we first ran MCMC to make draws from the posterior distributions of the pertinent parameters in  $\theta$ . For each such parameter, an estimate is formed based on the mean value of the random draws (after appropriate burn in). We will call the estimated parameter set  $\hat{\theta}$ .

Next, Equation (1) is used to find an estimate,  $\hat{Z}(s_p, t_p; \mathbf{z}, \hat{\theta})$ , at each predictive location  $(s_p, t_p)$ . For simplicity, we restrict our estimation region to  $\mathcal{D}$  at time  $T = 25$ . Those values are then compared against the simulated ground truth by examining the mean and variance of the absolute error values ( $|\hat{Z} - \mathbf{z}_p|$ ).

For the leave-one-out analysis, we compute a vector of predictions,  $\hat{Z}(s_i, t_i; \mathbf{z}_{-i}, \hat{\theta})$ , where  $\mathbf{z}_{-i}$  denotes the vector  $\mathbf{z}$  with the  $i$ th element removed. These predictions are compared against the measurements, using the first two moments of the absolute errors,  $|\hat{Z} - \mathbf{z}|$ . Fig. 6 depicts the distributions of absolute errors for each correlation function while Fig. 7 shows the effects of correlation kernel on the shape of the reconstructed field. In this carefully constructed simulation, the benefits of the Lagrangian frame of reference are clear. We next examine results from a multi-asset experiment in the coastal waters of Monterey Bay, where the situation is more challenging.

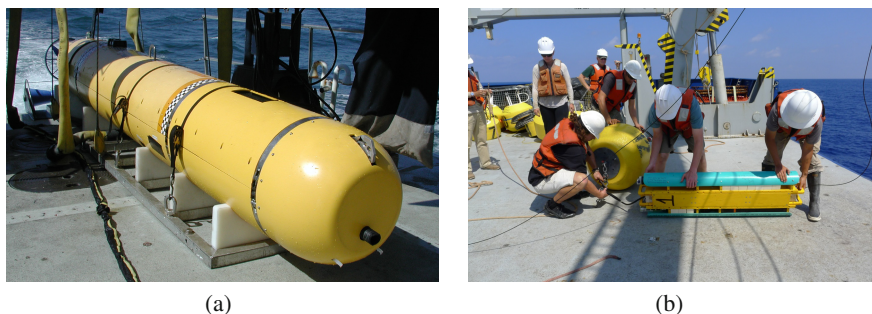


**Fig. 7** Examples of the field reconstruction over  $\mathcal{D}$  at time 25. In 7(a) we see the ground truth for this predictive region. The reconstructed fields follow for correlation kernel  $C_{ESEP}$  in Fig. 7(b),  $C_{ESEPLSP}$  in Fig. 7(c), and  $C_{LSP}$  in Fig. 7(d). In each case, we have depicted the simulated trajectory of the AUV up to step 25 in the Eulerian coordinates. The two omitted Eulerian kernels are moderately well represented by  $C_{ESEP}$  here, and the Lagrangian ones by  $C_{LSP}$ . Note that because of the advection, the samples taken early in the mission (at the north edge of the figures) bear little relation to the ground truth and predictions at those spatial locations at the current time.

## 5 Experiments

In CANON, we have focused on using our Dorado AUV shown in Fig. 8(a) in our experiments at sea in Monterey Bay, California. In tracking patches, typically one or more GPS equipped drifters have been used. Position updates from the drifter are used to evaluate surface current displacement which can then be sent to our AUV. T-REX onboard the AUV uses this information to compute a path in-situ, around this moving body which can be either directly translated in the Eulerian frame – resulting in a direct transposition of the waypoints around the drifter’s last position – or distorted to be projected in the Lagrangian frame [44, 45].

The WireWalker profiling system, developed at Scripps Institution of Oceanography, utilizes surface wind-waves to power a profiling vehicle vertically through



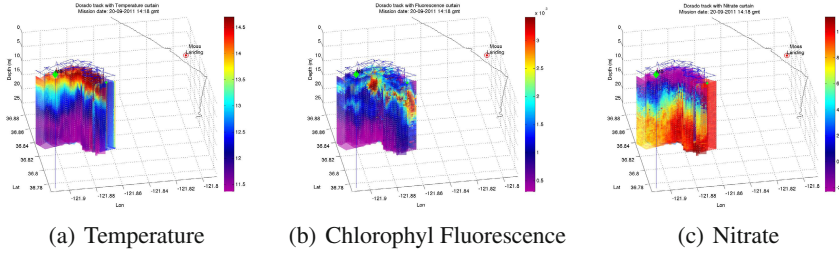
**Fig. 8** MBARI's Dorado AUV platform in 8(a). The WireWalker is shown in 8(b) on the deck of the Scripps research vessel R/V *Roger Revelle* with its buoy, used in the September 2011 experiments. Fig. 8(b) image courtesy: Tony Aja, Scripps Institute of Oceanography

the water column [18, 19]. The design is purely mechanical, lending itself to robust performance, and is equally functional in moored and Lagrangian modes. WireWalker deployments have been conducted in open ocean and coastal settings and have proven their utility in measuring biological and physical dynamics on vertical scales of centimeters and temporal scales of minutes. For example, a canonical month long deployment on a 50m wire will produce as many as 25,000 individual, full water column profiles, with vertical resolution  $< 5\text{cm}$  [46, 47].

In a September 2011 experiment, a WireWalker profiler was deployed on a 22m wire (total profiling distance of  $\sim 20\text{m}$ ). The WireWalker was equipped with a Seabird 49 conductivity, temperature, and depth (CTD) sensor and a Turner designs Cyclops 7 analogue chlorophyll-a fluorometer. Data was recorded onboard the WireWalker at 16Hz. Nominal upcast speed was 0.3m/s with a profile completed on average once every  $\sim 3$  minutes. The WireWalker was deployed with a 75cm diameter surface buoy, tracked using an Iridium satellite uplink with GPS positioning. The WireWalker is expected to follow a pseudo-Lagrangian path, integrating across the sampled portion of the water column. Fig. 9 shows data collected over the night of September 19<sup>th</sup>. Over the span of 20 hours the AUV performed 12 survey boxes of  $1\text{km} \times 1\text{km}$  around the WireWalker without Lagrangian distortion<sup>2</sup> centered by the latest WireWalker position updates received from shore. At the end of each survey T-REX on the vehicle requested the next position update. Both the WireWalker and the AUV shared similar CTD sensors calibrated a priori, which made data analysis tractable.

In addition to temperature, we compare chlorophyll fluorescence measurements across the two devices. However, with different fluorometers and the significant influence of pitch angle and speed (neither of which was constant across platforms) some work was required to reconcile the data sets from the AUV and the WireWalker. For our purposes, which is to compare space-time models rather

<sup>2</sup> This choice was made to be able to predict AUV waypoints and ensure its safety as the mission was relatively close to shore.



**Fig. 9** Interpolated results from September 2011 T-REX AUV mission in Monterey Bay with the WireWalker deployed over a 20 hour overnight mission

than precise chlorophyll measurement, it suffices to transform one of the data sets to approximately match the distribution of the other. Chlorophyll fluorescence is known to be generally lognormally distributed, so we take the log of both data sources and transform the result from the AUV such that the first two moments of the resulting data vector match those of the log transformed measurements from the WireWalker.

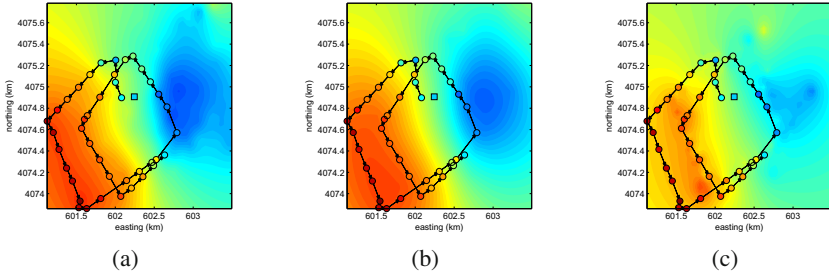
### 6 Analysis of Experimental Results

We next use the methods discussed in Sections 3 and 4 to examine some of the results from the drifter following experiment with the WireWalker. To simplify computation and presentation, we restrict analysis to horizontal slices of the water column at a series of fixed depths. We will focus on an 8.5 hour mission segment starting at 11am on September 20, 2011, and consider reconstructions of temperature, salinity and chlorophyll fluorescence.

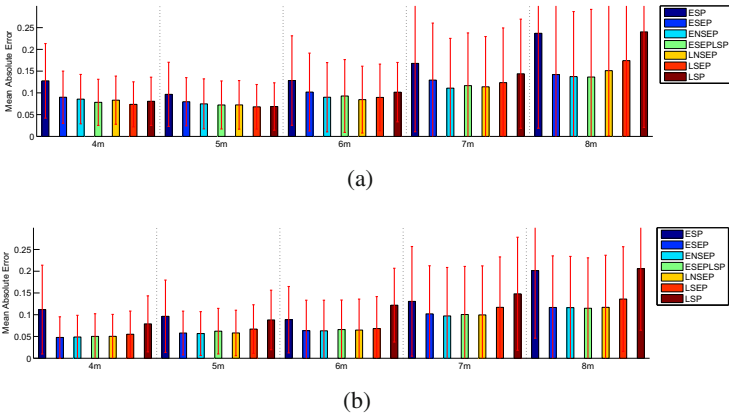
Fig. 10 depicts examples of the Chlorophyll Fluorescence field reconstruction at 6 meters, based on three of the correlation kernels. For chlorophyll fluorescence, the posterior spatial ranges stayed between 100 and 2000 meters, depending on the other parameters and the depth slice. Temporal correlation ranges were between .05 and .56 days. The spatial to temporal mixing coefficient,  $\lambda$ , was strictly within (.5, .65) for all MCMC runs with the  $C_{ESEPLSP}$  kernel. Thus, at least for chlorophyll, both coordinate systems contributed *evenly* to that kernel. In Fig. 11, we show the results of comparison against ground truth and the leave-one-out validation.

By these metrics, it would appear that the differences between separable and non-separable (and mixed) kernels are relatively small, and that all do a far better job than the purely spatial analysis. It can also be seen that the leave-one-out errors are moderately better than the comparisons against WireWalker data, both in terms of the mean error as well as the variance of errors, although perhaps not so much as one might expect.

Fig. 12 depicts examples of the salinity field reconstruction over the same depth slice. The posterior spatial ranges for salinity were between 30 and 500 meters, the

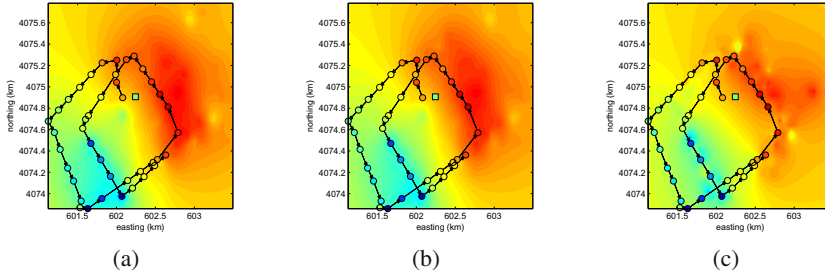


**Fig. 10** Examples of Chlorophyll Fluorescence field reconstruction at a fixed prediction time,  $t_p$ . The figures correspond to reconstructions with correlation kernel  $C_{ESEP}$  in Fig. 10(a),  $C_{ESEPLSP}$  in Fig. 10(b), and  $C_{LSP}$  in Fig. 10(c). In each case, only a short piece of the AUV trajectory is depicted, leading up to  $t_p$ . Many more samples before and after are used in the statistical reconstruction. The square in the center marks ground truth measured by the *WireWalker* at  $t_p$ . As opposed to the simulated data, the difference between Eulerian and Lagrangian reconstructions is less distinct, but there is a much stronger difference between the space-time kernels and the purely spatial ones. The non-separable reconstructions appear similar to the result from  $C_{ESEPLSP}$ .

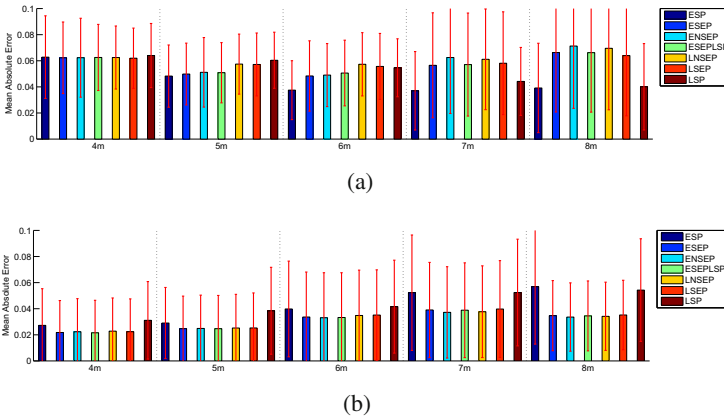


**Fig. 11** Comparison of predictive error distributions of Chlorophyll Fluorescence based on the different correlation models over 5 depth slices (grouped along the x axis). In 11(a) we compare predictions against ground truth from the *WireWalker*. In 11(b) the leave-one-out errors are generated by estimating the value of the  $i$ th measurement based on all other measurements and repeating for all measurements. Box heights depict the mean of the absolute errors, error bars show one standard deviation.

temporal correlation ranges between .03 and .14 days, and  $\lambda$ , was within (.1, .35). This seems to suggest that the measurements are more coherent in the Eulerian frame of reference, but the ground truth comparisons in Fig. 13 do not lend much support to this theory. For the leave-one-out comparison, there does not seem to be

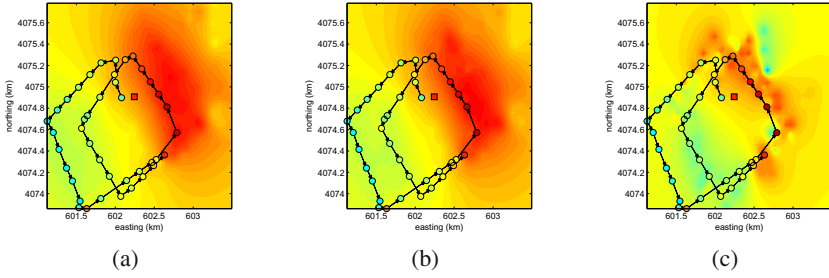


**Fig. 12** Examples of the Salinity field reconstruction at prediction time  $t_p$ . The figures correspond to reconstructions with correlation kernel  $C_{ESEP}$  in Fig. 12(a),  $C_{ESEPLSP}$  in Fig. 12(b), and  $C_{LSP}$  in Fig. 12(c). In each case, only a short piece of the AUV trajectory is depicted, leading up to  $t_p$ . Many more samples before and after are used in the statistical reconstruction. The ground truth provided by the `WireWalker` at  $t_p$  is shown with the square.



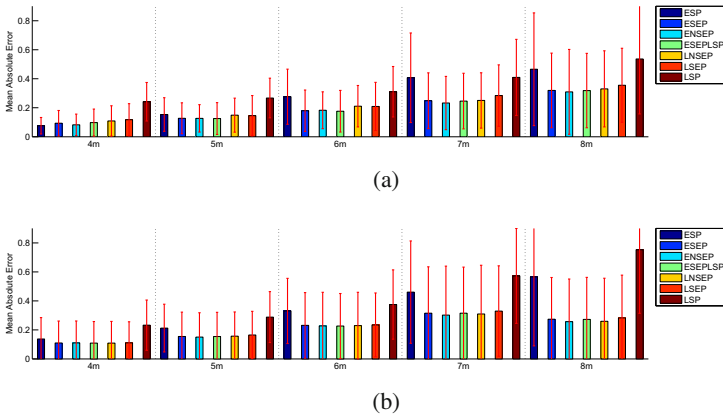
**Fig. 13** Comparison of predictive error distributions of Salinity based on the different correlation models over 5 depth slices (grouped along the x axis). In 13(a) we compare predictions against ground truth from the `WireWalker`. In 13(b) the leave-one-out errors are generated by estimating  $z_i$  from  $\mathbf{z}_{-i}$  for all  $i$ . The box heights show the mean of the absolute errors, while the error bars show one standard deviation.

much difference between the various space-time kernels, however the distinction between leave-one-out and `WireWalker` comparison is dramatic. This is to be expected from the fact that the minimum distance from `WireWalker` to AUV is approximately 500 meters which is the upper limit of the spatial correlation range. The fact that the spatial kernels do better than the space-time ones when compared against the `WireWalker` but worse in the case of the leave-one-out comparison suggests that there may be a certain amount of over fitting on the part of the space-time kernels.



**Fig. 14** Examples of the water Temperature field reconstruction at prediction time  $t_p$ . The figures correspond to reconstructions with correlation kernel  $C_{ESEP}$  in Fig. 14(a),  $C_{ESEPLSP}$  in Fig. 14(b), and  $C_{LSP}$  in Fig. 14(c). In each case, only a short piece of the AUV trajectory is depicted, leading up to  $t_p$ . Many more samples before and after are used in the statistical reconstruction. The *WireWalker* ground truth at  $t_p$  is depicted by the square.

Fig. 14 depicts examples of the Temperature field reconstruction over the same depth slice. For temperature, the posterior spatial ranges were also between 30 and 500 meters, the temporal correlation ranges between .02 and .15 days, and  $\lambda$  was within  $(0, .15)$ . Again, this might suggest that the Eulerian kernels will perform significantly better than the Lagrangian counterparts, but this is not clearly evident in the comparison results shown in Fig. 15. Here the striking thing is that in most



**Fig. 15** Comparison of predictive error distributions of water Temperature based on the different correlation models over 5 depth slices (grouped along the x axis). In 15(a) we compare predictions against ground truth from the *WireWalker*. In 15(b) the leave-one-out errors are generated by estimating the value of the  $i$ th measurement based on all other measurements and repeating for all measurements. Box heights show the mean absolute error, error bars one standard deviation.

of the cases the `WireWalker` comparisons are better (both in terms of mean and variation) than the leave-one-out comparisons.

To sum up these results, it is not clear that the space-time separability of these three variables (for this particular mission) is significantly impacted by whether the correlation kernels are defined in Lagrangian or Eulerian space, or a linear combination of the two. There is clearly a significant benefit (in most cases) to incorporating temporal correlation, but there does not appear to be sufficient evidence, using the methods in this paper, to suggest that the space-time fields studied are distinctly non separable.

## 7 Conclusions and Future Work

We set out to quantify the notion that following a drifting device in the real ocean and using a pseudo-Lagrangian context for field reconstruction may improve the quality of estimation. The hypothesis is that a field which was initially not separable in space and time could be made approximately separable by a complete or partial transformation into drifter-relative coordinates. In an idealized simulation, the results are clear. When the Lagrangian context completely removes temporal variation, significant improvements in error distributions were found. The removal of advection and subsequent treatment in a purely spatial frame of reference is clearly preferable to any space-time treatment for this simplified case.

When the method was extended to a real world experiment in coastal waters, the results were less clear. We found little difference in terms of these error metrics between separable and non-separable treatments. This could be the result of a number of factors. A simple answer is that the fields in question are actually separable in Eulerian space, or are no more separable in the pseudo-Lagrangian frame of reference. This could result from the `WireWalker` being a poor proxy for overall advection in this study, since it is driven by some average current over its depth range. It could be that the turbulent dynamics of the coastal ocean are too complex to get any benefit out of a simple horizontal approximation. It is also possible that a more comprehensive space-time model (e.g., one which incorporated mean field estimation, horizontal anisotropy, or the full three dimensional measurement space) would yield a more conclusive result. An altogether different possibility is that the density of samples in the space-time domain is insufficient to make the distinctions we are trying to make.

Future work will address these deficiencies. We will take into account the full three spatial dimensions, and consider more complete models for the mean field. We also plan to compare model uncertainty in a fully Bayesian setting against the comparison metrics used here. Further experiments will be run to determine if there are circumstances such as smaller survey patterns or stronger current which make the distinction between separable and non-separable reconstructions more pronounced. Yet another idea for future work is to carefully target the survey pattern such that results may be considered approximately gridded in either Eulerian or Lagrangian space. In this case, methods such as that proposed in [37] may be used to



approximate separability, or even to determine how “close” the field is to separable (in terms of a matrix distance metric).

**Acknowledgements.** This work was supported in part by the National Science Foundation under grant #1124975 for the COMECO: Collaborative Multi-Robot Exploration of the Coastal Ocean project. MBARI authors are also funded by a block grant from the David and Lucile Packard Foundation to MBARI. The authors thank their colleagues in the CANON program as well as the crew of the R/V *Zephyr* for supporting their field experiments.

## References

1. Ferrari, R.: A Frontal Challenge for Climate Models. *Science* 332, 316–317 (2011)
2. D’Asaro, E., Lee, C., Rainville, L., Harcourt, R., Thomas, L.: Enhanced turbulence and energy dissipation at ocean fronts. *Science* 10 (March 2011)
3. Barth, J., Herbert, D., Dale, A., Ullman, D.: Direct observations of along-isopycnal upwelling and diapycnal velocity at a shelfbreak front. *J. Physical Oceanography* 34, 543–565 (2004)
4. Levy, M.: Mesoscale variability of phytoplankton and of new production: Impact of the large-scale nutrient distribution. *J. Geophysical Research-Oceans* 108 (2003)
5. Levy, M., Klein, P., Jelloul, M.B.: New production stimulated by high-frequency winds in a turbulent mesoscale eddy field. *Geophysical Research Letters* 36 (2009)
6. Pallás-Sanz, E., Johnston, T.M.S., Rudnick, D.L.: Frontal dynamics in a California Current System shallow front: 1. Frontal processes and tracer structure. *J. Geophysical Research* 115 (2010)
7. Barth, J., Cowles, T., Kosro, P., Shearman, R., Huyer, A., Smith, R.: Injection of carbon from the shelf to offshore beneath the euphotic zone in the California current. *J. Geophysical Research-Oceans* 107 (2002)
8. Franks, P.J.: Sink or swim: accumulation of biomass at fronts. *Marine Ecology Progress Series*, vol. 82, pp. 1–12 (1992)
9. CANON: Controlled, Agile and Novel Observing Network, <http://www.mbari.org/canon/>
10. McGann, C., Py, F., Rajan, K., Thomas, H., Henthorn, R., McEwen, R.: A Deliberative Architecture for AUV Control. In: Intl. Conf. on Robotics and Automation (ICRA), Pasadena (May 2008)
11. McGann, C., Py, F., Rajan, K., Ryan, J.P., Henthorn, R.: Adaptive Control for Autonomous Underwater Vehicles. In: Proceedings of Assoc. for the Advancement of AI National Conference, Chicago, IL (2008)
12. McGann, C., Py, F., Rajan, K., Ryan, J.P., Thomas, H., Henthorn, R., McEwen, R.: Preliminary Results for Model-Based Adaptive Control of an Autonomous Underwater Vehicle. In: Intl. Symp. on Experimental Robotics (ISER), Athens (2008)
13. Py, F., Rajan, K., McGann, C.: A Systematic Agent Framework for Situated Autonomous Systems. In: 9th International Conf. on Autonomous Agents and Multiagent Systems, Toronto, Canada (May 2010)
14. Rajan, K., Py, F.: T-REX: Partitioned Inference for AUV Mission Control. In: Roberts, G.N., Sutton, R. (eds.) *Further Advances in Unmanned Marine Vehicles*. IEE (2012) (to be Published)
15. Cressie, N.A.C.: *Statistics for spatial data*. Wiley Interscience (1993)

16. Das, J., Py, F., Maughan, T., O'Reilly, T., Messie, M., Ryan, J., Rajan, K., Sukhatme, G.: Simultaneous Tracking and Sampling of Dynamic Oceanographic Features with Autonomous Underwater Vehicles and Lagrangian Drifters. In: *Intl. Symp. on Experimental Robotics (ISER)*, New Delhi, India (2010)
17. Das, J., Py, F., Maughan, T., Messie, M., O'Reilly, T., Ryan, J., Sukhatme, G.S., Rajan, K.: Coordinated Sampling of Dynamic Oceanographic Features with AUVs and Drifters. *Intl. J. of Robotics Research* 31, 626–646 (2012)
18. Rainville, L., Pinkel, R.: Wirewalker: An autonomous wave-powered vertical profiler. *J. Atmos. and Ocean. Technology* 18, 1048–1051 (2001)
19. Pinkel, R., Goldin, M.A., Smith, J.A., Sun, O.M., Aja, A.A., Bui, M.N., Huguen, T.: The Wirewalker: A Vertically Profiling Instrument Carrier Powered by Ocean Waves. *J. Atmos. and Ocean. Technology* 28(3), 426–435 (2011)
20. Chavez, F.P., Pennington, J.T., Herliem, R., Jannasch, H., Thurmond, G., Friederich, G.E.: Moorings and Drifters for Real-Time Interdisciplinary Oceanography. *Journal of Atmospheric and Oceanic Technology* 14(5), 1199–1211 (1997)
21. Molcard, A., Piterbarg, L.I., Griffa, A., Ozgokmen, T.M., Mariano, A.: Assimilation of drifter observations for the reconstruction of the eulerian circulation field. *J. Geophys. Res.* 108(C3), 11–121 (2003)
22. Salman, H., Jones, C., Ide, K.: Using Flow Geometry for Drifter Deployment in Lagrangian data assimilation. *Tellus* 60A, 321–335 (2008)
23. Rudnick, D., Perry, M.: ALPS: Autonomous and Lagrangian Platforms and Sensors, Workshop Report. Tech. Rep. (2003), <http://www.geo-prose.com/ALPS>
24. Zhang, Y., Bellingham, J.G.: An efficient method of selecting ocean observing locations for capturing the leading modes and reconstructing the full field. *Journal of Geophysical Research* 113(C4) (April 2008)
25. Wikle, C.K.: *Spatio-temporal Methods in Climatology*, pp. 1–12 (September 2011)
26. Dou, Y., Le, N.D., Zidek, J.V.: Bayesian Empirical Orthogonal Functions. Tech. Rep. (January 2009)
27. Zhang, Y., Baggeroer, A.B., Bellingham, J.G.: Spectral-Feature Classification of Oceanographic Processes Using an Autonomous Underwater Vehicle. *IEEE Journal of Ocean Engineering* 26(4), 726–741 (2001)
28. Handcock, M.S., Stein, M.L.: A Bayesian Analysis of Kriging. *Technometrics* 35(4), 403–410 (1993)
29. Wikle, C.K.: Hierarchical Models in Environmental Science. *International Statistical Review* 71(2), 181–199 (2003)
30. Cressie, N., Calder, C.A., Clark, J.S., Hoef, J.M.V., Wikle, C.K.: Accounting for uncertainty in ecological analysis: the strengths and limitations of hierarchical statistical modeling. *Ecological Applications* 19(3), 553–570 (2009)
31. Amani, A., Lebel, T.: Lagrangian kriging for the estimation of Sahelian rainfall at small time steps. *Journal of Hydrology* 192, 125–157 (1997)
32. Seed, A.W., Bowler, N.: A Model to Generate Stochastic Nowcasts of Rainfall. In: *AMS 31st International Conference on Radar Meteorology* (August 2003)
33. Seed, A.W.: A Dynamic and Spatial Scaling Approach to Advection Forecasting. *American Meteorological Society* 42(3), 381–388 (2003)
34. Jenkinson, M., Smith, S.: A global optimisation method for robust affine registration of brain images. *Medical Image Analysis* 5, 143–156 (2001)
35. Gneiting, T., Genton, M.G., Guttorp, P.: Geostatistical Space-Time Models, Stationarity, Separability, and Full Symmetry. In: Finkenstaedt, B., Held, L., Isham, V. (eds.) *Statistical Methods for Spatio-Temporal Systems*, pp. 151–175 (2006)

36. Taylor, G.I.: The Spectrum of Turbulence. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 164(919), 476–490 (1938)
37. Genton, M.G.: Separable approximations of space-time covariance matrices. *Environmetrics* 18(7), 681–695 (2007)
38. Bevilacqua, M., Mateu, J., Porcu, E., Zhang, H., Zini, A.: Weighted composite likelihood-based tests for space-time separability of covariance functions. *Statistics and Computing* 20(3), 283–293 (2009)
39. Fuentes, M.: Testing for separability of spatial–temporal covariance functions. *Journal of Statistical Planning and Inference* 136, 447–466 (2006)
40. Mitchell, M.W., Genton, M.G., Gumpertz, M.L.: A likelihood ratio test for separability of covariances. *Journal of Multivariate Analysis* 97(5), 1025–1043 (2006)
41. Stein, M.L.: Space-Time Covariance Functions. *Journal of the American Statistical Association* 100(469), 1–13 (2005)
42. Abrahamsen, P.: A Review of Gaussian Random Fields and Correlation Functions, Oslo, Norway, Tech. Rep. (April 1997)
43. Belkin, I.M., Cornillon, P.C., Sherman, K.: Fronts in large marine ecosystems. *Progress in Oceanography* 81(1-4), 223–236 (2009), *Comparative Marine Ecosystem Structure and Function: Descriptors and Characteristics*
44. Das, J., Maughan, T., McCann, M., Godin, M., O’Reilly, T., Messie, M., Bahr, F., Gomes, K., Py, F., Bellingham, J., Sukhatme, G., Rajan, K.: Towards mixed-initiative, multi-robot field experiments: Design, deployment, and lessons learned. In: *Proc. Intelligent Robots and Systems (IROS)*, San Francisco, California (2011)
45. Das, J., Py, F., Maughan, T., Messie, M., O’Reilly, T., Ryan, J., Sukhatme, G.S., Rajan, K.: Coordinated sampling of dynamic oceanographic features with auvs and drifters. In: *Proc. IROS Workshop on Robotics for Environmental Monitoring (WREM 2011)*, San Francisco, California (2011)
46. Lucas, A., Dupont, C.L., Tai, V., Largier, J.L., Palenik, B., Franks, P.J.S.: The Green Ribbon: Multiscale physical control of phytoplankton productivity and community structure over a narrow continental shelf. *Limnology & Oceanography* 56, 611–626 (2011)
47. Lucas, A., Franks, P.J.S., Dupont, C.L.: Horizontal Internal-tide Fluxes support elevated phytoplankton productivity over the inner continental shelf. *Limnology & Oceanography: Fluids & Environments* 1, 56–74 (2011)

# Autonomous, Localization-Free Underwater Data Muling Using Acoustic and Optical Communication

Marek Doniec, Iulian Topor, Mandar Chitre, and Daniela Rus

**Abstract.** We present a fully autonomous data muling system consisting of hardware and algorithms. The system allows a robot to autonomously find a sensor node and use high bandwidth, short range optical communication to download 1.2 MB of data from the sensor node and then transport the data back to a base station. The hardware of the system consists of an autonomous underwater vehicle (AUV) paired with an underwater sensor node. The robot and the sensor node use two modes of communication - acoustic for long-range communication and optical for high bandwidth communication. No positioning system is required. Acoustic ranging is used between the sensor node and the AUV. The AUV uses the ranging information to find the sensor node by means of either stochastic gradient descent, or a particle filter. Once it comes close enough to the sensor node where it can use the optical channel it switches to position keeping by means of stochastic gradient descent on the signal quality of the optical link. During this time the optical link is used to download data. Fountain codes are used for data transfer to maximize throughput while minimizing protocol requirements. The system is evaluated in three separate experiments using our Autonomous Modular Optical Underwater Robot (AMOUR), a PANDA sensor node, the UNET acoustic modem, and the AquaOptical modem. In the first experiment AMOUR uses acoustic gradient descent to find the PANDA node starting from a distance of at least 25 m and then switches to optical position keeping during which it downloads a 1.2 MB large file. This experiment is completed 10 times successfully. In the second experiment AMOUR is manually steered above the PANDA node and then autonomously maintains position using

---

Marek Doniec · Daniela Rus

Computer Science and Artificial Intelligence Laboratory,  
Massachusetts Institute of Technology

e-mail: doniec@mit.edu, rus@csail.mit.edu

Iulian Topor · Mandar Chitre

Acoustic Research Lab, Tropical Marine Science Institute,  
National University of Singapore, Singapore

e-mail: iulian@arlab.nus.edu.sg, mandar@nus.edu.sg

the quality of the optical link as a measurement. This experiment is performed two times for 10 minutes. The final experiment does not make use of the optical modems and evaluate the performance of the particle filter in finding the PANDA node. This experiment is performed 5 times successfully.

## 1 Introduction

Our goal is to develop technologies that enable users to interact with ocean observatories. In an ocean observatory robots and in-situ sensors collect information about the underwater environment and deliver this information to remote users, much like a web-cam delivers remote data to users on the ground. In this paper we focus on developing effective technologies for wireless data transmission underwater. When the amount of data from an ocean observatory is large (e.g. in the case of image feeds), low-bandwidth acoustic communication is not adequate. We instead propose using optical data muling with a robot equipped with an optical modem that can retrieve data fast from underwater nodes with line-of-site connection to the robot. An important problem is locating the underwater sensor node. When distances between the robot and the nodes are large, and their locations are unknown, positioning the data muling robot within optical communication range is challenging. In this paper we present a solution to autonomous data muling underwater, where the node's location is unknown. The algorithm has three phases. In the first phase, acoustic communication is used to bring the data muling robot within some close range of the desired sensor where it can detect the optical signal. In the second phase, the robot does a local search using the optical signal strength to precisely locate the sensor and position itself within communication range. In the third phase the robot uses optical communication to collect the data from the sensor. In practice, phase two and three overlap once the signal strength becomes strong enough to transmit data.

Previous work has looked at the theoretical performance of data muling [10] and the optimization of the path taken between nodes [7]. In both cases the locations of the nodes are assumed to be known. Data muling with an underwater robot has been previously shown in [5]. The nodes were found using a spiral search that looked for a valid optical signal. A method for homing to a single beacon using acoustic ranging based on an Extended Kalman Filter with a fixed robot maneuver for initialization is presented and evaluated in simulation in [13]. In [9] the authors present an Extended Kalman Filter approach to localizing a moving vehicle using range-only measurements to a group of beacons. They use particle filters to initialize the beacons location. In [1] a high-frequency acoustic network is suggested, that offers range and bandwidth performance between conventional acoustic and optical rates.

We implemented and experimentally evaluated the data muling system described in this paper. This work uses a new version of the Pop-up Ambient Noise Data Acquisition sensor node called UNET-PANDA, which is presented, along with the acoustic modem used, in [2]. For simplicity UNET-PANDA is referred to as PANDA in the remainder of the paper. The optical modem has been described in [3]. The Autonomous Modular Optical Underwater Robot (AMOUR) was presented in [4].

**Algorithm 1.** Acoustic Stochastic Gradient Descent

---

```

1:  $YAW_{robot} \leftarrow \pi * Random(-1.0...1.0)$ 
2:  $SPEED_{robot} \leftarrow 0.25$  Knots forward
3:  $RANGE_{th} \leftarrow inf$ 
4: while No optical link available do
5:   Receive  $RANGE_m$ 
6:   if  $RANGE_m \geq RANGE_{th} + 1m$  then
7:      $YAW_{robot} \leftarrow YAW_{robot} + \pi + \pi * Random(-0.5...0.5)$ 
8:      $RANGE_{th} \leftarrow RANGE_m$ 
9:   end if
10:  if  $RANGE_m < RANGE_{th}$  then
11:     $RANGE_{th} \leftarrow RANGE_m$ 
12:  end if
13: end while
14: Begin Optical Gradient Descent

```

---

**Algorithm 2.** Optical Stochastic Gradient Descent

---

```

1: Retain  $YAW_{robot}$  from Algorithm 1.
2: Retain  $SPEED_{robot}$  from Algorithm 1.
3:  $SSI_{th} \leftarrow 0$ 
4: while Optical Link Established do
5:   Wait 0.25 Seconds. Measure  $SSI_m$ 
6:   if  $SSI_m \leq 0.9 * SSI_{th}$  then
7:      $YAW_{robot} \leftarrow YAW_{robot} + \pi + \pi * Random(-0.5...0.5)$ 
8:      $SSI_{th} \leftarrow SSI_m$ 
9:   end if
10:  if  $SSI_m > SSI_{th}$  then
11:     $SSI_{th} \leftarrow SSI_m$ 
12:  end if
13: end while
14: Switch back to Acoustic Gradient Descent

```

---

Our implementation of the data muling system was repeatedly able to acoustically locate the sensor node from distances of 25 m and 100 m and to download a 1.2 MB data file optically once the node was found.

## 2 Problem Statement

We consider a sensor node that is deployed at a fixed location on the seafloor. We assume that the sensor node is equipped with an acoustic modem and an optical modem. We use the acoustic modem for low data rate ( $\leq 1$  Kbps) and long-range ( $\geq 100$  m) communications. We use the optical modem for high data rate ( $\geq 1$  Mbps) and short-range ( $\leq 100$  m) communications. We do not require a precise external positioning system but we assume that a coarse location estimate of the node exists. By coarse we mean that the margin of error for this position estimate is within the acoustic communication range. This is usually on the order of hundreds of meters to a few kilometers, though acoustic communication ranges of over 100 km are possible [11]. Examples in which such a situation can arise are (1) when a node is deployed in deep waters from a boat and drifts before it finally reaches the ocean floor; (2) when a node is deployed by an autonomous underwater vehicle using dead reckoning the placement can have a large error; (3) when a node is not rigidly moored and its position changes with time because of water currents.

Further, we assume that an autonomous hovering underwater vehicle (AUV) is equipped with identical acoustic and optical modems and capable of communicating through these with the sensor node when in range. Hovering enables the vehicle to hold its attitude and depth statically and to execute surge (forward / backward) velocity commands.

Our problem statement concerns the case in which the sensor node is collecting data at a faster rate than can be transmitted using the long distance acoustic channel. Kilfoyle *et al.* show empirically that the product of acoustic communication rate and bandwidth rarely exceeds 40 km-Kbps for state of the art acoustic modems [6]. For a single sensor separated by 5 km from the user this would result in a communication rate of 8 Kbps. If we consider an application that collects ambient acoustic signals or video our data stream will far exceed the available acoustic channel capacity.

### 3 Technical Approach

We developed a combined acousto-optical communication network capable of large scale data recovery that does not require precise localization of the robot nor the sensor node. The robot uses acoustic communication and ranging to come close to the sensor node. High bandwidth optical channel allows the robot to download the payload data. More specifically, our approach to data muling is as follows:

1. We use acoustic ranging between the robot and the PANDA sensor node. The acoustic modem on the PANDA transmits a ranging beacon every 6 s that is received by the acoustic modem on the robot and provides it with a range measurement. The robot uses the stochastic gradient descent algorithm shown in Algorithm 1 to travel close to the PANDA.
2. At all times the PANDA is streaming the payload data using the optical modem and random linear rateless erasure codes known as Luby transform (LT) codes [8]. The payload data is a random file consisting of 2048 blocks of 576 bytes. The LT-Codes require on average an overhead of 3 %, so about 2109 packets have to be received by the robot to decode the entire file. Because of the nature of the LT-Codes it does not matter which packets are received.
3. Once the robot is close enough to the sensor node to receive an error-free packet it switches from acoustic gradient descent to maintaining position using the optical gradient descent algorithm described in Algorithm 2. If the optical connection breaks at any point in time we return to step 1.
4. While AMOUR is in optical communication range every packet is used to (a) measure the signal strength and (b) decode the payload data if the CRC matches.
5. The experiment is considered to have completed successfully once the entire 1.2 MB file has been received and decoded by the robot. In a real world scenario the robot would now continue on the approximate location of the next sensor node to begin acoustic gradient descent there.

### 4 Performance Improvement with a Particle Filter

The stochastic gradient descent approach described in Section 3 has no memory of previous decisions. The only state variables are the current heading and a range threshold used to make the decision whether to keep going straight or to turn. When the algorithm encounters an increasing range it changes the direction of the robot

in a random direction at least 90 degrees different from the current direction of travel. This new choice of direction takes into account only the most recent few measurements as reflected in the threshold stored. Because so little information is taken into account, bad choices are made frequently. Further, even when the robot is moving in the right direction a single spurious measurement caused by noise can make it veer of the correct course. The algorithm will recover from this mistake with high probability as the ranges will keep increasing from here on, but this comes at the cost of time and energy. It also causes a large variance in the time that it takes to find the target sensor node.

A more effective algorithm should keep a belief of where the robot is relative to the sensor node and update this belief with every measurement. An Extended Kalman Filter (EKF) delivers such a behavior. It represents the current belief of the robot's location as a mean and covariance. Because of this it needs to be initialized, for example by performing a circular maneuver such as in [13]. Further, because we are representing the robot's state with a multidimensional Gaussian, we cannot represent multimodal distributions, for example when we have a baseline ambiguity because our vehicle has been traveling straight.

In order to represent multimodal distributions we implement a particle filter algorithm, as shown in Algorithm 3. The filter represents the current belief using  $N$

---

**Algorithm 3.** Acoustic Particle Filter
 

---

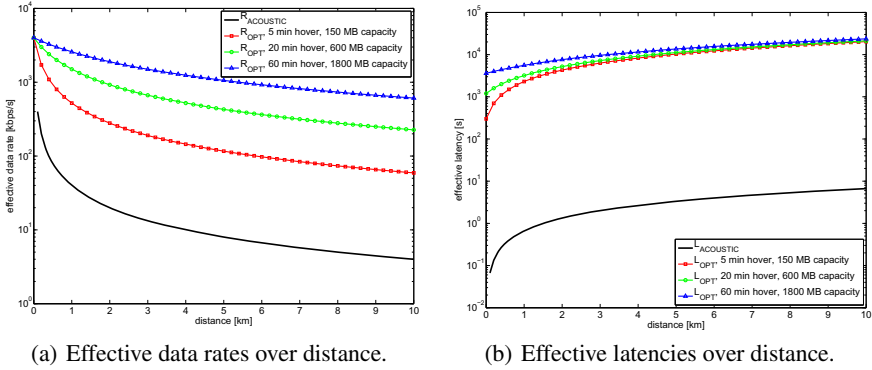
```

1:  $YAW_{robot} \leftarrow \pi * Random(-1.0..1.0)$ ,  $SPEED_{robot} \leftarrow 0.25$  Knots forward
2: Receive first measurement  $RANGE_m$ 
3: for  $k = 1 \dots N$  do
4:    $\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = RANGE_m \cdot \begin{bmatrix} \cos(\alpha_k) \\ \sin(\alpha_k) \end{bmatrix}$ , where  $\alpha_k = Random(-\pi \dots \pi)$ 
5: end for
6: while No optical link available do
7:   for  $k = 1 \dots N$  do
8:     Independently draw  $e_x$  and  $e_y$  from  $\mathcal{N}(0.0, \sigma_{robot})$ 
9:      $\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + SPEED_{robot} \cdot \begin{bmatrix} \cos(YAW_{robot}) \\ \sin(YAW_{robot}) \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix}$ 
10:   end for
11:   if Received new measurement  $RANGE_m$  then
12:     for  $k = 1 \dots N$  do
13:        $W_k = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_{range}}} \cdot \exp(\frac{-D_k^2}{2 \cdot \sigma_{range}})$ , where  $D_k = RANGE_m - \sqrt{X_k^2 + Y_k^2}$ 
14:     end for
15:     RESAMPLE particles using weights  $W_k, k \in \{1 \dots N\}$ .
16:   end if
17:    $\begin{bmatrix} \bar{Z}_X \\ \bar{Z}_Y \end{bmatrix} = \frac{1}{N} \sum_{n=1}^N \begin{bmatrix} -X_k \\ -Y_k \end{bmatrix}$ ,  $\mu_\theta = atan(\bar{Z}_Y, \bar{Z}_X)$ ,  $\sigma_\theta = \sqrt{\ln(1/\|\bar{Z}\|)}$ 
18:    $YAW_{robot} = \mu_\theta + \sigma_\theta / 4$ 
19: end while
20: Begin Optical Gradient Descent

```

---





**Fig. 1** The left graph shows effective data rates for given distances between the sensor node and the user. The x-axis shows the distance in km and the y-axis the data rate. Black shows data rates for acoustic communication. The colored lines show data rates for data muling with different times spent hovering above the sensor to download the data. The right graph shows effective latencies for the same cases as in the left figure. The x-axis shows the distance in km and the y-axis the resulting latency in seconds. The latencies are reported as the entire round trip time (worst-case latency).

particles. The sensor node position is assumed to be fixed at the origin. The filter localizes the robot relative to the sensor node. Each particle stores one guess  $[X_k Y_k]$  of the robot’s location. The algorithm initializes the particles when the first range measurement  $R_m$  is received by randomly placing all of the particles on a circle of radius  $R_m$  around the origin (lines 3 to 5). Once the particles have been initialized, we perform a prediction step every  $100ms$ , taking into account robot movement noise with a standard deviation  $\sigma_{robot}$  (lines 7 to 10). When a new range measurement is received we compute the probability of observing such a range for every particle taking into account the measurement noise  $\sigma_{range}$ . The particles are then re-weighted according to the algorithm presented in Thrun *et al.* [12]. At the end of every iteration we compute a new heading for the robot. This is done by computing the heading required for every particle to travel towards the node. We assume these headings form a wrapped normal distribution and we compute its mean and standard deviation (line 17). Setting the robot heading to the mean would result in the particles traveling directly towards the node and can create a baseline ambiguity. Thus, we chose to set the new headings as the mean plus the standard deviation divided by a factor of 4. The more uncertain the particle filter is, the more the robot will deviate from the straight path, and this in turn helps resolve the baseline ambiguity.

### 5 Theoretical Performance

Figure 1(a) shows the upper bounds of achievable data rates for our approach plotted against the travel distance for the AUV (one way). The black line shows the achievable data rates using the acoustic channel computed as

$$R_{acoustic}(d) = \frac{40,000 \text{ m-Kbps}}{d}.$$

according to Kilfoyle *et al.* [6]. In color are shown the upper bounds for different time intervals that the AUV spends above the sensor nodes. We compute the upper bounds of the effective data rates for data muling as follows. We assume the AUV travels on a direct path (best possible case) to the sensor node at a speed of  $v_{AUV} = 1$  m/s. This is the speed at which AMOUR can travel. We assume the optical data rate to be  $r_{OPT} = 4$  Mbps. This is the data rate of AquaOptical as used during the experiments. We call  $d$  the distance of the sensor from the user (one way trip in meters) and  $t_{hover}$  the time that the robot hovers above the sensor node to download the data. Under these assumptions the total travel time of the robot from the user to the node and back, including the time spent hovering above the sensor node, is

$$t_{travel}(d) = t_{hover} + \frac{2 \cdot d}{v_{AUV}}$$

This results in an effective data rate of

$$r_{OPT,t_{hover}}(d) = \frac{t_{hover} \cdot r_{OPT}}{t_{travel}(d)}.$$

Figure 1(b) shows the resulting latencies under the same assumptions. For the acoustic communication we compute the latency as travel time of sound in water, i.e.

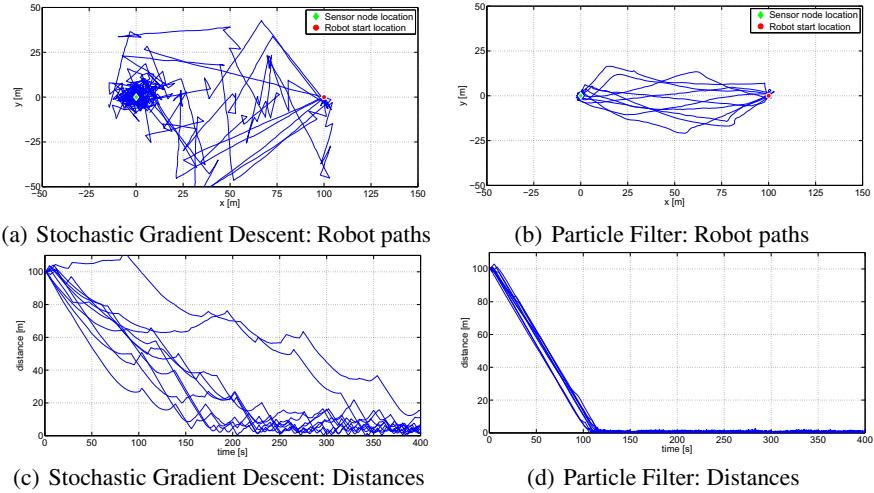
$$L_{acoustic}(d) = \frac{d}{1,500 \text{ m/s}}$$

The optical latency is equivalent to  $t_{travel}(d)$ . Figure 1(a) shows that the achievable data rate when using data muling far exceeds the currently available acoustic data rates. This effect can even be amplified by using multiple AUVs that can travel in parallel to either a single or multiple sensor nodes. Using acoustic communication neighboring nodes often have to share the medium reducing the effective data rate per node. The disadvantage of data muling is its higher latency as seen in Fig. 1(b).

## 6 Simulations

We evaluated both the acoustic stochastic gradient descent algorithm (Algorithm 1), and the acoustic particle filter algorithm (Algorithm 3) in simulation. In each simulation the robot state was represented as  $[X_{robot} \ Y_{robot} \ YAW_{robot}]$ . The robot was simulated with a constant speed of  $SPEED_{robot} = 1$  m/s. Independent white Gaussian noise with a standard deviation  $\sigma_{robot}$  in meters was added to the robot's position every second to simulate movement errors. Thus, every second the new robot position was computed as

$$\begin{bmatrix} X_{robot}(t+1) \\ Y_{robot}(t+1) \end{bmatrix} = \begin{bmatrix} X_{robot}(t) \\ Y_{robot}(t) \end{bmatrix} + SPEED_{robot} \cdot \begin{bmatrix} \cos(YAW_{robot}) \\ \sin(YAW_{robot}) \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix}$$



**Fig. 2** Sample simulation results of 10 stochastic gradient descents and 10 particle filter node localizations using acoustic ranging in both cases. All simulations were performed with a robot speed of 1 m/s, measurement noise  $\sigma_{range} = 1$  m and robot motion noise  $\sigma_{robot} = 0.1$  m. Range measurements occurred every 1 s. Plots (a) and (b) show the resulting robot paths. The x and y-axes show displacement in meters. The sensor node is located at the origin (green circle) and the robot starts at location (0 m, 50 m) (red diamond). Each continuous blue line denotes one simulation run. Plots (c) and (d) show the corresponding distances of the robot from the sensor node in m on the y-axis over time in seconds on the x-axis.

where  $e_x$  and  $e_y$  are independently drawn from  $\mathcal{N}(0.0, \sigma_{robot})$ . Measurements were simulated every second with added Gaussian noise with a standard deviation  $\sigma_{range}$ . Each new measurement  $RANGE_m$  is computed as

$$RANGE_m = \sqrt{X_{robot}^2 + Y_{robot}^2 + e_r}$$

where  $e_r$  is drawn from  $\mathcal{N}(0.0, \sigma_{range})$ .

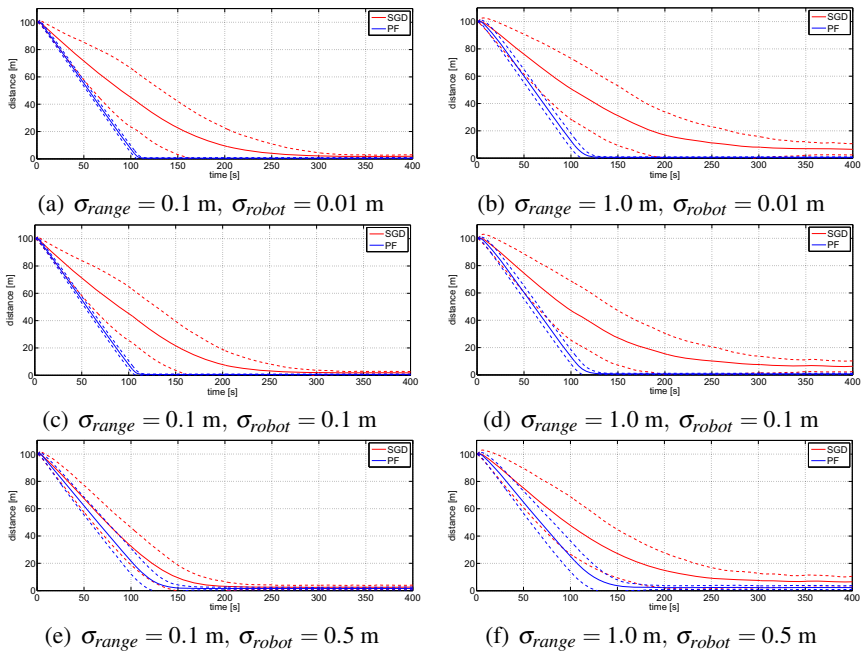
Figure 2 shows two sets of 10 simulated paths taken by the robot using stochastic gradient descent (Fig. 2(a)) and a particle filter (Fig. 2(b)). The simulations were performed using Algorithm 1 and 3. The parameters for these simulations were  $\sigma_{range} = 1$  m and  $\sigma_{robot} = 0.1$  m. These plots visualize the characteristic difference in paths generated by the stochastic gradient descent and the particle filter. When the stochastic gradient descent encounters an increasing range, it picks a new direction almost entirely at random. The particle filter, on the other hand, continuously merges all gathered information about the sensor node location and continuously updates the robot’s heading resulting in a more direct path. Plotted in Figures 2(c) and 2(d) are the corresponding distances of the robot to the sensor node over time.

Finally we conducted six sets of simulation runs with parameters chosen as  $(\sigma_{range}, \sigma_{robot}) \in \{0.1 \text{ m}, 1.0 \text{ m}\} \times \{0.01 \text{ m}, 0.1 \text{ m}, 1.0 \text{ m}\}$ . For each set we simulated 1000 runs using the stochastic gradient descent algorithm and 1000 runs using the particle filter. Figure 3 shows the results for all these runs grouped in six plots

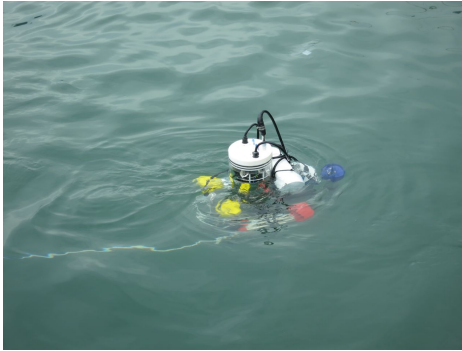
according to parameter choices. Each plot shows the mean distance over time (solid line) with  $1\sigma$  boundaries (dashed lines). The stochastic gradient descent results are plotted in red and the particle filter results are plotted in blue. In all six cases the particle filter outperforms the stochastic gradient descent. When the noise is low (Figures 3(a) and 3(c)), the particle filter takes on average 104 s for the robot to come to within 5 m of the sensor node. This is only 10 % more than the theoretical minimum, which is 95 s since the robot starts 95 m away and travels at 1 m/s.

## 7 Hardware

We used our in-house developed hardware for the experiments presented in this paper. The Acoustic Research Lab at National University Singapore developed the UNET2 Acoustic Modems. The UNET2 modems use a carrier frequency of 27 kHz with a transmission bandwidth of 18 kHz. The maximum power level for



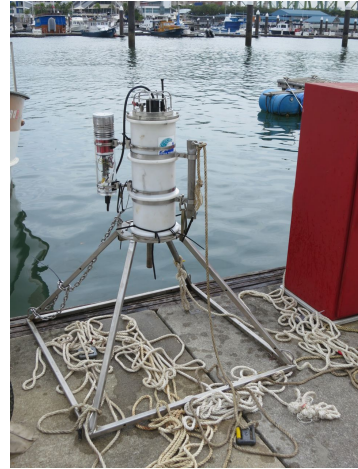
**Fig. 3** Simulation results. Each plot corresponds to one of six different choices for  $(\sigma_{range}, \sigma_{robot})$ , the measurement noise and robot motion noise used in the simulation. All simulations were performed with a robot speed of 1 m/s and range measurements occurred every 1 s. The x-axis corresponds to time in seconds and the y-axis corresponds to the distance in meters of the simulated robot to the sensor node. Plotted in red are the results of stochastic gradient descent and in blue the results of the particle filter. For each algorithm and choice of  $(\sigma_{range}, \sigma_{robot})$  1000 simulations were performed for a total of 12000 simulations. The solid lines correspond to the mean, the dotted lines represent the  $1\sigma$  boundaries.



(a) Picture of AMOUR 6 [4].



(b) Picture of experimental site.



(c) Picture of PANDA with Optical Modem.

**Fig. 4** (a) AMOUR 6 in the water with acoustic and optical modems attached. (b) Experimental site. The PANDA was deployed in the center of the basin shown between the two docks. (c) PANDA node (white cylinder on tripod) with Optical Modem attached on the left.

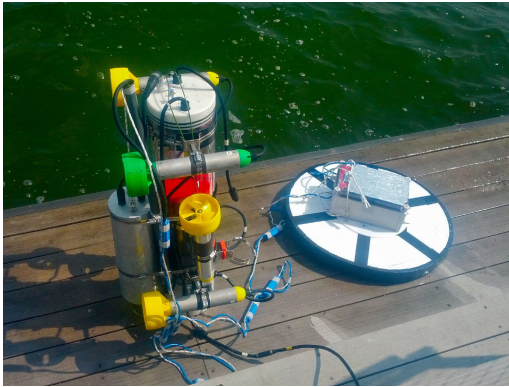
transmissions is 180 dB measured at 1 m. We use Orthogonal Frequency Division Multiplexing with 256 carriers per symbol. For the inner code we chose a 1/3-rate convolution code and for the outer code we chose a 12/23 Golay code. The acoustic modem on the PANDA transmits an 18 byte long ranging beacon every 6 s that is received by the acoustic modem on the robot and provides it with a range measurement.

The Distributed Robotics Lab at MIT developed the AquaOptical modem and the Autonomous Modular Optical Underwater Robot (AMOUR) used during the experiments. The AquaOptical modem communicates using visible blue light with a wavelength of 470nm. The signal is amplitude modulated using on-off shift keying (OOK). Packets are delimited with a 13-bit Barker code and data is encoded using Manchester code. Each optical packet transmitted contained 576 bytes payload data plus 32 bytes of configuration data (i.e. source and destination address, packet size, 32-bit CRC checksum, degree and seed used for the LT-Codes). The robot consists of a set of thrusters that can be attached in arbitrary locations to the robot's main body, which contains computation, power electronics, and a battery.

During the experiments two different hardware configurations were used. In the first configuration one acoustic and one optical modem were attached to the robot



(a) Panoramic picture of experimental site at Pandan Reservoir.



(b) Picture of AMOUR 6 and floating Wifi.



(c) Picture of PANDA.

**Fig. 5** (a) Experimental site at Pandan Reservoir in Singapore. (b) AMOUR 6 with acoustic modem attached with the WiFi bouy next to the robot. (c) PANDA node with floats and weight.

and it was configured with 5 thrusters allowing it to maintain attitude while traveling at a forward speed of 0.25 m/s. This configuration is shown in the water in Figure 4(a). One optical modem was attached to the PANDA node as shown in Figure 4(c). In the second configuration no optical modems were used. The robot carried one acoustic modem and had 6 thrusters attached. This allowed the robot to travel at a forward speed of 0.5 m/s. The robot and the PANDA can be seen in Figures 5(b) and 5(c).

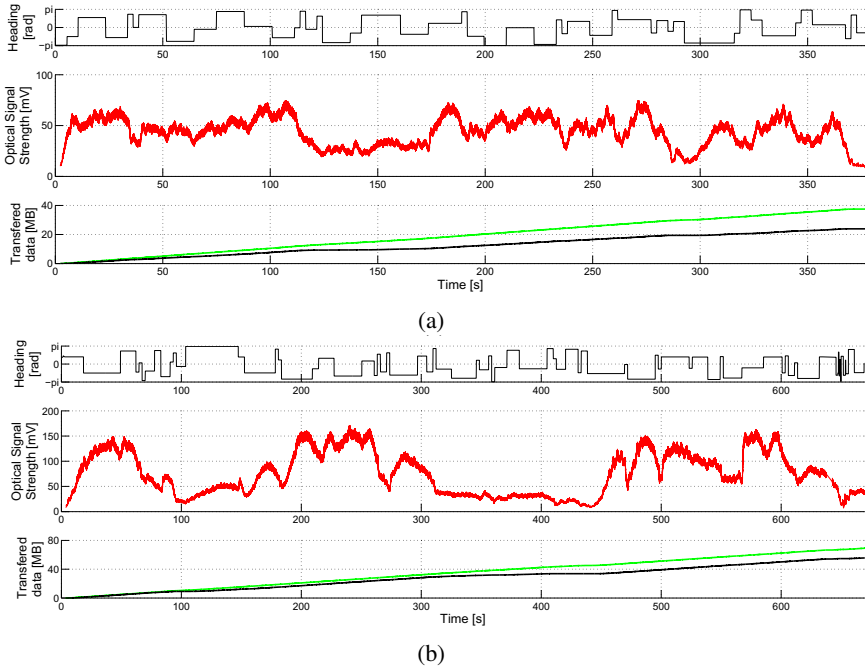
## 8 Experiments

We conducted two sets of experiments to demonstrate the system’s ability to localize a sensor node using a robot and recover data from it. A third set of experiments was conducted to evaluate the performance of the particle filter. In this work we did not focus on the return of the robot to the base station.

The first two sets of experiments were conducted from a dock at the Republic of Singapore Yacht Club (Figure 4(b)). The water depth was about 7 m and we estimated visibility at about 2 m. The PANDA with the optical modem was mounted on a tripod to guarantee that they should be pointing upright after being lowered to the ground. This setup can be seen in Figure 4(c). Our vehicle AMOUR carrying

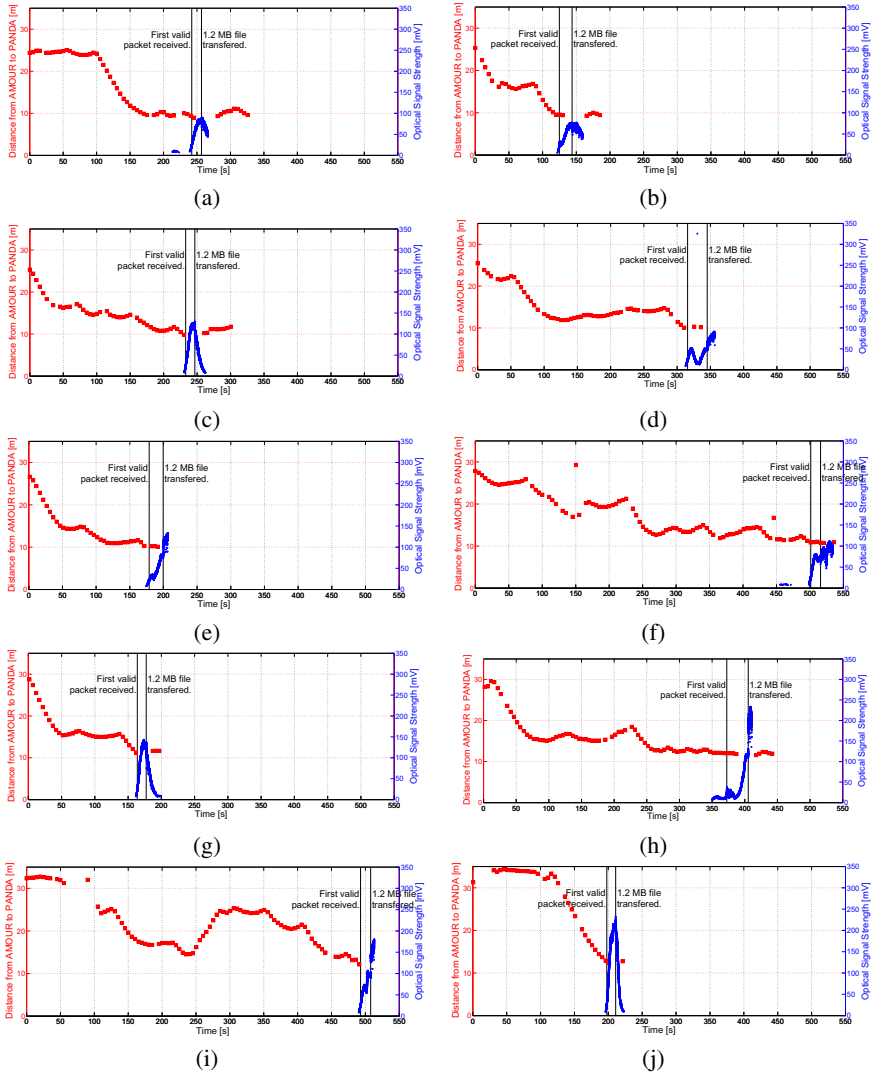
the acoustic and optical modems can be seen in Figure 4(a). It was tethered for data collection and security, but operated autonomously during the experiments. The robot speed was set at about 0.25 m/s to ensure safe operation and to keep distance changes at a reasonable rate between updates. Generating the LT-Codes as described in Section 3 requires substantial computation. Because of this we needed to reduce the number of packets transmitted from the optical modem on the PANDA to 392 packets a second. Including overhead this corresponds to a bit rate of 1.84 MBit/s. The remainder of the optical channel (2.16 MBit/s) was not utilized.

The first set of experiments consisted of manually placing the robot close to the PANDA node and using optical gradient descent to maintain a position close to the PANDA node. This experiment was conducted two times, one time with the robot at



**Fig. 6** Results of two optical gradient descent experiment runs. The first run is plotted in (a), the robot operated at the water surface. The second run is plotted in (b), the robot operated at a depth of 1.5 m. At the beginning of each experiment the robot was manually steered close to the PANDA to establish an optical link. The optical gradient descent algorithm then controlled the robot to stay close to the PANDA. In all plots the x-axis indicates the time in seconds since the beginning of the experiment. The top graph for each experiment (black) shows the heading of the robot as computed by the optical gradient descent algorithm. The middle graph (red) shows the measured signal strength. The bottom graph shows the amount of data received in MB (green) and the amount of data received error-free in MB (black). Packet size was 576 bytes with a 4 byte CRC.





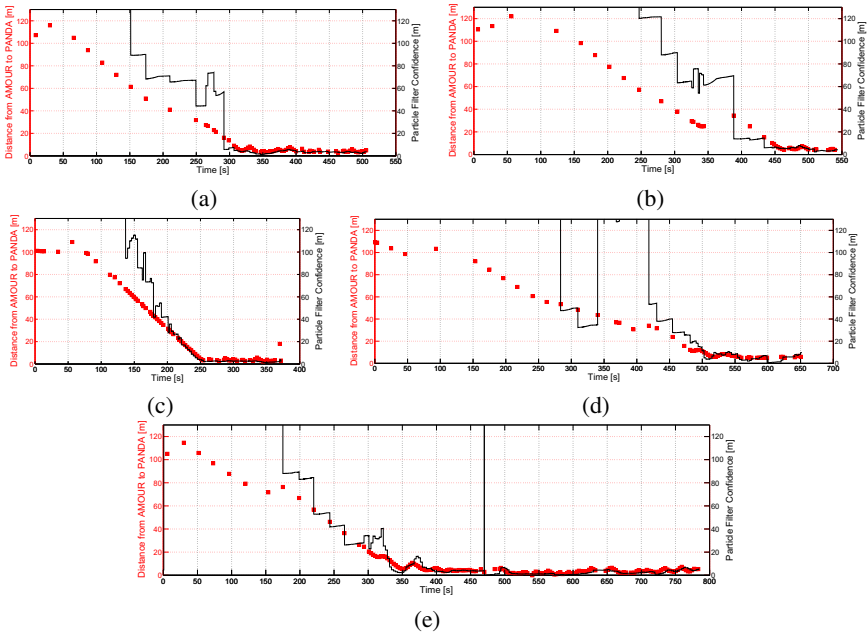
**Fig. 7** Results of data muling experiments. Each graph (a)-(j) shows one experiment. The x-axis shows time in seconds since beginning of the experiment. The red curve (left y-axis) shows the distance between AMOUR and the PANDA node. Each red square corresponds to one range measurement received by AMOUR. The blue curve (right y-axis) shows the optical signal strength between PANDA and AMOUR. A link was established whenever there is non-zero signal strength. Two horizontal black lines mark the receipt of the first error free packet (left line) and the receipt of the final error free packet needed to decode the 1.2 MB test file (right line).



the water surface and the second time with the robot keeping a depth of 1.5 m under the water surface. Given a water depth of about 7 m, a height of the PANDA of about 1 m, and a robot height of slightly below 1 m, the first and second experiment had a minimum distance of 5 m between the optical modems and 3.5 m, respectively.

In the second set of experiments we manually positioned the robot at a distance of about 25 m away from the PANDA, dove it to 2 m depth where we started the acoustic gradient descent algorithm. This experiment was conducted 12 times, of which two were aborted because the robot's tether got entangled with obstacles in the harbor (the robot would dive under the docks sometimes due to the random nature of stochastic gradient descent). We used the other 10 experiments for evaluation.

The third set of experiments to measure the performance of the particle filter was conducted at the Pandan Reservoir in Singapore (Figure 5(a)). The reservoir covers an area of over 1 km<sup>2</sup> and has a depth of about 4 m close to the shore. During this set of experiments we used a floating buoy that carried a long-range WiFi. The buoy was tethered to the robot with a 5 m long Ethernet cable and additionally secured with a rope. This allowed for remote operation of the robot while it was able to move



**Fig. 8** Results of acoustic particle filter experiments. Each graph (a)-(e) shows one experiment. The x-axis shows time in seconds since beginning of the experiment. The red curve (left y-axis) shows the distance between AMOUR and the PANDA node. Each red square corresponds to one range measurement received by AMOUR. The black curve (right y-axis) shows the confidence of the particle filter (square root of the determinant of the covariance of all particle positions, lower is better). The spike visible at second 420 of plot (e) was caused by a spurious range measurement.

freely through the water without having a long tether that could get entangled in the many buoys that are present at the reservoir. To prepare the experiments the robot was used to transport the PANDA about 40 m off shore and drop it there. A weight attached to the bottom of the PANDA together with floats attached at the top ensured that it would sink to the bottom of the reservoir but remain upright (Figure 5(c)). A rope was permanently attached to the PANDA that allowed us to recover it manually after the experiments. During each experiment the robot was manually positioned at a distance of at least 100 m away from the PANDA and the node localization algorithm based on a particle filter was started. At all times the robot traveled with a speed of 0.5 m/s at a depth of 1 m. At distances greater than 20 m the acoustic beacons were often corrupted by the noise of the robot's thrusters. To alleviate this problem we chose to send beacons every 3 s during this experiment instead of every 6 s as in the previous experiments. Further, if the robot did not receive a valid beacon for more than 20 s it stopped its thrusters significantly reducing the acoustic noise levels. This ensured that even at distances beyond 20m we would receive ranging beacons no less than twice per minute. The experiment was conducted 5 times, all of which were used for evaluation.

## 9 Results

The results of the optical gradient descent experiments can be seen in Figure 6. In the first run the robot maintained position for over 6 min before it lost track of the optical signal. During this time the optical modem on the PANDA transmitted 52.6 MB of payload data and on AMOUR received 37.5 MB of which 23.87 MB were error-free packets (one packet was 576 bytes large). In the second run it maintained position successfully for 11 min after which we stopped the experiment. During this time the PANDA transmitted 93.4 MB of payload data and AMOUR received 69.1 MB of which 55.6 MB were error-free packets. The rate of error-free packets was higher in the second run because the robot was operating at a lower depth closer to the transmitter, which resulted in higher signal strength at the receiver.

The results of the second set of experiments can be seen in Figure 7. In all 10 experiments the robot successfully found the PANDA within 2.5 to 8 min and proceeded to download the 1.2 MB file within an additional 10 to 35 s.

Figure 8 shows the results of the third set of experiments in which a particle filter was used. In all 5 experiments the robot successfully found the PANDA within 4.2 to 9 min. This is a significant improvement over stochastic gradient descent when considering that the robot was coming from a distance 4 times larger than in the second set of experiments.

## 10 Main Experimental Insights

The proposed data muling system using bi-modal acousto-optical communication allows for large scale data recovery and eliminates the need for precise localization of the node and robot. It allows quick in-situ deployment of nodes and successive

autonomous data recovery. In all gradient descent experiments the robot successfully found the underwater sensor node within a few minutes using acoustic gradient descent and proceeded to download a 1.2 MB file within 10 to 35 s using the optical link. Further we demonstrated that we can use the optical signal strength to maintain the robot's position close to the position of the sensor node. We also experimentally evaluated the use of a particle filter to locate the node using only acoustic ranging. Both in simulation and experimentally the particle filter performed better than stochastic gradient descent.

If the PANDA had been able to generate LT-Codes at the full rate of 4 MBit/sec then our throughput would have been 3.2 times higher than measured. It should be noted that this was purely a limitation on the computational side and not a limitation of the optical or acoustic modem itself. Also, since we do not use error coding and correction, all packets with a single bit error were discarded. This amounted to 13.6 MB of 37.5 MB and 13.5 MB of 69.1 MB in payload data lost. With the expense of more computational resources this bandwidth can be almost entirely utilized.

Future improvements of the system include the usage of the acoustic link to turn on and off both the optical modem and the acoustic beacons (or at least reducing their frequency) to save battery life while the robot is not in range. We also plan to extend the presented data muling system to three dimensions, which will allow for the nodes to be deployed at greater depths. The particle filter algorithm can be extended to utilize optical signal strength measurements in order to improve the robot's position keeping above the node. Further, we plan to extend experiments into the open ocean where the algorithm can be tested at distances of multiple kilometers.

**Acknowledgements.** Support for this research has been provided in part by the MURI Antidote grant N00014-09-1-1031 and NSF grants IIS-1133224 and IIS-1117178. We are grateful for this support. We would like to thank Mohan Panayamadam, Andy Marchese, TeongBeng Koay, and Puthenpurayil Unnikrishnan Saneesh for their assistance.

## References

1. Benson, C., Dunbar, R., Ryan, M., Huntington, E., Frater, M.: Towards a dense high-capacity underwater acoustic network. In: 2010 IEEE International Conference on Communication Systems (ICCS), pp. 386–389 (2010), doi:10.1109/ICCS.2010.5686512
2. Chitre, M., Topor, I., Koay, T.B.: The unet-2 modem - an extensible tool for underwater networking research. In: OCEANS 2012, Yeosu, pp. 1–9 (2012)
3. Doniec, M., Rus, D.: Bidirectional optical communication with Aquaoptical II. In: 2010 IEEE International Conference on Communication Systems (ICCS), pp. 390–394 (2010), doi:10.1109/ICCS.2010.5686513
4. Doniec, M., Vasilescu, I., Detweiler, C., Rus, D.: Complete se(3) underwater robot control with arbitrary thruster configurations. In: Proc. of the International Conference on Robotics and Automation, Anchorage, Alaska (2010)
5. Dunbabin, M., Corke, P., Vasilescu, I., Rus, D.: Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. In: Proc. IEEE ICRA 2006, Orlando, Florida, pp. 2091–2098 (2006)

6. Kilfoyle, D., Baggeroer, A.: The state of the art in underwater acoustic telemetry. *IEEE Journal of Oceanic Engineering* 25(1), 4–27 (2000), doi:10.1109/48.820733
7. Li, K., Shen, C.C., Chen, G.: Energy-constrained bi-objective data muling in underwater wireless sensor networks. In: 2010 IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS), pp. 332–341 (2010), doi:10.1109/MASS.2010.5664026
8. Luby, M.: Lt codes. In: Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 271–280 (2002), doi:10.1109/SFCS.2002.1181950
9. Olson, E., Leonard, J., Teller, S.: Robust range-only beacon localization. In: Proceedings of Autonomous Underwater Vehicles 2004 (2004)
10. Shah, R., Roy, S., Jain, S., Brunette, W.: Data mules: modeling a three-tier architecture for sparse sensor networks. In: Proceedings of the First IEEE 2003 IEEE International Workshop on Sensor Network Protocols and Applications, pp. 30–41 (2003), doi:10.1109/SNPA.2003.1203354
11. Stojanovic, M.: Recent advances in high-speed underwater acoustic communications. *IEEE Journal of Oceanic Engineering* 21(2), 125–136 (1996), doi:10.1109/48.486787
12. Thrun, S.: Probabilistic robotics. *Commun. ACM* 45(3), 52–57 (2002) doi:10.1145/504729.504754, <http://doi.acm.org/10.1145/504729.504754>
13. Vaganay, J., Baccou, P., Jouvencel, B.: Homing by acoustic ranging to a single beacon. In: *OCEANS 2000*, vol. 2, pp. 1457–1462 (2000)

# Local-Search Strategy for Active Localization of Multiple Invasive Fish

Joshua Vander Hook, Pratap Tokekar, Elliot Branson, Przemyslaw G. Bajer, Peter W. Sorensen, and Volkan Isler

**Abstract.** In this paper, we study a problem encountered during our ongoing efforts to locate radio-tagged fish aggregations with robots. The problem lies at the intersection of search-based methods whose objective is to detect a target, and active target localization methods whose objective is to precisely localize a target given its initial estimate. Real-world sensing constraints such as limited and unknown range, large measurement time, and ambiguity in bearing measurements make it imperative to have an intermediate initialization phase to transition from search to localization. We present a local search strategy aimed at reliably initializing an estimate for a single target based on observations from field experiments. We then extend this strategy to initialize multiple targets, exploiting the proximity of nearby aggregated tagged fish to decrease the cost of initialization per target. We evaluate the performance of our algorithm through simulations and demonstrate its utility through a field experiment where the robot successfully detects, initializes and then localizes nearby targets.

## 1 Introduction

We are developing a robotic system (Figure 1) and algorithms [10, 11] to enable a mobile sensor network to monitor the common carp (*Cyprinus carpio*), an invasive fish. The common carp is an ecologically damaging freshwater fish found in many regions around the world [14]. Biologists are interested in developing efficient methods for controlling carp populations. To this end, they catch a small sample of the population and implant each fish with radio transmitters (tags). These tagged fish are reintroduced to the lakes and periodically tracked using radio receivers over

---

Joshua Vander Hook · Pratap Tokekar · Elliot Branson · Volkan Isler  
Department of Computer Science, University of Minnesota, USA  
e-mail: {jvander, tokekar, ebranson, isler}@cs.umn.edu

Przemyslaw Bajer · Peter Sorensen  
Department of Fisheries, Wildlife and Conservation Biology, University of Minnesota, USA  
e-mail: {bajer003, soren003}@umn.edu



**Fig. 1** Our robotic system consists of radio tags, a radio antenna and receiver mounted on autonomous boat in summer and wheeled rover in winter (to operate on frozen lakes)

the course of a year. When multiple tagged fish seem to aggregate, it is assumed a larger population is nearby. When these large aggregations of carp are found, typically during the winter, they can be removed by netting. This provides a safe and environmentally-friendly method for controlling the population of carp.

The radio tags (Figure 1) are small, low duty-cycle transmitters which are implanted under the skin of the fish. Each tag emits a pulsed signal on a dedicated frequency approximately once per second. A human operator carries a loop antenna and a receiver which converts the signal to a received signal strength indicator (RSSI). By monitoring the RSSI and rotating a directionally sensitive antenna, the operator can discern a bearing to the radio tag. Typically a human operator will take 2-3 bearing measurements to estimate the location of one tag. However, this manual tracking approach is tedious, time consuming and possibly inaccurate at times. Therefore, we believe that this problem is a good application for robotics.

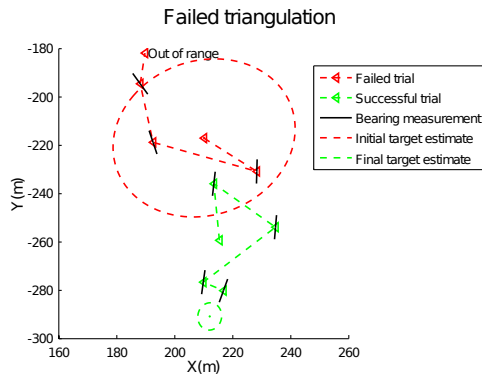
Our overall objective in this application is as follows: Given a list of  $N$  frequencies (one per tagged fish), each of which can be detected by the robot at a unique range  $r_i$ , localize each target to a desired accuracy in bounded time. In Section 3.1, we discuss our previous work where we partition this problem in two separate phases: (i) *Search* phase where the objective is to find a location for the robot within the sensing range of each target, and (ii) *Localization* phase where the robot uses bearing measurements to reduce the uncertainty in the target's estimate.

During field tests of this system, we found that the localization routine was sensitive to the accuracy of the initial estimate. Constructing a consistent, reasonably certain prior estimate in limited time has proven to be a difficult task. The problem becomes further challenging because the sensing ranges of individual tags can vary based on the depth of the fish, the age of the tag, and other environmental factors. For example, Figure 2 shows a field trial where the robot could not complete the triangulation due to an incorrect initialization. The target was initialized with a 2D Gaussian distribution centered at the location where the robot first moved into the sensing range of the tag, with a variance based on empirical estimates of the sensing range. However, the variance was set too low and as such the initial estimate was not consistent. During triangulation, the robot moved to a location which fell outside the sensing range of the target, and the final estimate was wrong. The robot successfully triangulated the same tag in another run where the initial estimate (not shown for

clarity) was consistent. This indicates the importance of starting with a good initial estimate. Therefore, we present a local search strategy which, after detecting a target during the search phase,

1. Initializes a consistent estimate of the target location,
2. Maps a region from which bearing measurements are likely to succeed,
3. Exploits clustering behavior of the fish to locate nearby targets efficiently.

After presenting the details of the search strategy and its analysis in Section 4, we evaluate the strategy through simulations (Section 5), and present results from a field experiment (Section 6). The field trial demonstrates that our proposed initialization strategy is effective, and promising for large-scale future experiments. We believe our proposed approach of search, initialization, and localization should be applicable for other applications where one or more robots are tasked with accurately locating one or more targets in bounded time.



**Fig. 2** Failed triangulation due to incorrect initialization for trials conducted on Lake Staring, MN. The initial estimate for the first trial was inconsistent and resulted in the localization to diverge and move the robot out of the tag's sensing range. During a second trial, with a consistent initial estimate, the target was successfully localized.

## 2 Previous Work

Recently, there has been significant interest in developing algorithms for locating transmitting radio sources using mobile robots. Song et al. [8] considered the problem of localizing an unknown number of transient radio sources using a mobile robot. They used an occupancy grid in a Bayesian framework to update the probability of a radio source being located in a given grid cell. They further proposed a path-planning algorithm for the robot to improve the convergence time for locating all sources. In [6], Kim et al. presented a centralized multi-robot search algorithm for the same problem setting, where the robots are controlled in pairs to allow detection of unknown transmission powers from the radio sources.

In [9], Tekdas et al. consider the problem of finding a point of high signal strength inside the sensing disc of transmitting sources. They assume a prior estimate of the source's location is given but sensing range is unknown. Here, we consider the problem of finding a good point to begin triangulation, while estimating sensing range and target location simultaneously.

Fink and Kumar [3] presented methods to build a radio signal strength map in an unknown indoor environment and presented control laws for mobile robots to seek the transmitting radio source. Recently, Twigg et al. [12] addressed the problem of exploration while seeking a radio source. The algorithm builds a gradient of the RSSI by collecting samples locally. Their work involves indoor environments and areas with significant multi-path effects, and so is not directly applicable to our work. In addition, the directional sensitivity of our antenna makes it difficult to determine and follow a gradient.

The problem of simultaneously localizing a robot and multiple transmitting sources was considered in [4]. It was assumed that range could be explicitly recovered from the transmissions, and an arbitrary robot path was reconstructed while simultaneously estimating the position of each radio. An iterative, offline algorithm was proposed and evaluated. This problem is fundamentally different because we cannot recover range directly, and must solve the problem online, i.e., as measurements become available. Furthermore, we have direct control over the robot's path. In fact, defining the robot's path to aid the estimation problem is the what we address in the following sections.

### 3 Motivation

In this section, we present the details of our system and then discuss some intuitive methods for addressing the problem under consideration.

#### 3.1 System

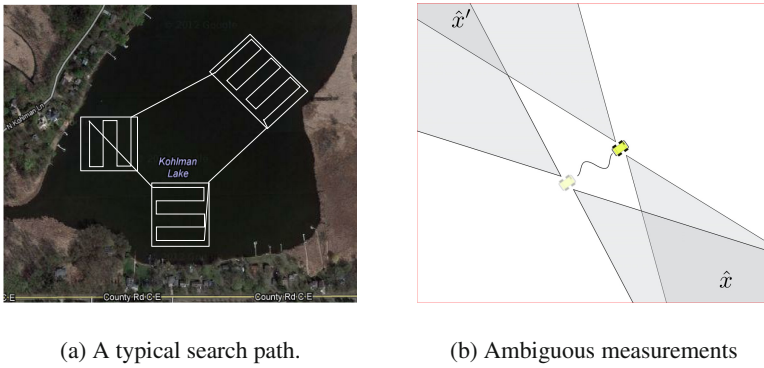
Our system consists of a wheeled rover to traverse frozen lakes and locate fish aggregations. We deploy a similar system during the summer using robotic boats. Our mobile rover is the A100 Husky by Clearpath Robotics, and our robotic boat is an OceanScience QBoat. Both chassis are fitted with a loop antenna, a servo motor to rotate the antenna, a radio receiver, and a laptop computer. The robots estimate their own pose and navigate using an Extended Kalman Filter (EKF) combining information from a Global Positioning System unit and a digital compass (on the boat) or encoders (on the wheeled rover).

The radio tags are shown in Figure 1 (Left). Each tag is programmed with a specific frequency in the 48-50MHz range and emits 30-50 signals per minute. We operate on lakes which have 10-20 tagged fish and the list of frequencies present in each lake is known a priori. Signals from the tags attenuate as a function of the humidity, salinity of the water, ice or snow thickness, and the depth of the fish it is attached to. These factors cause variations in the range at which tags can be detected.



Therefore we do not use the absolute signal strength to estimate range, and instead use the directional nature of the antenna to estimate bearing.

The radio antenna and receiver are pictured in Figure 1 (middle and right), atop both robots. The sensitivity of the antenna varies with the relative angle with the tag. We rotate the antenna using a servo motor in 15 degree steps over 180 degrees. We sample the signal strength at each step and fit a smooth function to the data to estimate the direction with maximum RSSI. This direction is treated as the bearing towards the target. Because of the low signal rate, obtaining a bearing measurement takes about 1-2 minutes. Empirically we have found the bearing measurements to follow an approximately Gaussian distribution around the true target bearing ( $\sigma \approx 15^\circ$ ). However, bearing measurements constructed in this way are ambiguous, or  $\pi$ -periodic. For any estimated bearing  $z$ ,  $z + \pi$  is also a valid bearing measurement (see Figure 3(b)).



**Fig. 3** Examples of search patterns (Figure 3(a)) and ambiguous bearing measurements (Figure 3(b)). High sensor noise, ambiguity and unknown sensing range makes it difficult to transition from search to localization.

In our previous trials we observed that the tags’ radio signal is undetectable unless we are within 100-200 meters. This provides a natural task partitioning: *Search* and *Localization* [11]. The goal of the search phase is to cover the regions of the lake that are likely to contain tagged fish and move the robot to within sensing range of each tag. We then switch to *Localization* where the goal is to obtain multiple bearing measurements to localize the tag to a desired precision. Once a target is localized, the robot can resume its search for other tags. During the search phase, we simply wait for a detection of a non-zero RSSI value, which takes significantly less time than obtaining a full bearing measurement.

Our current localization algorithm uses an EKF to estimate the position of the tag [13]. The localization subroutine takes time proportional to the area of initial uncertainty and the distance between the initial estimate and the robot. In simulation and experiments this method performs well, but only if the initial estimate of the

target is consistent and not significantly uncertain. Obtaining an initial estimate of the target location with bounded uncertainty is challenging, as we discuss next.

### 3.2 *The Initialization Problem*

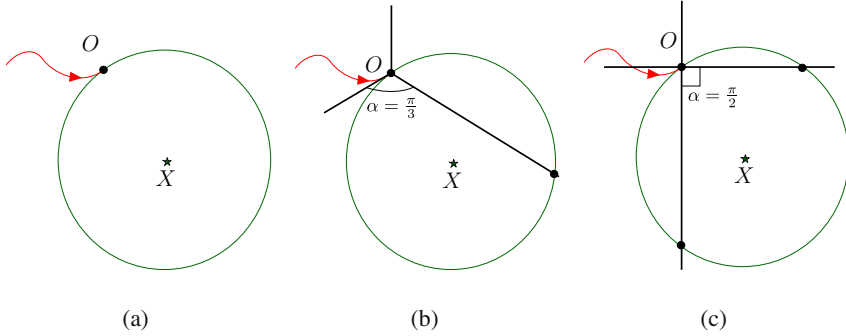
Before the localization algorithm can be deployed to precisely estimate tag locations, we must initialize a prior estimate as input. We briefly present some intuitive methods we have tried and discuss why they fail.

**Measurement-Based.** As often recommended in bearing-only tracking literature, a small number of bearing measurements can be collected and processed in a batch. Given a set of  $k$  measurements  $\mathbf{Z} = \{z_1, \dots, z_k\}$ , we maximize the likelihood,  $p(\mathbf{Z}|x)$  over target locations  $x$ . In practice, limited sensing range and long measurement time make this strategy infeasible. Also, consider Figure 3(b). The two dark regions show areas which are likely to contain the true target and we cannot easily determine which hypothesis is the origin of the measurements ( $\hat{x}$  or  $\hat{x}'$ ). A third measurement, taken from a large baseline could disambiguate the two. However, a large baseline is likely to move the robot outside the sensing range of the target, producing no information while paying the full cost of a bearing measurement. Another solution could be to take a fixed number of measurements around the initial detection point. Again, the long bearing measurement time makes this an expensive strategy which must be repeated for each nearby tag. Further, it is not clear how these additional measurement locations should be chosen to guarantee a good estimate of the target.

**Initial Hypothesis.** In contrast to the above, we can initialize a hypothesis by taking two measurements as shown in Figure 3(b). By drawing a wedge surrounding each measurement to represent its uncertainty, we can obtain an intersection representing the target hypothesis. We can fit a Gaussian distribution to this intersection area and use as an initial estimate. This is not robust in practice, since the intersection can be unbounded. Additionally, we have two intersection areas leading to two initial hypothesis. As such, this method provides no guarantees about initial estimate uncertainty or range.

**Signal-Strength Based.** We can attempt to use the signal strength to resolve the ambiguity of each measurement. The robot could travel toward one hypothesis and measure the signal strength. We expect the signal strength to increase if the robot travels towards the correct hypothesis. In practice, we found this strategy to be sensitive to sensor noise from the unknown and possibly complex spatial signal strength patterns. We found that for small movements near the edge of the sensing range this method was unreliable.

Each of these initialization methods fails to provide a guarantee of time cost, uncertainty, or consistency of the estimate. In the next section, we describe our solution to this problem which relies on a local search strategy.



**Fig. 4** (a) While on its coverage path (curved arrow), the robot, at  $O$ , detects a non-zero signal for some frequency  $X$ . (b) The initialization strategy determines the sensing circle for  $X$  by moving along search paths as shown until  $X$  is not detectable again. Shown is a case where three search paths fail to uniquely identify the sensing circle. (c) An example of a Four-path search.

### 4 Local Search

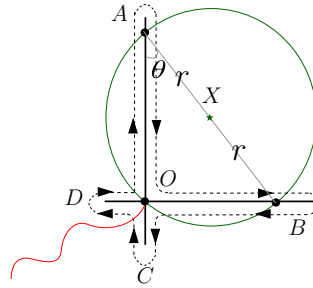
The goal of the local search is two-fold: (1) determine whether an aggregation exists nearby and which targets are contained within the aggregation, and (2) form good initial estimates (mean and covariance) for each target in the aggregation. The initialization phase begins as soon as the robot first detects a non-zero RSSI from a radio tag while on the search path (Figure 4(a)). We assume that the detected tag  $X$  is at the center of a sensing circle  $C_X$  of radius  $r$ . Our objective is to establish an initial estimate of  $X$  and  $r$ . In this section, we first present our local search initialization strategy for a single target (i.e.  $X$ ). We bound the worst-case and average-case time required for this strategy. We then extend this strategy for the case of an aggregation of multiple tagged fish.

#### 4.1 Single-Target Local Search

Note that both  $X$  (the origin of  $C_X$ ) and  $r$  are unknown. By finding three points on the perimeter of  $C_X$  we can solve for  $X$  and  $r$ . To find these points, the local search proceeds as follows:

1. From the point of first detection ( $O$ ), the robot moves in a fixed direction with respect to the global frame (e.g., North or angle  $\alpha$ ).
2. When the robot can no longer detect the target  $X$  (position  $A$  in Figure 5) it reverses direction and returns to  $O$ .

The line segment traversed in these two steps is called as a *search path*. To analyze the time cost of this strategy, we establish the minimum number of search paths needed to find at least three points on the boundary of  $C_X$ . We can see that at least four equally spaced search paths are necessary and sufficient from Figure 4(b).



**Fig. 5** The robot continues along an arbitrary but fixed direction until it cannot detect the signal from  $X$  (at position  $A$ ). The robot then returns to  $O$  and repeats the same strategy along a perpendicular line ( $B$ ). In general, the  $O$  can lie in the interior of the sensing circle, hence the robot also searches along  $C$  and  $D$ .

We now establish the cost of using four search paths to find  $X$  and  $r$ . The analysis follows Figure 5. Let angle  $OAX$  be  $\theta$ . By design, the angle  $AOB$  is  $\frac{\pi}{2}$ . The distance  $|AB|$  is  $2r$  and segment  $OA$  has length  $2r \cos \theta$  while  $OB$  has length  $2r \sin \theta$ . Assume the robot moves with velocity  $v$ . Each of these lines must be traversed twice, for a total required time of,

$$T_{\text{single}} = \frac{4r}{v} \cos \theta + \frac{4r}{v} \sin \theta + 4 \cdot \varepsilon \tag{1}$$

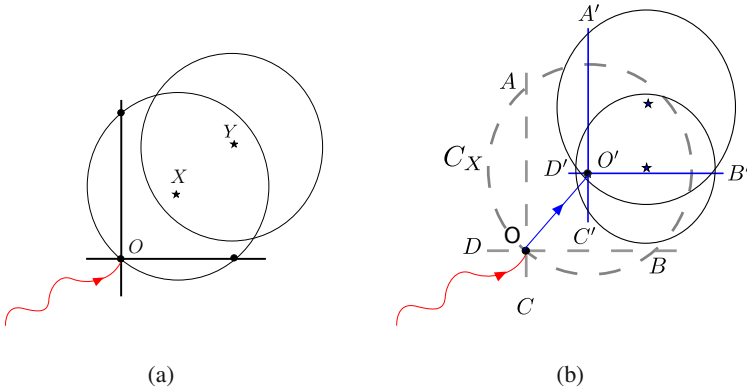
where  $\varepsilon$  is the time taken to recognize the robot has left  $C_X$ , turn around, and re-enter  $C_X$ . Note that  $\theta$  is unknown and can take any value between  $0$  and  $2\pi$ , depending on the relative orientation of the target position with respect to the first search direction. To obtain the worst-case cost, we maximize the cost function with respect to  $\theta$ . A straightforward derivation shows the cost is maximum when  $\theta = 45$  degrees for a maximum cost of,

$$\max_{\theta} T_{\text{single}} = 2^{\frac{5}{2}} \frac{r}{v} + 4 \cdot \varepsilon \tag{2}$$

The expected search time, assuming  $\theta$  is uniform in the range  $[0, 2\pi]$  is  $\mathbf{E}[T_{\text{single}}] = 2\frac{r}{v} + 4\varepsilon$ .

### 4.2 Multi-target Local Search

To extend the local search strategy to multiple targets, we need a model for fish aggregations. While common carp are relatively broadly dispersed during summers, they tend to form tight aggregations under ice-covered lakes in winters [1, 2, 5, 7]. For example, while average distances between radio-tagged carp during summers are 300-500 meters, in winters, these distances decrease to 50-100 meters [1]. In some cases, entire populations of carp, usually several thousands of fish, have been



**Fig. 6** (a) The single target search fails to intersect all the sensing circles in the case of an aggregation. (b) An example of searching for an aggregation using two separate search steps. The robot first finds the boundary of  $C_X$  (dashed), centered at  $O'$ . Then, after moving to  $O'$ , searches along four paths to identify the boundaries of each sensing circle.

shown to aggregate in areas that are only  $100 \times 100$  meters in size [1]. We formalize the notion of an aggregation using the following definition.

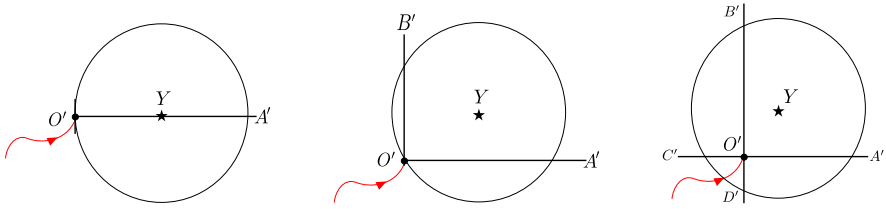
**Definition 1.** Let  $\mathcal{L} = \{X_1, \dots, X_i, \dots, X_N\}$  be a set of tagged fish,  $r_i$  be the sensing radius of  $X_i$ , and  $r^* = \min_i r_i$ .  $\mathcal{L}$  is called an aggregation if,  $\forall i, j, \|X_i - X_j\|_2 < r^*$

Under this definition, we cannot directly use the local search strategy for a single target for multiple targets. Figure 6(a) illustrates an example case where the four search paths do not intersect the sensing circle of  $Y$  present in the aggregation.

We propose the following strategy: By Definition 1, for any target  $x$ , the distance to all other targets to  $x$  is less than  $r^*$ . Returning to the case of one target shown in Figure 5, we see that four search paths can provide an estimate of a target location as the center of the estimated sensing disk. In general, since we don't know which fish are contained in the aggregation, it might be necessary to search for all frequencies. As a practical step, we make the assumption that the true location of the first fish  $X$  is close to the center  $O'$  of the estimated sensing circle. This allows us to move to  $O'$  and determine which fish are nearby. We can then perform another multi-path search to map the boundaries of all nearby frequencies (see Figure 6(b)). We call the resulting algorithm *Four-Path*.

Assuming we begin a search from the target location  $X$ , we can show that four paths are sufficient to detect the boundaries of each sensing circle in the aggregation. Consider Figure 7, which illustrates the possible configurations of the rest of the targets with respect to the first. We have three cases:

- The target  $Y$  is aligned with the search path starting at  $O'$ , and we detect two points of  $C_Y$ . This case has a unique solution:  $Y$  is at  $\frac{1}{2}|XA'|$  along  $XA'$ .
- $O'$  is on the boundary of  $C_Y$ . In this case we detect three points  $O', A',$  and  $B'$ . We can solve  $C_Y$  directly.



**Fig. 7** In general, the starting location of local search can lie anywhere on the boundary or interior of the sensing circle. In each case, we obtain a different number of points as shown. For all cases, we can determine the sensing circle uniquely.

- $O'$  is inside the circle  $C_Y$ . We can detect four points at  $A', \dots, D'$ , and solve the sensing circle  $C_Y$  using least-squares fitting.

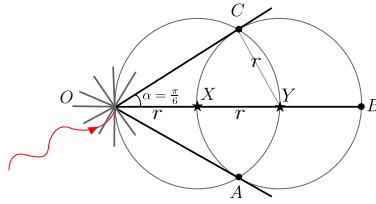
Each search path begins at  $O' \approx X$ . The robot moves until it cannot detect any nearby tags. By Definition 1, this can be a maximum of  $2r$  in any direction (traveled twice) for a total cost of  $16\frac{r}{v} + 4 \cdot \epsilon$ . A total of five targets are required to achieve the worst-case cost. Adding this to the worst-case cost of the initial search, plus the maximum displacement between the points  $O$  and  $O'$  gives,

$$T_{\text{multi}} = 17\frac{r}{v} + 2\frac{5}{2}\frac{r}{v} + 4 \cdot \epsilon. \tag{3}$$

### 4.3 Discussion

The cost shown by Equation (3) may seem large. For example, given our system,  $v$  is approximately 2 meters per second and, for comparison, assume  $r$  is near 100 meters. Thus the total cost is approximately 19 minutes for the worst-case 5 targets. While we are not concerned with the aggregation displacing in this time, this may cause unnecessary drain on the limited operational life of the robot. To put this in context, compare this to the cost of taking two bearing measurements to initialize each target individually. Recall from Section 3.1 that a bearing measurement takes approximately 1-2 minutes. At least two measurements are required, resulting in 10-20 minutes for 5 targets, not counting the time to displace between measurement locations. By amortizing the cost of a local search on a per-target basis, it is clear the search-based strategy will incur a lower cost to initialize larger aggregations.

To see the relative advantage of a two-phase search, consider the work required by a single-phase search. That is, upon detecting a non-zero signal strength, we could search along  $K > 4$  search paths and attempt to intersect each nearby sensing circle. The necessary number of search paths can be found as follows. Refer to Figure 8. In this example two targets,  $X$  and  $Y$  are arranged along the  $x$  axis with respect to the starting location  $O$ . Assume the first search path moves along the  $x$  axis and the next search path is offset by an angle  $\alpha$ . Then, to intersect the circle  $C_Y$  we require  $2r \sin \alpha = r$ . Solving, we get  $\alpha = 30$  degrees, i.e.,  $K \geq 12$  search paths over 360 degrees. We call the resulting algorithm *Twelve-Path*. Note, unlike



**Fig. 8** To extend the single target local search strategy, we need at least twelve search paths (separated by less than  $\frac{\pi}{6}$ ) to intersect each sensing circle at least thrice

the Four-Path strategy, we must sample the entire list of frequencies in the lake over each of the twelve paths because we do not know until we are finished which tags belong to the aggregation. Hence the time taken to sample a frequency, and the total number of targets in the lake affect the cost of this strategy.

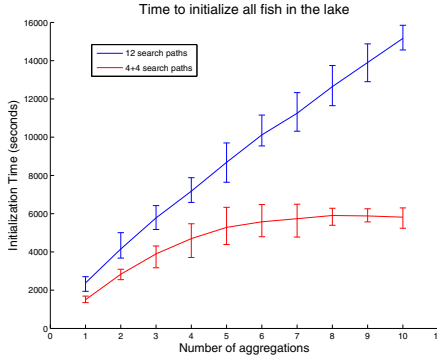
Because the distribution of the targets both in and between aggregations plays a large role in the *expected* search time, we compare these strategies in simulations.

### 5 Simulations

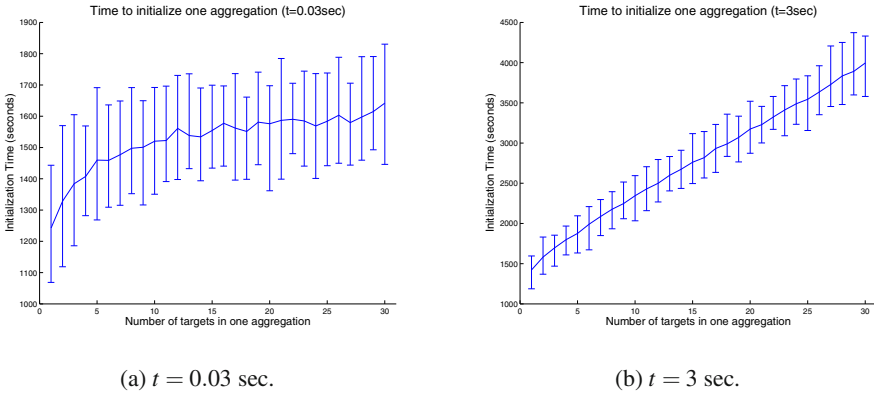
In the analysis presented in the previous section we assumed the time required to sample a frequency ( $t$ ) was negligible. In practice, we may periodically stop the robot while sampling the frequencies to avoid radio interference from the electric drives, which takes some time. Second, we assumed the same sensing range  $r$  for all tags, when in practice it can be different for each tag. Finally, we evaluated the cost to initialize the targets in a single aggregation. In general, there can be more than one aggregation in the lake, each possibly containing different numbers of tagged fish. In this section, we investigate the role of the time spent in sampling the frequencies, the effect of multiple aggregations on total cost, and the effect of different sensing ranges on the time to initialize all targets.

We conducted simulations as follows. To evaluate a varying sensing range,  $r$  is drawn uniformly at random between  $[50, 100]$ m for each tag. We vary the number of aggregations from 1 – 10 (with at least one fish each). The remaining fish are assigned randomly. The direction in which the robot enters the detection disk of the first target for each aggregation is also drawn uniformly at random between 0 and  $2\pi$  radians. The velocity of the robot is given as  $v$  and is assumed fixed.

We compare Twelve-Path and Four-Path strategies presented in the previous section. Recall that the Twelve-Path (Figure 8) strategy moves along twelve search paths from the point of first detection, while sampling on the entire list of frequencies present in the lake. The Four-Path strategy (see Figure 6(b)) estimates the sensing circle for first tag detected, moves to the center of this estimated circle, samples all frequencies once to detect the list of frequencies present in the aggregation, and then moves along four search paths to estimate the sensing disks for only the subset of tags detected in the aggregation. Both produce an estimate of the sensing range and position of each nearby tag.



**Fig. 9** Simulation comparing the time taken to initialize all 10 fish in the lake, as the number of aggregations varies. The Four-Path strategy performs better than the Twelve-Path. The bars indicate the minimum and maximum times, and the trend line plots the mean time of 50 instances.



**Fig. 10** Mean, minimum and maximum time taken as the number of fish increases in one aggregation for the Four-Path strategy. For lower sampling time, the time to travel dominates and thus scales well for larger aggregations.

In Figure 9, we compare the mean, min and max time taken for executing both strategies for 50 iterations, as a function of the aggregation size  $M$  with total number of fish,  $N = 10$ . The sampling time per frequency is  $t = 0.03$  sec (we obtain similar results for other choices of sampling time). We observe that the Four-Path strategy takes less time, as compared to the Twelve-Path strategy.

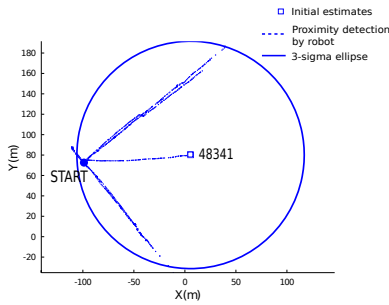
Figure 10 shows the time taken by the Four-Path strategy when size of one aggregation is increased (as opposed to the number of aggregations in Figure 9). For lower sampling time, we observe that the time to travel over the search paths dominates the time to sample for various frequencies. Since the distance traveled by the



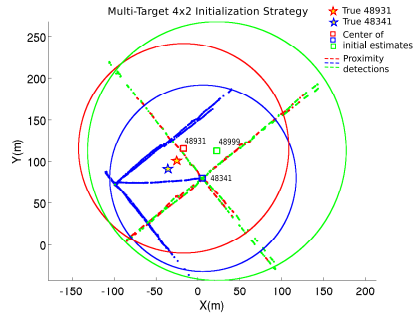
robot doesn't change significantly with increasing number of fish in the aggregation (by Definition 1), we see that the time taken scales well.

## 6 Experiments

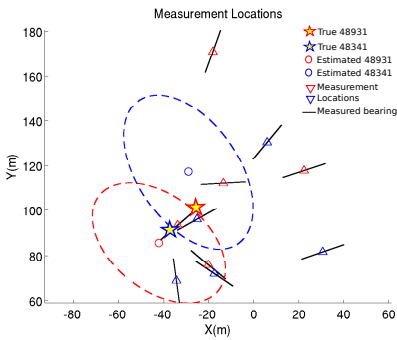
We implemented our initialization strategy on the mobile chassis shown in Figure 1. Three tags were deployed on Lake Gervais, MN, and their true locations were recorded for comparison (see Figure 11). The robot first detected the tag with frequency 48341 at the location marked START in Figure 11(a). The robot then executed the Four-Path strategy. After completing the first phase of the Four-Path strategy, we fit a circle to the points where we stopped detecting the signal for 48341 as shown. This circle was used as the  $3\text{-}\sigma$  uncertainty ellipse of a 2D Gaussian



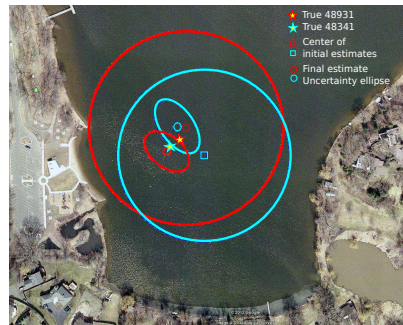
(a) First Local Search (Section 4.1)



(b) Aggregation Search (Section 4.2)



(c) Localization Output (see [13])



(d) Final Estimates vs True

**Fig. 11** A successful experiment demonstrating the local search strategy and localization steps

distribution with the center of the circle used as the mean for initializing the estimate for this tag. The robot then traveled to the center of this circle and sampled the list of frequencies to detect nearby tags. The robot detected signals for frequencies 48931 and 48999 (48999 was due to radio interference and not an actual tag—the Localization strategy received no valid measurements and discarded this estimate).

The robot then executed the second phase of the Four-Path strategy, where it searched for frequencies detected at the center of the initial circle as shown in Figure 11(b). The corresponding hypothesis for all tags are shown relative to the true locations. Using this initial hypothesis, the robot then executed the active localization algorithm described in [13]. Figure 11(c) shows the execution of this localization algorithm, the measurement locations selected for each tag (triangles), and the bearing measured (black lines).

The final estimates for the two actual tags in the aggregation after five measurements (48341 and 48931) are shown using the  $3\text{-}\sigma$  uncertainty ellipse. Figure 11(d) shows the GPS location of the tags along with the initial and final estimates. The final covariance for 48341 had eigenvalues  $56\text{m}^2$  and  $168\text{m}^2$  (corresponding to an error ellipse with radii 7m and 12m), starting from an initial covariance with eigenvalues  $1380\text{m}^2$ . The final covariance for 48931 had eigenvalues  $49\text{m}^2$  and  $127\text{m}^2$  (radii 7m and 11m), starting from an initial covariance with eigenvalues  $1758\text{m}^2$ . The final error for 48341 and 48931 were 27m and 23m respectively.

## 7 Conclusions

We are working toward the goal of localizing multiple targets in a known environment in bounded time. The complicated interplay of target distribution, sensing range, measurement noise, and ambiguous measurement model makes each phase independently interesting. Here we presented a strategy to initialize consistent hypotheses for multiple targets in an aggregation. In our future work, we plan to extend the our system to multiple robots and incorporate fish mobility models. To extend this algorithm to multiple robots, we must account for communication constraints between the robots and develop allocation algorithms which guarantee the work is distributed evenly. For mobile targets, we must both develop motion models for fish and develop new search and localization algorithms based on these models. One possible approach is to model the fish as adversarial—part of our ongoing work.

**Acknowledgements.** The authors would like to thank Brett Miller and Mary Haedrick for their involvement in gathering telemetry data. This work is supported in part by NSF Awards #1111638, #0916209, #0917676 and #0936710.

## References

1. Bajer, P.G., Chizinski, C.J., Sorensen, P.W.: Using the Judas technique to locate and remove wintertime aggregations of invasive common carp. *Fisheries Management and Ecology* 18(6), 497–505 (2011)

2. Bajer, P.G., Sorensen, P.W.: Recruitment and abundance of an invasive fish, the common carp, is driven by its propensity to invade and reproduce in basins that experience wintertime hypoxia in interconnected lakes. *Biological Invasions* 12(5), 1101–1112 (2010)
3. Fink, J., Kumar, V.: Online methods for radio signal mapping with mobile robots. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 1940–1945 (2010)
4. Hollinger, G., Djughash, J.: Target tracking without line of sight using range from radio. *Autonomous Robots*, 1–14 (July 2011)
5. Johnsen, P., Hasler, A.: Winter aggregations of carp (*Cyprinus carpio*) as revealed by ultrasonic tracking. *Transactions of the American Fisheries Society* 106(6), 556–559 (1977)
6. Kim, C., Song, D., Xu, Y., Yi, J.: Localization of multiple unknown transient radio sources using multiple paired mobile robots with limited sensing ranges. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 5167–5172 (2011)
7. Penne, C., Pierce, C.: Seasonal Distribution, Aggregation, and Habitat Selection of Common Carp in Clear Lake, Iowa. *Transactions of the American Fisheries Society* 137(4), 1050–1062 (2008)
8. Song, D., Kim, C.-Y., Yi, J.: Simultaneous localization of multiple unknown and transient radio sources using a mobile robot. *IEEE Transactions on Robotics* 28(3), 668–680 (2012)
9. Tekdas, O., Karnad, N., Isler, V.: Efficient strategies for collecting data from wireless sensor network nodes using mobile robots. In: *14th International Symposium on Robotics Research, ISRR* (2009)
10. Tokekar, P., Bhadauria, D., Studenski, A., Isler, V.: A Robotic System for Monitoring Carp in Minnesota Lakes. *Journal of Field Robotics* 27(6), 779–789 (2010)
11. Tokekar, P., Branson, E., Vander Hook, J., Isler, V.: Coverage and active localization for monitoring invasive fish with an autonomous boat. *IEEE Robotics and Automation Magazine* (2012) (to appear)
12. Twigg, J., Fink, J., Paul, L., Sadler, B.: RSS Gradient-Assisted Frontier Exploration and Radio Source Localization. In: *Proc. IEEE International Conference on Robotics and Automation* (2012)
13. Vander Hook, J., Tokekar, P., Isler, V.: Cautious Greedy Strategy for Bearing-based Active Localization: Experiments and Theoretical Analysis. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 1787–1792 (2012)
14. Weber, M.J., Brown, M.L.: Effects of Common Carp on Aquatic Ecosystems 80 Years after Carp as a Dominant: Ecological Insights for Fisheries Management. *Reviews in Fisheries Science* 17(4), 524–537 (2009)

# **Part XIII: ISER Session Summary on “Sensing and Navigation”**

Liz Murphy

CyPhy Lab, Queensland University of Technology

## **Session Summary**

Recent years have seen the sensing and navigation field move to consolidate past accomplishments into robust and persistent navigation systems that are able to leverage continually evolving sensor technology. While interest now lies beyond sensing the world as obstacles or free space and instead centers on constructing continuous models of the environment that are robust to environmental changes, challenges such as automatic sensor calibration remain pervasive. The sensing and navigation session of ISER 2012 contained 4 papers spanning the full spectrum of the field; from sensor calibration through to navigation systems. The first speaker, Liz Murphy from QUT, presented a comparison of visual odometry techniques that experimentally evaluated the strengths and weaknesses of two different algorithmic approaches to odometry in indoor environments and demonstrated the possibility of achieving more robust VO through combining these approaches. The paper highlighted that issues such as poor illumination and restricted fields of view still pose problems for visual odometry systems. The second speaker, Steve Martin from QUT, outlined a method for constructing traversability maps for robotic vehicles online. In this work, traversability was measured onboard the vehicle using metrics such as power consumption, slip and vehicle orientation. Maps were constructed from sparse measurements using Gaussian Process techniques. Experiments were conducted to verify the usefulness of both the traversability metrics as predictors of terrain type and of the traversability maps' utility for planning. The third paper, presented by Jessie Levinson of Stanford, demonstrated the automatic calibration of coupled 2D camera and 3D laser systems in arbitrary scenes with no known features. The technique estimates the six dimensional transform between the two sensors – assumed to have overlapping fields of view and synchronized data capture – using the assumption that depth discontinuities in the laser scan should map to edges in the camera image. Vastly more accurate and faster calibrations than comparable manual techniques were reported. The final paper, presented by Jan Elseberg from Jacobs University in Bremen, discussed the simultaneous automatic calibration of all sensors – such as 3D laser, IMU and odometry, on a mobile scanning platform. This was achieved by optimizing a quality measure on the constructed point clouds, and was accomplished without the need for additional hardware or modifications to the environment. The system was validated on two different robotic platforms, in both cases finding more accurate calibration parameters than manual techniques.

# Experimental Comparison of Odometry Approaches

Liz Murphy, Timothy Morris, Ugo Fabrizi, Michael Warren, Michael Milford, Ben Upercroft, Michael Bosse, and Peter Corke

**Abstract.** Odometry is an important input to robot navigation systems, and we are interested in the performance of vision-only techniques. In this paper we experimentally evaluate and compare the performance of wheel odometry, monocular feature-based visual odometry, monocular patch-based visual odometry, and a technique that fuses wheel odometry and visual odometry, on a mobile robot operating in a typical indoor environment.

## 1 Motivation and Problem Statement



**Fig. 1** The Adept Guiabot robot used in this paper

Today most research robots, the prototypes of future commercial systems, use laser rangefinders (LRFs) as the sensor that informs the essential parts of a navigation system: odometry, place recognition and mapping. LRFs are self-contained, reliable and provide metric information about the world, but since they are based on mature electro/optical/mechanical technologies will always be more expensive than cameras. They also only capture just a slice through the scene whereas a camera can instantaneously capture a view of a whole area which is important for fast moving robots. Imaging 3D sensors such as flash LIDAR (eg. Swiss Ranger) or Kinect technology are entirely solid-state but function poorly in the presence of daylight.

---

Liz Murphy · Timothy Morris · Ugo Fabrizi · Michael Warren · Michael Milford · Ben Upercroft · Peter Corke  
CyPhy Lab, Queensland University of Technology  
e-mail: [firstname.lastname@qut.edu.au](mailto:firstname.lastname@qut.edu.au)

Michael Bosse  
Autonomous Systems Lab, CSIRO, 1 Technology Court, Pullenvale, QLD, 4069 Australia  
e-mail: [mike.bosse@csiro.au](mailto:mike.bosse@csiro.au)

Theoretically vision is able to perform the tasks of odometry and place recognition, and the sensors are in principle cheaper and smaller than LRFs. Vision has other advantages: it is a passive sensor and thus immune from the interference problems that will occur when multiple robots emitting infra-red light are operating in close proximity; it provides rich scene information such as color and texture which can assist semantic tagging; and when combined with motion it can generate 3D scene structure. However vision is rarely reported for indoor navigation and this paper details early results from a larger project aimed at developing a purely vision-based large-scale and long-term mobile robot navigation system that can work both indoors and outdoors.

This paper is concerned specifically with maximizing the accuracy of odometry; we view odometry as an essential part of any navigation system and recognize that external mechanisms to “close-the-loop” can improve odometry performance significantly, but the focus here is on maximizing odometry performance in isolation. We compare the performance of several odometry approaches: raw wheel odometry, patch based monocular visual odometry (VO), state-of-the-art feature-based monocular VO, a method of fusing wheel odometry and VO using confidence metrics generated from the VO, and laser scan matching which we refer to as ground truth. The performance of these odometry techniques is evaluated in two distinct indoor environments that present a number of different challenges.

## 2 Related Work

Visual data is rich in information, and in recent years visual navigation systems have been able to create accurate large-scale maps whilst maintaining precise position and orientation estimates for indoor and outdoor mobile robots [1, 2, 3]. While these use loop closure detection to correct drift in the visual odometry estimates, we focus on visual odometry alone in this paper.

There have been a number of different groups that have presented successful approaches for VO. Some initial efforts were focused on the use of single uncalibrated moving cameras [4]. Issues such as projective drift in addition to the very difficult task of practical auto-calibration, led subsequent authors towards a focus on calibrated methods for visual motion estimation [5, 6].

In [7] both single camera and stereo-rig approaches to the task of VO are presented. The conclusions of [7] indicate that the single camera method is less reliable due to inherent ambiguities in calculating the essential matrix between pairs of images to initialise the structure and motion [8]. Despite these conclusions, two successful monocular systems, MonoSLAM [9] and Parallel Tracking and Mapping (PTAM) [10], have been demonstrated since. Stereo VO indoors has recently been demonstrated to very good effect with minimal features [11, 12]. Although, stereo cameras constrain the solution, we analyse a monocular setup, and in addition, fuse wheel odometry to correct some of these problems.

The majority of visual SLAM and visual odometry techniques today rely on some form of feature detection, whether that be SIFT [13], SURF [14], or some other

feature detector. To obtain accurate visual odometry, features in an image must be detected reliably and accurately over a short period of time to generate feature tracks.

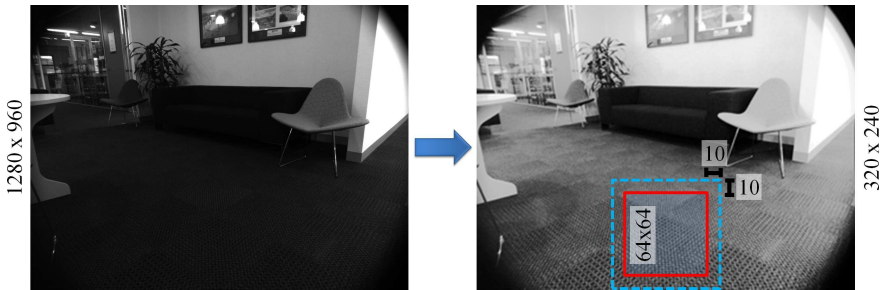
However, recent biological-based systems such as RatSLAM have shown good performance over a wide range of environments and robotic platforms using very low resolution, poor quality visual input. RatSLAM typically takes as input 1000 pixel images or simple image intensity profiles, and performs monocular patch-based VO using template matching, yet is able to successfully map changing indoor environments over long periods of time [15].

### 3 Technical Approach

In this section we describe the vision-based odometry algorithms that we are evaluating: two monocular VO systems (patch-based VO and VO using Structure from Motion) and an odometry fusion technique. For ground truth we use the well known laser-scan matching algorithm (GMapping [16]).

#### 3.1 Patch-Based Visual Odometry

The patch-based visual odometry system detailed in this paper is a modified version of the system deployed on a quad rotor in [15]. Compared to a quad rotor, using a ground-based robot means that camera-scene geometry can be assumed known and constant so the patch-based visual odometry system produces correctly scaled velocity information. The approach assumes a non-holonomic platform with a camera mounted at a constant height above the ground surface. Histogram equalization is applied to full resolution images before down sampling (in order to minimize loss of detail when reducing the image size) to a resolution of  $320 \times 240$  pixels (see Figure 2). Then the movement of a single  $64 \times 64$  image patch is tracked over consecutive



**Fig. 2** Patch-based visual odometry tracks the motion of a single small patch over consecutive histogram equalized, resolution reduced images. Horizontal motion is scaled to produced a measure of yaw, while vertical motion is scaled to produce a measure of translational speed.

frames, by calculating the Sum of Absolute Differences (SAD) measure between the current patch and the corresponding patch in the previous image, over a range of horizontal and vertical offsets:

$$f(\Delta x, \Delta y, I^j, I^k) = \frac{1}{r^2} \left| p_{x+\Delta x, y+\Delta y}^j - p_{xy}^k \right| \quad (1)$$

where  $I^j$  and  $I^k$  are the past and current patches,  $r$  is the square patch side length in pixels,  $p$  is the pixel intensity, and  $\delta x$  and  $\delta y$  are the patch offsets. The actual motion of the patch is calculated by minimizing  $f()$ :

$$(\Delta x_m, \Delta y_m) = \underset{\Delta x, \Delta y \in [-\rho, \rho]}{\operatorname{argmin}} f(\Delta x, \Delta y, I^j, I^k) \quad (2)$$

where  $\rho$  is the range of patch offsets, set to 10 pixels for these experiments described here. Vertical displacements of the patch are multiplied by a translational gain constant to provide the translational speed, while horizontal displacements of the patch are multiplied by a yaw gain constant to provide yaw rate information. The translational gain can be calibrated either through knowledge of the camera's height above the ground plane and its pose or experimentally — in this paper the gain was calibrated using a traverse of a known length straight corridor. An estimate of the yaw gain was calculated using the image resolution and camera's field of view, and refined using a sequence of robot rotations that were known to result in zero net angular change.

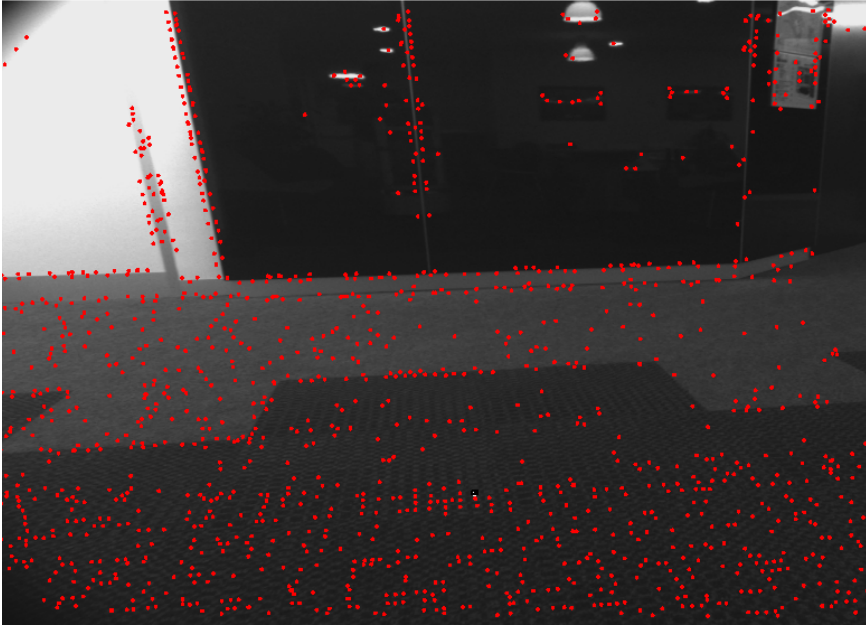
### 3.2 Structure from Motion VO

The tracking process for the Structure from Motion (SfM) VO is the monocular version of the VO described in [2]. The VO algorithm starts by detecting SURF [14] features in the images at each time-point (Fig. 3). A GPU-based feature matching scheme is then used to match features from the base camera ( $P_{(base,i)}$ ) to those at the previous time-point ( $P_{(base,i-1)}$ ). Successful matches are those where the ratio of the Euclidean distances between the best and second best descriptor candidates exceeds 0.6 [13].

Given that a set of features has been tracked successfully from the previous set of frames, it is now possible to estimate the new location of the camera rig. The most common method [7, 17, 18] is to determine the camera's pose from the existing scene structure [19]. This has the advantage of using the well conditioned scene structure from previous time-points, without having to rely as heavily on the new track data being introduced. The particular algorithm used in this case is from [20] and is combined with a MLESAC [21] hypothesis and test paradigm in order to account for incorrect feature tracks and erroneous scene structure.

Within the MLESAC routine, further speed ups are achieved by using a novel modification of the  $T_{d,d}$  test [22] to allow for early detection of poor pose estimates. This greatly reduces the need to calculate all the reprojection errors for unlikely pose hypotheses. The modification replaces the chi-squared test from [22] with a test that





**Fig. 3** Typical set of SURF features detected in an indoor image. Note the lack of suitable features due to minimal texture in this environment.

the average (Huber) truncated cost of the reprojection error for the  $d$  tested points does not exceed the average reprojection error of the best solution found so far. This modification attempts to enforce that only improved estimates for the camera’s pose be tested in full.

### 3.3 Wheel and Visual Odometry Fusion

Relying on Visual Odometry alone can result in either catastrophic failures or large deviations from the true pose when the number of tracked features is small, such as occurs when turning at the end of a corridor. To minimise these types of failures, we investigated the fusion of wheel odometry with VO.

The fusion was based on a weighted average of the wheel and visual odometry, where the weighting was determined by the estimated error computed by the visual odometry technique. As the estimated VO error increased, the wheel odometry was weighted more highly. The estimated error was mapped to a value in the range  $[0, 1]$  using the  $arctan$  of the error to give a confidence or weight  $\alpha$  for the weighted average computation:

$$\hat{X} = \alpha X_{VO} + (1 - \alpha) X_{wheel} \tag{3}$$

where  $\hat{X}$  is the estimated pose of the robot.

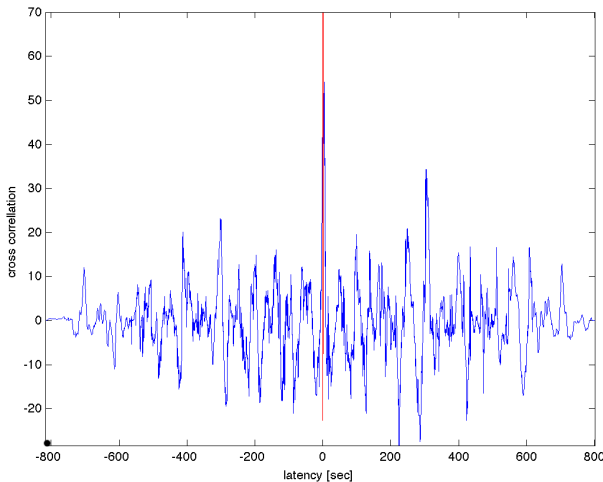
The confidence measure of the patch-based VO,

$$S = \frac{1 - d_{min}}{\bar{d}} \quad (4)$$

is based on the best SAD matching score, where  $d_{min}$  is the minimum Sum of Absolute Differences (SAD) score and  $\bar{d}$  is the average SAD score calculated over all offsets.

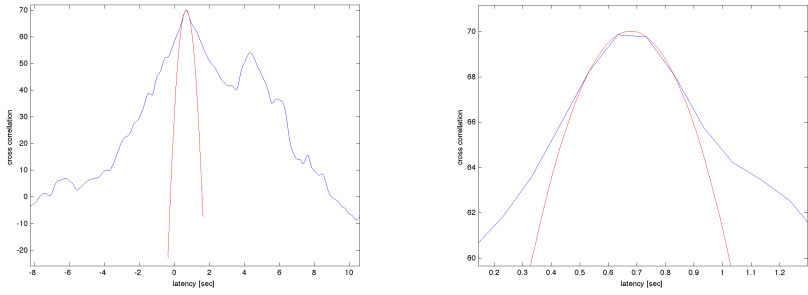
The SfM VO confidence level is based on the number of structure points added for a new camera position which are consistent with previous cameras and associated structure. Bundle Adjustment over the last 20 camera positions is used to compute the number of inlier points. A sharp drop in the number of new structure points has proven to be a good indicator of failure in the SfM VO. However, sharp drops can also be attributed to quick turns where very little structure from the previous camera positions are observed.

Both monocular VO estimates are rescaled at each camera frame using the wheel odometry at the same time step. To ensure accurate rescaling and fusion, both visual and wheel odometry data streams had to be well synchronised. This was achieved by computing the correlation between two subsets of each of the data streams (Fig. 4).



**Fig. 4** Correlation between the wheel and visual odometry data streams

A parabola was fitted to the peak with the maximum indicating the time offset between the two data streams (Fig. 5).



(a) Parabolic best fit to peak (b) Magnified view of parabolic best fit

Fig. 5 Magnified view of the peak in the correlation and the parabolic best fit to the peak

### 4 Experiments

All data was collected on a MobileRobots’ Research Guiabot shown in Figure 1. The Guiabot has a SICK LMS200 laser scanner which was used for ground truth acquisition indoors. A Point Grey Grasshopper 1.4M pixel monochrome firewire camera with a 2/3” CCD was used for image acquisition at 12.5 frames per second. The robot has two onboard computers, both running ROS, and both vision and laser data were logged continuously during operation. Wheel odometry is computed by the Guiabot which fuses wheel encoders and heading gyroscope estimates. All data processing was done offline. Camera exposure time and gain settings were controlled by an algorithm that optimized exposure by ignoring bright image regions (such as windows) while keeping an upper bound on exposure time to eliminate motion blur.

The experiments took place on two separate floors of a building. The first, *Level 7*, represents an open plan, feature-rich environment that the robot can navigate without making sharp turns. By contrast, *Level 11* features narrow corridors with long sections of plain white walls and navigating this environment required the robot to make sharp 180° turns (Fig. 6).

Ground truth was calculated using the logged transform and laser data in conjunction with GMapping [16] to make a map of the area covered in the experimental traverse. The resulting map was then provided to AMCL [23] to recover the path taken by the robot.

Analysis of the performance of the two VO algorithms (SFM and Patch-based) was performed using a windowing technique similar to that of Johnson [24]. The windowing technique works to remove the effect of global drift from the analysis. For each point on the VO trajectory, the corresponding point ( $gt_{start}$ , reached at time  $t_{start}$ ) on the ground truth trajectory is found by interpolating linearly in time. The ground truth is traversed some *window* distance (in our case 20 metres) and the time  $t_{end}$ , and location  $gt_{end}$  is noted. We then apply the transforms contained in the original VO trajectory between times  $t_{start}$  and  $t_{end}$  to the equivalent ground truth



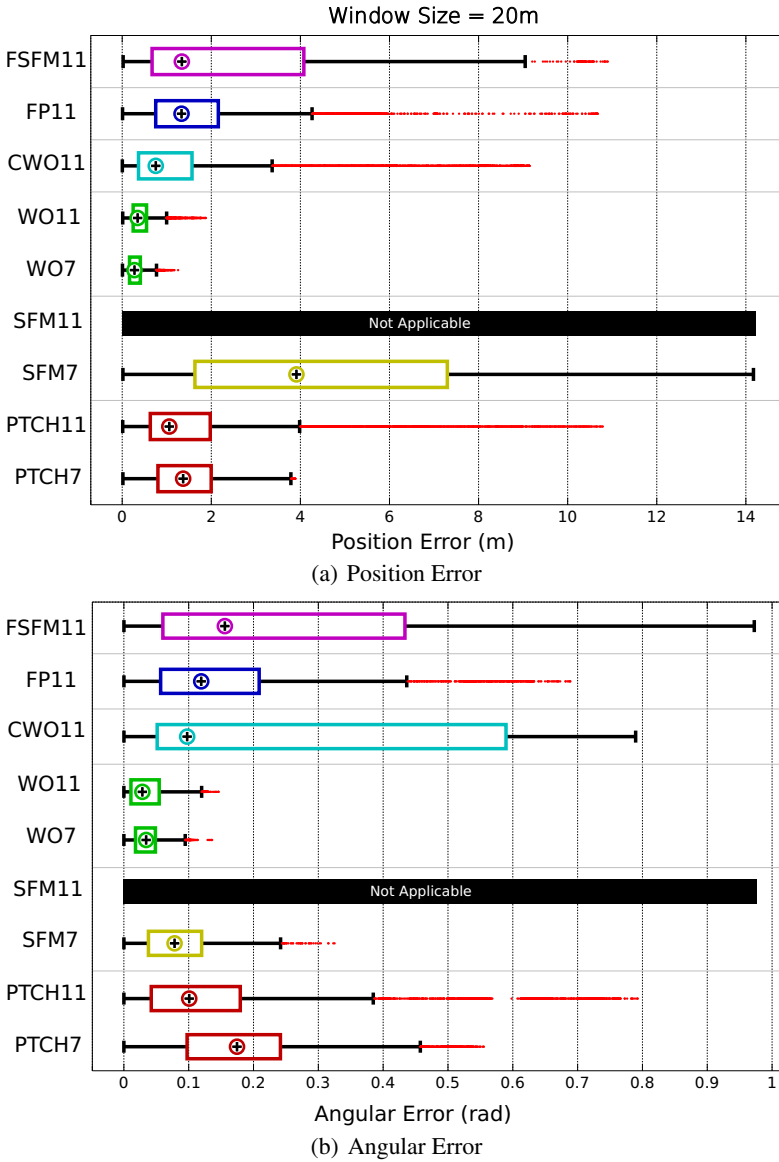
**Fig. 6** A typical image from our dataset. These types of images proved very difficult for the feature-based SfM VO.

point  $gt_{start}$ , to produce our drift-free VO position  $VO_{end}$ . The windowing error is the difference between  $VO_{end}$  and  $gt_{end}$ . Note that we found the window size had little effect on the normalized window-relative errors.

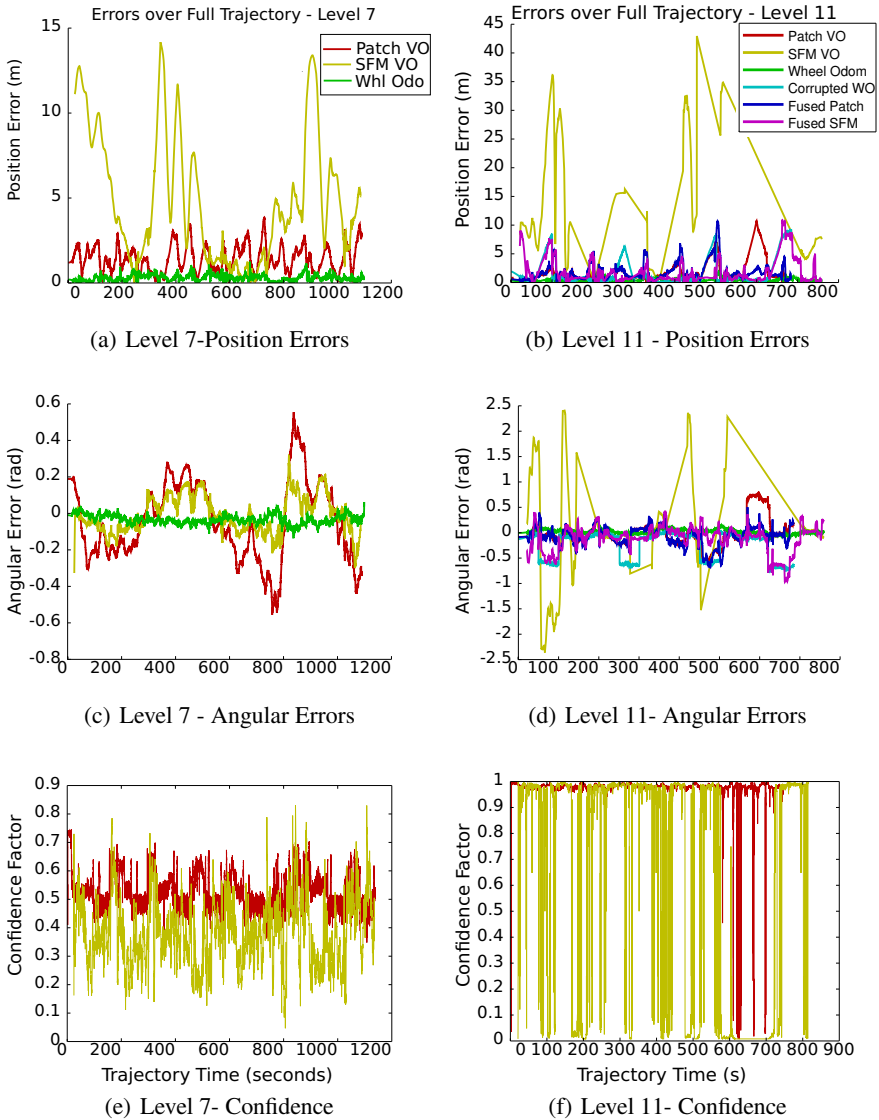
This process is repeated for every point in the VO trajectory for which we could reliably interpolate ground truth.

## 5 Results

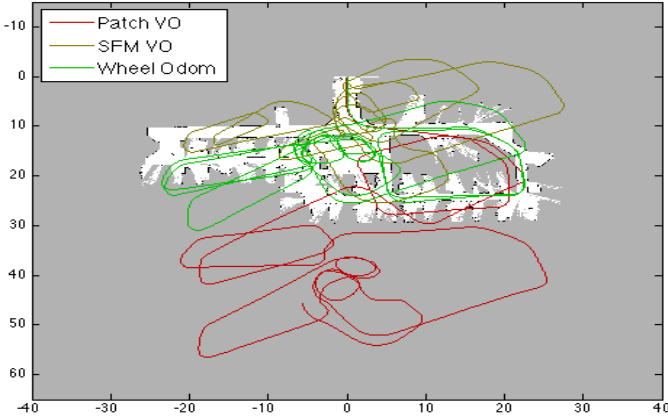
The performance of the odometry techniques was evaluated using the metric detailed in Section 4. Figures 7-10 show the results of the analysis and underpin our comparison of the techniques. Analysis of Figure 7 reveals that the wheel odometry gathered on both level's 7 & 11 (WO7 & WO11) performs extremely well. Intuitively, there is no benefit to be gained from fusing sensors unless there is significant overlap in the performance of individual sensors. The results in Figure 8 show no areas of the trajectory where Structure from Motion and Patch-Based VO (SFM11, SFM7, PTCH11, PTCH7) exhibit good performance while wheel odometry (W11, W07) is performing poorly. As such we have introduced artificial wheel odometry failure at points within the level 11 dataset (see Figures 8(b) and 8(d) CWO11, and Figure 10) which are subsequently analysed with the proposed fusion with results in Figures 8(b), 8(d) and 8(f).



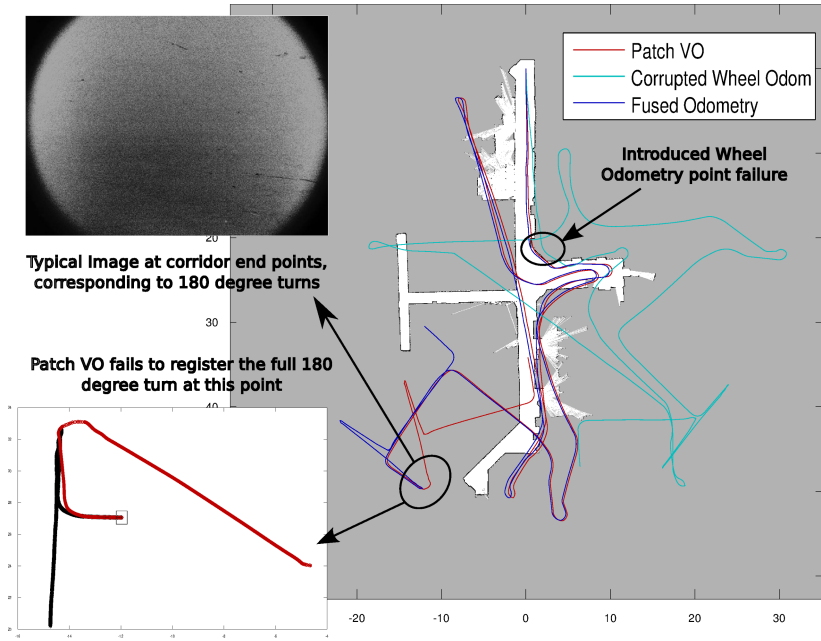
**Fig. 7** The two box plots compare the position errors and angular errors across odometry techniques. From the top, we compare Fused (with artificially corrupted Wheel Odometry) Structure from Motion VO on the more-challenging 11th floor (FSFM11), similarly Fused Patch-Based VO on Level 11 (FP11), the corrupted Wheel Odometry on Level 11 (CWO11), the raw Wheel Odometry on Levels 11 and 7 (WO11,WO7), raw Structure from Motion VO on both floors (SFM11,SFM7) and raw Patch-based VO (PTCH11,PTCH7). The circled cross denotes the median value of the absolute error and the box denotes the region where 25% to 75% of the error data points lie. Red dots to the right of the whiskers denote the spread of outliers.



**Fig. 8** (a)-(d) show how the window-error varies over the trajectory taken by the robot. Points of extreme error on the Patch VO and SfM VO curves tend to correspond to turns of greater than  $90^\circ$ . Note the scale of the error for the Level 7 dataset is much smaller, and there are no gross failures (see Figure 9) so the results of the fusion technique add little improvement and are not shown on results for this Level. In (b) and (d) it is obvious that the fusion technique greatly improves the performance of the SfM VO. Figures (e) and (f) show that drops in the confidence factor coincide with large errors in the VO, but they are not closely correlated. At  $t=700s$  in (d) note how the error in the fused SfM is high as a result of low confidence in the SfM VO result coinciding with a point error in the corrupted wheel odometry.



**Fig. 9** Overlay of level 7 odometry sources used in experiments. Analysis of this level identified only global drift and no significant point failures to which a fusion approach could recover.



**Fig. 10** A geometric view of the VO fusion method operating on the Level 11 environment. Here, corrupted Wheel Odometry is fused with Patch-based VO. The inset images highlight the difficult areas of the Level 11 environment - both VO techniques struggled with tight cornering in narrow corridors. The top inset shows the nature of the featureless images that the robot sees when turning in close proximity to white walls at the corridor ends - these may continue for tens of frames. The bottom inset shows the result of a window-error calculation at one of these failure points - note the large drift in positional error shown.

Figure 7 illustrates that Patch-based VO exhibits low positional error compared with SfM VO, but that SfM VO outperforms it in terms of angular error — the fusion of these two odometry sources to create a more accurate source is therefore a viable future option. Figures 8(b) and 8(d) illustrate a flaw in our fusion technique — basing the confidence purely on metrics derived from VO means that when both VO confidence is low and wheel odometry encounters a point of failure, the resulting system error is high. Future work will need to incorporate some form of Wheel Odometry sourced confidence measure to ensure good results, or express low confidence in the result — should both odometry sources happen to fail simultaneously.

## 6 Main Experimental Insights

The high level insights are that we find patch and feature-based VO are complementary, wheel based odometry is surprisingly good, and that there are pathological situations that severely challenge a visual odometry only approach. Patch-based VO outperforms SfM VO for position error but not angular error, and that SfM VO failed to produce a contiguous result for Level 11 so an overall metric could not be computed.

A significant surprise was the good performance of wheel odometry; perhaps aided by the significant weight of this robot, the carpeted floors on which it operated, and inbuilt gyroscopic fusion. Another important insight gained during the experiments was the challenge of obtaining images of sufficient quality for the visual odometry algorithms. Indoor environments are poorly illuminated compared to outdoors and the light gathering ability of the camera is important. The trend to high-resolution sensors means that pixel size and hence light gathering ability is low. We used a camera with the largest pixels we could buy — a 2/3" format with pixel size of  $6.45 \times 6.45 \mu m$  which have more than four times the area of the  $3 \times 3 \mu m$  pixels in a typical 1/2" sensor. Larger pixels means that exposure time can be kept smaller which reduces image blur. Blur reduces reduces high-frequency scene content and the number of detected features. Similarly, maximizing the field of view of the camera is important in indoor environments, to avoid the feature-poor imagery such as that shown in Figure 10 which is a challenge to any VO system. We also found that the camera's inbuilt exposure control strategies were unsuitable — it responded to image highlights by reducing exposure making large areas of interest too dark. We implemented our own exposure controller as a separate ROS process that periodically samples the image stream and adjusts the exposure according to a median statistic and placed an upper bound on exposure time.

**Acknowledgement.** This research was supported under Australian Research Council's Discovery Projects funding scheme (project number DP110103006).



## References

1. Konolige, K., Agrawal, M.: Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics* 24(5), 1066–1077 (2008)
2. Warren, M., McKinnon, D., He, H., Upcroft, B.: Unaided stereo vision based pose estimation. In: *Australasian Conference on Robotics and Automation, ARAA*, Brisbane (2010)
3. Sibley, G., Mei, C., Reid, I., Newman, P.: Adaptive relative bundle adjustment. In: *Proceedings of Robotics: Science and Systems*, Seattle, USA (June 2009)
4. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. *International Journal of Computer Vision* 59(3), 207–232 (2004)
5. Olson, C., Matthies, L., Schoppers, H., Maimone, M.: Robust stereo ego-motion for long distance navigation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 453–458 (2000)
6. Corke, P., Strelow, D., Singh, S.: Omnidirectional visual odometry for a planetary rover. In: *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, Sendai, Japan (September 2004)
7. Nistér, D., Naroditsky, O., Bergen, J.R.: Visual odometry. In: *CVPR* (1), pp. 652–659 (2004)
8. Nistér, D.: An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(6), 756–777 (2004)
9. Davison, A., Reid, I., Molton, N., Stasse, O.: Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(6), 1052–1067 (2007)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, Nara, Japan (November 2007)
11. Clipp, B., Lim, J., Frahm, J.M., Pollefeys, M.: Parallel, real-time visual slam. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, pp. 3961–3968 (2010)
12. Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., Fua, P.: View-based maps. *The International Journal of Robotics Research* 29(8), 941–957 (2010)
13. Lowe, D.: Object recognition from local scale-invariant features. In: *IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 (1999)
14. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* 110, 346–359 (2008)
15. Milford, M., Schill, F., Corke, P., Mahony, R., Wyeth, G.: Aerial SLAM with a single camera using visual expectation. In: *International Conference on Robotics and Automation*, Shanghai, China (May 2011)
16. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics* 23(1), 34–46 (2007)
17. Konolige, K., Agrawal, M., Sol, J.: Large scale visual odometry for rough terrain. In: *Proc. International Symposium on Robotics Research*, Hiroshima, Japan (2007)
18. Mei, C., Sibley, G., Cummins, M., Newman, P., Reid, I.: A constant time efficient stereo SLAM system. In: *British Machine Vision Conference (BMVC)*, London (2009)
19. Haralick, R.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. Comput. Vision* 13, 331–356 (1994)

20. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
21. Torr, P.H.S., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* 78, 138–156 (2000)
22. Matas, J., Chum, O.: Randomized ransac with td,d test. *Image and Vision Computing* 22(10), 837–842 (2004)
23. Fox, D.: KLD-sampling: Adaptive particle filters. In: *Advances in Neural Information Processing Systems*, pp. 713–720. MIT Press (2001)
24. Johnson, A., Goldberg, S., Cheng, Y., Matthies, L.: Robust and efficient stereo feature tracking for visual odometry. In: *IEEE International Conference on Robotics and Automation*, pp. 39–46 (May 2008)

# Building Large Scale Traversability Maps Using Vehicle Experience

Steven Martin, Liz Murphy, and Peter Corke

**Abstract.** Traversability maps are a global spatial representation of the relative difficulty in driving through a local region. These maps support simple optimisation of robot paths and have been very popular in path planning techniques. Despite the popularity of these maps, the methods for generating global traversability maps have been limited to using a-priori information. This paper explores the construction of large scale traversability maps for a vehicle performing a repeated activity in a bounded working environment, such as a repeated delivery task. We evaluate the use of vehicle power consumption, longitudinal slip, lateral slip and vehicle orientation to classify the traversability and incorporate this into a map generated from sparse information.

## 1 Introduction

When deploying long-term outdoor robots it is important that they operate safely and efficiently in order to maximise their life span and performance. In the context of mobile robots this means optimising paths to minimise energy, time and the risk of failure. In path planning the minimum cost route is often computed from traversability, a continuous scalar metric representing the cost to traverse a region.

Typically traversability is calculated from maps or sensors (such as satellite imagery, onboard cameras and LIDAR) which examine the terrain prior to traversal thus allowing potential obstacles and difficult regions to be avoided at runtime. The main focus of current traversability metrics has been on optimising a single traversal through the environment but for many long term deployments it is likely that a robot will be performing repeated tasks within a region which opens the possibility of learning from experience and improving performance.

Current traversability metrics do not incorporate information about the vehicle's prior experience yet this can provide valuable information to build or

---

Steven Martin · Liz Murphy · Peter Corke  
Queensland University of Technology, Brisbane, Australia  
e-mail: {steven.martin, liz.murphy, peter.corke}@qut.edu.au

reinforce the estimated traversability map. In this paper we describe algorithms to create traversability maps at run time from various sensor data streams and present traversability maps built from robot experience.

Unlike other learning or classification systems maps based on vehicle experience do not rely on expert knowledge or training and have a continuous signal over all terrain independent of terrain type. We argue that this approach provides a more accurate reflection of the actual cost since the metric is based on actual robot experience of the terrain and does not have a bias toward any terrain type.

Most outdoor robots are equipped with sensors such as GPS, accelerometers, gyroscopes, odometry and motor power which can be used to estimate traversability. We use these onboard sensors to compute four traversability metrics from which we estimate costmaps using a Gaussian process regression and demonstrate how these are refined over multiple traversals. We then combine the individual metrics into a weighted traversability function. The weights are trained on a dataset in a structured classified region and applied to the exploration of an unknown environment.

Section 2 presents an overview of prior work on traversability and its estimation using a-priori data and proprioceptive sensors. Section 3 describes the traversability metrics used and provides some background on the Gaussian regression process for interpolation and Section 4 details the platform and experimental areas. Section 5 presents the experimental results, analysis and discussion, and in Section 6 we present conclusions and future work.

## 2 Related Work

Traversability is a measure of the difficulty of driving through a local region and is used by mobile robot path planners to avoid obstacles or hazards and optimise paths. Traversability is usually determined by calculating a metric directly from the sensor data or by making a classification of the type of terrain the vehicle is driving over.

### 2.1 *Direct Traversability Metrics*

The most common method of calculating traversability directly is to use the 3D structure of the surface. This can be gathered locally using nodding or spinning LIDAR, stereo, structured light cameras or globally from aerial LIDAR.

A simple surface roughness-based traversability metric was used by [4] to adapt the speed of mobile robot based to the terrain immediately in front of it. In [10] three traversability metrics were developed which use roughness of the field to determine the robot's ability to cover the area and cross regions. Roughness has also been used by rover style robots, for example [14] used stereo vision to determine local roughness as well as terrain slope in order to plan paths which avoided rocky and sloped areas.

Another method using the 3D structure to determine the traversability is to examine the configuration of the robot on the surface. This approach was employed

by [6] and [7] which assessed the vehicle's stability or tractive force at a point and this was equated to traversability. This is also useful for reconfigurable robots which can adapt their suspension or geometry to optimise the traversability in different regions [6].

Another approach to calculating traversability directly is based on terrain colour analysis [17]. However this makes strong assumptions about the association between colour and terrain and hence traversability. Learning techniques can be applied but the use of imagery usually requires some form of pixel-based terrain classification.

## ***2.2 Classification***

A more popular approach has been to classify terrain. Robots often work in structured environments, the most ubiquitous being road networks, where the surface types are known. This allows classifications of terrain types to be made and used in planning. The primary sensor modality for classification has been imagery. Attempts have also been made to link a priori information from satellite images with local classifications to improve long distance traversability estimates.

Classification is often simplified to a binary problem of whether terrain is locally traversable or untraversable. This type of traversability assessment has been demonstrated by [8] using a neural-network-based approach and similar classifiers have been described by [12] and [16]. A binary classification is valuable for hazard avoidance but does not provide any additional information about the cost of regions.

Classification from satellite and aerial LIDAR into ground structure such as buildings, road and vegetation and then a global traversability has been shown by [15] and demonstrated on the Crusher platform [13]. An alternative to image-based classification is to use proprioceptive sensors. In [5] the use of a tactile sensor was demonstrated to classify the terrain type according to sensed vibration.

A purely classification-based assessment has the disadvantage of relying completely on training data and expert knowledge of the traversability cost of each terrain type, and ignoring intra-class variation. The classification approach can be augmented to use proprioceptive sensors such as the wheel slip, vibration, energy consumption to infer whether or not the classification is successful and possibly to update the cost of the terrain class.

## ***2.3 Self-supervised Classification***

A novel approach to self supervised classification was demonstrated by [1] which used onboard stereo imagery to determine a local traversability map and incorporated vehicle slip to supervise learning of the terrain classification. The work of [2] extends [13] to include map online learning that learns the association between a local traversability from onboard sensors and the classification of terrain from satellite imagery. A similar approach was applied to a lunar rover platform by [3].

The focus of much prior work has been on the situation of a vehicle performing a single traverse through a region. Based on this assumption the proprioceptive sensors are much more desirable as they allow hazardous areas to be avoided. The issues with training and misclassification of regions has been improved by incorporating proprioceptive sensors and also providing a better grounding of the classified terrains cost.

### 3 Technical Approach

Typical robots are equipped with a variety of sensors, many of which can be used to assess traversability of the terrain they are driving over. For example GPS, accelerometers, gyroscopes, odometry and motor current sensors provide rich information about the vehicle's motion with respect to the plan, terrain induced vibration, and the work required.

In this paper we compute four traversability metrics to describe the vehicle's performance along the path: power consumption, longitudinal slip, lateral slip and vehicle orientation. These metrics are designed to capture the properties of the terrain as seen by the vehicle. The power consumption provides a direct energy cost for traversing each region. The slip metrics provide an indication of the amount of tractive force the vehicle can exert, while the orientation shows the stability or risk of rollover of the vehicle. Each of these metrics is described below.

#### 3.1 Traversability Metrics

The power metric is calculated from the vehicle's electrical power consumption to give a measure of energy used per meter travelled

$$T_P = \frac{VI}{|v|} \text{ [J/m]} \quad (1)$$

where  $V$  is the battery voltage,  $I$  is the current to the drive motor, and  $v$  is the vehicle velocity estimated from GPS. The velocity is estimated within the GPS receiver based on the Doppler shift of the GPS signals which provides a much more accurate measurement than the position derivative [11].

The longitudinal slip metric is

$$T_l = \frac{\omega R_u - |v|}{\omega R_u} \quad (2)$$

which is non-dimensional and where  $\omega$  is the angular velocity of the drive wheels and  $R_u$  is the radius of the wheel.

The angular slip metric is

$$T_{\dot{\theta}} = \frac{\frac{\tan(\theta)}{W_b} v - \dot{\theta}_z}{\frac{\tan(\theta)}{W_b} v} \tag{3}$$

which is non-dimensional and where  $\theta$  is the steering angle,  $\dot{\theta}_z$  is the rotational velocity about the vehicle frame z-axis in the and  $W_b$  is the wheel base length.

The local slope of the terrain impacts mobility and reduces the normal force on the wheels, resulting in lower tractive force and an increased risk of rollover. As the IMU is fixed in the vehicle frame it can be used to calculate the normal force directly. The orientation traversability metric is

$$T_o = 1 - \frac{|a_z - g|}{g} \tag{4}$$

which is non-dimensional and where  $a_z$  is the vertical acceleration in the vehicle frame and  $g$  is gravity. This metric does account for acceleration of the vehicle, however acceleration which changes the normal force is expected to affect the traversability.

### 3.2 Interpolation

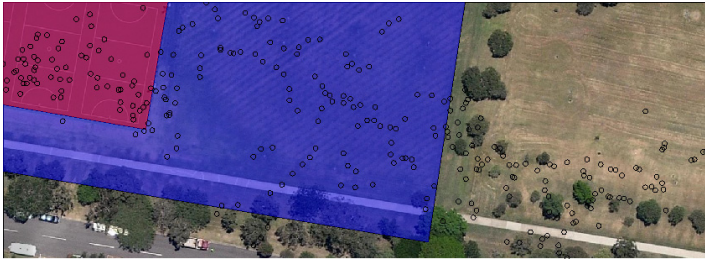
The data points which are captured from robot experience are sparsely distributed in the region over which the robot operates, and noisy due to variation in terrain and vehicle motion. To make these measurements into a continuous spatial function, useful for path planning, it is necessary to smooth and interpolate the raw data. Gaussian processes (GP) with a squared exponential kernel have been popular in applications such as reconstructing terrain geometry [18] where they interpolate and reconstruct the underlying non-linear characteristics of the surface. In prior work heteroscedastic and homoscedastic Gaussian processes have been used to generalise probabilistic traversability costmaps from sparse measurement data. Based on these results we choose a squared exponential kernel with hyperparameters estimated using cross-validation and Geisser’s Predictive Probability criteria for optimisation.

The GP model is used to learn the relationship between an input vector of the robots position and the measurement of traversability. The traversability is normalised from 0 to 1 where 1 is the most favourable traversability measurement for each metric during the exploration phase and 0 is the least favourable measurement. The GP model is trained on the traversability from the exploration dataset and then used to predict the traversability and variance on a 1m grid for the region.

## 4 Experimental Setup

The rOscar platform, a small scale car-like vehicle shown in figure 1, was used to map the traversability in two parkland areas. The vehicle was used to perform a

**Fig. 1** The rOscar platform is a 1/8th scale RC car platform. It is equipped with a Hokoyu UTM-30LX laser scanner, a Microstrain 3DM-GX2 IMU, odometry and current sensors. The onboard processor is a Gumstix Overo running ROS and running drivers as published in the cyphy ROS repository. See [9] for more details.



**Fig. 2** *Graceville*: The terrain traversability samples are shown in the test area (black) on three terrain regions. These were manually identified as concrete (red), short manicured grass (blue) and long unkept grass terrain.

random exploration strategy (using GPS waypoints) with a superimposed high frequency sinusoidal steering demand to excite slip dynamics. The vehicle was driven at an approximate speed of 5m/s (18 km/h, 11 mph) and the traversability metrics are calculated at 1Hz.

The *Graceville* data set spans an area of  $230 \times 96$  meters. It contains three distinct areas consisting of concrete netball courts, short grass netball courts and a long grass untended field. A total of 287 slip measurements were taken on the field. The different regions are shown and highlighted in Figure 2. We expected that the concrete, short and long grass areas would provide a good environment for demonstrating the slip and energy traversability metrics. This data set was used to evaluate the four proposed traversability metrics and costmap regression techniques.

The *Everton Park* data covers  $208 \times 134$  meters. It is a suburban park consisting of mainly grass and shrubbery. A total of 567 slip measurements were taken in the park over a period of approximately 2 hours. This unstructured park area provided a more interesting experimental environment for testing the metrics. After the exploration phase was conducted 8 paths were then traversed to randomly selected



**Fig. 3** *Everton Park* : The terrain traversability samples are shown in the test area (black) on three terrain regions



waypoints with a modified excitation to assess the correlation between the generated traversability maps and the vehicle’s experience along the specific paths.

The exploration strategy consisted of selecting random waypoints within an exploration area and then performing a sinusoidal path directed towards these locations. The traversability metrics were calculated when the vehicle was deemed to be in an approximate steady state with sufficient angular velocity to excite the slip dynamics. Whether the vehicle was in steady state was determined by its deviation from the commanded velocity. When performing test paths through the environment the frequency of the oscillation was increased to allow the vehicle to drive an approximately straight path and the traversability recorded at the peaks when commanded angular velocity was at a maximum. For the tested paths the vehicle was always assumed to be in an approximate steady state at the peaks.

## 5 Results

### 5.1 Metrics

The *Graveville* dataset was used to assess how well the traversability metrics reflected the actual terrain. As the terrain had been pre-classified onsite, GPS locations were used to partition the traversability measurements according to terrain type.

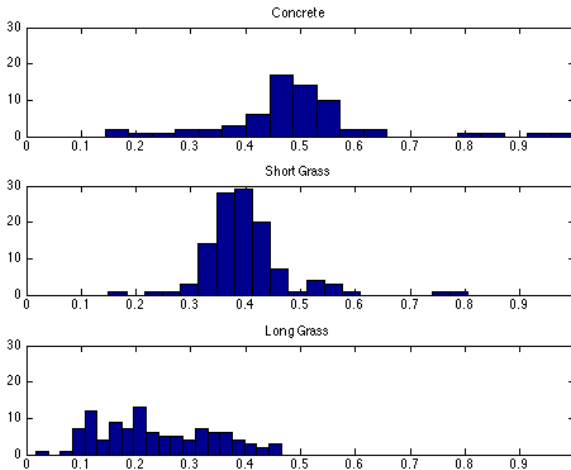
We expected the known regions of concrete, manicured grass and long grass to have quite different levels of traversability. Our goal was not to classify terrain

**Table 1** Summary of the mean and standard deviation for classified surfaces

	Concrete		Short Grass		Long Grass	
	Avg.	Std. Dev	Avg.	Std. Dev	Avg.	Std. Dev
$T_P$	0.9591	0.0231	0.9223	0.0289	0.8111	0.1221
$T_{\hat{\theta}}$	0.4560	0.1618	0.4094	0.0919	0.2252	0.1247
$T_l$	0.4907	0.1454	0.4014	0.0811	0.2341	0.1054
$T_o$	0.9356	0.1211	0.9155	0.0979	0.9121	0.0737

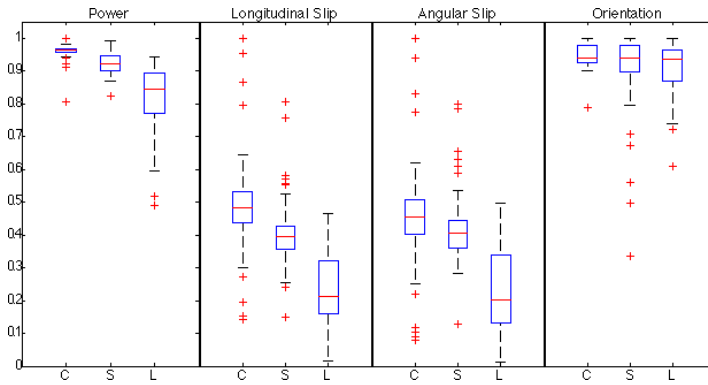
but to observe an expected ranking for each terrain type and the variation of the metrics over these regions. The concrete surface was expected to have the highest traversability followed by the short grass and then the long grass. Similarly the variation or noise was expected to be smallest for concrete and largest for the long grass.

The statistics for the sample points on terrain types is summarised in Table 1. Histograms showing the  $T_l$  measure for different terrain types is shown in Figure 4. Figure 5 uses box-whisker plots to summarise and compares the different traversability metrics for each of the known terrain types. The metrics are normalised from 0 to 1 where 1 indicates the most traversable region.



**Fig. 4** Histogram of the distribution of samples of  $T_l$  on terrain types

The expected rankings can be seen in table 1 and the box and whisker plot in figure 5 — both show the expected traversability ranking across terrain types although there is a significant number of outliers and overlap in the measurement intervals. The histograms of  $T_l$ , figure 4, also shows the distribution is unimodal and Gaussian and similar results were found for other metrics.



**Fig. 5** Box plots of traversability metrics grouped by metric and showing terrains in the order concrete (C), short grass (S) and long grass (L)

For  $T_o$  the calculated metrics all appear to approach the upper bound of one, this is likely due to the experimental area being approximately flat. The wider range of values and lower average could be attributed to the local roughness in the grass areas which was expected due to the small scale of the vehicle relative to terrain variation.

### 5.2 Interpolation

Gaussian processes are used to identify the underlying signal from sparse and noisy data. Figure 6 shows the traversability maps for the *Graceville* test region after regression.

From observations the  $T_p$  metric provides the best distinction between the terrain types. It has the lowest standard deviation on the two homogeneous terrains and the traversability map resembles the manually classified terrain in figure 2. The slip metrics,  $T_{\theta}$  &  $T_l$ , also show some distinction between the terrain types however the similarity of the signals and the variation within the terrain groups perhaps indicates some dependence on the path driven. The orientation metric is largely constant across the explored region, which is expected since the test region was flat.

As expected the variance of the metrics increases with distance from the sampled data, as shown in figures 6 & 7, however it can also be seen that in the absence of information the traversability tends to zero or one and may contribute to under or over estimation of the traversability in unexplored regions.

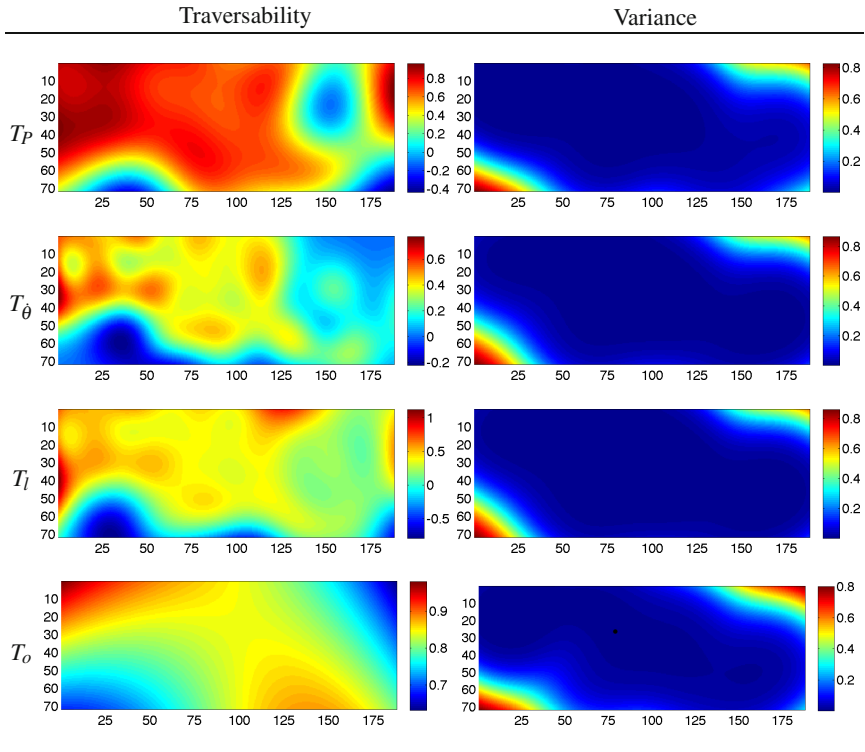
The correlation of the costmaps was calculated by

$$C = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right) \left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}} \tag{5}$$

**Table 2** 2D Correlation coefficient of the interpolated costmaps for *Graceville*, left, and *Everton Park* Right

	$T_P$	$T_{\hat{\theta}}$	$T_l$	$T_o$			$T_P$	$T_{\hat{\theta}}$	$T_l$	$T_o$
$T_P$	1.0	-	-	-	$T_P$	1.0	-	-	-	-
$T_{\hat{\theta}}$	0.69	1.0	-	-	$T_{\hat{\theta}}$	0.051	1.0	-	-	-
$T_l$	0.66	0.82	1.0	-	$T_l$	0.12	0.92	1.0	-	-
$T_o$	0.41	0.71	0.46	1.0	$T_o$	0.21	0.65	0.76	1.0	-

where  $A$  and  $B$  are the predicted mean matrices of traversability, and detailed in table 2. There is a strong correlation between the  $T_P$  and  $T_l$  metrics in the *Graceville* dataset. In the *Everton Park* dataset there is also a strong correlation between the  $T_{\hat{\theta}}$  and  $T_l$  metrics, but a reduced correlation between  $T_P$  and  $T_{\hat{\theta}}$ ,  $T_l$  and  $T_o$ . We believe that this is due to the *Everton Park* area being non-flat and power metric being dependant on the mode of traversal.



**Fig. 6** Traversability and variance of the traversability metrics for the *Graceville* dataset

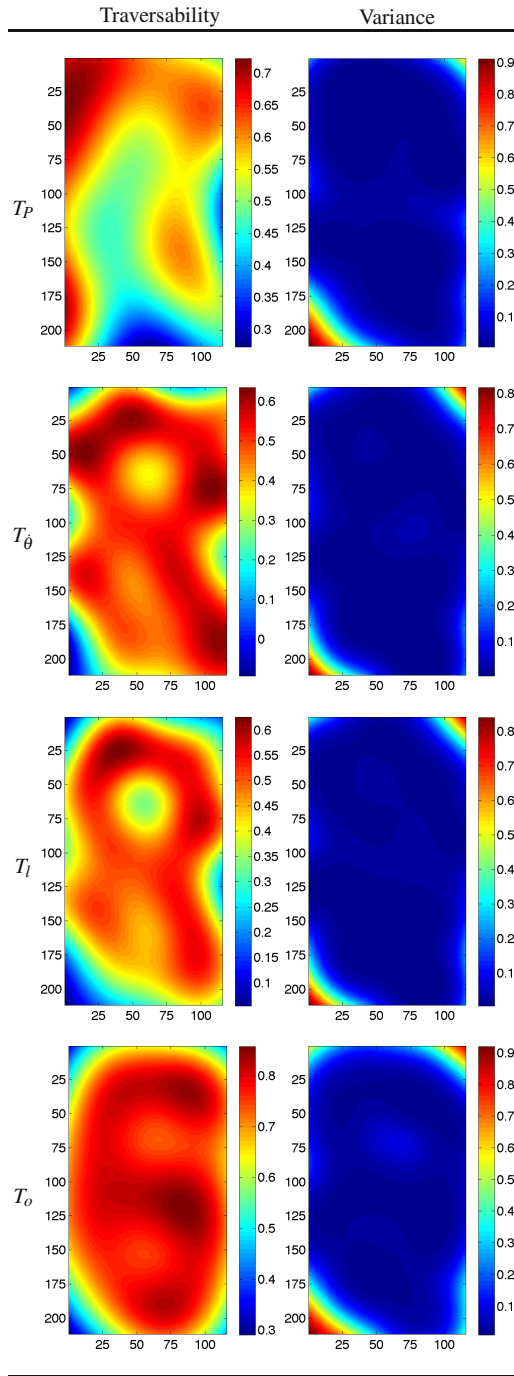


Fig. 7 Traversability and variance of the traversability metrics for the *Everton Park* dataset

### 5.3 Testing Paths

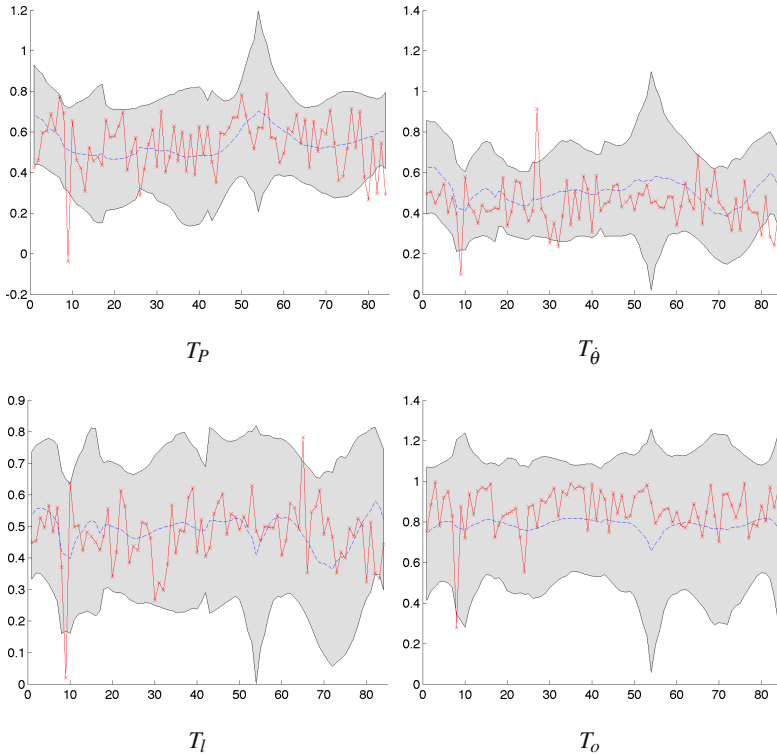
The *Everton Park* dataset was gathered to test the application of the metrics to natural terrain. In natural terrain it is difficult to classify which areas are likely to be costly, as the colours can often be ambiguous and the qualities of preferable terrain may not be known prior to operation.

To test the cost map the vehicle was drive to 8 randomly chosen waypoints within the explored region, as shown in 8. The frequency of the sinusoidal path was increased to give allow detection of the traversability metrics while travelling in an approximate straight line. Figure 9 shows the predicted and actual traversability over the path.

While the measured values are mostly within the 95% confidence interval, the noise on the signal is much more significant than any local variation within the metric. It could be argued that the as the overall costmap utilises a large number of samples it is likely to resemble the true traversability of the terrain but at the scale the vehicle is operating the variation in global traversability is largely insignificant compared to variation locally.



**Fig. 8** *Everton Park* : The traversed path is shown in black and the goal waypoints indicated by the markers



**Fig. 9** The estimated traversability (blue) with 95% confidence interval (grey) compared with measured traversability (red)

## 6 Conclusions and Future Work

In this paper we have investigated the online computation of terrain traversability metrics and their efficacy for path planning. We evaluated four different metrics that are readily computed from common mobile robot sensors with data collected in from different areas that the robot randomly explored.

The first experiment investigated whether the online metrics were indicative of known surface types. While there was some correlation and the power measure was the most discriminative the data shows a significant amount of variation which we believe is due to the small scale of the vehicle with respect to the terrain — a tussock of grass is of similar scale to the vehicle.

The second experiment investigated whether the metrics could be used to identify regions which were preferable in natural terrain which would be difficult for current classification techniques. The region could be mapped and the data used to predict which areas would be more traversable but it was difficult to verify this as the noise on the signal overshadowed the variation globally.

The small scale vehicle was chosen for this work as it was expected that variation in the terrain would have a larger impact on the vehicles performance and therefore be easier to detect. While this was shown in the *Graveville* data set it is also evident that the variation in the terrain at this scale is a problem. Future work plans to examine the use of larger vehicles in natural terrain and also evaluate the performance against classification from satellite imagery.

The use angular and longitudinal slip measurements were also very closely correlated for both data sets indicating that one of these metrics gives all necessary information about the terrain. For future work the longitudinal slip metric is preferable as the vehicle does not need to introduce curvature to its path to provide excitation for angular slip. The inclusion of additional sensor inputs for learning locally measured traversability is also of interest and future work plans to includes using satellite imagery and local imagery for classification.

**Acknowledgements.** The authors would like to acknowledge the CSIRO: Autonomous Systems Lab for supporting this work.

## References

1. Angelova, A., Matthies, L., Helmick, D., Perona, P.: Learning slip behavior using automatic mechanical supervision. In: 2007 IEEE International Conference on Robotics and Automation, pp. 1741–1748. IEEE (2007)
2. Bagnell, J.A., Bradley, D., Silver, D., Sofman, B., Stentz, A.: Learning for autonomous navigation. *IEEE Robotics & Automation Magazine* 17, 74–84 (2010)
3. Brooks, C., Iagnemma, K.: Self-supervised terrain classification for planetary surface exploration rovers. *Journal of Field Robotics*
4. Castelнови, M., Arkin, R., Collins, T.: Reactive speed control system based on terrain roughness detection, pp. 891–896 (2005)
5. Giguere, P., Dudek, G.: Surface identification using simple contact dynamics for mobile robots. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 3301–3306. IEEE (2009)
6. Iagnemma, K., Dubowsky, S.: *Mobile Robots in Rough Terrain*. Springer (2004)
7. Ishigami, G., Nagatani, K., Yoshida, K.: Path planning and evaluation for planetary rovers based on dynamic mobility index. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 601–606. IEEE (2011)
8. Konolige, K., Agrawal, M., Blas, M., Bolles, R., Gerkey, B., Solà, J., Sundaesan, A.: Mapping, navigation, and learning for off-road traversal. *Journal of Field Robotics* 26(1), 88–113 (2009)
9. Martin, S., Corke, P.: Path planning using surface shape and ground properties. In: Drummond, T. (ed.) *Australasian Conference on Robotics and Automation (ACRA 2011)*, pp. 1–7. ARAA, Monash University (2011), <http://eprints.qut.edu.au/47115/>
10. Molino, V., Madhavan, R., Messina, E., Downs, T., Jacoff, A., Balakirsky, S.: Traversability Metrics for Urban Search and Rescue Robots On Rough Terrain. In: *Proceedings of the Performance Metrics for Intelligent Systems* (2006)
11. Serrano, L., Kim, D., Langley, R., Itani, K., Ueno, M.: A gps velocity sensor: how accurate can it be?—a first look. In: *ION NTM*, pp. 875–885 (2004)



12. Shneier, M., Shackelford, W., Hong, T., Chang, T.: Performance evaluation of a terrain traversability learning algorithm in the DARPA LAGR program. Tech. rep., National Inst. of Standards and Technology Gaithersburg Md (2009)
13. Silver, D., Sofman, B., Vandapel, N., Bagnell, J., Stentz, A.: Experimental analysis of overhead data processing to support long range navigation, pp. 2443–2450 (2006)
14. Singh, S., Simmons, R., Smith, T., Stentz, A., Verma, V., Yahja, A., Schwehr, K.: Recent progress in local and global traversability for planetary rovers. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2000, vol. 2, pp. 1194–1200 (2000), doi:10.1109/ROBOT.2000.844761
15. Sofman, B., Bagnell, J., Stentz, A., Vandapel, N.: Terrain classification from aerial data to support ground vehicle navigation. Tech. rep., Robotics Institute, Carnegie Mellon University (2005)
16. Talukder, A., Manduchi, R., Castano, R., Owens, K., Matthies, L., Castano, A., Hogg, R.: Autonomous terrain characterisation and modelling for dynamic control of unmanned vehicles. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 708–713. IEEE (2002)
17. Ulrich, I., Nourbakhsh, I.: Appearance-based obstacle detection with monocular color vision. In: Proceedings of the National Conference on Artificial Intelligence, pp. 866–871. AAAI Press, MIT Press, Menlo Park, Cambridge (1999, 2000)
18. Vasudevan, S., Ramos, F., Nettleton, E., Durrant-Whyte, H.: Large-scale terrain modeling from multiple sensors with dependent gaussian processes. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1215–1221 (2010), doi:10.1109/IROS.2010.5650769

# Automatic and Full Calibration of Mobile Laser Scanning Systems

Jan Elseberg, Dorit Borrmann, and Andreas Nüchter

**Abstract.** Mobile scanning, i.e., the practice of mounting laser scanners on moving platforms is an efficient way to acquire accurate and dense 3D point clouds of outdoor environments for urban and regional planning and architecture. The mobile scenario puts high requirements on the accuracy of the calibration of the measurement system, as small calibration inaccuracies lead to large errors in the resulting point cloud. We propose a novel algorithm for the calibration of a mobile scanning system that estimates the calibration parameters for *all* sensor components simultaneously without relying on additional hardware. We evaluate the calibration algorithm on several real world data sets where ground truth is available via an accurate geodetic model.

## 1 Introduction

Laser range scanning provides an efficient way to actively acquire accurate and dense 3D point clouds of object surfaces or environments. Mobile scanning, i.e., the practice of mounting the range sensors on a moving platform or robot is the state of the art method for modeling in architecture and for urban and regional planning. The mobile scenario puts high requirements on the accuracy of the calibration of the measurement system, as small calibration inaccuracies lead to large errors in the resulting point cloud. Modern systems like the Riegl VMX-450 and the Lynx Mobile Mapper as produced by Optech work along the same basic principal. They combine a high precision GPS, a highly accurate Inertial Measurement Unit (IMU) and the odometry of the vehicle to compute the fully timestamped trajectory. Using a process called motion compensation this trajectory is then used to “unwind” the laser range. In this paper we propose a novel algorithm for the calibration of a mobile scanning system that estimates the calibration parameters for *all* sensor components simultaneously without relying on additional hardware. We have developed

---

Jan Elseberg · Dorit Borrmann · Andreas Nüchter  
Jacobs University Bremen gGmbH, Campus Ring 1, 28759 Bremen  
e-mail: j.elseberg@jacobs-university.de

a general framework for calibrating mobile platforms that estimates *all* configuration parameters including that of the odometry in a unified fashion. We evaluate our algorithm data sets acquired by our own mobile platforms Irma3D and Lars3D as seen in Fig. 1. The main sensor of each robot is a Riegl VZ-400 laser scanner. Instead of a 2D laser scanner common in a mobile scanning system, this is a 3D laser scanner. This design retains the high degree of automation and speed of common mobile laser scanning systems. Furthermore, only a single laser range sensor is required to produce models with minimal data shadowing.

## 2 State of the Art

In order for the mobile laser scanner to acquire high quality range measurement data the position and orientation of every individual sensors must be known. This paper is concerned with algorithmic calibration of these systems, i.e. algorithms to establish the parameters that best describe sensor displacements based on the sensor data itself. This process takes place after a calibration with external instruments was performed to finetune the roughly measured parameters.

The most basic sensor of the vehicle is its odometry, which estimates the vehicle's pose by measuring and extrapolating the wheel rotations. Martinelli et al. presented a method to calibrate the odometry readings using an augmented Kalman filter [4]. The algorithm estimates the odometry parameters on the basis of pose information acquired by the SLAM system. This relies on the strong assumption that calibration errors of the odometry do not influence the accuracy of the other sensors. This does not hold for mobile scanners where errors in the odometry lead to errors in the reconstructed 3D model and thereby to inaccuracies of the SLAM system. As odometry is the least reliable estimator of a vehicles pose, mobile laser scanners are usually also equipped with more precise positioning sensors, e.g., an IMU or a GPS device. Traditionally, these are calibrated against other positioning devices whose pose in relation to the vehicle is already known [2]. Nebot and Durrant-Whyte presented a method for calibrating a low cost six degrees-of-freedom IMU on a land vehicle [6]. Initially the vehicle and the IMU are at rest to remove gravitational bias from the IMU measurements. Then the redundancies in the measurement are exploited to estimate the calibration parameters of the IMU during a test drive.

The final sensor to calibrate is the laser measurement device. Classically, this is done using a process called boresight alignment. Boresight calibration is the technique of finding the rotational parameters of the range sensor with respect to the already calibrated IMU/GPS unit. Skaloud and Schaer describe a calibration method where an airplane equipped with a laser scanner makes several flights over a residential area [11]. Planes are extracted from the roofs in every flyover. Then, the planes are matched against each other to minimize the calibration error. A similar method was developed by Rieger et al. for kinematic laser scanning [9]. Here the vehicle drives past the same house several times. Again the planar surface of the building is exploited to estimate the calibration parameters of the laser scanner. The laser scanners calibration parameters can also be estimated when the vehicle

itself is stationary. Talaya et al. presented a calibration method for estimating the boresight parameters of a laser scanner by registering several scans of the environment at different positions [12]. The position and orientation of the vehicle is known at any one point and the scans are registered using landmarks. Recently Underwood et al. presented an approach for calibrating several range scanners to each other with no information about the pose of the vehicle [13]. The vehicle scans a previously known environment from several positions. The range data is then manually labeled, so that the ground truth for each data point is known and an error metric can be constructed. Minimizing the error then yields optimal calibration parameters for the range sensors.

Our calibration system is similar in that multiple sensors are calibrated using a quality metric on the constructed point cloud. However, we require no manual selection of points or any special environment for the calibration. Instead, we employ a quality metric that is similar to the one used by Sheehan et al. [10]. They calibrate a laser scanner of their own design by computing the minimum of the quality metric with respect to the internal calibration parameters. We apply it in our own calibration framework and also improve upon the metric by reducing the time complexity that is involved in its evaluation. Furthermore, we estimate all calibration parameters for all sensors simultaneously. This allows us to forego a separate calibration process for every subsystem.

Aside from sensor misalignment a second source of errors are timing related issues. On a mobile platform the several subsystems need to be synchronized to a common time frame. This can be achieved with pure hardware via triggering or with mixes of hard and software like pulse per second (PPS) or the network time protocol [5]. Good online synchronization is not always available for all sensors. Olson [7] has developed a solution for the synchronization of clocks that can be applied after the fact.

### 3 Technical Approach

Calibration is the process of estimating the parameters of a sensor system. The closer these values are to the true physical quantities, the more accurate the final measurements of the sensor system will be. In the context of mobile laser scanners, there are several types of parameters of note. First, the geometrical alignment of each subsystem with respect to the vehicle. There are many frames of references on a mobile platform. The challenge of establishing the transformation between the vehicle and the global reference system is referred to as the simultaneous localization and mapping (SLAM) problem. The localization is subject to the accuracy of the positioning systems themselves and is not discussed in this paper. For proper data acquisition, the full 6 degrees of freedom (DOF) pose  $\mathbf{V}_s = (t_{x,s}, t_{y,s}, t_{z,s}, \theta_{x,s}, \theta_{y,s}, \theta_{z,s})$  of each sensor  $s$  with respect to the vehicle frame is essential. Incorrect geometrical calibration leads to incorrect trajectory estimation and systematic errors in the final point cloud.

Aside from the general alignment parameters, there are internal parameters that are specific to certain sensors. These quantities influence only the measurement accuracy of the sensor they are related to and could in principle be calibrated independently of the rest of the system. The laser measurements are dependant on the internal alignment of the emitter and the mirror geometry of the laser scanner whereas the odometry is dependant on wheel circumference  $w$  and axis length  $a$  of the car. In practice, for many sensors the modification of the calibration parameters is unnecessary or infeasible, as the internal calibration is often performed by the manufacturers themselves. The odometry parameters are therefore the only internal parameters that we are concerned with in this paper.

Finally, systematic timing errors due to latencies can be counteracted by offset parameters  $o_s$ . We assume all sensor measurements are timestamped. Time frames are synchronized by an offset that represents the minimal inherent delay between a measurement and its reception in the system. While our proposed algorithm is capable of adjusting any parameter, systematical synchronization errors are minor for mobile laser scanning systems and do not contribute to the quality of the final point cloud in a significant way.

The principal behind our approach is to find the calibration parameters that produce the most accurate point cloud possible. The parameters for all sensors  $s$  are concatenated to construct the calibration vector  $\mathbf{C} = (a, w, \mathbf{W}_0, o_0, \dots, \mathbf{W}_n, o_n)$ . Then the process of extracting the point cloud  $\mathbf{P} = \{\mathbf{p}_0, \dots, \mathbf{p}_m\}$  with  $\mathbf{p}_i = (x_i, y_i, z_i)$  from the entirety of measurements  $\mathbf{M}$  of each sensor can be said to be a function  $f(\mathbf{M}, \mathbf{C}) = \mathbf{P}$ . In the case of a mobile scanner this function includes extrapolating measurements from available positioning systems to create the trajectory of the vehicle. SLAM algorithms may also be used to arrive at the final point cloud.

To find the optimal calibration we must define an appropriate quality measure on  $\mathbf{P}$ . We employ a general quality measure that is somewhat similar to the one used by Sheehan et al. [10]. We model the points  $\mathbf{p}_i$  as drawn from a probability distribution function (*pdf*)  $d(\mathbf{l})$  which represent the probability that a specific location  $\mathbf{l}$  has been measured. The pdf can be approximated as

$$d(\mathbf{l}) = \frac{1}{m} \sum_j^m G(\mathbf{l} - \mathbf{p}_j, \sigma^2 \mathbf{I}) \quad (1)$$

where  $G(\mu, \sigma^2 \mathbf{I})$  is a Gaussian with mean  $\mu$  and covariance  $\sigma^2 \mathbf{I}$ . Calibration errors lead to surfaces appearing at multiple positions. The entropy of  $d(\mathbf{l})$  increases with these errors and decreases the more compact the point cloud is. Thus, an entropy measure on  $d(\mathbf{l})$  is also a quality measure for  $\mathbf{P}$ . Sheehan et al. derive the following simplified entropy measure, which depends on only the pairwise distance of the sample points:

$$E_2(\mathbf{P}) = - \sum_i^m \sum_j^m G(\mathbf{p}_i - \mathbf{p}_j, 2\sigma^2 \mathbf{I}) \quad (2)$$

In Sheehan et al.'s calibration, the Jacobian of  $E_2$  with respect to the calibration parameters of their system is used to apply Newton's method for optimization. This is not possible in our case for several reasons. First, our calibration is supposed to be general, i.e., no definitive system to calibrate for is given. Second, the inclusion of parameters for the positioning systems make the derivation of the Jacobian infeasible. This is due to the fact that in order to compute a global measurement  $\mathbf{p}_i$  at time  $t_i$  the pose estimate of the vehicle at that time must be known. However, to compute  $\mathbf{W}_i$  all sensor measurements prior to  $t_i$  may be taken into account. Furthermore, the presence of multiple positioning sensors requires sensor fusion, thereby increasing the non-linearity and complexity of the entropy measure even more. In addition, we acquire a large number of sample points, usually in the order of several millions for properly calibrating an entire mobile platform. Thus, the quality measure  $E_2(\mathbf{P})$  is infeasible for the calibration using large point clouds. One way of dealing with this problem is to reduce the number of points. We propose to uniformly subsample the entire point cloud. This is achieved by first binning the point cloud in a regular 3d grid and then randomly selecting a number of points in each voxel. Both the number of points and the side length of a voxel can vary to allow for many possible point densities. An additional advantage of the uniformity of the subsampling is that surfaces closer to the scanner do not unfairly contribute more to the quality measure than surfaces that are farther away. Furthermore, we propose to simplify the measure by not using every possible pair of points. For every point  $\mathbf{p}_i$  that remains from the initial sub sampling we determine its closest point  $\mathbf{q}_i \in P$  such that  $|t_i - t_j| > \delta$ . Here,  $\delta$  is the minimal amount of time that must have elapsed for the laser scanner to have measured the same point on the surface again. Temporally close measurements are usually spatially close as well, so they must be excluded to prevent them from dominating the quality measure. By selecting only closest point pairs we filter out a very large number of pairs of points which do not correspond to the same feature in the environment. Thus, the geometry of the environment should have less effect on the quality metric. This also helps to prevent other negative effects like the collapse of points from two or more separate objects in the scene onto the same space to lead to a higher quality according to the metric. We seek to find

$$\hat{\mathbf{C}} = \underset{\mathbf{C}}{\operatorname{argmin}} E(f(\mathbf{M}, \mathbf{C})) \quad (3)$$

where

$$E(f(\mathbf{M}, \mathbf{C})) = - \sum_i^m G(\mathbf{p}_i - \mathbf{q}_i, 2\sigma^2\mathbf{I}). \quad (4)$$

Standard minimization algorithms require the computation of the derivative of the error function with respect to the calibration parameters. This is infeasible for the function  $E(f(\mathbf{M}, \mathbf{C}))$  as it involves complex algorithms for filtering data and fusing multiple modalities of measurements. We employ Powell's method for optimizing  $E$  as it does not require  $E'$  [8]. Instead, we must merely provide an initial estimate of  $\mathbf{C}$ , which is readily obtainable by manual estimation.

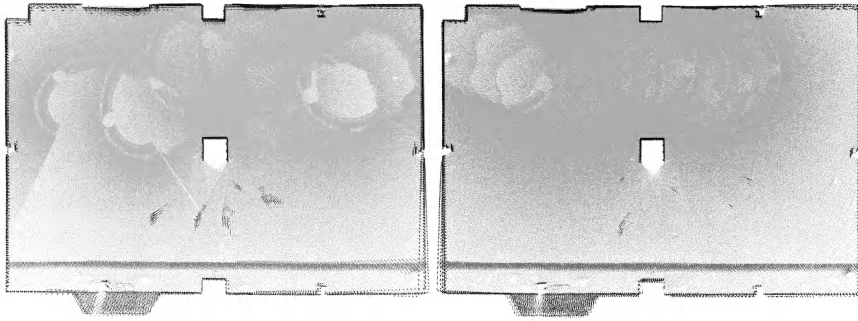
## 4 Experiments

Experimental data was acquired by the mobile robots seen in Fig. 1. The main sensor of each robot is the Riegl VZ-400 laser scanner. The scanner is the large cylinder mounted on the top of platforms. Irma3D is built of a Volksbot RT-3 chassis whereas Lars3D is based on a custom-built Volksbot chassis. Both mobile scanners utilize Olson's time synchronization algorithm [7] to reduce latency errors.



**Fig. 1** Left: The robots we used in our calibration experiments. From left to right: Achim3D, Lars3D and Irma3D. The red/silver cylinder on top of the robots is the Riegl VZ-400 laser scanner. Right: The geodetically measured model of the basement (colored by reflectance).





**Fig. 2** Topview on 2 point clouds acquired by our robots. The robots were calibrated with the proposed calibration algorithm. Although the scan quality is good, some non-calibration errors remain due to slipping wheels and erroneous position estimation.

Several data sets using both robots were acquired in an empty basement room (see Fig. 1). All data sets were acquired in continuous mode, i.e., the laser scanner rotates around its vertical axis while the robot moves simultaneously. The robots moved in “serpentine” trajectories, i.e., taking left and right turns as well as segments where the heading remains unchanged. In this paper we present 4 data sets, 2 for each robot with 3 to 7 million points each. For these data sets, the robot drove in a relatively slow manner, to ensure minimal wheel slippage.

For each configuration a manual estimation of the calibration parameters has been performed in advance of the experiments. We applied our automatic calibration technique to each data set. To evaluate the quality of the resulting point clouds we compare them with a high precision model of the room as acquired by the Riegl VZ-400 laser scanner. The accuracy of the scanner and thus the model is 5mm. We compare the acquired point clouds with the model in the following fashion. After calibration, the point cloud is matched to the model using ICP [3] from the 3D Toolkit (3DTK [1]). Then we compute point to plane distances on each of the 4 walls as well as the ceiling and floor of the room.

The point clouds obtained with our automatically determined calibration parameters are shown in Fig. 2. The results of the direct comparison between the point clouds after automatic and manual calibration and the model of the room are shown in Fig. 3-6.

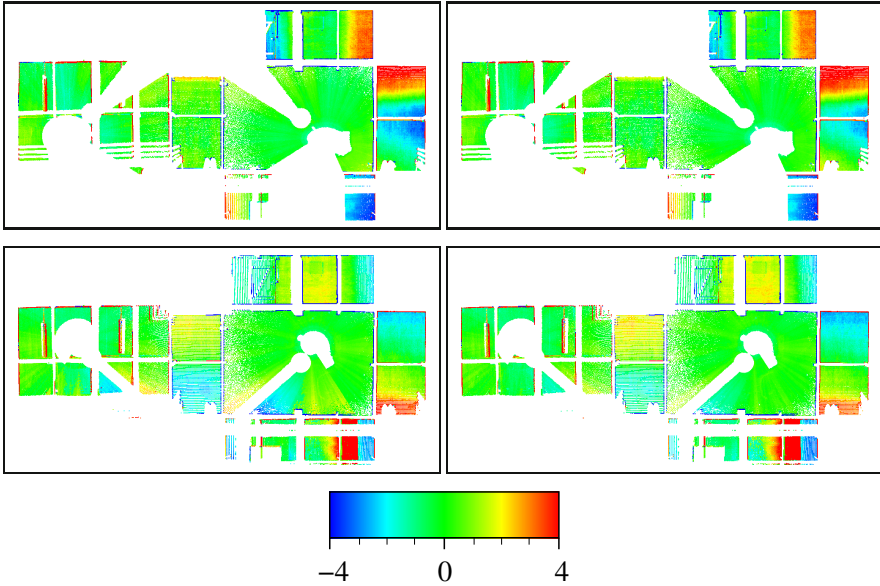
The deviations between model and point cloud are plotted in color coded images, i.e., green for absolute errors less than 1 cm, yellow to red for large positive errors and cyan to blue for large negative errors. White areas indicate that no point was measured at the corresponding location. The full color scale is also given in Fig.3.

On the whole, the quality of the scans improves remarkably with the automatically determined parameters when compared to the manual estimation. Absolute errors are generally within 1 cm. Occasionally the deviations exceed that boundary. Very rarely they are above 3 cm. These large scale errors are explained by uncertainties in the odometry measurements leading to erroneous estimations of the robot

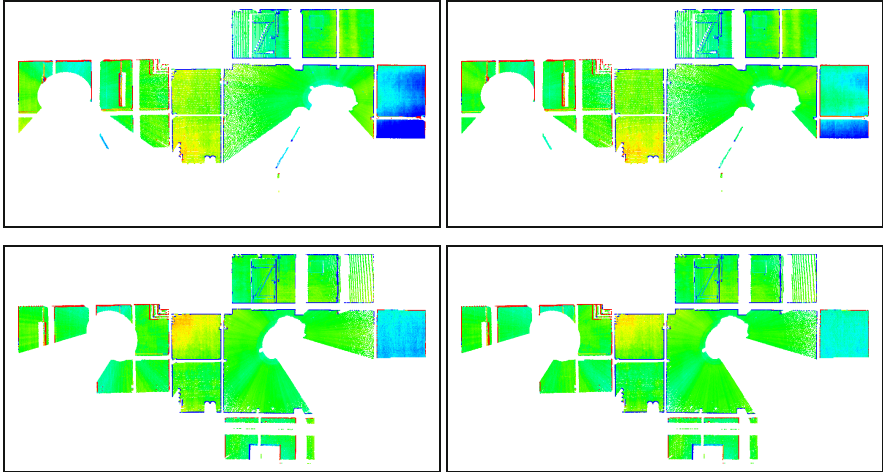


**Table 1** Deviations in cm per cm<sup>2</sup>

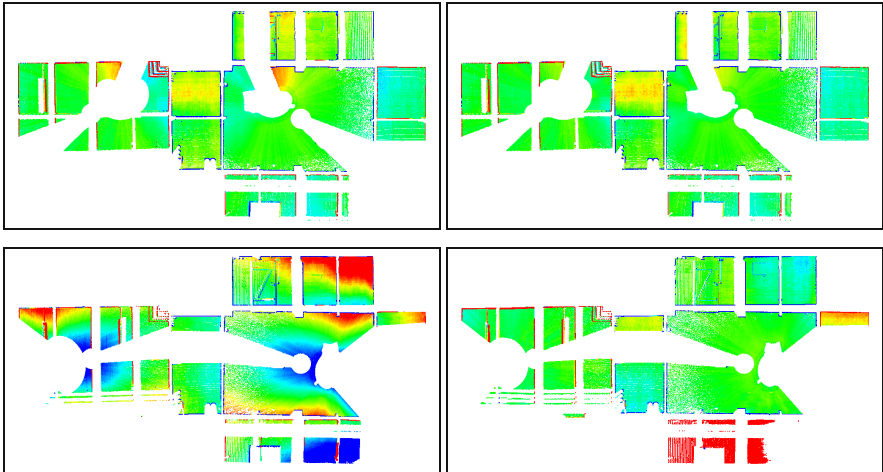
Robot	Data Set	manual	automatic
Irma3D	1	0.8517	0.8121
	2	0.9176	0.8744
	average	0.8920	0.8502
Lars3D	1	1.1161	0.9084
	2	1.1843	0.7878
	average	1.1730	0.8290
average		1.0346	0.8383

**Fig. 3** Comparison of the acquired laser scans with the model using the manual (left) and the automatic calibration (right) for representative excerpts of the first Irma3D data set. Deviations in cm are color coded as indicated on the bottom. Best viewed in color.

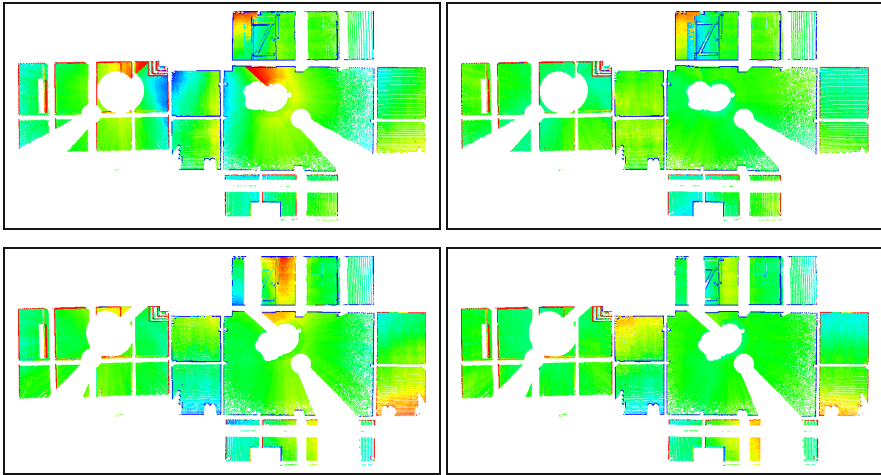
pose along the trajectory of the robot. Slipping wheels and shaking of the robot base are not detectable by any of the on-board sensors. Thus errors in the pose estimation carry over to erroneous point clouds and are not related to erroneous calibration parameters. The average deviations per cm<sup>2</sup> are collected in Table 1. For each data set the automatic calibration could successfully reduce the scan quality. The final error with the automatic calibration for both data sets and both robots are similar suggesting reliability. The combined mean error for all data sets was reduced by about 2 mm from 10.3 mm to 8.3 mm.



**Fig. 4** Comparison of the acquired laser scans with the model using the manual (left) and the automatic calibration (right) for representative excerpts of the second Irma3D data set. Deviations are color coded as indicated in Fig. 3.



**Fig. 5** Comparison of the acquired laser scans with the model using the manual (left) and the automatic calibration (right) for representative excerpts of the first Lars3D data set. Deviations are color coded as indicated in Fig. 3.



**Fig. 6** Comparison of the acquired laser scans with the model using the manual (top) and the automatic calibration (right) for representative excerpts of the second Lars3D data set. Deviations are color coded as indicated in Fig. 3.

## 5 Conclusions

The fully automatic calibration algorithm for mobile laser scanners as presented in this paper has proven to increase point cloud quality. It is capable of finding calibration parameters more precisely than manual estimation. The algorithm is robust against non-systematic errors that are necessarily present in any mobile laser scanning system, such as laser scanner noise and erroneous pose estimations due to slipping wheels. A minor issue of the algorithm is the dependency on the environment. In our experiments the robot primarily moved on a level surface. This means some calibration parameters, i.e., the upwards translation of the laser scanner in relation to the vehicle frame is less constrained by the quality measure as other parameters. In the future we plan to design an experiment with non-planar motion to estimate all calibration parameters more precisely.

## References

1. Nüchter, A., et al.: 3DTK – The 3D Toolkit (February 2011), <http://slam6d.sourceforge.net/>
2. Baziw, J., Leondes, C.: In-Flight Alignment and Calibration of Inertial Measurement Units. Part II: Experimental Results. *IEEE Transactions on Aerospace and Electronic Systems* 8(4), 450 (1972)
3. Besl, P., McKay, N.: A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2), 239–256 (1992)

4. Martinelli, A., Tomatis, N., Tapus, A., Siegwart, R.: Simultaneous Localization and Odometry Calibration for Mobile Robot. In: International Conference on Intelligent Robots and Systems, Las Vegas (2003)
5. Mills, D.L.: Internet Time Synchronization: The Network Time Protocol. *IEEE Transactions on Communications* 39, 1482–1493 (1991)
6. Nebot, E., Durrant-Whyte, H.: Initial Calibration and Alignment of Low-Cost Inertial Navigation Units for Land Vehicle Applications. *Journal of Robotic Systems* 16(2), 81–92 (1999)
7. Olson, E.: A Passive Solution to the Sensor Synchronization Problem. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (October 2010)
8. Powell, M.J.D.: An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. *The Computer Journal* 7(2), 155–162 (1964)
9. Rieger, P., Studnicka, N., Pfennigbauer, M.: BoreSight Alignment Method for Mobile Laser Scanning Systems. In: Proc. of the RSPRS Conference (2008)
10. Sheehan, M., Harrison, A., Newman, P.: Self-calibration for a 3d laser. *The International Journal of Robotics Research* (2011)
11. Skaloud, J., Schaer, P.: Towards Automated LiDAR BoreSight Self-calibration. In: MMS 2007, Padova (2007)
12. Talaya, J., Alamus, R., Bisch, E., Serra, A., Kornus, W., Baron, A.: Integration of a Terrestrial Laser Scanner with GPS/IMU Orientation Sensors. *International Archives of Photogrammetry and Remote Sensing* 5(17) (2004)
13. Underwood, J.P., Hill, A., Peynot, T., Scheduling, S.J.: Error Modeling and Calibration of Exteroceptive Sensors for Accurate Mapping Applications. *Journal of Field Robotics* 27(1), 2–20 (2009)

# Part XIV: ISER Session Summary on “Human Robot Interaction”

Alonzo Kelly

Carnegie Mellon University

## Session Summary

This session concentrated mostly on topics related to the problem of facilitating more effective communication between robots and humans.

“Hallucinating Humans for Learning Robotic Placement of Objects” by Yun Jiang, and Ashutosh Saxena considers the problem of how robots can be more adaptive to the needs of humans. By imagining likely poses for humans and using knowledge of how humans interact with specific objects like telephones (manipulate) and television (we look at them), reasonable placements for objects can be inferred. A robot so-endowed might be able to tidy up a room or put away the dishes, for example.

“Hand Shape Classification with a Wrist Contour Sensor” by Rui Fukui, Masahiko Watanabe, Masamichi Shimosaka, and Tomomasa Sato considers a unique sensor, a special wrist band, that can infer the configuration of the hand by exploiting the fact that tendons for each finger in the wrist change diameter as the fingers flex. A machine learning approach is used and the classic challenges of user independence and classification accuracy are addressed.

“Experimental Validation of Operator Aids for High Speed Vehicle Teleoperation.” by Alonzo Kelly, Nicholas Chan, Herman, Herman, Randy Warner considers the problem of teleoperating small (1 meter scale) mobile robots at relatively high speeds. Various operator aiding technologies are proposed and evaluated in a user study.

“Intention-Aware Pedestrian Avoidance” by Tirthankar Bandyopadhyay, Zhuang Jie Chong , David Hsu, Marcelo Ang, Daniela Rus, and Emilio Frazzoli addresses the need for cognitive models in order to predict the actions of pedestrians effectively in situations like traffic where pedestrians must be relatively close (i.e on the sidewalk) to fast vehicles. Intentions (e.g. intended destinations) are an effective way to do this. An efficient factored Markov decision process formulation is shown to be an efficient approach.

“The UBC Visual Robot Survey: A benchmark for robot category recognition” by David Meger, James Little proposes a collection of data sets containing both depth and appearance information. The data set is specialized for certain computer vision tasks that are relevant to human-inhabited environments and it supports a particular form of “simulation using data” that is can be very effective.

# Hallucinating Humans for Learning Robotic Placement of Objects

Yun Jiang and Ashutosh Saxena

**Abstract.** While a significant body of work has been done on grasping objects, there is little prior work on placing and arranging objects in the environment. In this work, we consider placing multiple objects in complex placing areas, where neither the object nor the placing area may have been seen by the robot before. Specifically, the placements should not only be stable, but should also follow human usage preferences. We present learning and inference algorithms that consider these aspects in placing. In detail, given a set of 3D scenes containing objects, our method, based on Dirichlet process mixture models, samples human poses in each scene and learns how objects relate to those human poses. Then given a new room, our algorithm is able to select meaningful human poses and use them to determine where to place new objects. We evaluate our approach on a variety of scenes in simulation, as well as on robotic experiments.

## 1 Introduction

“*Tidy my room.*” “*Put the dishes away.*” — While these tasks would have been easy for *Rosie* robot from *The Jetsons* TV show, they are quite challenging for our robots to perform. Not only would they need to have the basic manipulation skills of picking up and placing objects, but they would also have to perform them in a way that respects human preferences, such as not placing a laptop in a dish-rack or placing the dishes under the bed.

Over the last few decades, there has been a significant body of work on robotic grasping of objects (e.g., [1–10]). However, there is little previous work on teaching robots where and how to *place* the objects after picking them up. Placing an object is challenging for a robot because of the following reasons:

---

Yun Jiang · Ashutosh Saxena  
Department of Computer Science,  
Cornell University, Ithaca, NY 14853, USA  
e-mail: {yunjiang, asaxena}@cs.cornell.edu

- *Stability*. An object needs to be placed in a correct orientation for it to be stable. For example, while a martini glass could be placed upright on a flat surface, it is stable when hanging upside down in a wine glass rack.
- *Novel objects and placing areas*. An unstructured human environment comprises a large number of objects and placing areas, both of which may have complex geometry and may not have been seen by the robot before. Inferring stable placements in such situation requires robust algorithms that generalize well.
- *Human preferences*. The objects should be placed in meaningful locations and orientations that follow human preferences. For example, a laptop should be facing the chair when placed on a table.

In our recent work [11, 12], we proposed a learning algorithm for placing objects stably and in their preferred orientations in a given placing area (see Fig. 1b). Our approach was based on supervised learning that used a graphical model to learn a functional mapping from features extracted from 3D point-clouds to a placement’s quality score. When multiple objects and placing areas were present, the inference was formulated as an integer linear program that maximized the total placement quality score. Our formulation also allowed linear stacking of objects. This enabled our robot to place objects (even the ones that were not seen previously by the robot) in areas such as dish-racks, a stemware-holder and a fridge, etc. While our model in [12] captured certain semantic preferences, it did not consider human usage preferences and therefore placements were often not meaningful. For example, placing food up in a fridge that is hard to reach (see Fig. 1b), or placing a mouse and keyboard far away from each other making them impossible to be used together.

In this work, our goal is to learn meaningful object placements that follow human preferences, such as the arrangement in Fig. 1c. The key idea that makes a placement meaningful is how it will be used by humans: Every object is meant to be used by a human in a certain way, at a certain location and for a certain activity. For example, in an office, a keyboard is placed on a table below a monitor because a person typically uses the keyboard while sitting in the chair and watching the monitor. Such usage preferences are sometimes also called object “affordances” [13]. One naïve way to encode them would be looking up a dataset that shows examples of humans using each object. Unfortunately, no such dataset exists and the effort to construct a comprehensive one would be prohibitive.

Instead of relying on a dataset of real humans manipulating objects in 3D environments, we work with a dataset that only has arrangements of objects in different scenes.<sup>1</sup> Then, in order to learn the human usage preferences, we would ‘*hallucinate*’ human poses in the 3D scene, and learn the object affordances using an unsupervised learning algorithm.

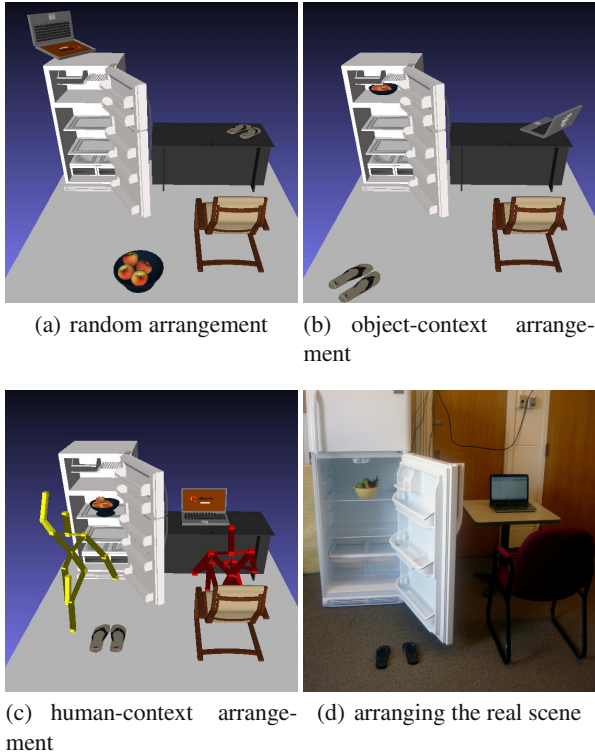
*A hallucination is a fact, not an error; what is erroneous is a judgment based upon it.*

Bertrand Russell.

What would be the key here is to learn which human poses are more likely than others and how they interact with the objects. To do this, we first define a potential

---

<sup>1</sup> Such datasets are readily available on the Internet, e.g., Google 3D warehouse.



**Fig. 1** An example of arranging three objects in a room. (a) A random arrangement may lead to unstable placement (such as the laptop tilted on the fridge) and stable but unreasonable placement (such as shoes on the table and food on the ground). (b) Therefore, we adopt supervised learning based on a variety of appearance and shape features for finding stable orientations and meaningful locations for placing the objects. However, the placed objects are still hard for human to access, such as the fruit stored high up on the fridge and the laptop towards the wall. (c) We thus further improve the arrangement by considering the relationship between the objects and humans (such as a sitting pose in the chair and a reaching pose in front of the fridge). (d) Our approach does not require human poses at present but samples them based on Dirichlet processes and learned potential functions. The last arrangement not only places the objects appropriately but is also ready for human to use.

function giving a score for an object and a human pose, based on their spatial features. We consider the human poses as latent variables, and model them as mixture components in a Dirichlet process (DP) mixture model and consider arranging the objects as a generative process: a room first generates several human poses; then each object chooses a human pose and is drawn from the potential function parameterized by this human pose. This model allows different objects to be used by the same human pose (e.g., using a monitor, keyboard and mouse at the same time), while a room can have as many human poses as needed (one of the DP mixture



model's property) [13]. Given the most likely placements, our robot then uses path planning algorithms to compute specific placing trajectories and execute them.

Our algorithm thus learns the preferred object arrangements from the 3D scenes collected from the Internet. We first evaluate our algorithm on such datasets consisting of 20 different rooms and compare the inferred arrangements to the ground truth. We also test on five scenes using real point-clouds. Finally, we perform our algorithms on our robot on actual placements in three real scenarios.

## 2 Related Work

There is little work in robotic placing and arrangement of objects and most existing methods are restricted to placing (or moving) objects on flat horizontal surfaces. Edsinger and Kemp [14] and Schuster et al. [15] focused on finding flat clutter-free areas where an object could be placed. Our work considers arranging objects in the whole room with significantly more complex placing areas in terms of geometry.

Placing objects also requires planning and high-level reasoning about the sequence of actions to be performed. Lozano-Perez [16] proposed a task-level (in contrast with motion-level) planning system for picking and placing objects on a table. Sugie et al. [17] used rule-based planning in order to push objects on a table surface. There are some recent works using symbolic reasoning engines to plan complex manipulations for human activities, such as setting a dinner table (e.g. [18–20]). However, these works focus on generating parameterized actions and task-level plans instead of finding specific placements, and hence are complementary to ours.

In our own recent work [11, 12], we employed 3D stability and geometric features to find stable and preferred placements. However, without taking human context into consideration, the generated strategy was often not good enough. In this paper, we discuss a method for combining the stability with human usage preference, and compare our approach to one that does not consider the human usage preferences in Section 4.

In this paper, learning the human usage preference, i.e., the relationship between the objects and the humans is the key. In a way, this could be called 'human context.' In other fields, such as computer vision, the idea of 'context' has helped quite a bit in improving tasks such as object recognition. For example, using estimated 3D geometric properties from images can be useful for object detection [21–26]. In [27–29], contextual information was employed to estimate semantic labels in 3D point-clouds of indoor environments. Fisher et. al. [30, 31] designed a context-based search engine using geometric cues and spatial relationships to find the proper object for a given scene. Unlike our work, their goal was only to retrieve the object but not to place it afterwards. These works are different from ours not only because they address different problems, but also because none of these works used the 'human context.'

We use sampling techniques to sample the human poses, which are never observed. In general, sampling techniques are quite common in the area of path planning [32, 33], where it is the robot pose that is sampled for constructing a path.

Often modeling and sampling of human poses is also done in the area of computer graphics and animation [34], and solving the kinematics and dynamics issues of robots operating in presence of humans [35], and analyzing human body poses and actions [36–38]. However, to the best of our knowledge, our work is the first one that samples such human poses for capturing context among objects.

### 3 Object and Human Context

An object when placed in an environment depends both on its interaction with the placing area and its interaction with the humans. In the following sections, we first briefly review our potential function that captures the object context—relationship between the object and placing areas [11, 12]. Then we discuss how to encode human context (such as human usage preferences and access effort) in our algorithm.

Specifically, we formulate a general placing problem as follows: There are  $n$  objects  $\mathcal{O} = \{O_1, \dots, O_n\}$  to be placed in  $m$  placing areas  $\mathcal{E} = \{E_1, \dots, E_m\}$ , all of which are represented as point-clouds. A placement of  $O_i$  is specified by its location  $\ell_i$  and orientation/configuration  $c_i$ . Moreover, a placement is often associated with certain human pose for certain purpose. Let  $\mathcal{H} = \{H_1, \dots\}$  to denote all the possible human poses. Our goal is to, for each object  $O_i$ , find 1) a placing area  $E_j$  to place it at and the specific placement  $(\ell_i, c_i)$ , and 2) a relevant human pose  $H_k$  that explains the placement well.

#### 3.1 Object Context

By object context, we mean the relationship/interaction between the object and the placing area that determines whether the placing area can hold the object stably and, more importantly, meaningfully. For instance, books should be placed on a shelf or a table, plates are better inserted in a dish-rack, and shoes should be put on the ground instead of on a table or in a dishrack.

We capture this object-environment relationship (or object-object relationship when the objects are stacked on top of each other) using a supervised learning algorithm that learns a functional mapping,  $\Psi_{\text{object}}(O_i, E_j, \ell_i, c_i)$ , from a set of features representing the placement to a placing quality score. A larger value of  $\Psi_{\text{object}}(\cdot)$  indicates a better placement. (Our goal then becomes to maximize the value of this function during learning and inference.)

We decompose the function into two terms:

$$\Psi_{\text{object}}(O_i, E_j, \ell_i, c_i) = \Psi_{\text{stability}}(O_i, E_j, \ell_i, c_i) \Psi_{\text{semantics}}(O_i, E_j). \quad (1)$$

We develop a variety of appearance and shape features to capture the stability and semantic preferences respectively [12].

As we observed in a series of experiments in [12], using this algorithm can help us to predict preferred placements for various objects and different scenes. However, because we model each object independently of others, certain connections among

the objects are lost in this approach, making the arrangement often disorganized and pointless. For example, a keyboard and mouse are placed far away from each other and the desk-light faces towards the wall. The goal of capturing the connection between different objects making them usable after placing is the key motivation for introducing the human context.

### 3.2 Human Context

The arrangements and connections among objects can be naturally explained by human poses and activities. For example, a monitor on the desk would mean that a human skeleton may be in front of it in a sitting pose. Then the sitting skeleton could further suggest to place a mouse and a keyboard close to the hand and therefore at the edge of the desk. Although the human poses are not present during the arrangement, *hallucinating* them would help the robots to place objects in a human-friendly way.

We consider an arrangement as an outcome of the following generative process: A scene generates a set of possible human poses in the scene based on certain criteria (such as reachability or usage of existing objects); then use the human poses to determine where to place the new objects. There are two components required for this approach: (a) modeling how the objects relate to human poses based on criteria such as their affordances, ease of use and reachability, and (b) learning a distribution of human poses in the scene.

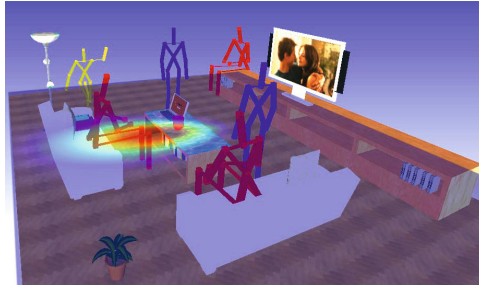
We first capture the object’s usage preferences by a potential function that models how a human pose  $H_k$  is related to an object  $O_i$ . (Again, a higher value of this function indicates a better match between an object and the human pose.)

$$\Psi_{\text{human}}(O_i, H_k, l_i, c_i) = \Psi_{\text{loc}}(O_i, H_k, l_i) \Psi_{\text{ori}}(O_i, H_k, c_i). \quad (2)$$

Here,  $\Psi_{\text{loc}}$  and  $\Psi_{\text{ori}}$  represent the preferences in the relative location and the orientation of the object from a human pose respectively. For example, a TV has a usage score that is high in the front at a certain distance and falls off as you go to the side (see the projected heat map in Fig. 2). The potential function also indicates more meaningful and relevant human poses from the innumerable possible poses in an environment.

For instance, in a room such as the one shown in Fig. 2, the reaching pose (in yellow) and the sitting pose on the TV stand are less important because they do not relate to any object, while a sitting pose on the couch is important because it has high scores with several objects in the scene—the cushion, laptop, TV, etc.

Note that the human poses in our problem are latent, and therefore we model them using a mixture model. The model comprises an infinite number of human poses and each object selects a human pose according to a mixture proportion  $\pi$ . As a result, an object is affected by multiple human poses. For example, in Fig. 2, the TV’s location is determined by all the human poses (sitting on the couch or next to the TV, standing to the coffee table and so on). However, since the one on the couch is more important than others, its corresponding proportion defined in  $\pi$  will be higher and thus put more influence upon the TV. After considering all possible



**Fig. 2** While there could be innumerable possible human poses in a room, only a few of them are meaningful, such as the one on the couch who is related to many objects. We sample human poses based on the objects’ affordances. For example, the learned potential function for the TV (shown in the projected heat map) has high values in the front with certain distance and thus the sitting human pose is sampled with high probability.

human poses (i.e., marginalizing out  $H_k$  and  $\pi$  in the mixture model), we define the likelihood of an arrangement,  $\mathcal{O}$ , of  $n$  objects (in human context) as<sup>2</sup>

$$p(\mathcal{O}) = \int_{\pi} p(\pi) \prod_{i=1}^n \sum_{k=1}^{\infty} \left( p(O_i | H_k) P_0(H_k) \pi_k \right) d\pi, \quad (3)$$

where  $P_0$  is the prior of human poses, and  $P(O_i | H_k) \propto \Psi_{\text{human}}(O_i, H_k)$ . We adopt DP mixture model so that  $\pi$  can have unbounded length and be constructed using stick-breaking processes [39].

The inference problem is to find  $\mathcal{O}$  with the maximum likelihood. Although (3) is intractable to compute, it can be approximated using a sampling scheme. We use Gibbs sampling with auxiliary parameters [40], where in each round we sample which human pose to select for each object, the object placements and the human poses according to their conditional distribution (see [13] for more details).

To differentiate the preference in selecting human poses for different types of objects, we add type-specific parameters  $\Theta$  in the potential function and learn them from the labeled data. During training, given the objects in the scenes, we learn the parameters using the maximum likelihood estimation based on human poses sampled from the DP. In detail, we use human poses sampled from a DP, denoted by  $H^1, \dots, H^s$  as our observations. The optimal  $\Theta$  is then computed by solving the following optimization problem:

$$\Theta^* = \arg \max_{\Theta} \sum_{\text{scenes}} \sum_{j=1}^s \sum_{i=1}^n \log \Psi_{\text{human}}(O_i, H_i^j; \Theta). \quad (4)$$

<sup>2</sup> We abuse the notation  $O_i$  in this section to indicate the object’s placement, including  $\ell_i$  and  $c_i$ .

In our previous work [11, 12], we considered only the object context  $\Psi_{\text{object}}(\cdot)$ . In this current work, we primarily use the human context  $\Psi_{\text{human}}(\cdot)$ , and only combine it with some of the object context defined heuristically (see [13]). Jointly learning both the object and human context is an interesting direction for future work.

Once we have obtained the likelihood of the arrangements  $p(\mathcal{O})$ , we need to perform planning to realize the desired placements.

### 3.3 Planning

After finding the potential object placements that have high scores, the robot still faces two challenges in realizing the placing: First, high-scored placements may not be reachable/executable by the robot due to its kinematic constraints; Second, placing certain objects first may impede placing other objects later. Thus, the order in which the objects are placed becomes important and we need to find a valid placing order and target locations efficiently.

We address the first challenge by filtering out the placements that are not physically realizable by the robot due to its kinematic constraints (while considering placement of each object independently of the others).

For the second challenge, we adopt the classic backtracking search for finding a valid placing sequence. Particularly, in each search step, given the already-placed objects  $\mathcal{P}$ , we need to de-

termine which object to be placed next (indexed by  $i$ ) and also where to place it (denoted by  $O_i$ ). While this search space is enormous, we can cut the redundancy using the following fact: *Any superset of an infeasible plan is also infeasible*. We maintain a set of all infeasible plans encountered so far, denoted by  $\mathcal{F}$  (see Algorithm 1). Before trying to place a new object  $O_i$ , we check that if  $\mathcal{P} \cup O_i$  becomes a superset of any elements in  $\mathcal{F}$ . Only if not, a path planning algorithm (in our case, rBiRRT in OpenRAVE [41]) is then used to verify the validity of placing at  $O_i$  and the search continues for other objects.

## 4 Experiments

We perform three experiments as follows. First, we verify our human-context learning algorithm in arranging 20 different rooms, represented as 3D models. Second, we compare the object context and human context in different scenes in real point-clouds. Third, we perform robotic experiments on our Kodiak (PR2) robot based on the learned arrangements.

---

#### Algorithm 1. TryPlace( $\mathcal{P}$ , ObjNotPlaced)

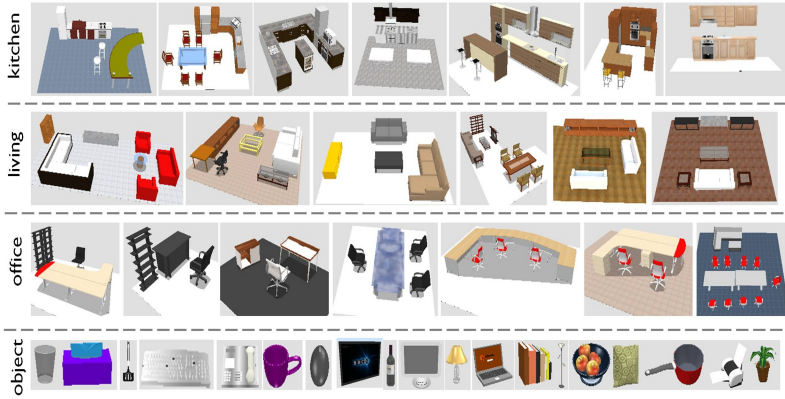
---

```

1 if ObjNotPlaced =  $\emptyset$  then
2   | Succeed
3 for  $i \in \text{ObjNotPlaced}$  do
4   for  $O_i \in \text{PossiblePlacements}_i$  do
5     if IsSuperSetOf( $\mathcal{P} \cup O_i, \mathcal{F}$ ) then
6       | continue
7     if feasible( $O_i, \mathcal{P}$ ) then
8       | TryPlace( $\mathcal{P} \cup O_i, \text{ObjNotPlaced} \setminus O_i$ )
9  $\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathcal{P}\}$ 

```

---



**Fig. 3** Our dataset contains 20 scenes (7 kitchens, 6 living rooms and 7 offices) and 47 objects from 19 categories that are commonly seen in these scenes [13]

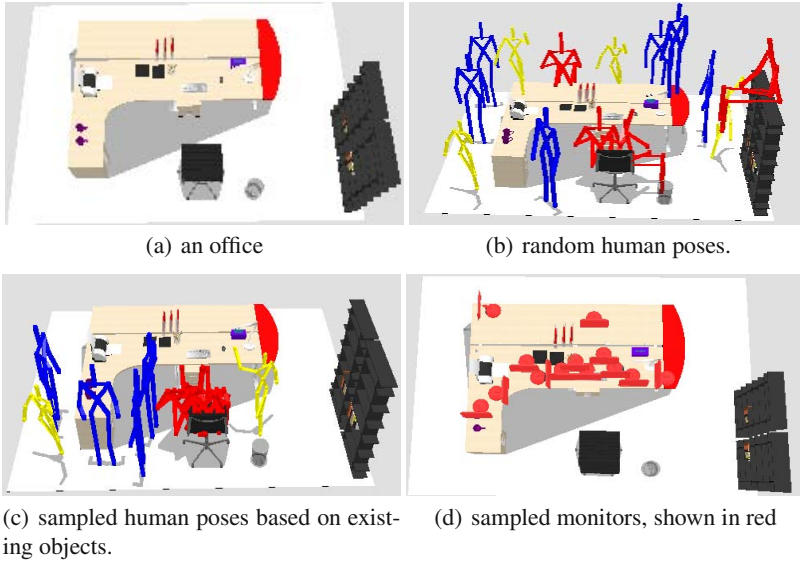
### 4.1 Arranging Rooms under Human Context

In order to verify that our DP-based learning algorithm can generate reasonable human poses as well as object placements, we evaluated it on a dataset containing 20 scenes from three categories (living room, kitchen and office) and of 47 daily objects from 19 types (listed in Fig. 6) such as dish-ware, fruit, computers, desk-lights, etc. [13]. Fig. 3 shows a snapshot of our dataset. Some example good arrangements of each room were labeled by three to five subjects (not associated with the project).

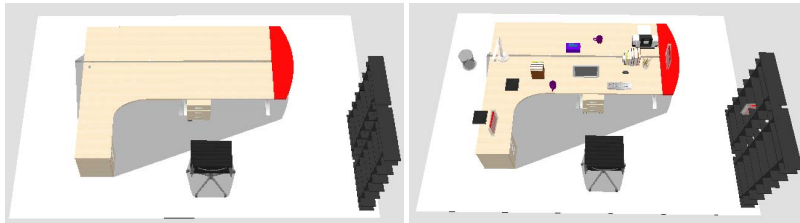
We conduct 5-fold cross validation on 20 rooms so that the test rooms have never been seen by the algorithm. We consider two placing scenarios: placing objects in filled rooms and empty rooms. In the first case, the task is to place one type of objects while other types are given (placed). In the second case, no object is in the test rooms at all.

Figure 4 shows an example of our algorithm inferring meaningful human poses and object placements. Given an office such as Fig. 4(a), if we randomly sample human poses regardless the existing objects, then many unreasonable human poses appear. For example, in Fig. 4(b), we have standing poses (in blue) oriented randomly and some sitting poses (in red) at absurd locations such as on top of the table and book shelf and reaching poses (in yellow) on the table as well. However, if we sample human poses based on the learned potential function (2), then we obtain human poses in meaningful places such as sitting in the chair or standing close to the object (see Fig. 4c). Note that now the distribution of both location and orientation of human poses has changed due to the  $\Psi_{loc}$  and  $\Psi_{ori}$  terms in the potential function.

We then sample the monitor’s location according to these human poses. Figure 4d shows that the distribution is biased towards the inner side of the L-desk, especially concentrated in front of the chair. This is because that sitting poses are more related to monitors. Moreover, the preference of monitor placed on the table (as compared with being placed on the ground) is naturally learned through our human access effort rather than hand-script rules. Another interesting observation is that



**Fig. 4** Sampling results of a test room. Given an office as (a), sampling human poses randomly results in several poses at absurd locations, such as on the top of the shelf in (b). Our algorithm, on the other hand, samples more relevant human poses in (c) and thus is able to sample more locations for placing a monitor in front of the chair than other places in (d).

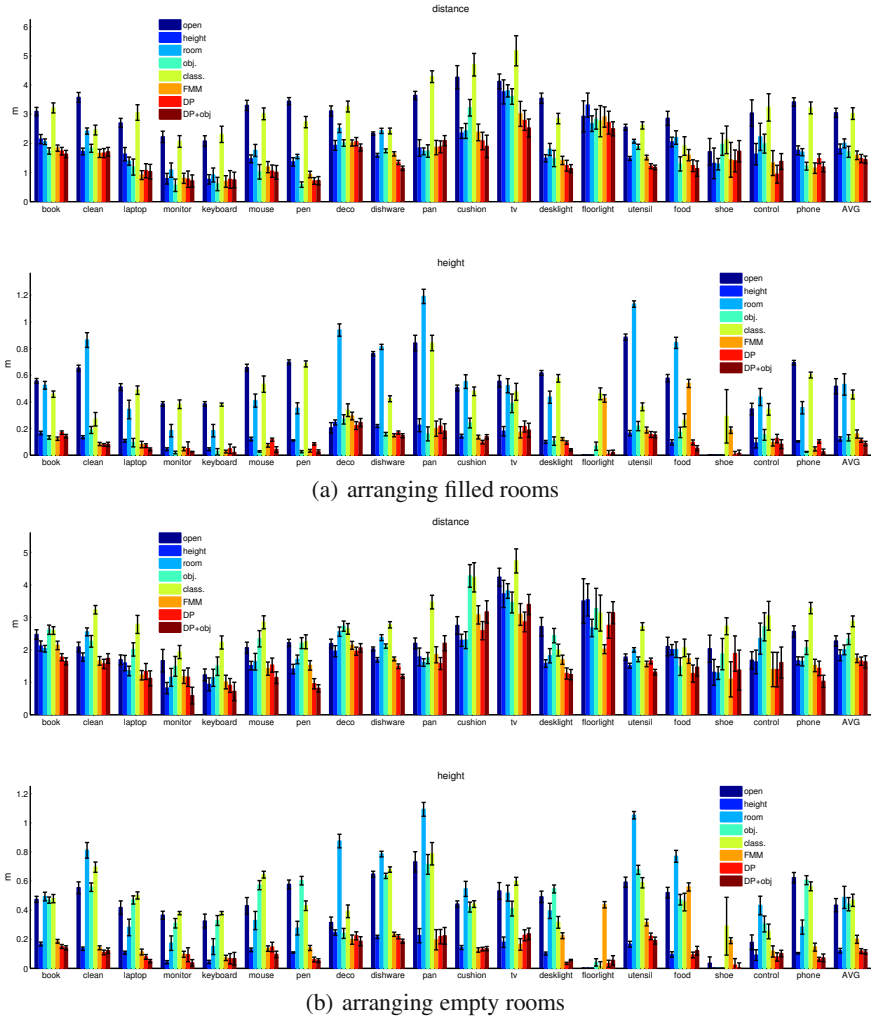


**Fig. 5** Predicted arrangement for 17 objects in an empty rooms. Using our method, all the objects are correctly placed in their preferred areas, such as the trashcan on the ground and the books on the table. Some object-object relationships are also captured without modeling them explicitly, such as the monitor being placed close to the keyboard and their relative orientation to the chair.

most samples are near the keyboard. This shows that the monitor-keyboard relationship can be linked through human poses naturally, without needing to explicitly model it.

Fig. 5 shows one sampled arrangements when placing in an empty room. Although the monitor and keyboard are not perfectly aligned, they are still placed roughly in front of the chair, with correct orientations. All the objects are placed in the correct placing areas, such as trashcan on the ground and the desk-light on the table. The trashcan being far from the chair is mainly due to some sampled human poses around that location.





**Fig. 6** Results of arranging filled rooms (top) and empty rooms (bottom), evaluated by the difference in location and height in meters. The error bar shows one standard error.

We now give the quantitative results on the whole dataset in Fig. 6. Arrangements are evaluated by two metrics: difference in *location* and *height* between the prediction and the ground-truth. We compare our method with six baselines [13], including using object context ('obj'). We additionally present another algorithm in which we combine the distribution of objects generated through human poses  $O \propto \Psi_{\text{human}}(O, H; \Theta)$  with a distribution generated through object - object context  $O \propto \Psi_{\text{obj}}(O, \mathcal{G})$  ( $\mathcal{G}$  is the set of given objects) using a mixture model:  $O \propto \omega \Psi_{\text{human}}(\cdot) + (1 - \omega) \Psi_{\text{obj}}(\cdot)$ . We give a comparison of methods of using object context only, human context only and their combination in our experiments.



In the task of arranging filled rooms (shown in Fig. 6a), using object context (‘obj’) benefited from the strong spatial relationships among objects and hence beat other baseline methods, especially for the laptop, monitor, keyboard and mouse types. However, our methods based on human context (last three bars) still outperformed the object context. They significantly improved placements of the objects that have weaker connection to others, such as book, TV, decoration and shoes.

The task of arranging objects in an empty room (Fig. 6b) raises many challenges when placing the first few objects as no object context would be available. Not surprisingly, we found that the object-context method performed poorly, even worse than using just height as a reference (‘height’). Although our methods also performed worse than the previous scenario, they could still sample human poses based on the furniture in the scene and thus predicted better locations for objects. Our experiments also showed that the finite mixture model using human context (‘FMM’) performed better than other baselines, but not as well as the ones our method using DPs.

In both tasks, our human-context algorithm successfully predicted object placements within 1.6 meters on average. The average error in height was only 0.1 meters. By combining human- and object-context, the error was further reduced—indicating that they provide some complementary context.

**Robotic Simulation Experiment.** In order to study how the desired placements are affected by the robot constraints (see Section 3.3), we tested arranging these synthetic scenes using Kodiak (PR2) in simulation. Table 1 shows that the location errors increase only slightly for arranging filled rooms as well as empty rooms, but the errors in height increase significantly. This is mostly because of the kinematic constraints of the robot. How to incorporate robotic constraints into our current score function is an interesting direction for future work.

**Table 1** Comparison between the predicted placements with and without the robotic constraints (verified in simulation). Unit is meters. Note that only those objects that are physically movable by the robot are considered.

		book	clean tool	mouse	pen	dish- ware	pan	cush- ion	uten- sil	food	shoe	re- mote	AVG
arranging filled rooms													
location	without constraints	1.63	1.71	1.00	0.72	1.15	2.09	1.90	1.17	1.13	1.73	1.38	1.42
	with constraints	1.87	2.26	0.96	0.69	1.63	2.45	2.38	1.03	1.34	2.01	1.24	1.62
height	without constraints	0.14	0.08	0.04	0.03	0.15	0.18	0.14	0.15	0.05	0.01	0.08	0.10
	with constraints	0.32	0.52	0.14	0.17	0.27	0.42	0.33	0.31	0.17	0.18	0.22	0.28
arranging empty rooms													
location	without constraints	1.65	1.74	1.15	0.82	1.19	2.21	3.17	1.32	1.47	1.38	1.61	1.61
	with constraints	1.97	2.31	1.51	1.22	1.89	2.34	3.01	1.89	1.55	1.55	1.72	1.91
height	without constraints	0.14	0.12	0.10	0.05	0.19	0.22	0.14	0.19	0.12	0.00	0.10	0.13
	with constraints	0.36	0.59	0.18	0.21	0.39	0.44	0.35	0.33	0.47	0.19	0.26	0.34

## 4.2 Arranging Real Scenes

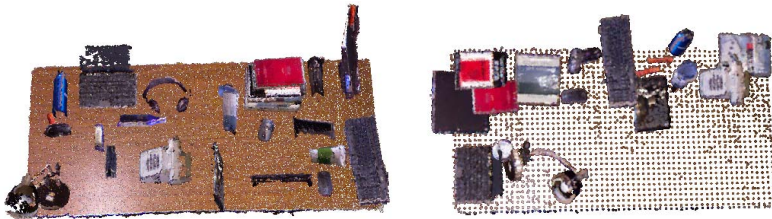
In this experiment, we compare the two algorithms—using object context [12] and using human context [13]—in arranging real scenes. The dataset [12] contains 3 different offices and 2 different apartments where the placing areas such as tables, cabinets, floor, and drawers were segmented out. We evaluated the quality of the final placing layout by asking two human subjects (one male and one female, not associated with the project) to label placement for each object semantically correct or not, and also report a qualitative metric score on how good the overall placing was (0 to 5 scale).

Results are shown in Table 2. Both methods arranged objects more meaningfully than heuristic rules (not reported here, see [13]), i.e., books were stacked together, while a keyboard, laptop and mouse were placed close to each other. The human context, however, performed much better by, for example, placing shoes at the bottom level of a shelf, while food and books are on the middle level or on a table. The approach of using object context only sometimes put the laptop on a shelf making it difficult for human to access.

Fig. 7 shows a comparison in arranging office2. Compared to using object context, human context links the mouses and keyboard together as well as lamp and the laptop. The laptop is now at the edge of the table and thus becomes accessible for humans. Other objects are all close to the edge, unlike objects scattered uniformly in the left figure making the bottle and mouse in the center hard to reach.

**Table 2** Results on arranging five real point-cloud scenes (3 offices & 2 apartments). The number of objects for placing are 4, 18, 18, 21 and 18 in each scene respectively. **Co**: % of semantically correct placements, **Sc**: average score (0-5).

	office1		office2		office3		apt1		apt2		Average	
	Co	Sc	Co	Sc	Co	Sc	Co	Sc	Co	Sc	Co	Sc
obj context [12]	100	4.5	100	4.2	87	3.5	65	3.2	75	3.0	85	3.7
Human context (FMM)	100	3.5	100	2.0	83	3.8	63	3.5	63	3.0	82	3.2
Human context (DP)	100	<b>5.0</b>	100	4.3	91	4.0	74	3.5	<b>88</b>	<b>4.3</b>	90	4.2
Human (DP) + obj context	100	4.8	100	<b>4.5</b>	<b>92</b>	<b>4.5</b>	<b>89</b>	<b>4.1</b>	81	3.5	<b>92</b>	<b>4.3</b>



**Fig. 7** Results of placing multiple objects on an office desk, when considering object context [12] (left) and considering human context (right). While objects are scattered in the left arrangement, the right arrangement prefers placing objects at the edge of the desk for easy access.



Fig. 8 Our Kodiak robot arranging several objects in three different scenarios

### 4.3 Robotic Experiments

We verified our approach on our Kodiak (PR2) robot in three scenarios: 1) placing five objects (a beer bottle, cup, soda can, hand torch and shoe) in a kitchen with a fridge and a table; 2) placing six objects (a mouse, a pen, a trashbin and three books) in an office with a table and a bookshelf; 3) placing five objects (a cup, tissue box, book, soda can and throw pillow) in a living room with a couch and a coffee table.

Given the predicted arrangements, the robot uses a *pre-determined grasp* to pick up every object, and executes the plan (see Section 3.3) for moving the object to its designated location. Fig. 8 shows some screenshots of our robot performing the object arrangements. We found that all the objects were placed at the locations consistent with the simulation experiments. For the videos, see <http://pr.cs.cornell.edu/placingobjects/>.

There were certain failures however caused by the limitation of our learning algorithm. For example, the beer bottle was placed on the couch instead of the table. This was because the physical properties of the surfaces (e.g., hard vs soft) are not explicitly modeled. This may potentially be avoided by including semantic information or appearance features of the furniture.

## 5 Discussion and Conclusions

We considered arranging multiple objects in complex placing areas, while following human usage preferences. Motivated by the fact that objects are often arranged for certain human activities, we developed an approach based on sampling meaningful latent human poses and using them to determine objects' placements. In detail, we

designed a potential function for capturing the human-object relationship and used Dirichlet processes to sample human poses and placements jointly. We verified our approach on a variety of scenes in simulation as well as on a real robot.

In this work, we have focussed on learning the object arrangements from a human usage perspective. We believe that integrating the object detection, grasping and placing jointly is a challenging direction for future work. Furthermore, one can also potentially incorporate control and planning into our model in order to obtain placements that are easily executed by the robot.

## References

1. Cutkosky, M.R.: *Robotic Grasping and Fine Manipulation*. Kluwer Academic Publishers, Norwell (1985)
2. Salganicoff, M., Ungar, L.H., Bajcsy, R.: Active learning for vision-based robot grasping. *Machine Learning* 23, 251–278 (1996)
3. Bicchi, A., Kumar, V.: Robotic grasping and contact: A review. In: *ICRA*, pp. 348–353 (2000)
4. Hsiao, K., Nangeroni, P., Huber, M., Saxena, A., Ng, A.Y.: Reactive grasping using optical proximity sensors. In: *ICRA* (2009)
5. Rodriguez, A., Mason, M., Ferry, S.: From caging to grasping. In: *Proceedings of Robotics: Science and Systems* (June 2011)
6. Saxena, A., Driemeyer, J., Kearns, J., Ng, A.: Robotic grasping of novel objects. In: *Neural Information Processing Systems* (2006)
7. Saxena, A., Driemeyer, J., Ng, A.: Robotic grasping of novel objects using vision. *The International Journal of Robotics Research* 27(2), 157 (2008)
8. Saxena, A., Wong, L., Ng, A.Y.: Learning grasp strategies with partial shape information. In: *AAAI* (2008)
9. Le, Q., Kamm, D., Kara, A., Ng, A.: Learning to grasp objects with multiple contact points. In: *ICRA* (2010)
10. Jiang, Y., Moseson, S., Saxena, A.: Efficient grasping from rgb-d images: Learning using a new rectangle representation. In: *ICRA* (2011)
11. Jiang, Y., Zheng, C., Lim, M., Saxena, A.: Learning to place new objects. In: *ICRA* (2012)
12. Jiang, Y., Lim, M., Zheng, C., Saxena, A.: Learning to place new objects in a scene. *The International Journal of Robotics Research, IJRR* (2012)
13. Jiang, Y., Lim, M., Saxena, A.: Learning object arrangements in 3d scenes using human context. In: *ICML* (2012)
14. Edsinger, A., Kemp, C.: Manipulation in human environments. In: *Int'l. Conf. Humanoid Robots* (2006)
15. Schuster, M., Okerman, J., Nguyen, H., Rehg, J., Kemp, C.: Perceiving clutter and surfaces for object placement in indoor environments. In: *Int'l. Conf. Humanoid Robots* (2010)
16. Lozano-Pérez, T., Jones, J., Mazer, E., O'Donnell, P.: Task-level planning of pick-and-place robot motions. *Computer* 22(3), 21–29 (2002)
17. Sugie, H., Inagaki, Y., Ono, S., Aisu, H., Unemi, T.: Placing objects with multiple mobile robots-mutual help using intention inference. In: *ICRA* (1995)

18. Jain, D., Mosenlechner, L., Beetz, M.: Equipping robot control programs with first-order probabilistic reasoning capabilities. In: ICRA (2009)
19. Mösenlechner, L., Beetz, M.: Parameterizing Actions to have the Appropriate Effects. In: IROS (2011)
20. Aker, E., Erdogan, A., Erdem, E., Patoglu, V.: Housekeeping with multiple autonomous robots: Knowledge representation and automated reasoning for a tightly integrated robot control architecture. In: IROS (2011)
21. Torralba, A., Murphy, K., Freeman, W.T.: Using the forest to see the trees: object recognition in context. *Communications of the ACM, Research Highlights* 53(3), 107–114 (2010)
22. Saxena, A., Chung, S., Ng, A.: 3-d depth reconstruction from a single still image. *International Journal of Computer Vision* 76(1), 53–69 (2008)
23. Saxena, A., Sun, M., Ng, A.: Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(5), 824–840 (2009)
24. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded classification models: Combining models for holistic scene understanding. In: *Neural Information Processing Systems* (2008)
25. Li, C., Kowdle, A., Saxena, A., Chen, T.: Towards holistic scene understanding: Feedback enabled cascaded classification models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(7), 1394–1408 (2012)
26. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: *ICCV* (2009)
27. Xiong, X., Huber, D.: Using context to create semantic 3d models of indoor environments. In: *BMVC* (2010)
28. Koppula, H., Anand, A., Joachims, T., Saxena, A.: Semantic labeling of 3d point clouds for indoor scenes. In: *NIPS* (2011)
29. Anand, A., Koppula, H., Joachims, T., Saxena, A.: Contextually guided semantic labeling and search for 3d point clouds. *IJRR* (2012)
30. Fisher, M., Hanrahan, P.: Context-based search for 3d models. *ACM TOG* 29(6) (2010)
31. Fisher, M., Savva, M., Hanrahan, P.: Characterizing structural relationships in scenes using graph kernels. In: *SIGGRAPH* (2011)
32. Nabbe, B., Kumar, S., Hebert, M.: Path planning with hallucinated worlds. In: *IROS* (2004)
33. Kuffner Jr., J., LaValle, S.: Rrt-connect: An efficient approach to single-query path planning. In: *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2000*, vol. 2, pp. 995–1001. IEEE (2000)
34. Stoll, C., Gall, J., de Aguiar, E., Thrun, S., Theobalt, C.: Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)* (2010)
35. Shin, D., Sardellitti, I., Park, Y.-L., Khatib, O., Cutkosky, M.: Design and control of a bio-inspired human-friendly robot. *The International Journal of Robotics Research* 29(5), 571–584 (2010)
36. Sung, J., Ponce, C., Selman, B., Saxena, A.: Unstructured human activity detection from rgbd images. In: *International Conference on Robotics and Automation, ICRA* (2012)
37. Ly, D., Saxena, A., Lipson, H.: Co-evolutionary predictors for kinematic pose inference from rgbd images. In: *Genetic and Evolutionary Computation Conference, GECCO* (2012)

38. Demircan, E., Besier, T.F., Khatib, O.: Muscle force transmission to operational space accelerations during elite golf swings. In: Proc. of the IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, pp. 1464–1469 (May 2012)
39. Teh, Y.W.: Dirichlet process. Encyclopedia of Machine Learning, 280–287 (2010)
40. Neal, R.: Markov chain sampling methods for dirichlet process mixture models. Journal of Computational and Graphical Statistics, 249–265 (2000)
41. Diankov, R., Kuffner, J.: Openrave: A planning architecture for autonomous robotics, Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34 (2008)

# Hand Shape Classification with a Wrist Contour Sensor (Comparison of Feature Types and Observation of Resemblance among Subjects)

Rui Fukui, Masahiko Watanabe, Masamichi Shimosaka, and Tomomasa Sato

**Abstract.** Hand gesture can express rich information. However, existing hand shape recognition methods have several problems. In order to utilize hand gesture in a home automation, we have focused on "wrist contour", and have developed a wrist-watch-type device that measures wrist contour using photo reflector arrays. In this paper, we try on two challenges: the first is improvement of the hand shape recognition performance, and the second is making clear the effect of personal difference and finding a key to overcome the difference. We collect wrist contour data from 28 subjects and conduct two kinds of experiments. As for the first challenge, three different feature types are compared. The experimental results extract several important contour statistics and the classification rate itself is also improved by introducing multiple subjects' data for training. As for the second challenge, we compose a resemblance matrix to evaluate resemblance among subjects. The results indicate that training data selection is important to improve the classification performance, especially when we don't have time to collect enough training data for a new user.

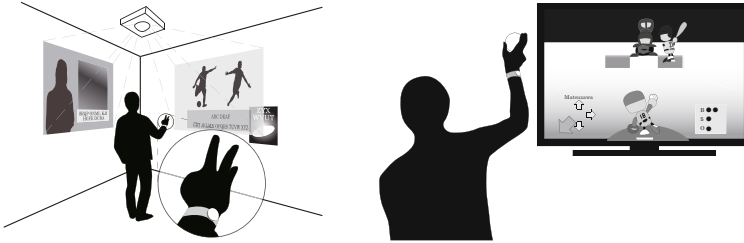
## 1 Introduction

Gesture recognition is getting more popular for home use. Popular gesture devices such as Microsoft Kinect(R) use body movement (i.e. arm movements) as an input. They are specialized in recognition of large body movements, that means they are not good at monitoring small body movements (e.g. hand motion) because of the resolution and accuracy limitations. Even though hand shape can express rich information with small movements, lack of concise hand shape recognition method prevents us from utilizing the beneficial information[8].

---

Rui Fukui · Masahiko Watanabe · Masamichi Shimosaka · Tomomasa Sato  
The University of Tokyo,  
7-3-1 Hongo Bunkyo-ku, Tokyo, Japan  
e-mail: {fukui, masahiko, simosaka}@ics.t.u-tokyo.ac.jp,  
tomomasasato@jcom.home.ne.jp

Therefore, we propose a novel method that realizes hand shape recognition with only a wrist-watch-type device. Such recognition method enables us to realize many applications: remote control of home electronics, gaming interface and so on (Fig. 1).



**Fig. 1** Application images: Hand gesture control of displays and recognition of ball grip in baseball game

### 1.1 Hand Shape Recognition Methods

There are three major hand shape recognition methods.

1. Data glove [1][2]: A user attaches a glove-type device with bend sensors at each finger joint. This method realizes high recognition performance, but the glove restricts hand movements and attaching a glove is a little troublesome for daily activities.
2. Camera vision [3][7][6]: This system recognizes hand shapes using image processing. If a camera acquires an image from appropriate direction, it exerts high performance, but it highly depends on the relative position and direction of user’s hand and the camera.
3. Electromyogram (EMG) [9][11]: With signals from electrode attached to user’s arm, it detects myoelectric potential and estimates hand movements. EMG sensor can realize a wearable system. Hence, it does not restrict user position and direction. However, it needs many initial configuration and calibration processes to realize sufficient performance: a user has to clean up his or her skin, the sensor is very naive to attached point, and a recognition system needs a lot of calibration data.

When introducing a hand gesture recognition device into a home, these methods have three problems: influence on activity, complex initial configuration, and insufficient recognition performance.

To overcome these problems, we focus on “wrist contour” We designate a wrist cross-section contour as a wrist contour. The wrist contour has various shapes because finger movements are induced by activities of tendons and muscles near the wrist as shown in Fig. 2.

As a similar approach, Rekimoto[10] developed a wrist-watch-type device with capacitive sensors measuring wrist surface in three points, and recognized two hand



shapes (grasping and pointing). In our previous work[5], we developed a wrist-watch-type device (Fig. 3) with two array of 75 photo reflectors, and conducted hand shape classification experiments. The performance of eight hand shapes' classification was 73.2% (training data include subject's own data), or 47.8% (training data exclude subject's own data). The value is not sufficient for practical use, especially personal difference is the bottom neck of the performance.

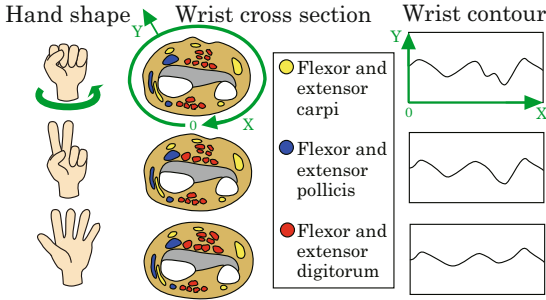


Fig. 2 Principle of wrist contour variation with hand shape



Fig. 3 The wrist-watch-type device

### 1.2 Challenges

In this paper, we try on two challenges: the first is improvement of classification performance, and the second is making clear the effect of personal difference and finding a key to overcome the difference.

As for the first challenge, the wrist contour data differ slightly in attached conditions even in the same subject. Besides, even if the measuring device is almost appressed to the wrist, slippage (moving in a radial or circumferential direction) may occur. Therefore, we need to investigate a robust normalizing procedure that can extract stable characteristics and need to design more solid feature extraction process.

Regarding the second challenge, we want to utilize other subjects' data that is acquired previously, and this approach can release a new user from collecting numerous training data.

## 2 Approaches for Efficient Utilization of Wrist Contour Data

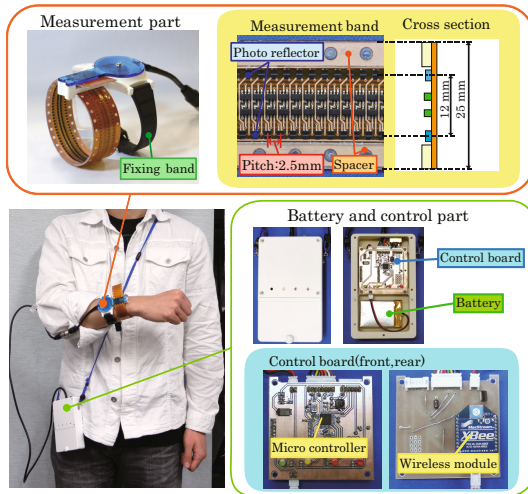
In order to measure wrist contour precisely, we developed a wrist contour measuring device. From the data collected with the device, features are extracted for hand shape classification methods. The key elements for our approach are described in this section.

### 2.1 Wrist Contour Measuring Device

The device consists of two parts: the wrist-watch-type measurement part, and the battery and control part (Fig. 4). The measurement part has a measurement band; a flexible band with photo reflectors (infrared-light distance sensors). The band can measure distances between the band and surface of the wrist using photo reflector arrays. The band has two arrays and each array has 75 photo reflectors. With a wireless module mounted in battery and control part, the device communicates with PC wirelessly.

The basic specifications are as follows. Please refer to our previous work [5] for the details of the device.

- Measurement area: ~185mm
- Measurement pitch: 2.5mm
- Distance resolution: 0.1mm (~3.5mm range)
- Sampling rate: 10Hz



**Fig. 4** Composition of wrist contour measuring device. The measurement part is connected to the battery and control part by a wire. The Fixing band assists the attachment of the measurement band and reduces slippage.

### 2.2 Feature Extraction Process

Wrist contour raw data examples are shown in the upper side of Fig. 5. There are small difference among the raw data of hand classes, therefore feature extraction process is essential. We prepare two potential feature types.

One feature type is “normalized contour data”. Because each muscle and tendon is different in thickness, each sensor element has different variation range of distance. The process samples the maximum and minimum distance for each sensor

element, and normalizes distance data into 0 to 1 (Fig. 5). With this process, small variations can be emphasized. On the other hand, the slippage of the band might be great noise.

Another feature type is “contour statistics”. They are statistics from wrist contour distance data, such as sum of distances, maximum distance, histograms and so on (Fig. 6). Each statistics are normalized by calibration data (wrist contour data of Fist and Open hand). With this approach, we try to overcome slippage or personal differences.

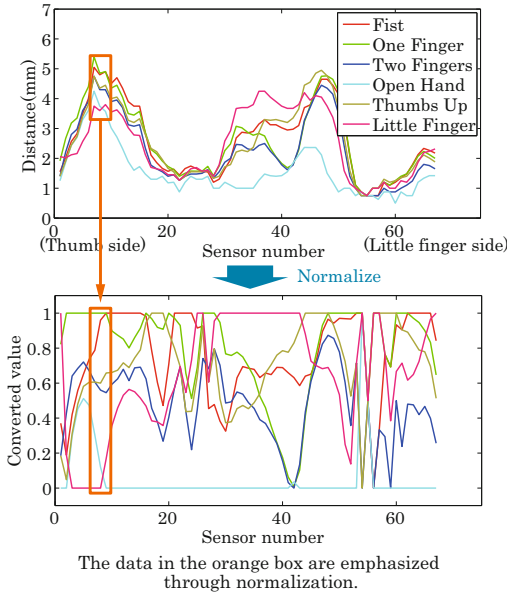


Fig. 5 Example of wrist contour data normalization

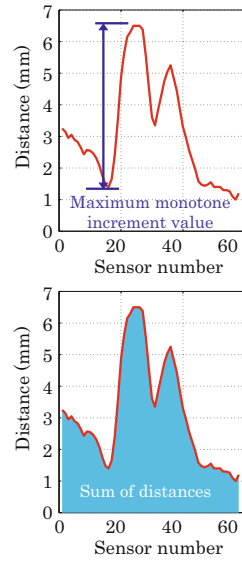


Fig. 6 Example of statistics from wrist contour data

### 2.3 Classification Method

As classification methods, k-Nearest Neighbor (k-NN) method and AdaBoost method are used.

In k-NN method, the test data is labeled by votes of k nearest samples. All Euclidean distances between the test sample and training samples in the feature space are calculated, and nearest k training samples have right to vote.

AdaBoost is a kind of boosting method, which makes some weak learners (in our implementation, decision stumps) [4]. The test data is labeled by weak learners' weighted votes. Weights on weak learners are tuned to fit to training data in the training process.

### 2.4 Personal Resemblance

Resemblance among subjects is the main topic to be investigated. That is because when other subjects' data are used as training data, the recognition performance differs drastically according to the combination of test subject and training subjects. We try to observe the resemblance among subjects by examining the relationships between classification performance and combination of subjects.

## 3 Experiments

We collected wrist contour data from 28 subjects. With the data, two kinds of hand shape classification experiments are conducted: (1) Comparison of two feature types using three training data groups, and (2) Evaluation of resemblance among subjects.

### 3.1 Wrist Contour Data Collection

We collected wrist contour data from 28 subjects, male and female of 20's to 50's. The arm posture and wrist pronation is fixed as shown in Fig. 7 because a wrist contour varies with wrist pronation. Data collecting procedure is as follows:

- Step 1. The measurement part is attached on the wrist in rough alignment.
- Step 2. A display shows a hand shape illustration to the subject, and the subject imitates the hand shape, and then wrist contour data is recorded.
- Step 3. After recording wrist contour data of all six hand shapes (one set, shown in Fig. 8), the measuring device is taken off.
- Step 4. Repeat Step 1~3 six times for each subject.

Finally, 1008 wrist contour data were collected; 28 subjects × 6 wrist contour data sets × 6 hand shapes.

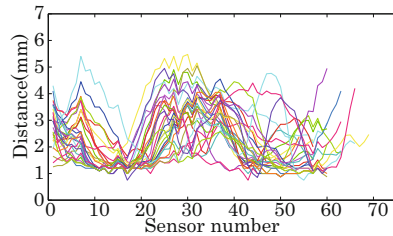
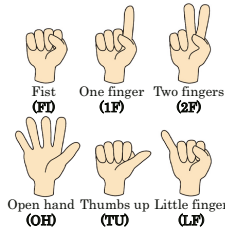


Fig. 7 The posture when collecting wrist data

Fig. 8 Hand shape images of six hand classes (one set)

Fig. 9 Wrist contour raw data of 28 subjects. (One data per subject. Hand shape: One finger)

## 3.2 *Classification Experiment (1) Comparison of Two Feature Types*

In order to examine robustness of the described feature extraction processes, two types and one derived type of features are compared.

### 3.2.1 **Setup of Classification Experiment (1)**

Feature types to be compared are as follows.

- A. Normalized contour data 45 dimensions: each contour data are linearly converted to the smallest wrist contour size.
- B. Contour statistics 92 dimensions
- C. Selected contour statistics 5 dimensions: This feature type is derived from feature type B. When classifying from contour statistics using k-NN method, useful features might be hidden by other less useful features. Also, when classifying using AdaBoost, large number of features might cause over-fitting. Therefore we conduct another experiment of using only five contour statistics that have large separation metrics. The separation metrics is calculated as between-class variance divided by in-class variance.

Two classification methods are used: k-NN method and AdaBoost method. When using AdaBoost, the system produces six one-versus-the-rest classifiers and an output class is determined by maximum output value in six classifiers.

Therefore, the number of combinations of feature types and classification methods are  $3 \times 2 = 6$ .

As for training data, three groups are prepared.

- Group 1 Subject's own data 5 sets.
- Group 2 Subject's own data 5 sets and other subjects'  $27 \times 6$  sets.
- Group 3 Other subjects'  $27 \times 6$  sets.

The classification performance is evaluated by classification rate (number of correctly classified samples / number of inputted samples  $\times 100$  [%]).

### 3.2.2 **Result of Classification Experiment (1)**

Result of classification experiment (1) is shown in Fig. 10. The table explains the result in each training data groups. As expected, the combination of feature type B and k-NN method exerts low score in all groups.

In group 1 (the subject's own data), feature type A exerts higher performance in classification rate than feature type B and C. Incidentally, results of experiments using AdaBoost (especially feature type A and B) exert lower performance. This is because there are not enough training data and that causes over-fitting. In group 2 (the subject's own data and other subjects' data), no remarkable difference among three feature types is observed. In group 3 (other subjects' data), feature type B and C exert better performance than feature type A.

Tbl. 1 shows the confusion matrix of one experiment (B. Statistics + AdaBoost). Most hand shapes are correctly classified on some level, however, confusions among some similar classes (e.g. 1F and LF) are observable.

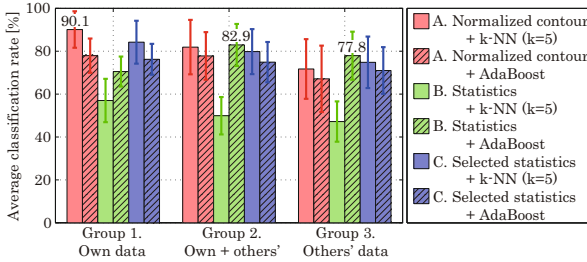


Fig. 10 Result of classification experiment (1)

Table 1 Confusion matrix of B. Statistics + AdaBoost

		Output class						Rate [%]
		FI	1F	2F	OH	TU	LF	
Accurate class	FI	160	1	0	0	5	2	95.2
	1F	0	94	16	0	25	33	56
	2F	0	12	136	7	0	13	81
	OH	0	0	4	163	0	1	97
	TU	6	19	0	0	127	16	75.6
	LF	2	41	8	0	12	105	62.5

The insights from the results are as follows. First, useful features are different according to training data group. For instance, when using subject’s own data as training data, the combination of feature type A (normalized contour data) and k-NN classification method exerted relatively high performance of 90.1% classification rate. On the other hand, when using other subjects’ data as training data, the performance is improved using feature type B (contour statistics); classification rate is 77.9% using AdaBoost. Second, the weights of AdaBoost weak learners indicates that sum of distances, maximum monotone increasing value and sum of differences of histogram are useful contour statistics.

As a result, it can be said the performance is improved: with subject’s own data, 90.1% in classification of six hand shapes is thought to be enough for use. However, the classification rate of 77.9% with other subjects’ data is not enough for practical use, so we should consider how to use other subjects’ data more efficiently.

### 3.3 Classification Experiment (2) Evaluation of Resemblance among Subjects

In order to examine resemblance in wrist contours among subjects, classification experiment is conducted in one-on-one combinations of training and test data.

#### 3.3.1 Setup of Classification Experiment (2)

Classification rates are used for evaluation of resemblance among subjects. We configure the resemblance evaluation matrix (28 subjects × 28 subjects). To make the matrix, all combination (28 × 27) classification experiments are conducted. For each experiment, training data only include one subject data. An element (S1, S2) of the resemblance evaluation matrix is classification rate when using S1’s data as test data, and S2’s data as training data.

Two feature types (A and C) are used and k-NN method is used as classification method. This is because when classifying with AdaBoost, learners are over-fitted to small number of training data.

### 3.3.2 Result of Classification Experiment (2)

One result of classification experiment (2) is shown in Tbl. 2; the result of the experiment using feature type C (selected contour statistics). The row represents the test subject and the column represents the training subject. For example, when the test subject is subject 3 and the training subject is subject 9, the classification rate is 80.6% (the green box).

The average classification rate of all experiments is 59.8%. The rate of more than 80% elements is 11.2% (85/756), and the rate of less than 40% elements is 12.8% (97/756).

Meanwhile, the result of experiment using feature A is as follows: the average classification rate of all experiments is 54.3%, the rate of more than 80% elements is 5.6% (42/756), and the rate of less than 40% elements is 20.2% (153/756). In addition, each element of the resemblance evaluation matrixes is far different from the one of the result using feature C.

**Table 2** Resemblance evaluation matrix (using feature C)

	training subject number																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	55.6	41.7	86.1	50.0	50.0	52.8	58.3	50.0	50.0	30.6	38.9	55.6	47.2	47.2	50.0	58.3	41.7	44.4	58.3	58.3	55.6	44.4	41.7	41.7	41.7	58.3	50.0	
2	38.9	66.7	80.6	83.3	72.2	72.2	38.9	77.8	83.3	72.2	63.9	52.8	66.7	66.7	38.9	80.6	66.7	69.4	58.3	38.9	83.3	72.2	83.3	38.9	58.3	33.3	83.3	
3	47.2	66.7	61.1	77.8	72.2	66.7	27.8	80.6	89.4	77.8	61.1	55.6	77.8	63.9	50.0	66.7	69.4	63.9	69.4	30.6	66.7	77.8	55.6	75.0	30.6	75.0		
4	33.3	83.3	77.8	83.3	86.1	77.8	41.7	72.2	80.6	80.6	55.6	58.3	69.4	61.1	58.3	86.1	52.8	88.9	61.1	50.0	86.1	63.9	83.3	44.4	77.8	50.0	80.6	
5	33.3	83.3	77.8	88.9	75.0	80.6	52.8	83.3	77.8	80.6	58.3	61.1	72.2	66.7	58.3	77.8	75.0	80.6	66.7	41.7	86.1	75.0	86.1	44.4	83.3	33.3	83.3	
6	33.3	75.0	91.7	63.9	83.3	75.0	38.9	80.6	66.7	77.8	55.6	55.6	83.3	63.9	61.1	69.4	66.7	72.2	69.4	36.1	77.8	77.8	83.3	52.8	75.0	33.3	66.7	
7	30.6	58.3	69.4	63.9	72.2	72.2	44.4	58.3	75.0	77.8	44.4	58.3	75.0	52.8	44.4	55.6	61.1	80.6	58.3	33.3	88.9	91.7	86.1	66.7	75.0	33.3	83.3	
8	75.0	66.7	44.4	50.0	63.9	47.2	63.9	50.0	55.6	44.4	36.1	69.4	61.1	61.1	72.2	58.3	66.7	41.7	55.6	72.2	50.0	36.1	41.7	55.6	55.6	58.3	58.3	
9	47.2	80.6	66.7	58.3	80.6	52.8	41.7	63.9	55.6	47.2	55.6	58.3	61.1	47.2	55.6	63.9	61.1	66.7	44.4	63.9	63.9	61.1	44.4	61.1	36.1	61.1		
10	72.2	72.2	63.9	69.4	69.4	77.8	69.4	25.0	58.3	69.4	63.9	38.9	58.3	47.2	22.2	63.9	44.4	52.8	55.6	16.7	75.0	69.4	72.2	30.6	66.7	19.4	83.3	
11	38.9	63.9	69.4	52.8	63.9	61.1	69.4	41.7	69.4	50.0	38.9	63.9	63.9	52.8	52.8	44.4	61.1	75.0	47.2	41.7	69.4	72.2	66.7	61.1	69.4	47.2	72.2	
12	36.1	80.6	63.9	55.6	69.4	66.7	58.3	27.8	69.4	66.7	72.2	44.4	50.0	63.9	41.7	83.3	55.6	44.4	47.2	36.1	80.6	38.9	66.7	47.2	63.9	30.6	80.6	
13	55.6	52.8	61.1	52.8	55.6	55.6	61.1	72.2	66.7	50.0	63.9	38.9	52.8	47.2	77.8	58.3	63.9	63.9	50.0	63.9	63.9	55.6	58.3	63.9	77.8	55.6	77.8	
14	36.1	63.9	86.1	55.6	72.2	75.0	69.4	58.3	75.0	58.3	69.4	47.2	75.0	63.9	69.4	55.6	75.0	72.2	66.7	44.4	63.9	75.0	69.4	61.1	66.7	44.4	69.4	
15	22.2	50.0	44.4	33.3	36.1	52.8	44.4	16.7	55.6	47.2	36.1	52.8	19.4	38.9	16.7	47.2	33.3	36.1	38.9	16.7	33.3	41.7	36.1	22.2	30.6	16.7	44.4	
16	58.3	63.9	61.1	58.3	63.9	75.0	63.9	75.0	72.2	61.1	77.8	30.6	83.3	66.7	50.0	66.7	63.9	77.8	63.9	75.0	69.4	69.4	58.3	63.9	69.4	66.7	75.0	
17	36.1	91.7	66.7	91.7	86.1	72.2	66.7	36.1	75.0	83.3	63.9	72.2	50.0	63.9	61.1	50.0	66.7	50.0	72.2	44.4	86.1	69.4	83.3	52.8	69.4	38.9	75.0	
18	38.9	61.1	69.4	44.4	86.1	44.4	75.0	61.1	69.4	44.4	58.3	36.1	66.7	77.8	58.3	66.7	50.0	52.8	69.4	50.0	41.7	66.7	58.3	58.3	75.0	36.1	58.3	
19	36.1	66.7	63.9	52.8	69.4	61.1	61.1	30.6	63.9	61.1	77.8	41.7	41.7	69.4	50.0	44.4	58.3	58.3	52.8	25.0	72.2	72.2	69.4	55.6	66.7	27.8	75.0	
20	47.2	80.6	83.3	61.1	72.2	75.0	63.9	50.0	75.0	69.4	58.3	61.1	66.7	69.4	55.6	75.0	69.4	72.2	61.1	41.7	66.7	72.2	63.9	38.9	69.4	38.9	75.0	
21	66.7	52.8	50.0	44.4	55.6	47.2	52.8	77.8	69.4	50.0	41.7	47.2	75.0	52.8	50.0	75.0	50.0	47.2	50.0	52.8	33.3	61.1	77.8	75.0	72.2	44.4		
22	38.9	88.9	75.0	58.3	80.6	83.3	47.2	72.2	80.6	83.3	69.4	47.2	55.6	75.0	61.1	58.3	83.3	66.7	77.8	69.4	33.3	72.2	83.3	58.3	66.7	30.6	66.7	
23	36.1	66.7	75.0	66.7	63.9	77.8	86.1	33.3	55.6	69.4	75.0	38.9	58.3	83.3	50.0	66.7	50.0	52.8	83.3	55.6	30.6	77.8	77.8	63.9	75.0	33.3	83.3	
24	36.1	72.2	80.6	63.9	83.3	77.8	83.3	19.4	69.4	80.6	88.9	55.6	41.7	80.6	63.9	27.8	77.8	50.0	80.6	58.3	19.4	80.6	83.3	61.1	63.9	19.4	88.9	
25	47.2	44.4	61.1	41.7	55.6	44.4	52.8	61.1	50.0	36.1	69.4	38.9	61.1	58.3	50.0	63.9	47.2	61.1	63.9	47.2	66.7	47.2	61.1	58.3	72.2	52.8	58.3	
26	50.0	63.9	77.8	58.3	83.3	61.1	66.7	58.3	66.7	47.2	75.0	38.9	72.2	69.4	52.8	66.7	55.6	75.0	72.2	58.3	50.0	58.3	72.2	83.3	55.6	38.9	63.9	
27	77.8	50.0	50.8	58.9	50.0	47.2	50.0	66.7	50.0	55.6	33.3	25.0	58.3	50.0	50.0	50.0	50.0	44.4	50.0	33.3	80.6	50.0	47.2	44.4	69.4	47.2	50.0	
28	38.9	69.4	83.3	50.0	80.6	58.3	75.0	33.3	66.7	69.4	75.0	52.8	44.4	66.7	58.3	41.7	72.2	66.7	80.6	58.3	30.6	69.4	75.0	80.6	50.0	63.9	38.9	

Each element represents classification rate [%].  
 Elements more than 80%,  
 Elements less than 40%.

Several insights can be drawn from the results. First, the fact that the rate of useful combinations (more than 80%) is 11.2% and the rate of useless combinations (less than 40%) is 12.8% indicates the necessity of a training data selection process. Second, the resemblance evaluation matrix of feature type C and the one of feature type A are far different, that means the definition of resemblance deeply depends on the type of features. Therefore the training data selection processes need to be

designed respectively to the feature type. Third, when examining each subject, there are some subjects that exert low classification rate with any combinations, so we need to collect more data to fill the “lack” of current data.

When applying results of experiment (2) to improve classification performance, some approaches can be candidate. One approach is making groups of highly resemble subjects, and then training data or classifiers are selected according to the group that new subject belongs to. It is important to determine what model to use as group, and number of groups, and design of group classifiers. Another candidate approach is like a kind of filtering that selects each subject to use as training data [12].

## 4 Conclusion and Future Work

We focus on wrist contour for hand shape recognition in order to overcome problems of existing recognition methods. As a measurement device, we developed a wrist-watch-type device with photo reflector arrays. Using the device, wrist contour data of six hand shapes are collected from 28 subjects. With the collected data, we conducted several experiments for comparison of feature types, and evaluated resemblance among subjects.

Through classification experiment (1), it can be said the performance is improved. However, the classification rate of 77.9% with other subjects' data is not enough for practical use and more efficient usage of other subjects' data is essential. Through classification experiment (2), the need of training data selection process are confirmed. We should try on some method of grouping or selecting training data.

Our future works are as follows. In terms of design, it is necessary to downsize the device and integrate all functions to a wrist-watch-type device. Additionally, we need to tackle the problem of wrist pronation changes by redesigning hardware and software. In terms of recognition performance, we need to try some approaches such as finding other new statistics or using another classification methods to improve performance. As mentioned above, an efficient training data selection process should be designed to utilize other subjects' data as training data.

**Acknowledgements.** This work was supported by JSPS KAKENHI Grant Number 23800012.

## References

1. CyberGloveSystems, <http://www.cyberglovesystems.com/>
2. Deller, M., Ebert, A., Bender, M., Hagen, H.: Flexible gesture recognition for immersive virtual environments. In: Proceedings of Tenth International Conference on Information Visualization, pp. 563–568 (2006)
3. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: A review on vision-based full DOF hand motion estimation. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 75 (2005)



4. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Proceedings of the Second European Conference on Computational Learning Theory, pp. 23–37 (1995)
5. Fukui, R., Watanabe, M., Gyota, T., Shimosaka, M., Sato, T.: Hand shape classification with a wrist contour sensor: Development of a prototype device. In: Proceedings of ACM Conference on Ubiquitous Computing, pp. 311–314 (2011)
6. Garg, P., Aggarwal, N., Sofat, S.: Vision based hand gesture recognition. *World Academy of Science Engineering and Technology*, 972–977 (2009)
7. Hoshino, K., Tamaki, E., Tanimoto, T.: Copycat hand robot hand imitating human motions at high speed and with high accuracy. *Advanced Robotics* 21, 1743–1761 (2007)
8. Huang, T.S., Pavlovic, V.I.: Hand gesture modeling, analysis, and synthesis. In: Proceedings of IEEE International Workshop on Automatic Face and Gesture Recognition, pp. 73–79 (1995)
9. Nagata, K., Adno, K., Yamada, M., Magatani, K.: A classification method of hand movements using multi channel electrode. In: Proceedings of IEEE-EMBS, Annual International Conference of the Engineering in Medicine and Biology Society, pp. 2375–2378 (2005)
10. Rekimoto, J.: GestureWrist and GesturePad: Unobtrusive wearable interaction devices. In: Proceedings of the 5th IEEE International Symposium on Wearable Computers, pp. 21–27 (2001)
11. Yoshikawa, M., Mikawa, M., Tanaka, K.: Real-time hand motion estimation using EMG signals with support vector machines. In: Proceedings of SICE-ICASE, International Joint Conference, pp. 593–598 (2006)
12. Yu, K., Tresp, V., Yu, S.: A nonparametric hierarchical bayesian framework for information filtering. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 353–360 (2004)

# Experimental Validation of Operator Aids for High Speed Vehicle Teleoperation

Alonzo Kelly, Nicholas Chan, Herman Herman, and Randy Warner

**Abstract.** Although fully autonomous robots continue to advance in ability, all points on the spectrum of cooperative interfaces between man and machine continue to have their place. We have developed a suite of operator assist technologies for a small (1 cubic meter volume) high speed robot that is intended to improve both speed and fidelity of control. These aids include fast stability control loops that run on the robot and graphical user interface enhancements that help the operator cope with lost peripheral vision, unstable video, and latency. After implementing the driving aids, we conducted an experiment where we evaluated the relative value of each from the perspective of their capacity to improve driving performance. Over a one week period, we tested 10 drivers in each of four driving configurations for three repetitions of a difficult test course. The results demonstrate that operators of all skill levels can benefit from the aids and that stabilized video and predictive displays are among the most valuable of the features we added.

## 1 Introduction

Teleoperation is a control concept that is as old as robotics — and for fundamental reasons. When the motivation for the use of a robot is to keep a human out of harm's way (e.g. nuclear servicing) or to place a robot where a human could never go (e.g. inside a blood vessel), the robot and the human are separated by assumption. Given that separation in space, the question of how they can effectively work together arises naturally. As research in autonomous robots has advanced, teleoperation has become merely one of many options, but those advances have neither rendered teleoperation irrelevant nor solved many of its fundamental challenges. Nonetheless, a robot which is more autonomous could potentially use its awareness of its surroundings and its state to conform to the needs and limitations of humans. Autonomy therefore has the potential to render the man-machine system more effective. This is hardly a new idea but different applications give rise to different

---

Alonzo Kelly · Nicholas Chan · Herman Herman · Randy Warner  
Robotics Institute, Carnegie Mellon University  
#10 Fortieth Street, Pittsburgh Pa, 15201  
e-mail: {alonzo, ncchan, herman, rwarner}@rec.ri.cmu.edu

realizations of the concept. This paper explores the potential of operator aiding in the context of small high speed wheeled mobile robots, with a particular emphasis on experimental validation.

Despite the promise of teleoperation, its history has been characterized by a constant struggle to solve many fundamental and difficult issues. For vehicles, the operator is removed from the rich sensory experience (visual, audio, olfactory, inertial) of sitting in the driver seat. The intrinsic limitations of the associated sensing, communication, and display technology then deprive the operator of all stimuli except a video. Unfortunately, video and associated displays is a poor surrogate for biological vision according to almost any chosen basis of comparison.

These technology limitations are responsible for both the reduced situation awareness of a remotely located driver, and the difficulty of driving competently at high speed. There is no inexpensive and effective means of remotely conveying the sensation of all of the acceleration, impact, and vibration associated with sitting in the driver seat. Likewise, it is well known that latency in the video (or any sensing) makes it very difficult to respond effectively to unpredictable disturbances in a feedback setting, whether there is a human in the loop or not.

## ***1.1 Problem Statement***

Nonetheless, displays can be annotated to include knowledge (available to the robot) that may not be discernible in the raw video. Some control loops can be closed on the vehicle where there is far less latency and others can include a predictive component that allows the operator to remove, in a classical feedforward manner, predictable errors before they occur.

Therefore, assistive technologies should be able to improve performance in a measureable way and our goal in this work was not only to implement these technologies on a challenging platform, but to measure their effectiveness in a controlled empirical setting.

## ***1.2 Related Work***

This work fuses ideas for teleoperation from the earliest days of robot manipulator research with techniques for video stabilization, ideas from modern gaming interfaces, and elements of electronic stability control (ESC). ESC is now available on most recently manufactured automobiles. We will use predictive displays, gyro-stabilized video, annotations over live video, and speed governing based on yawrate feedback. The effort described here was motivated in part by a desire to produce a second version of our teleoperation system described in [7]. We wanted this second version to be less expensive to produce and to be suitable for robots without lidar perception.

Almost three decades ago, the field of *teleroobotics* was a subfield of robotics pursuing a scientific understanding of the man-machine system. Numerous techniques for supervisory control and teleoperation of manipulators were outlined as early as the mid 1980s [12]. Virtual displays that are either predictive or used for preview have often been used to compensate for both delay and limited data

bandwidth when remotely operating manipulators. The concept of *teleprogramming* was an early form of model-based teleoperation [6] that used models to mitigate the effects of latency. More recently, more intuitive and task-centric interfaces have been used to operate manipulators over thousands of miles of separation [8].

Latency compensation in space applications has also been accomplished with motion preview and predictive displays [2]. In some cases, stereo graphics viewed in a stereoscopic display have been used to improve operator depth perception [9]. While all of the telerobotics work described so far has been applied to stationary manipulators in a stationary scene, the principles are extendable to moving sensors in a dynamic scene, if the image processing is efficient. For example, Ricks et al. used a predictive method that they dubbed *quickenning* to compensate for latency when teleoperating a mobile indoor robot [11].

The use of gyros to stabilize video was originally driven by the need to stabilize camcorders [10]. More recently, numerous techniques have been borrowed from computer vision which use the image data itself in order to estimate the motion of the camera [3]. Of course, inertial and visual cues can also be used at the same time [14].

One early use of augmented video like ours is augmented reality — the introduction of synthetic components into a largely real view. The display used may be head worn, handheld, or projected on a display surface. Numerous applications have been explored in medicine, manufacturing, visualization, entertainment and the military [1]. In our case, the live video is the real part and the graphical and textual annotation is the augmentation. When the augmentations are based on a rich underlying model, they are said to be knowledge-based [5]. Augmented video is also a favorite form of display in computer gaming. In that case, the video is also virtual.

Electronic stability control and roll stability control systems have been the subject of intense development by the automotive industry in the last 20 years. Our approach to ESC is a governor based on yaw rate error whereas active automotive systems are based on using braking to generate restoring moments [13]. Our approach to roll stability control is based on early work from legged mobile robots as realized in the algorithms in [4].

## 2 Technical Approach

The work described here investigates a number of techniques that promise to improve the performance of the man-machine system. These techniques can be grouped into those related to communications, control, and operator display. After describing the hardware and the overall rationale, the more important algorithms are described below.

### 2.1 Hardware Design

Our work was conducted on the Forerunner remote-controlled vehicle developed by RE<sup>2</sup> Inc (Figure 1). We chose this vehicle for its size and speed regime (max speed 25 km/hr/). The base platform provides control interfaces and computing to support

line-of-sight remote control. To this, we added a Core 2 Duo ULV 1.2 GHz CPU computer, a high dynamic range driving camera, an automotive obstacle avoidance radar, an inertial measurement unit, and 802.11-n wifi. The main computer was used to implement control algorithms and video compression.

Inertial navigation was performed in dedicated FPGA hardware. Another remote computer at the operator control station generated the graphical user interface. Our communications data rate was limited to a mere 0.75 megabits (not bytes) per second. Our approach for tolerating this limit was to use the latest (MPEG-4) video compression and to tune it for this application. Doing so permitted us to optimize video quality within the available resources.



**Fig. 1** Forerunner Remote Controlled Vehicle. All wheels are driven and steered. The vehicle is easily capable of 25 km./hr. speed so such ballistic motion is achievable.

## 2.2 Control Techniques

Several control techniques were used to assist the operator. A stability control system was used to help reduce risk of loss of yaw stability and of rollover, and a path following controller was used to reject associated disturbances at the vehicle level. The overall rationale for the use of these systems is as follows:

- *Feedback Control.* One of the most basic techniques is to close control loops locally on the vehicle where latency is low and reaction time is short. In this way, the vehicle is able to reject all disturbances that its feedback renders it competent to reject. The operator then has to deal only with the disturbances that remain, and these tend to be lower frequency and somewhat discernible from the operator display. In the case of safety systems like stability governing, the vehicle can be empowered to take control locally to prevent a mishap that would otherwise occur before the operator is even aware of the situation.
- *Model Predictive Control (MPC).* MPC has several uses. The most basic is the use of predictive models in multi-state control algorithms like path following. In this case the predictable effects of terrain following and wheel slip can be modelled to eliminate what would otherwise become error disturbances. Data bandwidth can be reduced as well. On-vehicle processing can be used to perform data-intensive predictions and then transmit only the results to the operator. For example, prediction calculations can include the effects of terrain slope, without having to transmit the terrain data to the operator.
- *Prediction Through Latency.* MPC can also be configured to account for predictable latencies - both uplink (of state) and downlink (of commands). Trajectory predictions performed on the vehicle can be shifted forward in time to reflect

uplink delay and predictions performed on the operator console can account for the command downlink delay. The delay itself can be predicted from past experience and/or from the measured delay of the most recent messages.

### 2.2.1 Inertial Navigation

We have implemented our own inertial navigation (INS) solution on other programs so we were able to re-use it for this application. The navigation system was used to provide position feedback for control purposes, to provide attitude feedback for video stabilization, and to provide specific force and angular velocity data for stability control. Use of our own INS permits us to optimize for GPS-denied performance and to integrate the solution for navigation and stability control in one package.

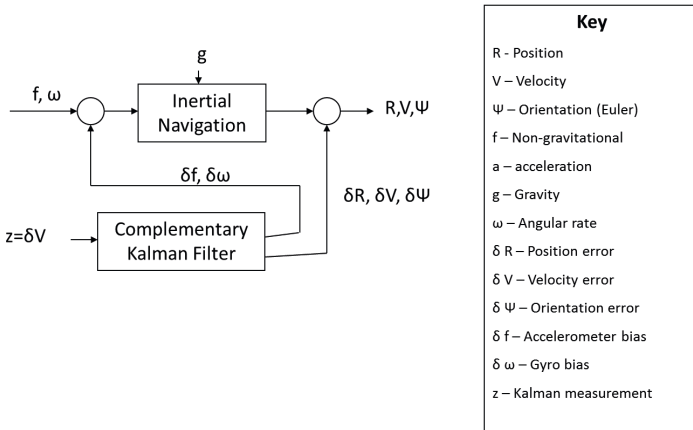


Fig. 2 Inertial Navigation System. A 15 state complementary filter configuration is used.

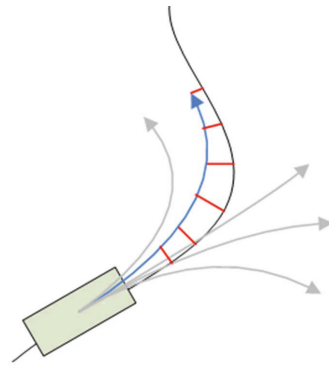
A relatively high performance Honeywell HG-1930 IMU was used. The design is a classical complementary Kalman filter (Figure 2) with 15 error states (position, orientation, velocity, accelerometer biases, gyro biases) and it is aided by measurements of the 4 wheel rotation rates and steer angles.

### 2.2.2 Stability Control

Exactly how a vehicle responds to high horizontal acceleration levels depends on at least the wheel support polygon, the center of gravity position, slope, terrain shape, and terrain friction. While the original intention was to implement a rollover prevention system, experimentation revealed that this vehicle is prone to spin out of control before wheel liftoff occurs. Accordingly, we implemented a yaw stability governor as well. It compared the commanded yawrate to the actual (as measured by the gyros in the IMU) and then imposed a computed limit on vehicle speed when the percent yawrate error exceeded a threshold.

### 2.2.3 Path Following

In addition to actuator level controls, and the operator’s own display-based adjustments, a model predictive path following controller was implemented on the vehicle. The operator’s driving commands, when converted to predicted response, are interpreted as a path to be followed. A clothoid is a path whose curvature function is of the form  $\kappa(s) = a + bs$  for initial curvature  $a$  and curvature gradient  $b$ . The algorithm (Figure 3) searches a discretized space of clothoids for the one which minimized the integrated pose error along the path and then sends the associated optimal control to the platform controller.



**Fig. 3** Path Follower. An MPC algorithm finds the optimal clothoid.

### 2.3 Display Techniques

Several display techniques were also important for assisting the operator. The overall rationale for the use of these systems is as follows:

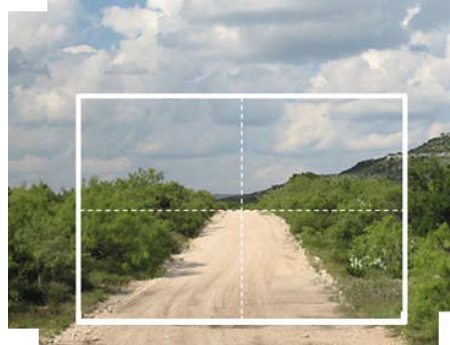
- *Inputs for Continuous Driving.* Of course, a very effective technique for remote driving with latency is to designate waypoints one at a time and wait until the vehicle achieves them. However, when driving continuously, the operator does not have the luxury of waiting for the display to stabilize (after motion stops) before injecting the next input — it is supposed to be a continuous process. One way to provide an intuitive input mechanism is to have the operator specify an instantaneous goal point in body coordinates. This input is static in a vehicle-fixed display and it can correspond precisely to the control horizon in MPC.
- *Predicted Path Display.* We furthermore chose to interpret the goal point as the desired endpoint of a vehicle trajectory. A predictive model of the vehicle is inverted as described above to produce the control that corresponds most closely to the desired path. The lateral position of the endpoint affects curvature and its distance downrange affects speed. The operator experience is that of literally steering this predicted endpoint. In this way, the mapping from what the operator wants to what the platform is commanded is automatic, state and terrain adaptive, and well calibrated. The result is a man-in-the-loop MPC system where the human continually adjusts the controls, optimizing on the fly, in the context of good predictions. The display discussion below reveals how the predictions are visually placed in the context of the vehicle surroundings on the screen.
- *Video Stabilization.* The context of a small vehicle driving fast over uneven terrain leads to a bumpy ride for the vehicle and a jumpy display for the operator. Accordingly, video stabilization was used to provide the operator with a



synthetically stable camera view. This feature was complementary to path prediction by providing a smoothly varying image of where the vehicle was headed, regardless of terrain following attitude changes.

### 2.3.1 Video Stabilization

Numerous options exist for the design of this feature. Our approach was based on our own prior efforts elsewhere because the software already existed. Each video frame was precisely tagged with the pose of the camera at the instant that the frame was acquired. The video was then rendered, based on the associated real camera attitude, onto a virtual billboard positioned a few meters in front of vehicle. The billboard was then viewed with a virtual camera at the true camera position — whose attitude was locally level (Figure 4). The operator perspective is that the video frames may move up and down slightly on the billboard but individual features remain fixed on the display. An added feature rendered the nose of the vehicle so that its attitude could be viewed in the same display.



**Fig. 4** Video Stabilization. A synthetic locally-level camera view is used so that terrain features remain fixed between video frames.

### 2.3.2 GUI with Video Overlay

The user interface (Figure 5) included the stabilized video as well as numerous overlays to convey such information as wheel slip (detected as yawrate error), radar-detected obstacles, attitude, speed, and proximity to rollover. Video overlays provide good use of screen real estate and convey extra information while permitting the operator to focus on the rapidly changing video. The predicted vehicle path was also overlaid on the video as shown. This technique allowed the operator to position the goal point precisely with respect to the objects in the scene.



**Fig. 5** Graphical User interface. A video game concept is used where annotations are overlaid with some transparency on the stabilized video.



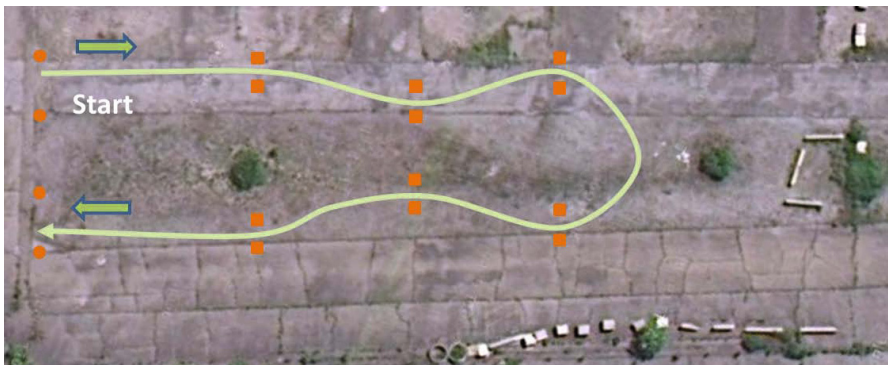
### 3 Experiments

It was already anecdotally clear that the assistive technology was valuable, so the test was designed to try to quantify that value. The principles of the experimental design included control of as much variables as possible while varying only the aiding modes that were available to the operator. During the testing, the speed was limited to 5 m/s and many operators achieved this speed, at times, due to the acceleration capability of the vehicle. We had damaged (and repaired) the vehicle with higher limits in earlier tests.

Communications latency was typically low, under 100 milliseconds for a round trip. Path prediction reduced the effects of latency by giving the operator the means to specify a destination, rather than an immediate velocity command. A destination in front of the vehicle is still valid even if it is received somewhat late, so the controller could still attempt to reach it.

Multiple driving modes were tested in random order to remove bias associated with learning the vehicle response and the test course. Operators were unable to see the course during the test though all could see it briefly before the test. Each operator was given the same briefing on the course and the technology before the test. Effects of cloud cover, precipitation, terrain friction etc. were mitigated by testing operators in all modes in a short period of time. Effects of vegetation were mitigated by driving the course often enough to trample the tall grass before any testing.

The test course (Figure 6) was designed to be short, but quite difficult to drive without the assistive technology. Narrow driving gates were constructed from bright cardboard boxes to enhance their visibility but they were designed to collapse easily on collision to avoid damaging the robot. Their precise positions were marked on the ground to ensure repeatability of course setup, because the gates were often hit by the vehicle. Half the course was grass and the other half was (old) pavement. Operators were told to drive as fast as possible without hitting the sides of any



**Fig. 6** Test Course - Overhead View. The small squares denote driving gates slightly wider than the vehicle. It took about 1 minute to drive the course when trying to drive fast. The course bounding rectangle is 185 ft along its longest dimension.

gates. None were given an opportunity to learn the feel of the system before the test began, although two drivers already knew the system well. All other drivers were robotics engineers with no knowledge of the system and varying experience in vehicle teleoperation. It was not our intention to evaluate learning curve. Rather, we concentrated on the effect the technology had on each driver as an individual, in the hope that it would help — regardless of skill level. Furthermore, the effects of cloud cover and sun angle, and perhaps other effects, could not be controlled over the course of the entire test. Therefore, comparisons of drivers to each other are not entirely free of such effects.

### 4 Results

Once initial tests determined that video stabilization and predictive display were the two most useful features, the final tests were designed to investigate these features more fully in order to produce a manageable number of tests. Ten subjects were tested and each drove the course three times in each of four configurations of the driving aids. That is, there were 12 tests performed for each of the 10 people. The configurations are summarized in Table 1. Video compression and stability control were on at all times.

Tests were conducted in mid summer at a test site in Hazelwood in Pittsburgh. We measured four principle observables: the total time to complete the course, the number of times a gate was hit by the vehicle, the number of times that the driver missed the gate entirely, and the curviness (integral of squared curvature with distance) of the path followed. The results averaged over all users are summarized in Table 2.

**Table 1** Test Configurations. These four combinations of assistive features were tested.

Attribute	Basic	Stabilized	Predictive	Both
Video Stabilization	no	yes	no	yes
Path Prediction	no	no	yes	yes
Path Follower	no	no	yes	yes

**Table 2** Test Results Averaged Over All Users. A clear trend of improved performance is evident with assistive features enabled, both individually and in combination.

Attribute	Basic	Stabilized	Predictive	Both
Time (secs)	49.9	44.6	39.1	36.7
# Hits	1.7	1.4	0.8	0.6
# Missed	0.3	0.2	0.0	0.0
Curviness	0.07	0.06	0.05	0.05

## 5 Main Experimental Insights

It is important to recognize that this was a difficult course to drive quickly and the vehicle could easily be driven beyond the speed threshold of stable control. Without video stabilization, the gates would jump around significantly in the field of view and it became less clear where the gate actually was in relation to the vehicle. Without prediction and path following, the perception of control fidelity was surprisingly low - meaning the vehicle appeared not to do what it was told to do. Latency was large enough to cause inexperienced operators to overcorrect, enter oscillation, and occasionally lose control entirely. Once the vehicle spun out of control, much time could be lost if it was already close to a gate and it had to be reversed to go through it. There was no rear camera for reverse driving. In any case, once the gate left the field of view due to a violent loss of control, the operator had to turn the vehicle in order to search the periphery of the camera field of view in order to find the gate again.

While the two most significant features added value (both individually and in combination) a fielded system would (based on our results) probably have all of them turned on, so we will concentrate on interpreting this case. With all features turned on a) 8 out of 10 users showed  $> 20\%$  improvement in time, b) 7 of 10 users showed a 25% improvement in the smoothness of the path driven and c) 8 of 10 users hit fewer obstacles. Whereas 4 users missed gates entirely with all features off, no users missed gates with all features on.

In considering the assistive features independently, the following results are noteworthy. Paths were smoother with path prediction only enabled and times were faster by 5% on average with video stabilization only enabled. Also, the two experienced users showed definite improvements with the use of the assistive technology, though the improvements were less pronounced in relative terms. It is difficult to determine to what degree this reflects reduced effectiveness of the technology with more experienced users or the fact that their unaided scores were already pretty good, and therefore harder to improve upon. The two users that did not hit fewer obstacles already hit very few so the relative improvements are less meaningful.

Users were also asked to complete an informal survey to provide their impressions of the usefulness of each control model. Users found that the path prediction feature made it easier to judge the motion of the vehicle. Times were measured to be faster with video stabilization turned on and users found that the feature made it easier to see the gates. In short, the two primary operator aids were found to be both individually useful and complementary.

## 6 Conclusion

In this work, we have produced empirical validation of the conjecture that semi-autonomous teleoperation of (even high speed) mobile robots can produce benefits both in terms of productivity and of safety. While that is not so surprising, we have conducted experiments to try to quantify the value of such improvements and we

have also assessed their value relative to each other. Our application context is that of a small, high speed, mobile robot, operating on nonflat terrain. Within that context, we have some evidence that all of the features we added were valuable. We left video compression on at all times because it is an established technique that we were not particularly interested in studying, though interesting studies have been done elsewhere. We left stability control on at all times because we felt it was too dangerous to the vehicle to do otherwise based on our preparations for the experiment. In a sense, both of these features were considered necessities for our context.

The remaining assistive features can be summarized as a control aid (path following and prediction) and a visualization aid (video stabilization). For all of our operators, regardless of skill level, these features were both individually valuable when used alone and complementary when used together. It is noteworthy that the "all features on" configuration can be viewed as a model predictive control system with a human in the loop. Not only was the path predicted well but it was presented in the context of live video of the objects in the scene. This made it possible for the operator to literally line up the robot path with the gap between obstacles, well in advance, and then refine the path based on a continuously updated, calibrated prediction of the "fit" of the robot to the gate. In this way, the problem becomes reduced to gently adjusting the path endpoint in a stable video rather than guessing the inputs required to make the obstacle gap appear in the center of the screen, at just the right time, as the robot drives through it.

After the tests, all operators expressed a preference to use the system with all features on at all times. While the level of improvement was not not extraordinary, we also did not try to maximize it. There are many realistic situations where the enhanced safety, higher speed, and more robust and precise control will all add up to an improved capacity to get a job done.

**Acknowledgments.** This CMU effort was sponsored by the U.S. Government under Other Transaction number W15QKN-08-9-000 1 between the Robotics Technology Consortium, Inc, and the Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

## References

1. Azuma, R.: A survey of augmented reality. *Presence* 6(4), 355–385 (1997)
2. Bejczy, A., Kim, W., Venema, S.: The phantom robot: predictive displays for teleoperation with time delay. In: *Proceedings of the 1990 IEEE International Conference on Robotics and Automation, 1990.*, vol. 1, pp. 546–551 (1990), doi:10.1109/ROBOT.1990.126037
3. Censi, A., Fusiello, A., Roberto, V.: Image stabilization by features tracking. In: *Proceedings of International Conference on Image Analysis and Processing*, pp. 665–667 (1999), doi:10.1109/ICIAP.1999.797671

4. Diaz-Calderon, A., Kelly, A.: On-line stability margin and attitude estimation for dynamic articulating mobile robots. *Int. J. Rob. Res.* 24, 845–866 (2005), <http://dl.acm.org/citation.cfm?id=1086867.1086870>, doi:10.1177/0278364905057865
5. Feiner, S., Macintyre, B., Seligmann, D.: Knowledge-based augmented reality. *Commun. ACM* 36, 53–62 (1993), <http://doi.acm.org/10.1145/159544.159587>
6. Funda, J.: Teleprogramming: Towards delay-invariant remote manipulation (1991), <http://search.proquest.com/docview/304008149?accountid=9902,304008149;1991;9125646;746704491;66569;141p.;Funda,Janez;30906811;English;CopyrightUMI-DissertationsPublishing1991;M1:Ph.D.;M3:9125646>
7. Kelly, A., Chan, N., Herman, H., Huber, D., Meyers, R., Rander, P., Warner, R., Ziglar, J., Capstick, E.: Real-time photorealistic virtualized reality interface for remote mobile robot control. *I. J. Robotic Res.* 30(3), 384–404 (2011)
8. Lloyd, J.E., Beis, J.S., Pai, D.K., Lowe, D.G.: Model-based telerobotics with vision (1997)
9. Milgram, P., Zhai, S., Drascic, D., Grodski, J.: Applications of augmented reality for human-robot communication. In: *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 1993*, vol. 3, pp. 1467–1472 (1993), doi:10.1109/IROS.1993.583833
10. Oshima, M., Hayashi, T., Fujioka, S., Inaji, T., Mitani, H., Kajino, J., Ikeda, K., Komoda, K.: Vhs camcorder with electronic image stabilizer. *IEEE Transactions on Consumer Electronics* 35(4), 749–758 (1989), doi:10.1109/30.106892
11. Ricks, B., Nielsen, C., Goodrich, M.: Ecological displays for robot interaction: a new perspective. In: *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, vol. 3, pp. 2855–2860 (2004), doi:10.1109/IROS.2004.1389842
12. Sheridan, T.: Human supervisory control of robot systems. In: *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 808–812 (1986), doi:10.1109/ROBOT.1986.1087506
13. Van Zanten, A., Erhardt, R., Pfaff, G.: Vdc, the vehicle dynamics control system of bosch. *Tech. Rep. 950759*, SAE (1995)
14. Zhu Xu, G., Yang, Y., Jin, J.: Camera stabilization based on 2.5d motion estimation and inertial motion filtering. In: *Proceedings of 1998 IEEE International Conference on Intelligent Vehicles*, vol. 2, pp. 329–334 (1998)

# Intention-Aware Pedestrian Avoidance

Tirthankar Bandyopadhyay, Chong Zhuang Jie, David Hsu,  
Marcelo H. Ang Jr., Daniela Rus, and Emilio Frazzoli

**Abstract.** A critical component of autonomous driving in urban environment is the vehicle's ability to interact safely and intelligently with the human drivers and on-road pedestrians. This requires identifying the human intentions in real time based on a limited observation history and reacting accordingly. In the context of pedestrian avoidance, traditional approaches like proximity based reactive avoidance, or taking the most likely behavior of the pedestrian into account, often fail to generate a safe and successful avoidance strategy. This is mainly because they fail to take into account the human intention and the inherent uncertainty resulting in identifying such intentions from direct observations.

---

Tirthankar Bandyopadhyay  
Future Urban Mobility,  
Singapore MIT Alliance for Research and Technology, Singapore 138602  
e-mail: [tirtha@smart.mit.edu](mailto:tirtha@smart.mit.edu)

Chong Zhuang Jie · Marcelo H. Ang Jr.  
Department of Mechanical Engineering,  
National University of Singapore, Singapore  
e-mail: [demianjie@hotmail.com](mailto:demianjie@hotmail.com)

David Hsu  
Department of Computer Science, National University of Singapore,  
Singapore 117417 Republic of Singapore  
e-mail: [dyhsu@comp.nus.edu.sg](mailto:dyhsu@comp.nus.edu.sg)

Daniela Rus  
Computer Science & Artificial Intelligence Laboratory,  
Massachusetts Institute of Technology, Cambridge, MA 02139  
e-mail: [rus@csail.mit.edu](mailto:rus@csail.mit.edu)

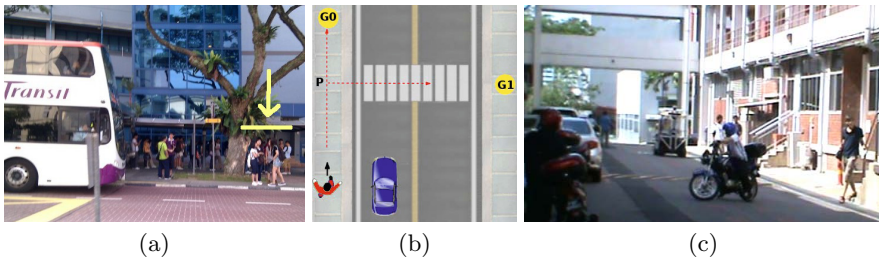
Emilio Frazzoli  
Laboratory for Information and Decision Systems,  
Massachusetts Institute of Technology, Cambridge, MA 02139  
e-mail: [frazzoli@mit.edu](mailto:frazzoli@mit.edu)

This work formulates the on-road pedestrian avoidance problem as an instance of the *Intention-Aware Motion Planning* (IAMP) problem, where the human intention uncertainty is incorporated in a principled manner into the planning framework. Assuming a set of all possible pedestrian intentions in the environment, IAMPs generate a Mixed Observable Markov Decision Process (MOMDP), (a factored variant of *Partially Observable Markov Decision Process* (POMDP)) with the human intentions being the *unobserved* variables. Solving the resulting MOMDP generates a robust pedestrian avoidance policy. In spite of the criticism of POMDPs to be computationally intractable in general, we show that with proper state factorization and latest sampling based approaches the policy can be executed online on a real vehicle on road. We demonstrate this by running the algorithm on a real pedestrian crossing in the NUS campus successfully handling the intentions for multiple pedestrians, even when they are jaywalking. In this paper, we present results in simulation to show the improved performance of the proposed approach over existing methods. Additionally, we present results validating experimentally the assumptions made in formulating the intention aware pedestrian avoidance problem.

This work presents a preliminary step towards safer and effective autonomous navigation in urban environments by incorporating the intentions of pedestrians and other drivers on the road.

## 1 Introduction

With robots venturing more into human spaces, it becomes imperative for the robots to predict the motion and intentions of people and other agents in the field for effective operation. A popular example is autonomous vehicles in urban environments which have to react with pedestrians, cyclists and other human drivers on the road. Identifying intentions is even more relevant in the case of autonomous vehicles because in many situations direct communication between the robots and people is not possible, *e.g.* between pedestrians and autonomous vehicles. A popular and simple approach of avoiding pedestrians is that of running a one step prediction interwoven with avoidance based on



**Fig. 1** Autonomous navigation in a crowded environment requires accounting for the pedestrian intentions

either potential fields [11], velocity obstacles [7] or by partial motion planning [2] *etc.* at a very high update rate. Due to the reactive approach, the success of such approaches in a crowded environment depends on the assumption that the dynamic nature of the environment will prevent the robot from getting stuck. However this assumption may not hold always. Take the scenario shown in Fig.1(a) of people standing very close to edge of the pedestrian sidewalk. While their position is close to the road, the pedestrian's intentions may not be to actually step on the road. Decisions purely based on the position and not the intention of the pedestrians may cause an autonomous vehicle to get stuck waiting for the pedestrian to cross. Human drivers however make a judgment over the pedestrian's intentions based on their activity (here waiting rather than trying to cross the road) and safely drive on.

Detecting a person's intentions or predicting his/her trajectory has been addressed using Hidden Markov Models [10, 17], non-parametric approaches like Inverse Reinforcement Learning [9], Gaussian Processes [6] and Bayesian Occupancy Filter [4] just to name a few. In most of these approaches the tools have been developed to identify a person's intention as an end goal. Only after the intention has been resolved sufficiently is the robot able to choose its action. In reality, the purpose of the robot is to navigate safely and it should only focus on resolving those pedestrian intentions relevant to the task. Integrating the intention prediction with motion planning provides a more effective approach. In many cases there might not be enough information for the intention to be completely resolved before the robot has to make a decision. Usually in such cases the robot takes actions against most likely intention [10]. However, not taking into the account the prediction limitation can lead to unsafe actions. Take the case of Fig.1(b) where the pedestrian walking along a sidewalk may move to either G0 or G1 along the marked trajectory. In essence no amount of sensing can effectively predict which goal the pedestrian is moving towards until he/she passes  $P$ . A false prediction of G0 due to motion stochasticity or sensing inaccuracy can cause potential collision or evoke emergency avoidance by the robot. In such scenarios it is imperative to not only take the prediction but also take the uncertainty associated with the prediction into the robot's decision process. This work formulates the on-road pedestrian avoidance problem as an instance of the *Intention-Aware Motion Planning* (IAMP) problem presented in [1], where the intention uncertainty is incorporated in a principled manner into the planning framework.

In many cases pedestrian motion models, their desired goals in the environment and their interaction with other entities can be learnt from the data collected by sensors in the environments [6]. We approach the problem assuming that the pedestrian motion models and their possible goals are available. The robot then has to reason about each sensed pedestrian's intention given the short observation history of the pedestrian's motion in making its decision. Even though in general exact solutions to POMDPs are intractable [14], with proper factorization of the observable and unobservable state variables

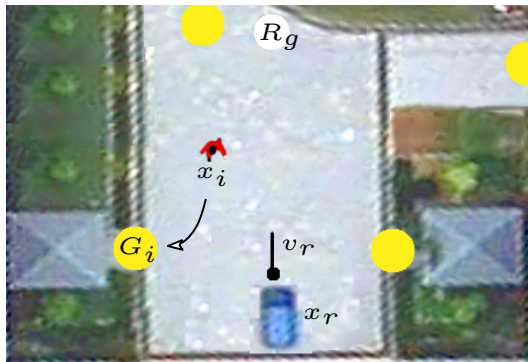


as a Mixed Observable MDP (MOMDP [13]) and sampling based approximate solutions (SARSOP [12]), we show that such an approach can be applied successfully on a real autonomous vehicle in a crowded environment (Fig.1c). Let us now state the problem statement formally:

*For a known environment  $\mathcal{W}$  and known set of possible pedestrian goal locations  $\Theta$  in  $\mathcal{W}$ , find an optimal policy  $\Pi$ , to minimize the time taken for the robot  $R$  to reach its goal  $R_g$  and avoiding collision in the presence of multiple pedestrians  $\mathbf{p}_i$  moving towards corresponding goals,  $g_i \in \Theta$  where all  $g_i$  are unknown to the robot.*

## 2 Technical Approach

The problem of avoiding a single pedestrian  $\mathbf{p}_i$  in the environment  $\mathcal{W}$ , can be described by a few variables, the robot’s state: location  $\mathbf{x}_r$ , velocity  $\mathbf{v}_r$ , and the pedestrian state: location  $\mathbf{x}_i$ , his/her intention  $g_i$  (here the goal location). In general,  $\mathbf{x}_r$ ,  $\mathbf{v}_r$  and  $\mathbf{x}_i$  can be estimated from a variety of sensors following an observation function  $Z : p(o|\mathbf{x}_r, \mathbf{v}_r, \mathbf{x}_i)$ . However under acceptable sensing accuracy we assume them to be fully observed. This helps us reduce the computational complexity of the problem and we show experimentally in Sec.3.2 that this assumption is indeed reasonable. On the other hand, there is no “intention sensor” for  $g_i$  and it is treated as unobservable variable. Let  $x \in \mathcal{X}$  represent the observed state variables and  $g_i \in \mathcal{Y}$  denote the unobserved pedestrian’s intention. The robot can pick actions  $a \in A : \{cruise, accelerate, decelerate\}$ , to control the vehicle. The choice of using such high level acceleration commands as compared to direct velocity commands mimics human driver behavior who control the brakes and acceleration rather than reason about the actual velocities. The robot is rewarded by a function  $R$  when it successfully navigates to its desired goal and is penalized for time delay and collision.



**Fig. 2** Pedestrian avoidance scenario in the quadrangle environment shows the relevant variables in the formulation

We formulate the pedestrian avoidance problem in autonomous navigation by a discrete MOMDP:  $\mathcal{M}_i : (\mathcal{X}, \mathcal{Y}, A, O, Z, T_X, T_Y, R, \gamma)$ , where  $O$  is the set of all possible observations and  $\gamma$  the discount factor. The transition function  $T_X(x, g_i, a, x') : p(x'|x, g_i, a)$  gives the transition of the observed variables from the current observed state  $x$  into the future observed state  $x'$  upon taking the action  $a$  in the state  $(x, g_i)$ . This incorporates the pedestrian and the robots motion models.

We assume the motion of the pedestrian to follow a trajectory towards its intended goal in a shortest possible path, a simplified model of social potential fields [8]. Note that the pedestrian may not follow the exact shortest path due to personal preferences, distracted walking, avoidance of other pedestrians and vehicles on the road. These unknown variations are modeled with the uncertainty distribution over the intended direction.

$$\mathbf{p}_i' = \mathbf{p}_i + v_p \Delta t \hat{\mathbf{n}}_i$$

where  $\hat{\mathbf{n}}_i \sim N(n(g_i), \sigma)$  is the heading of the pedestrian sampled from a distribution with the mean direction towards  $g_i$  and  $v_p \sim N(V_p, \sigma_2)$  is the velocity of the pedestrian sampled from a mean pedestrian velocity  $V_p$  calculated from interactions with the goal, the environment and robot position.

The robot's own motion model follows from a velocity controller. In this paper we consider the robot to be constrained along a single lane. This simplifies the analysis without loss of generality in the discussion.  $\mathbf{v}_r$  is the speed of the robot along the road.

$$\begin{aligned} \mathbf{x}_r' &= \mathbf{x}_r + \mathbf{v}_r \Delta t + \epsilon_1 \\ \mathbf{v}_r' &= \mathbf{v}_r + \dot{\mathbf{v}}_r(a) \Delta t + \epsilon_2 \end{aligned}$$

where  $\dot{\mathbf{v}}_r(a)$  denotes that the actual acceleration of the vehicle is influenced by the action taken in the previous step. The errors  $(\epsilon_1, \epsilon_2)$  are determined by the vehicle's controller characteristics.

The transition function  $T_Y(x, g_i, a, x', g_i') : p(g_i'|g_i)$ , shows the transition over the pedestrian intentions  $g_i$ . In general the pedestrian may change the intentions midway during execution and can be handled by our formulation. However in our analysis and results in this paper, we assume that during the course of the problem, pedestrian's intentions do not change. Interested readers are referred to [1] for a more general formulation of the problem of intention aware motion planning.

The focus of this paper is to formulate the problem of pedestrian avoidance as an intention aware motion planning problem and to show the effectiveness and feasibility of such an approach to autonomous navigation on the road. Once the problem is formulated as a MOMDP, any solver can be used to solve for the policy. In our case we use SARSOP [12], a leading point-based approximation algorithm, to solve our MOMDP model.

The corresponding MOMDP belief space is a union of lower dimensional belief subspaces over the goal set  $\Theta : \{g_i\}$  at each observed state  $x$ .

$$\mathcal{B} = \bigcup_{x \in \mathcal{X}} \mathcal{B}_\Theta(x)$$

Here the dimensionality of the belief subspace  $\mathcal{B}_\Theta$  is equal to the cardinality of the goal set  $\Theta$  less 1, clearly reducing the computational complexity of the problem. The MOMDP value function is represented by a collection of alpha vector sets  $\{\Gamma_\Theta(x) : x \in \mathcal{X}\}$ .

$$V(x, b_\Theta) = \max_{\alpha \in \Gamma_\Theta(x)} \{\alpha \cdot b_\Theta\} \quad (1)$$

The belief is initialized to a uniform distribution over all goals. The online execution is performed in two steps: *action selection* and *belief update*. The action corresponding to the alpha vector that maximizes Eq.1 is chosen based on the current belief,  $b_\Theta$ . The robot gathers observations resulting from its actions and updates the belief value (Eq.2) and the process repeats itself, until the robot reaches the goal.

$$b'_\Theta(g_i) = \eta T_Y(x, g_i, a, x', g'_i) b_\Theta(g_i) \quad (2)$$

$\eta$  being the normalizing constant.

### Handling Multiple Pedestrians

A naive way of adding multiple pedestrians directly into the state space causes the problem to become intractable quickly. In addition since pedestrians are detected asynchronously, the time of detection has to be also incorporated further adding additional dimensions to the problem space. To avoid this we address each pedestrian independently. Once a pedestrian is detected, a new MOMDP problem is generated with uniform beliefs on the possible goals. This requires maintaining belief of each pedestrians according to Eq.2. Using the same policy,  $\Pi$ , different actions are chosen based on the unique belief state for each pedestrian. In general there can be many ways of combining these actions, we choose a simple conservative approach to pick the safest action based on a safety metric  $S : \mathcal{X} \times A \rightarrow \mathbb{R}^+$ .

Let  $\mathcal{M}_i$  denote the problem generated due to  $\mathbf{p}_i$  and  $a_i$  denote the current action chosen for  $\mathcal{M}_i$ .

$$a = \arg \max_i \{S(a_i)\} \quad (3)$$

In general the safety metric can be defined on relative velocity or conservative lane changing or safety distance. In our campus environment the robot is constrained to move along a fixed lane, the only variability being controlling the speed. The safety metric we use is inversely mapped to the braking distance for the vehicle at the expected speed resulting from the decision and the speed controller.

### 3 Experiments and Results

We present the result of our intention aware pedestrian avoidance approach on the autonomous vehicle. Our autonomous system is a Yamaha golf-cart with drive by wire capability. Two onboard computers running ROS [16] on intel i5 processors with 8Gb RAM execute various perception, planning and control algorithms. The vehicle is programmed to run at a maximum speed of 2m/s autonomously in a section of the NUS campus and can seat upto 2 people (Fig.3a). The vehicle is equipped with 3 LIDARs and a webcam. The LIDARs are used for localization while the combination of LIDAR and camera data is processed to identify pedestrians. More details about the system and its architecture is presented in [3].

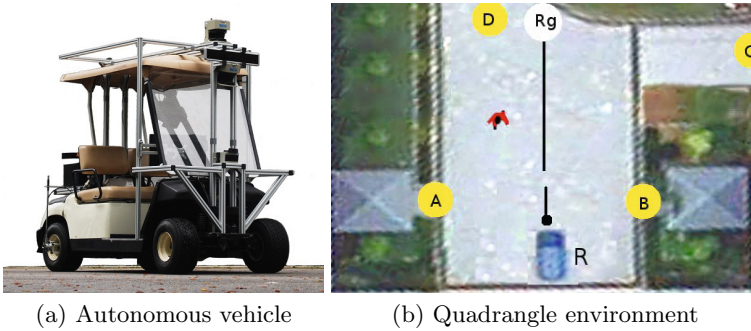


Fig. 3 Experimental Environments

#### 3.1 Qualitative Comparison

Fig.3b shows the quadrangle in the engineering faculty of NUS where the students enter through (C & D). (A & B) are entrances to a shaded study area. Let us analyze the policy generated for this environment. Fig.4 plots the decision executed by the robot at  $R$  moving forward with velocity 1m/s for various positions of the pedestrians for a particular belief value over goals (displayed on the goal regions). The '+' sign represents the decision to accelerate when a pedestrian is present at that particular location, a '-' sign deceleration and '.' represents a decision to cruise accordingly. Fig.4(a & b) compares the spatial distribution of decision when the belief value over goals changes from uniform to being higher on left (goals, A & D). As the robot becomes more confident about the pedestrian's intention, it's decisions are no longer overly conservative as shown by an increase in '+', helping it to navigate a more crowded environment. Note however that at locations where pedestrians stepping into the robot's path would require the robot to execute emergency avoidance, (marked by red box in Fig.4b) the robot's decisions are more cautious maintaining speed ('.') rather than accelerating ('+'). Note however that the lowest decision in the marked area is to accelerate. A quick

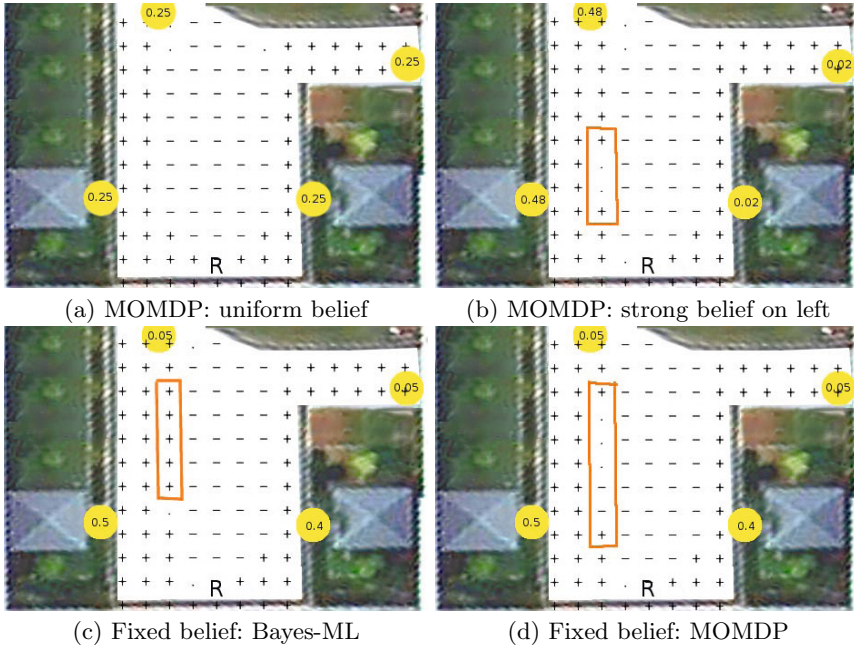


Fig. 4 Comparison

calculation of the location and relative speed shows that by accelerating, the robot can overtake the pedestrian in case it tries to move towards B and hence is a safe decision.

Algorithm	Time (s):	Accident (4500 runs)
	Mean (S.D)	
Bayes-ML	9.4 (6.4)	4.4 %
IA-MOMDP	9.6 (6.5)	3.4 %

Fig. 5 Performance comparison in simulation runs in quadrangle environment

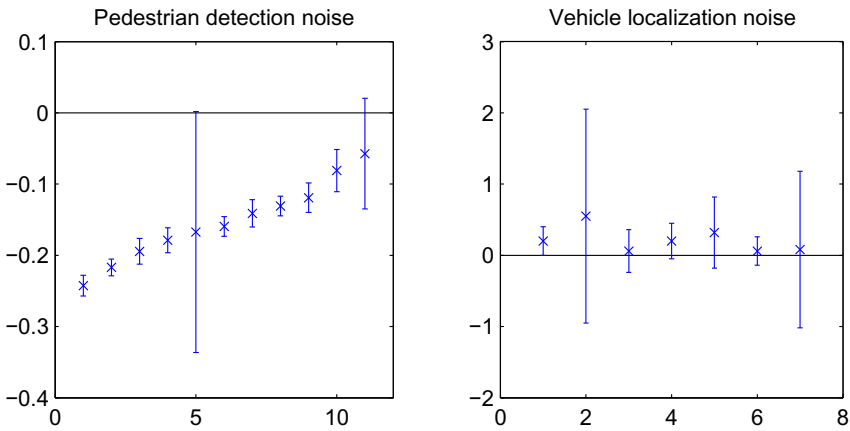
on goals A and B are both high A being marginally higher. Since Bayes-ML does not take B into consideration, it ignores, as shown by ‘.’ decision, the pedestrians at positions marked by red box. However, there is a significant chance of pedestrian at this location to go towards B. The MOMDP policy taking into account this possibility leads to a more conservative decision for the region under the same belief. To test how effectively does such a policy fare against Bayes-ML, we ran around 4500 simulations for pedestrians

We next compare the result of a simple maximum likelihood (Bayes-ML) approach where the robot first picks the most likely goal and subsequently chooses an action based on the MDP policy learnt for that particular goal. We first look at a case in Fig.4c where the prediction

starting from random locations assigned to random goals. The results are shown in Table.5. We see that for similar time taken by the robot the MOMDP policy encounters significantly lower simulated collision states.

### 3.2 Perception Accuracy

Pedestrians are detected using a single webcam calibrated and mounted on top of a SickLMS 200 in front of the autonomous vehicle. The range data from the laser is clustered based on spatial proximity and a HoG person classifier [5] is run through the corresponding sub-image to label the cluster as a pedestrian or non-pedestrian [3]. Once sufficient confidence is reached that a cluster belongs to a pedestrian, a proximity based nearest neighbor data association is applied to track the pedestrian. The laser system runs around 50Hz while the vision runs at 15Hz. We performed controlled experiments where a person stood at known locations and the pedestrian detector was initiated and the data recorded. The false negative rate from the vision system was 20% over the number of frames computed. However this only affects the initialization phase of pedestrian detection since, once detected the range based data association is able to reliably track the pedestrian. On average it took around 0.46s to reliably detect the pedestrian. The mean distance error in detection is plotted in Fig.6a for various control positions of the pedestrian in order of increasing distance (5m to 25m at different angles) from the sensor. The error in distance estimate is under 30cm. The data point 5 was close a background wall ( 2m) which created a larger variation in the estimate error. Fig.6b shows the localization estimate of the vehicle



**Fig. 6** Perception noise: the vertical axis shows error rate in meters, while horizontal axis marks the data point entries. The plots show that the noise is acceptable in assuming the pedestrian and vehicle positions as observed variables.

following a simple curb based localization approach [15]. The error is under 1m which is quite acceptable to a vehicle of footprint 3m x 1.5m. Note also that the pedestrian is detected relative to the vehicle in the map. Thus the localization error only affects the uncertainty of the pedestrian's goal estimate and not the proximity estimate of the pedestrian w.r.t. the robot which is crucial for the pedestrian safety.

The results clearly validate the assumption of the MOMDP formulation where the only unobservable variable is the pedestrian's intentions, while the position of pedestrians and robot are sensed within acceptable accuracy.

### 3.3 Stationary Vehicle: Belief Tracking Experiment

In order to test the results of the MOMDP policy on a real system, we first kept the robot stationary at  $R$  in the quadrangle environment (Fig.3b), look at the decisions being made.

The snapshots in Fig.7(a & b), display the belief and the decision generated. A bar graph is plotted associated for each pedestrian, the left being the belief that the pedestrian is going to a goal to the left of the robot's heading and the right correspondingly for the goals to the right. The red/green horizontal bar on the top denotes the decision made by the robot to STOP/GO w.r.t. the pedestrian.

The series of snapshots in Fig.7a, shows a couple of pedestrians (non-actors) walking in the quad. Each pedestrian is assigned a uniform belief upon detection. As more information is received the belief gets updated and the robot takes decisions accordingly. Note that the belief at snapshot (3-a) is higher over the goals (A & D) due to the stochasticity of the trajectory being followed. However, there is still a chance for him to move towards  $C$ . Taking this uncertainty into account, the decision of the robot is to STOP, which proves to be the right decision eventually.

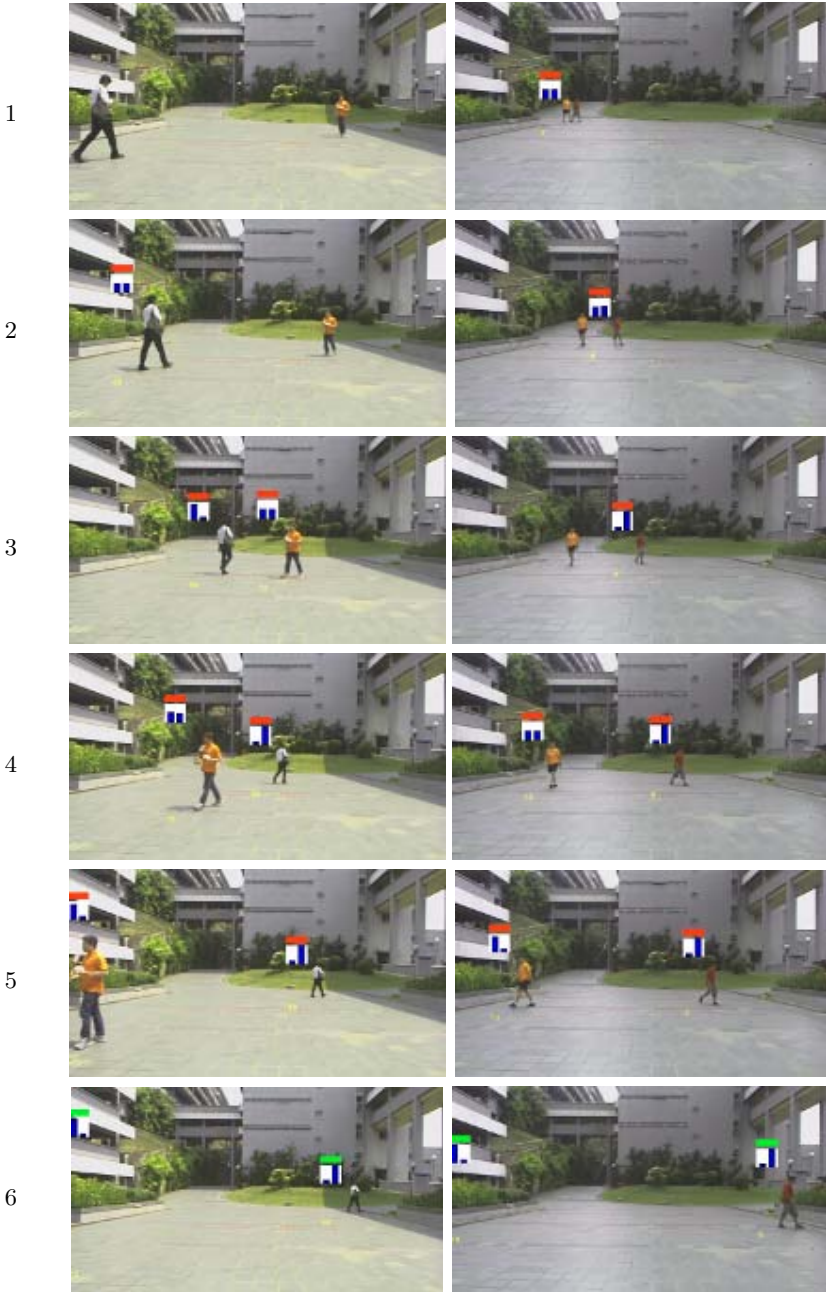
While we formulate the problem for a single pedestrian, clearly a group of pedestrians moving cohesively generates exactly the same avoidance problem. Our cluster based approach tracks the belief over the group of pedestrians as would on a single pedestrian thereby avoiding the explosion of the problem space with additional pedestrians. The ability to quickly detect and generate a MOMDP avoidance problem makes the approach robust to splitting and merging over the clusters. Snapshots in Fig.7b show a group of pedestrians splitting to move to different goals. At snapshot (4-b) we see that the split is detected and a new problem generated and resolved for the left pedestrian.

The videos of all the runs and more experimental results are available at ( <http://web.mit.edu/tirtha/Public> ). We see that the pedestrian tracker is able to keep detect, instantiate and maintain the beliefs and decisions over various pedestrians. This is even robust to temporary occlusion, merging and splitting of people in the crowd.



(a) belief tracking: cross

(b) belief tracking: split



**Fig. 7** Experimental Results (Best viewed in color). The videos of all the runs and more experiments are available at ( <http://web.mit.edu/tirtha/Public> ).





(a) A busy pedestrian crossing scene in the NUS campus that the vehicle has to navigate



(b) Avoidance for a single pedestrian (on-board camera picture merged at different times to show the evolution of the belief)



(c) Avoiding multiple pedestrians. (goals reordered and labeled differently than (b)). Each pedestrian generates a avoidance MOMDP and the beliefs are shown based on asynchronous set of observation history for each pedestrian.

**Fig. 8** Pedestrian crossing experiment

### 3.4 Moving Vehicle: Pedestrian Crossing Experiment

We tested the algorithm on a real pedestrian crossing shown in Figure 8(a). The robot not only needs to detect the pedestrians and their intentions on the pedestrian crossing but also deal with jay walkers, a common phenomena in a campus environment. Figure 8(b) shows the robot interacting with such a pedestrian. The analysis is presented for a single pedestrian for clarity. The windows in clockwise direction from top left show the onboard camera view, the simplified environment representation used for solving the policy (the orange trajectory shows the pedestrian track in this environment and the green box shows the robot's position), the speed controller command velocity generated and the belief plots showing the evolution of the belief over pedestrian's goal as the pedestrian moves towards its intended goal. In the belief plot the color of the graph encodes the goal id.

Snapshots of the pedestrian at different times are merged into one camera window to show the development of the the belief. Notice that when the pedestrian is detected, the initial belief over the goals are equal. Due to high initial uncertainty the robot comes to a stop to wait for the pedestrian and collect more information about its intention. As the pedestrian starts moving across the road, the belief values over G2 and G1 increase while that of G0 and G3 drop as it is more likely that the pedestrian wants to move towards the other side. However there is a chance that the pedestrian will turn back and so the vehicle remains stationary. Slowly the belief over G1 grows stronger and that on G2 drops as the pedestrian starts moving diagonally. As soon as the belief over G1 is sufficiently large and the pedestrian is sufficiently out of the vehicle's path the vehicle starts moving. Figure 8(c) shows the vehicle responding to multiple pedestrians. Note that the goals are ordered and labeled differently however the environment setup and analysis is the same.

## 4 Conclusion

The paper presented an approach to avoid pedestrians on the road by identifying their intentions based on their actions on the road. We show in Sec.3.1, that trying to analyze the pedestrian's intentions helps in a better response to pedestrians than a naive distance based reactive approach (Fig.4(a&b)). Also maintaining the uncertainty over pedestrians goals gives a more conservative avoidance policy (Fig.4(c&d)) which leads to a lower simulated collision rates (Table.5). We also show that with proper factorization of the problem in terms of observed and unobserved variables, we reduce the computational complexity making it feasible to run the policy online on a realistic scenario. We demonstrated this on a vehicle interacting with multiple pedestrians in a real pedestrian crossing.

The main assumption in this work is the availability of pedestrian motion models and their finite intention models. An immediate extension of this work being addressed currently is to try to learn the intention models from the data collected and integrate it into the planning paradigm.

Even though the approach was presented for pedestrians on the road, such an approach could also be utilized to identify the intention of other human drivers on the road and could lead to a principled way of interacting safely with other drivers on the road.

## References

1. Bandyopadhyay, T., Won, K.S., Frazzoli, E., Hsu, D., Lee, W.S., Rus, D.: Intention-aware motion planning. In: Frazzoli, E., Lozano-Perez, T., Roy, N., Rus, D. (eds.) *Algorithmic Foundations of Robotics X*. STAR, vol. 86, pp. 475–491. Springer, Heidelberg (2013)
2. Benenson, R., Petti, S., Fraichard, T., Parent, M.: Integrating perception and planning for autonomous navigation of urban vehicles. In: *Proceedings of the International Conference on Intelligent Robots and Systems*, Beijing, pp. 98–104. IROS (October 2006)
3. Chong, Z.J., Qin, B., Bandyopadhyay, T., Wongpiromsarn, T., Rankin, E.S., Ang Jr., M.H., Frazzoli, E., Rus, D., Hsu, D., Low, B.K.H.: Autonomous personal vehicle for the first- and last-mile transportation services. In: *Proceedings of the International Conference on Cybernetics and Intelligent Systems and International Conference on Robotics, Automation and Mechatronics*, Qingdao, China, September 17–19 (2011)
4. Coue, C., Pradalier, C., Laugier, C., Fraichard, T., Bessiere, P.: Bayesian occupancy filtering for multitarget tracking: An automotive application. *The International Journal of Robotics Research* 25(1), 19–30 (2006)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *CVPR*, pp. 886–893 (2005)
6. Ellis, D., Sommerlade, E., Reid, I.: Modelling pedestrian trajectory patterns with gaussian processes. In: *International Conference on Computer Vision Workshops*, pp. 1229–1234 (2009)
7. Gayle, R., Moss, W., Lin, M.C., Manocha, D.: Multi-robot coordination using generalized social potential fields. In: *International Conference on Robotics and Automation* (2009)
8. Helbing, D., Buzna, L., Johansson, A., Werner, T.: Self-organized pedestrian crowd dynamics and design solutions: Experiments, simulations and design solutions. *Transportation Science* 39(1), 1–24 (2005)
9. Henry, P., Vollmer, C., Ferris, B., Fox, D.: Learning to navigate through crowded environments. In: *Proceeding of the International Conference on Robotics and Automation* (2010)
10. Kelley, R., Nicolescu, M., Tavakkoli, A., Nicolescu, M., King, C., Bebis, G.: Understanding human intentions via hidden markov models in autonomous mobile robots. In: *International Conference on Human Robot Interaction*, Amsterdam, Netherlands, March 12–15. IEEE (2008)
11. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* 5(1), 90–98 (1986)

12. Kurniawati, H., Hsu, D., Lee, W.: Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In: Proceedings of the Robotics: Science and Systems (2008)
13. Ong, S., Png, S., Hsu, D., Lee, W.: Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research* 29(8), 1053–1068 (2010)
14. Papadimitriou, C., Tsisiklis, J.: The complexity of Markov decision processes. *Mathematics of Operations Research* 12(3), 441–450 (1987)
15. Qin, B., Chong, Z.J., Bandyopadhyay, T., Ang, M.H., Frazzoli, E., Rus, D.: Curb-intersection feature based monte carlo localization on urban roads. In: Proceedings of the International Conference on Robotics and Automation (2012)
16. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: Ros: an open-source robot operating system. In: Proceedings of the International Conference on Robotics and Automation (2009)
17. Vasquez, D., Fraichard, T., Laugier, C.: Incremental learning of statistical motion patterns with growing hidden markov models. *Transactions on Intelligent Transportation Systems* 10(3), 403–416 (2009)

# The UBC Visual Robot Survey: A Benchmark for Robot Category Recognition

David Meger and James J. Little

**Abstract.** Recognizing objects is a fundamental capability for robotic systems but comparing algorithms on similar testing situations remains a challenge. This makes characterizing the current state-of-the-art difficult and impedes progress on the task. We describe a recently proposed benchmark for robotic object recognition, named the UBC Visual Robot Survey, which is a robot-collected dataset of cluttered kitchen scenes. The dataset contains imagery and range data collected from a dense sampling of viewpoints. Objects have been placed in realistic configurations that result in clutter and occlusion, similar to common home settings. This data and accompanying tools for simulation from real data enable the study of robotic recognition methods. They specifically allow focus on specific concerns in robotics such as spatial evidence integration and active perception. We describe the method used to produce the dataset in detail, a suite of testing protocols and the current state-of-the-art performance on the dataset.

## 1 Robot Object Category Recognition

The locations and semantic labels of objects in the world are essential for robots during many real-world tasks. Estimating this information from a platform's sensors, known as *robot object category recognition*, is a challenging task due to: the wide variety of objects that share a semantic label; the indirect connections between object labels and raw sensory data (typically features are extracted and passed through a classification function); as well as clutter and occlusion in the world that leads to missing information. There are several uniquely robotic aspects to the problem,

---

David Meger · James J. Little  
The University of British Columbia  
201-2366 Main Mall  
Vancouver, Canada  
V6T 1Z4  
e-mail: {dpmeger, little}@cs.ubc.ca

including the need to actively move the robot's sensors in order to obtain informative viewpoints and the possibility to fuse information across sensing modalities and spatial locations.

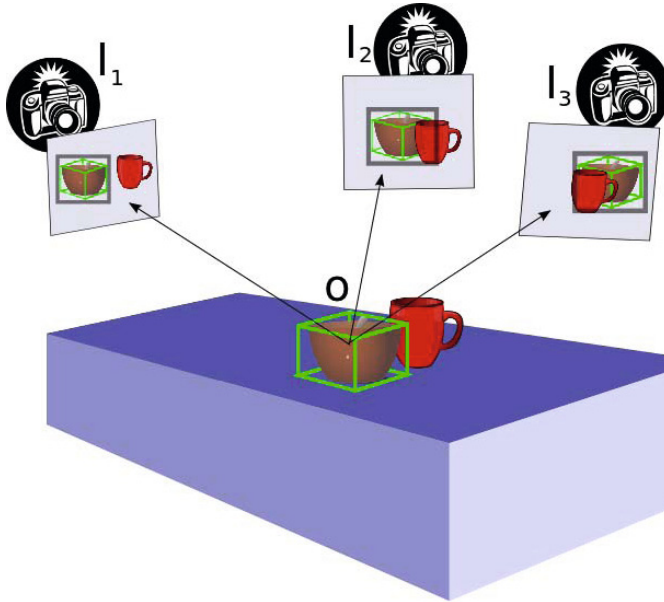
This paper describes a recently-established evaluation benchmark specifically tailored to the robot recognition problem, named the University of British Columbia Visual Robot Survey (UBC VRS). It has been motivated by the example of object recognition in Computer Vision, where rapid progress has been made through standardization around the Pascal Visual Object Classes (VOC) challenge [7], a high participation, yearly contest of ever-increasing difficulty, and benchmark tasks for distinguishing large numbers of object categories, such as Caltech 101 [8] and 256 [11]. Several robotics challenges exist, including the Semantic Robot Vision Challenge (SRVC) [1] and Solutions in Perception Challenge [2], which compare near real-time systems on robot recognition tasks at a particular venue once per year. These contests capture the full scope of robot recognition, but the requirement to travel to the contest location in order to participate limits their accessibility.

Several datasets based on RGB-D data such as that available from the Microsoft Kinect have recently been released. For example, the Berkeley 3D Object Dataset [14] is composed of many indoor scenes contributed by the community through *crowd-sourcing* and annotated by humans. While there are more images and more object types in this dataset than the one we present, each scene is captured from only a single viewpoint, which does not allow exploration of recognition methods involving robot motion. The Multi-View RGB-D Object Dataset by Lai *et al.* [15] includes a large number of scenes containing a single object on a turn-table, captured with an image-depth sensor from a number of viewpoints, as well as a smaller number of scenes containing multiple objects captured with hand-held trajectories. This dataset allows for rapid iteration and direct comparison between methods, but the single trajectory through each scenes precludes active perception.

The contribution of the UBC VRS evaluation benchmark is to allow the unique aspects of the robot recognition problem to be explored with statistical significance and repeatability. These aspects include: the use of 3D and visual sensory data; the ability to actively control the robot's path and influence the series of images obtained; and the challenge of cluttered scenes present in real environments.

## 2 UBC Visual Robot Survey Benchmark

While performing active perception, a robot moves through its 3D environment, controlling its own position as well as the orientation of its sensors. We have attempted to capture all information necessary to simulate (with real data) this perceptual experience for recognition systems at both training and test time. To this end, our database is created by recording the sensory experience of a physical robot following a trajectory that passes through a dense sampling of poses within a number of environments. The poses are registered to a consistent coordinate frame using a visual fiducial target of known geometry. A human manually annotates the locations of all object instances from several categories, both in the 3D coordinate frame



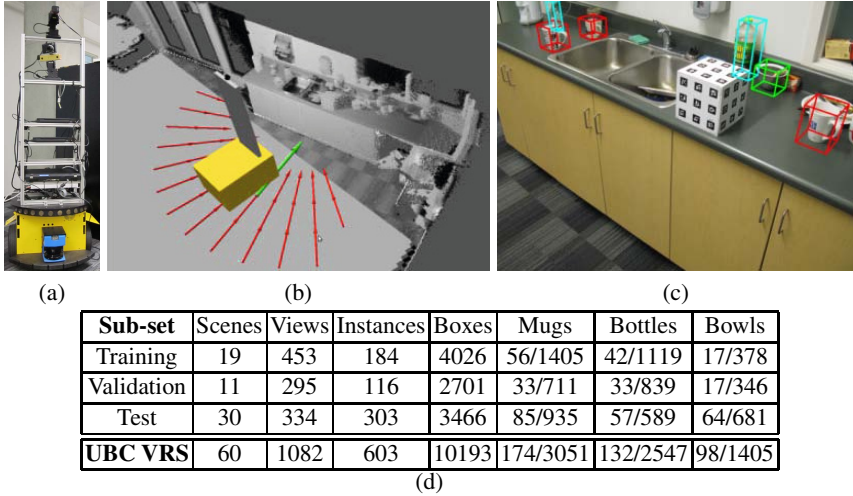
**Fig. 1** An overview of the information contained in the UBC VRS dataset. Our robot collects a number of images of a scene. Geometric registration allows 3D object information to be projected into each image. Accurate 2D bounding box annotations are also provided.

and within each collected image. Figure 1 illustrates the final product of this procedure, which is robot sensor data from a set of viewpoints of each scene, along with geometric knowledge linking that data into a common frame, and annotations both in 3D and 2D.

During training and testing of robot recognition algorithms, the recorded data is provided to recognition algorithms by a simulator that mimics a robot’s sensing and response to control input. We refer to this procedure as *simulation with real data*. Except for small limitations due to sampling discretizations, this perceptual experience is identical to the one a novel robot would experience in the same environments. This allows realistic evaluation of robot object recognition performance. Details on each stage in the process are provided in the remainder of this Section.

## 2.1 Robotic Data Collection

The sensor data that comprises the UBC VRS dataset was collected with the Curious George robot that is described in [16] and is shown in Figure 2(a). During data collection, the robot moves through a dense set of poses covering the space of possible visual experiences. We achieved this by planning a path consisting of three concentric circles. Along each circle, stop-points were located at an angular spacing of at most ten degrees (in some cases at a finer resolution). When the robot reached each



**Fig. 2** The UBC VRS Dataset. (a) The Curious George robot platform used for data collection. From top to bottom, the sensors include a tilting laser range-finder, digital camera, stereo camera and fixed laser range-finder. (b) A sample point cloud, and poses from the survey path followed by the robot. (c) A sample image with 3D wire-frames projected to display user-annotated ground truth volumes. (d) Summary statistics of the annotations available for the UBC VRS database. The final 3 columns represent the (unique instances / number of bounding boxes) that are present for the specified category.

stop point, it turned to face the center of the scene and it collected a single reading from each of its sensors. Figure 2(b) shows a sample path in one environment.

In the ideal case, this data collection method ensures that real sensor readings are available from a viewpoint within 5 degrees of any pose requested by a simulator. However, constraints of our robot and the environments prevented a complete sampling. Factors such as building layout, uneven floors, and furniture obstacles caused the robot’s navigation routines to skip some of the requested stop-points. Data from these skipped viewpoints is not available to recognition methods, which is also the case for real robotic systems exploring an environment. Recognition methods must therefore be robust to this realistic property of the dataset. Figure 2(d) displays the final number of images and scenes that were collected.

The Curious George robot has a variety of sensors suitable for object recognition. Images from the robot’s high-resolution digital camera capable of 10 megapixel imaging are down-sampled to 1600 by 1200 pixel resolution and stored, to balance overall data size with sufficient resolution to capture objects in detail. A planar laser range-finder was tilted with a continuous periodic command to capture an entire 3D sweep of the scene from each viewpoint. The set of scans was then assembled to form a cloud comprised of roughly 500,000 individual points. Each point is represented with a 3D (X,Y,Z) position as well as an intensity value measured by the laser. During data collection, the relative positions of the robot’s sensors were



calibrated as often as possible. This involved estimating the transformation relating the camera to the laser with the so-called Laser-Camera Calibration Toolbox [21]. However, a moderate degree of calibration error remains a factor, as is the case for many commodity robotic platforms.

## 2.2 Geometric Registration

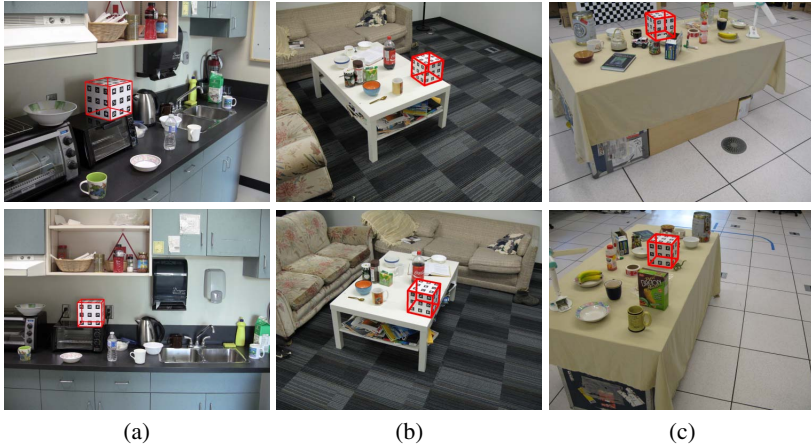
When a physical robot platform explores an environment, it has access to several forms of sensor feedback that can be used to determine its position. Also, it can actively control its position by issuing movement commands. In order to replicate this situation as closely as possible when performing recognition from our pre-recorded data, all information in the database is registered to a common base frame. First, the set of camera poses is registered using automatically detected fiducial marker points that correspond to known 3D target geometry to solve for the camera poses in a global frame. Then, the pre-calibrated relative sensor transformations (camera to X) are applied to globally register the remaining set of sensors. Using this common registration, both path information and robot control can be simulated, in combination with the real sensor data. This Section will describe the process for registering the camera poses in detail.

The cube-shaped target displayed in our example images (e.g. Figure 2(c)) is comprised of ARTag visual fiducial markers [10] and we have manufactured the cube target with precise 3D geometry. The ARTag library provides a marker detection scheme with virtually zero false positives that simultaneously localizes the corners of the fiducial patches in the image with sub-pixel accuracy. Each detected image location provides a 2D to 3D constraint on the extrinsic camera parameters (pose) using the typical pinhole camera projective equation:

$$\alpha \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where:  $x$  and  $y$  are the image coordinates of the detected corner pixel;  $K$  is the known intrinsic camera calibration containing the focal length, offsets and skew;  $R$  and  $t$  are the unknown rotation and translation which we seek;  $X$ ,  $Y$ , and  $Z$  are the 3D coordinates of the corner point using the known layout of the target; and  $\alpha$  represents projective scale. Numerous points are required to uniquely determine the camera pose, and our target provides between 36 and 108 visible corners, depending on the viewpoint. This yields a highly over-determined system. We estimate the solution using an approach similar to camera calibration methods such as [20], which involves making an initial guess using homography constraints (which exploits the known planarity of the target's faces), and then by refining the estimate using the Levenberg – Marquardt algorithm to minimize re-projection error.

We have validated this registration method on a number of test images by projecting known 3D points (e.g. a cube corner, or another point we have physically



**Fig. 3** Example results of automated geometric image registration. Each column holds two views of the same scene. Our system uses the estimated camera positions relative to a global frame along with previous intrinsic calibration estimates to render a wire-frame of the extents of the calibration target (shown in red where colour is available) into each view. Accurate alignment of the wire-frame to image content indicates accurate registration.

measured in 3D) into each of the images and manually observing the error in re-projection. The registration is typically accurate to within a pixel with the maximum error on the order of several pixels. Figure 3 illustrates the registration accuracy in a set of example images. Registration information is stored with the raw sensory data and both are used during annotation and simulation of robot motion for testing.

### 2.3 Object Annotation

In order to evaluate the performance of recognition algorithms, a human has annotated each scene and image in the dataset. We seek to describe objects both in 3D in the common registration coordinate frame as well as in 2D in each image. Annotating this information is a time-consuming process, but we have leveraged the registration information described above to ease the manual burden. We provide the annotator with a software tool to triangulate a number of 2D object points to locate the 3D centroid, a set of controls to fine tune the object’s orientation and scale in 3D, and a mechanism to refine an automatically initialized 2D bounding box for each object. We continue by providing more detail on both the 2D and 3D annotation procedure.

As mentioned previously, each image in our dataset has been accurately registered into a common coordinate frame. This allows projection of 3D information into each image, and it also permits triangulating a set of image points. The first step in our annotation process is for a human to mark a central and identifiable feature on an object in 3 or more images. We then solve for a 3D point that falls closest

to the rays through each marked pixel. As described in Hartley *et al.* [12], this involves finding the smallest singular vector of a matrix,  $A$ , formed by stacking rows that express constraints induced by the projection matrix and marked image points:

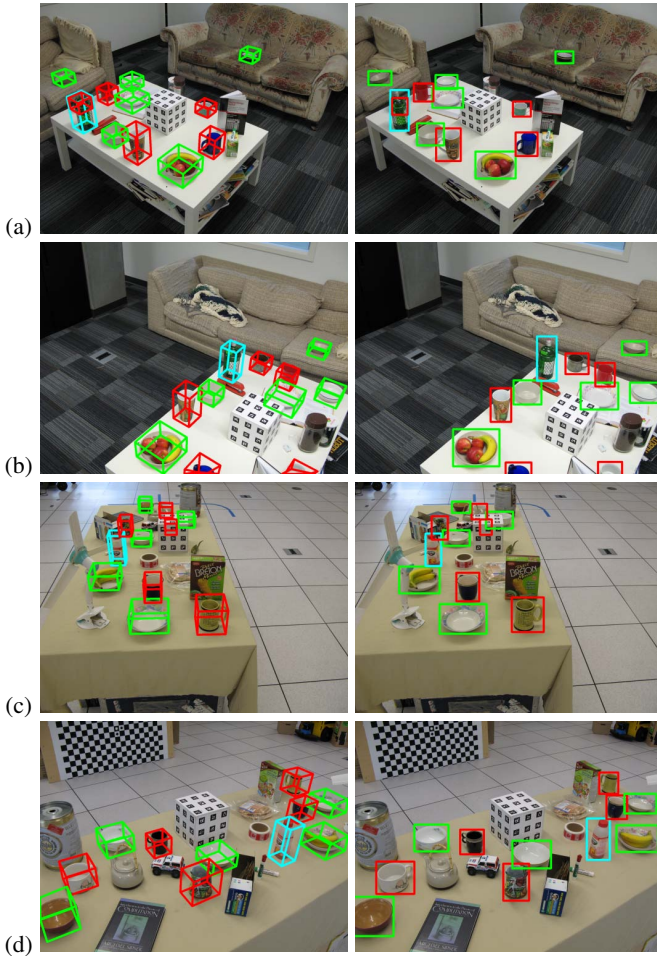
$$A \equiv \begin{bmatrix} xP_3 - P_1 \\ yP_3 - P_2 \end{bmatrix} \quad (2)$$

Here  $P$  is the three row by four column projection matrix combining extrinsic and intrinsic parameters:  $P \equiv K [R|t]$  and underscore notation indicates selecting a particular 1-indexed row of the matrix. The result of triangulation is an estimated 3D center point for the object. Our annotation tool instantiates a 3D object region composed of a 3D centroid initialized to the triangulated point, a 3D scale initialized to be the mean size of the object category, and an azimuth angle (that is rotation around the *up* or  $Z$  axis) initialized to zero. The annotator is then able to refine each dimension, but we have found that, if the image points are specified accurately at the outset, there is little extra effort required beyond specifying the true object orientation. Upon approving of all properties of the 3D annotation, the annotator saves the object volume and this is recorded along with the sensor data and registration information to be available at test time.

Our annotators have also provided 2D annotations of objects in every image in the dataset. Our 2D annotations share the format used by Pascal and other recognition challenges. That is the bounding box of the object is drawn, with extents tight to the image content. The object's category type is recorded as well as additional meta-information such as that the instance may be *difficult*, in that it is an uncommon representative of the class (e.g. a toy coffee mug in the shape of a cartoon character is a difficult mug), or that the instance is *occluded* in the image. The previously created 3D annotations are leveraged to expedite the process of creating 2D annotations. Volumes are projected into every image in which they are visible, and a bounding box that encompasses the 3D corners is created. The annotator's task is then simply to refine the precise image locations and meta-information values, rather than having to create each bounding box. This saves significant effort and reduces the probability that an image region will be missed due to human error.

At this stage, the annotator also often makes small adjustments to the 2D bounding box to ensure that it is pixel-tight to the underlying image content. This hand-adjustment is needed because we project imprecise shape models (a box-shaped 3D volume, rather than the object's true shape), and to account for any small errors introduced by 3D to 2D projection. Once again, when the annotator is satisfied with the quality of the data, the 2D box and meta-information are saved to the database.

The code and tools of our labeling pipeline can be re-used for any series of moderately well-registered images (such as video sequences, well-calibrated vehicles possessing accurate inertial positioning and a camera, or sets of highly overlapping photographs). It has been made open-source to the community and is available online along with the dataset, as described below.



**Fig. 4** Example annotations produced by a human. The left column shows 3D annotations projected onto the image (another verification of accurate registration) and the right column represents annotations that have been made directly on the 2D data, initialized by the projections. The first pair of rows, (a) and (b), are two views of the same scene, and the second pair of rows (c) and (d) are a second scene. The colours (where available) represent the object category with: bowls in green, mugs in red and bottles in blue.

### 2.4 Evaluation Protocols

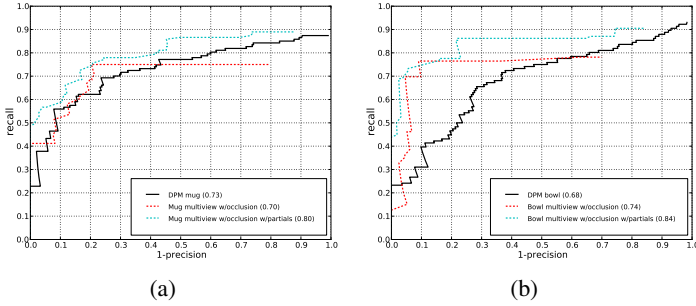
Our goal in collecting the UBC VRS dataset has been to facilitate scientific exploration of the robot recognition problem. That requires researchers to be able to compare the results of different methods evaluated on the same task. To that end, we describe a set of protocols that leverage the information provided in the dataset

to simulate a variety of tasks. In each case, we also outline the performance metric that is appropriate for benchmarking and comparison, using established guidelines where applicable. We focus on *simulate with real data* protocols since we believe this is crucial to addressing the complicated set of challenges facing a system that searches for objects in unstructured environments. This Section will describe the protocols in detail:

- *Passive single-view recognition*: means that each image is treated independently, as is common in the Computer Vision field. No registration or path information is available. In this case algorithms can localize objects with 2D bounding boxes in images, or by estimating 3D objects from 2D imagery and point clouds. The widely accepted metric for evaluation of such methods is precision and recall (PR) curves and the average precision (AP) statistic. Such curves are produced by varying a confidence threshold for the recognition method, and comparing which of the hypothesized 2D bounding boxes correctly overlaps an annotated (ground truth) object region. For each threshold, the ratio of true positives to total annotated objects is known as recall, while the ratio of true positives to number of hypothesized objects is precision. Average precision summarizes performance across all possible thresholds. Perfect performance on the task would give all of precision, recall, and average precision equal to 1.0.
- *Passive multi-view recognition*: provides the recognizer with data from a series of poses with variable length,  $N$ , from a path chosen by the simulator at random in each trial. Position information is provided to the recognizer so that evidence can be integrated spatially. This task captures the unique ability of a robot to move through the scene and gather data. Results may be reported as 3D object poses defined in the global registration frame, or as individual sets of 2D objects in each image. To standardize comparison, and to utilize the most accurate source of ground truth in the dataset, the primary evaluation should again be PR curves and AP statistics based on evaluation of 2D hypotheses. Where methods report 3D object poses, these should be projected to form 2D bounding boxes in each image. Our toolbox provides this functionality to aid fair evaluation.
- *Active multi-view recognition*: extends the passive protocol by requiring the recognition algorithm to provide a control input which simulates the robot's motion to a new pose and alters the image sequence accordingly. In general, a robot could take an unbounded number of steps through an environment before estimating the objects present, but we must standardize on specific path lengths for comparison with the passive multi-view protocol. Therefore, evaluation should be done after  $N$  steps are chosen by the system and the data is analyzed. All other evaluation details should remain the same as described in the previous item.

### 3 Experiments

At the time of submission of this abstract, several researchers have attempted to perform a subset of the robot recognition tasks possible with the UBC VRS dataset [17, 13, 18, 19]. This section will describe the results of these previous methods



**Fig. 5** Results of several versions of the the passive multi-view recognition method from [19] are compared against a state-of-the-art passive single-view method from [9] (labeled DPM) which has won the Pascal VOC challenge for several years. The summary statistic is Average Precision.

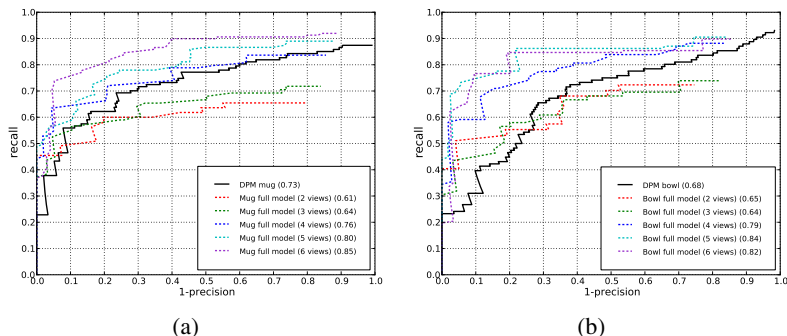
along with a brief description of the methods achieving the highest performance at the time of this publication. Further improvements will be recorded on the dataset’s website.

### 3.1 Passive Single Viewpoint Recognition

Meger *et al.* [19] report an average precision of 0.73 for mugs and 0.68 was achieved across the UBC VRS test set by the method of Felzenszwalb *et al.* known as the *Deformable-Parts Model* DPM. The DPM method is based on learning a latent-variable Support-Vector Machine (LVSVM) [3] classifier over an image feature similar to the well-known Histogram of Oriented Gradients (HOG) [4]. The authors attempted to train the DPM model on a variety of training data sets, and found maximum performance by selecting positive examples from both ImageNet [5] as well as from the UBC VRS training set, and by including a large number of relevant negative images from a variety of sources. Single-view recognition is not the focus of our dataset, but it is informative to observe whether an improvement can be made by fusing information from multiple viewpoints, and thus this DPM result serves as a baseline for comparison with other methods.

### 3.2 Passive Multiple Viewpoint Recognition

The highest passive multiple viewpoint recognition performance reported in previous work was obtained by [19] and we present the results of that paper here as a current benchmark. Their solution leveraged strong single-view DPM hypotheses in each image, and lifted this 2D appearance information to allow explicit per-object occlusion inference and the use of part-based detectors to improve accuracy on the



**Fig. 6** Results of passive multi-view recognition over various numbers of views per scene, from [19] are compared against a state-of-the-art passive single-view method from [9] (labeled DPM) which has won the Pascal VOC challenge for several years. The summary statistic is Average Precision.

large number of partially occluded object instances found in the dataset. Figures 5 and 6 illustrate the performance of this method on the UBC VRS test set. The first figure demonstrates results of the method when the simulator randomly provides images from five viewpoints per scene. By using multiple viewpoints, the method is able to achieve higher average precision than the single-view baseline provided by DPM. Also, occlusion reasoning is shown to give an improvement in the performance of the method, and additionally including part-based appearance information leads to the highest performance overall.

Figure 6 expands upon the previous result by examining the performance of the method as various numbers of viewpoints are made available for each scene. In all cases, the system achieves more recall in the high-precision region of the curves, but in some cases, the overall performance of the multi-view method is worse than the single-view baseline. This was explained by the difficulty in recovering 3D information from the weak geometric cues available in a small number of views that are often from a wide-baseline. However, it remains to be seen if future approaches will be able to achieve better performance from only two or three views of a scene. In all cases, the clear trend is that the method performs better as more views become available.

### 3.3 Active Multiple Viewpoint Recognition

As of the submission of this abstract, only a very preliminary attempt has been made on the active multi-view recognition task on a small portion of the dataset, by [17]. Active recognition (i.e. explicitly selecting the next best viewpoint at each step) is easily done with the UBC VRS dataset and is an area for future work.

## 4 Experimental Insights

While careful collection and annotation of a dataset with sufficient scale for meaningful evaluation is a large effort, the resulting repeatable evaluation will hopefully be of value to the robotics community<sup>1</sup>. Beyond data, a key contribution of our method is the labeling and evaluation pipelines, and the tools related to these can extend to a variety of additional data sources. For example, we have already succeeded in using the same tools to annotate and evaluate our approaches using Kinect data that was registered without the use of our fiducial marker (i.e. using the software of Endres *et al.* [6]) as well as outdoor data collected by an automobile with a highly accurate inertial measurement unit.

Pascal VOC has encouraged various authors to borrow and improve upon the best techniques from the winners of the previous years. A number of authors have already obtained the UBC VRS and are currently beginning to develop new solutions. Ideally, this will lead to additional performance improvements being published in coming years by a variety of authors. The data has been collected by a physical robotic platform along with modern sensors, control, and calibration. So, pursuit of such improvement is likely to provide direct benefits to the ability of many robots to perceive objects, in many environments.

**Acknowledgements.** The authors thank Scott Helmer and Marius Muja for their assistance with the early development of tools and collection of the first portion of data that lead to the final UBC VRS dataset. This research was funded, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References

1. Website, <http://www.semantic-robot-vision-challenge.org/>
2. Website, <http://solutionsinperception.org/index.html/>
3. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: *Advances in Neural Information Processing Systems* (2003)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, USA, vol. 2, pp. 886–893 (June 2005)
5. Deng, J., Li, K., Do, M., Su, H., Fei-Fei, L.: *Construction and Analysis of a Large Scale Image Ontology*. Vision Sciences Society (2009)
6. Endres, F., Hess, M., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the rgb-d slam system. In: *Proceedings of ICRA* (2012)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: *The PASCAL Visual Object Classes Challenge 2011 (VOC 2011) Results*. Website (2011)
8. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Generative-Model Based Vision* (2004)

---

<sup>1</sup> The data and code is are both completely open, at <http://www.cs.ubc.ca/labs/lci/vrs/index.html>



9. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (2010)
10. Fiala, M.: Artag, a fiducial marker system using digital techniques. In: *CVPR 2005*, vol. 1, pp. 590–596 (2005)
11. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
12. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press (2000) ISBN: 0521623049
13. Helmer, S., Meger, D., Muja, M., Little, J.J., Lowe, D.G.: Multiple viewpoint recognition and localization. In: *Proceedings of the Asian Computer Vision Conference* (2010)
14. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A category-level 3-d object dataset: Putting the kinect to work. In: *Proceedings of the ICCV Workshop on Consumer Depth Cameras in Computer Vision* (2011)
15. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: *IEEE International Conference on Robotics and Automation* (2011)
16. Meger, D., Forssén, P.-E., Lai, K., Helmer, S., McCann, S., Southey, T., Baumann, M., Little, J.J., Lowe, D.G., Dow, B.: Curious george: An attentive semantic robot. *Robotics and Autonomous Systems. Journal Special Issue on From Sensors to Human Spatial Concepts* 56(6), 503–511 (2008)
17. Meger, D., Gupta, A., Little, J.J.: Viewpoint detection models for sequential embodied object category recognition. In: *Proceedings of the International Conference on Robotics and Automation, ICRA* (2010)
18. Meger, D., Little, J.J.: Mobile 3d object detection in clutter. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, United States (2011)
19. Meger, D., Wojek, C., Schiele, B., Little, J.J.: Explicit occlusion reasoning for 3d object detection. In: *Proceedings of the 22nd British Machine Vision Conference, BMVC* (2011)
20. Tsai, R.Y., Lenz, R.K.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 323–344 (1987)
21. Unnikrishnan, R., Hebert, M.: Fast extrinsic calibration of a laser rangefinder to a camera. Technical Report CMU-RI-TR-05-09, Robotics Institute, Pittsburgh, PA (July 2005)

# Author Index

- Abbeel, Pieter 547  
Abbott, Jake J. 55  
Abelmann, Leon 617  
Abichandani, Pramod 183  
Allen, Peter K. 435  
Ang Jr., Marcelo H. 963  
Arrowsmith, Ramon 733
- Bagnell, J. Andrew 371  
Bahr, Alexander 259  
Bajer, Przemyslaw G. 859  
Bandyopadhyay, Tirthankar 963  
Barry, Jennifer 531  
Baumgartner, Eric 245  
Bayer, Justin 89  
Berenson, Dmitry 547  
Bergerman, Marcel 745  
Bo, Liefeng 387  
Bollini, Mario 481  
Borrmann, Dorit 907  
Bosse, Michael 877  
Bowman, Sean L. 303  
Branson, Elliot 859  
Buss, Martin 199
- Carton, Daniel 199  
Chan, Nicholas 951  
Charrow, Benjamin 337  
Chitre, Mandar 841  
Chopra, Nikhil 633, 683  
Christensen, Henrik I. 127, 231, 699  
Chuy, Oscar 651  
Clark, Jonathan E. 113  
Collins, Emmanuel 651
- Corke, Peter 877, 891  
Corona, Alejandro-Hinojosa 733
- Damani, Aayush 55  
Dantam, Neil 699  
Das, Jnaneshwar 259, 819  
Desai, Jaydev P. 563, 565  
Dollar, Aaron M. 27  
Dong, Lixing 435  
Doniec, Marek 841  
Drumwright, Evan 497, 499  
Dudek, Gregory 789  
Dunbabin, Matthew 775  
Dunlap, Damion 651
- Elseberg, Jan 907  
Escobedo, Arturo 449  
Evans, William C. 259
- Fabrizi, Ugo 877  
Fasola, Juan 463  
Ficuciello, Fanny 515  
Fox, Dieter 387, 403  
Frazzoli, Emilio 963  
Fukui, Rui 15, 939
- Gergondet, Pierre 215  
Giguère, Philippe 789  
Girdhar, Yogesh 789  
Goldberg, Ken 547  
Gosselin, Clément 1, 3, 635  
Gowal, Sven 353  
Graham, Rishi 819  
Guger, Christoph 215  
Gullapalli, Rao 565

- Herbst, Evan 403  
 Herman, Herman 951  
 Hernández, José Ramón Medina 73  
 Hesch, Joel A. 303  
 Hintermüller, Christoph 215  
 Hippenmeyer, Nicholas 155  
 Hirche, Sandra 73  
 Ho, Colin 759  
 Hollinger, Geoffrey A. 803  
 Howe, Robert D. 579  
 Hsiao, Kaijen 531  
 Hsieh, Mong-ying Ani 143, 183  
 Hsu, David 963
- Islar, Volkan 717, 859
- Jenkins, Tim 155  
 Jiang, Yun 921  
 Jie, Chong Zhuang 963
- Kadowaki, Keita 15  
 Kaelbling, Leslie Pack 531  
 Kelly, Alonzo 919, 951  
 Kesner, Samuel B. 579  
 Khalil, Islam S.M. 617  
 Khatib, Oussama 101, 155  
 Kheddar, Abderrahmane 215  
 Kim, Young-Ho 759  
 Koganezawa, Koichi 667  
 Kothmann, Bruce 289  
 Kottas, Dimitrios G. 303  
 Krishnan, Aravindhan K. 733  
 Kröger, Torsten 155  
 Kumar, Vijay 289, 337  
 Kushleyev, Aleksandr 289
- Lakaemper, Rolf 143  
 Lakatos, Dominic 89  
 Lakshmanan, Karthik 547  
 Laliberté, Thierry 3  
 Laugier, Christian 449  
 Lawitzky, Martin 73  
 Lecours, Alexandre 3, 635  
 Leontie, Roxana 499  
 Lessard, Frédéric 3  
 Leung, Clement 603  
 Little, James J. 979  
 Liu, Jun 603  
 Lozano-Pérez, Tomás 531
- Lu, Zhe 603  
 Lucas, Drew 819
- Ma, Raymond R. 27  
 Mallory, Kenneth 183  
 Marchese, Andrew D. 41  
 Martin, Steven 891  
 Martinoli, Alcherio 259, 321, 353  
 Mason, Matthew T. 369, 417  
 Matarić, Maja J 433, 463  
 Matuszek, Cynthia 403  
 Maughan, Thom 819  
 McMillan, Alan 565  
 Meger, David 979  
 Melchiorri, Claudio 515  
 Mellinger, Daniel 289  
 Michael, Nathan 273, 319, 337  
 Milford, Michael 877  
 Miller, Bruce 113  
 Minnig, Michael 259  
 Misra, Sarthak 617  
 Mitra, Urbashi 803  
 Morris, Timothy 877  
 Murphy, Liz 875, 877, 891
- Nambi, Manikantan 55  
 Narita, Takashi 101  
 Nieto-Granda, Carlos 231, 699  
 Nissen, Edwin 733  
 Niwa, Yamato 15  
 Nüchter, Andreas 907  
 Nuske, Stephen 715, 745
- Odhner, Lael U. 27  
 Onal, Cagdas D. 41  
 Ordonez, Camilo 113  
 Oslund, Ken 155  
 Ott, Lionel 129
- Palli, Gianluca 515  
 Paolini, Robert 417  
 Perrollaz, Mathias 245  
 Persson, M.S. 167  
 Pichel, Marc P. 617  
 Plonski, Patrick A. 717  
 Powers, Caitlin 289  
 Prorok, Amanda 321  
 Py, Frédéric 819

- Rajan, Kanna 819  
Ramos, Fabio 129  
Reeves, Nicolas 167  
Ren, Xiaofeng 387  
Rios-Martinez, Jorge 449  
Robinson, Matthew 245  
Rodriguez, Alberto 417  
Rogers III, John G. 231  
Roumeliotis, Stergios I. 303  
Roy, Nicholas 481  
Rus, Daniela 41, 481, 787, 841, 963  
Rüschén, Daniel 89  
Rusu, Radu Bogdan 155
- Sachdev, Apoorva 547  
Saripalli, Srikanth 733, 759  
Sato, Tomomasa 15, 939  
Saxena, Ashutosh 921  
Seelinger, Michael 245  
Shababo, Benjamin 435  
Sharf, Inna 167  
Sharma, Aneesh 651  
Shell, Dylan A. 499, 759  
Shen, Shaojie 273  
Shimosaka, Masamichi 15, 939  
Shin, Dongjun 101  
Shojaei-Baghini, Ehsan 591  
Siciliano, Bruno 515  
Silver, David 371  
Simha, Rahul 499  
Singh, Sanjiv 745  
Slater, Mel 215  
Smagt, Patrick van der 89  
Sorensen, Peter W. 859  
Spalanzani, Anne 449  
Srinivasa, Siddhartha S. 417  
Stein, Procópio 449  
Stentz, Anthony 371  
Stilman, Mike 699
- St-Ong, David 167  
Sukhatme, Gaurav S. 257, 259, 803  
Sun, Yu 591, 603
- Tallapragada, Pavankumar 683  
Tan, U-Xuan 565  
Tellex, Stefanie 481  
Tews, Ashley 775  
Thompson, Tyler 481  
Tokekar, Pratap 717, 859  
Topor, Iulian 841  
Torczynski, Dan 155  
Turnwald, Annemarie 199
- Upcroft, Ben 877
- Vander Hook, Joshua 859  
van der Smagt, Patrick 71  
Vasquez, Dizan 449  
Vogel, Jörn 89
- Wan, Weiwei 15  
Wang, Qi 745  
Warner, Randy 951  
Warren, Michael 877  
Watanabe, Masahiko 939  
Weisz, Jonathan 435  
West, Jeffrey 245  
Wollherr, Dirk 199  
Worcester, James 143
- Xie, Ziang 547
- Yang, Bo 565  
Yeh, Xiyang 101  
Yoder, John-David 245
- Zettlemoyer, Luke 403  
Zondervan, Lars 617