

Joaquim Filipe  
Kurosh Madani  
Oleg Gusikhin  
Jurek Sasiadek  
*Editors*

Informatics in Control,  
Automation and Robotics  
12th International  
Conference, ICINCO 2015  
Colmar, France, July 21–23,  
2015 Revised Selected  
Papers

# Lecture Notes in Electrical Engineering

Volume 383

## Board of Series editors

Leopoldo Angrisani, Napoli, Italy  
Marco Arteaga, Coyoacán, México  
Samarjit Chakraborty, München, Germany  
Jiming Chen, Hangzhou, P.R. China  
Tan Kay Chen, Singapore, Singapore  
Rüdiger Dillmann, Karlsruhe, Germany  
Haibin Duan, Beijing, China  
Gianluigi Ferrari, Parma, Italy  
Manuel Ferre, Madrid, Spain  
Sandra Hirche, München, Germany  
Faryar Jabbari, Irvine, USA  
Janusz Kacprzyk, Warsaw, Poland  
Alaa Khamis, New Cairo City, Egypt  
Torsten Kroeger, Stanford, USA  
Tan Cher Ming, Singapore, Singapore  
Wolfgang Minker, Ulm, Germany  
Pradeep Misra, Dayton, USA  
Sebastian Möller, Berlin, Germany  
Subhas Mukhopadhyay, Palmerston, New Zealand  
Cun-Zheng Ning, Tempe, USA  
Toyoaki Nishida, Sakyo-ku, Japan  
Bijaya Ketan Panigrahi, New Delhi, India  
Federica Pascucci, Roma, Italy  
Tariq Samad, Minneapolis, USA  
Gan Woon Seng, Nanyang Avenue, Singapore  
Germano Veiga, Porto, Portugal  
Haitao Wu, Beijing, China  
Junjie James Zhang, Charlotte, USA

### *About this Series*

“Lecture Notes in Electrical Engineering (LNEE)” is a book series which reports the latest research and developments in Electrical Engineering, namely:

- Communication, Networks, and Information Theory
- Computer Engineering
- Signal, Image, Speech and Information Processing
- Circuits and Systems
- Bioengineering

LNEE publishes authored monographs and contributed volumes which present cutting edge research information as well as new perspectives on classical fields, while maintaining Springer’s high standards of academic excellence. Also considered for publication are lecture materials, proceedings, and other related materials of exceptionally high quality and interest. The subject matter should be original and timely, reporting the latest research and developments in all areas of electrical engineering.

The audience for the books in LNEE consists of advanced level students, researchers, and industry professionals working at the forefront of their fields. Much like Springer’s other Lecture Notes series, LNEE will be distributed through Springer’s print and electronic publishing channels.

More information about this series at <http://www.springer.com/series/7818>

Joaquim Filipe · Kurosh Madani  
Oleg Gusikhin · Jurek Sasiadek  
Editors

Informatics in Control,  
Automation and Robotics  
12th International  
Conference, ICINCO 2015  
Colmar, France, July 21-23,  
2015 Revised Selected Papers

*Editors*

Joaquim Filipe  
Polytechnic Institute of Setúbal  
Setúbal  
Portugal

Oleg Gusikhin  
Ford Research and Advanced Engineering  
Dearborn, MI  
USA

Kurosh Madani  
University of Paris-EST Créteil (UPEC)  
Créteil  
France

Jurek Sasiadek  
Mechanical and Aerospace Engineering  
Carleton University  
Ottawa, ON  
Canada

ISSN 1876-1100                      ISSN 1876-1119 (electronic)  
Lecture Notes in Electrical Engineering  
ISBN 978-3-319-31896-7              ISBN 978-3-319-31898-1 (eBook)  
DOI 10.1007/978-3-319-31898-1

Library of Congress Control Number: 2016935980

© Springer International Publishing Switzerland 2016

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.

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.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature  
The registered company is Springer International Publishing AG Switzerland

# Preface

This book includes extended and revised versions of a set of selected papers from the 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2015) which is sponsored by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC) and co-organized by the University of Haute Alsace. ICINCO is held in cooperation with the ACM Special Interest Group on Artificial Intelligence (ACM SIGAI), EUROMICRO, Association for the Advancement of Artificial Intelligence (AAAI), Associação Portuguesa de Controlo Automático (APCA), The International Neural Network Society (INNS), Asia Pacific Neural Network Assembly (APNNA) and euRobotics AISBL (Association Internationale Sans But Lucratif). ICINCO is also technically co-sponsored by IEEE Control Systems Society (CSS) and IEEE Robotics and Automation Society and co-sponsored by International Federation of Automatic Control (IFAC).

ICINCO 2015 received 214 paper submissions, including special sessions, from 42 countries in all continents, of which 44 % were orally presented (14 % as full papers). In order to evaluate each submission, a double-blind paper review was performed by the Program Committee. As in previous editions of the conference, based on the reviewer's evaluations and the presentations, a short list of authors were invited to submit extended versions of their papers for this book. The selected papers reflect the interdisciplinary nature of the conference as well as the logic equilibrium of the four tracks: Intelligent Control Systems and Optimization, Robotics and Automation, Signal Processing, Sensors, Systems Modelling and Control, and Industrial Informatics. The diversity of topics is an important feature of this conference, enabling an overall perception of several important scientific and technological trends.

Furthermore, ICINCO 2015 included four keynote lectures, given by internationally recognized researchers: Krzysztof Kozłowski (Poznan University of Technology, Poland), Gerhard Schweitzer (ETH Zurich, Switzerland), Oliver Brock (TU Berlin, Germany), and Faïz Ben Amar (Université Pierre et Marie Curie, Institut Systèmes Intelligents et de Robotique, France). We would like to express

our appreciation to all of them and in particular to those who took the time to contribute to this book with a paper.

On behalf of the conference organizing committee, we would like to thank all participants, in particular all the authors, whose quality work is the essence of the conference and the members of the Program Committee, who helped us with their valuable expertise and diligence in reviewing the submitted papers. As we all know, organizing a conference requires the effort of many individuals. We wish also to thank all the members of our organizing committee, whose work and commitment were invaluable. Aiming to provide the potential readers with an objective overview of the latest advances in the four major topics of the conference mentioned above, we hope that this book will be relevant for all researchers and practitioners whose work is related to using informatics in control, robotics, or automation.

July 2015

Joaquim Filipe  
Kurosh Madani  
Oleg Gusikhin  
Jurek Sasiadek

# Organization

## Conference Co-Chairs

Joaquim Filipe, Polytechnic Institute of Setúbal/INSTICC, Portugal  
Kurosh Madani, University of Paris-EST Créteil (UPEC), France

## Program Co-Chairs

Oleg Gusikhin, Ford Research and Advanced Engineering, United States  
Jurek Sasiadek, Carleton University, Canada

## Program Committee

Andrew Adamatzky, University of the West of England, UK  
El-Houssaine Aghezzaf, Ghent University, Faculty of Engineering  
and Architecture, Belgium  
Eugenio Aguirre, University of Granada, Spain  
Adel Al-Jumaily, University of Technology Sydney, Australia  
Yacine Amirat, University Paris Est Créteil (UPEC), France  
Nicolas Andreff, Femto-ST Institute, France  
Rui Araujo, University of Coimbra, Portugal  
T. Asokan, Indian Institute of Technology Madras, India  
Ruth Bars, Budapest University of Technology and Economics, Hungary  
Laxmidhar Behera, Indian Institute of Technology, Kanpur, India  
Carsten Behn, Ilmenau University of Technology, Germany  
Karsten Berns, University of Kaiserslautern, Germany  
Mauro Birattari, IRIDIA-CoDE, Université Libre de Bruxelles, Belgium  
Jean-louis Boimond, ISTIA—LISA, France  
Marvin Bugeja, University of Malta, Malta  
Edmund Burke, University of Nottingham, UK

Kevin Burn, University of Sunderland, UK  
Kenneth Camilleri, University of Malta, Malta  
Giuseppe Carbone, LARM—Laboratory of Robotics and Mechatronics, Italy  
Alessandro Casavola, University of Calabria, Italy  
Riccardo Cassinis, University of Brescia, Italy  
Marco Castellani, University of Birmingham, UK  
Xu Cheng, Emerson Electric, United States  
Tsung-Che Chiang, National Taiwan Normal University, Taiwan  
Sung-Bae Cho, Yonsei University, Korea, Republic of  
Paul Christodoulides, Cyprus University of Technology, Cyprus  
Chengbin Chu, Ecole Centrale Paris, France  
Feng Chu, University of Evry Val d'Essonne, France  
Daisuke Chugo, Kwansai Gakuin University, Japan  
Carlos Coello, CINVESTAV-IPN, Mexico  
James M. Conrad, University of North Carolina at Charlotte, United States  
Hugo Costelha, Polytechnic Institute of Leiria, Portugal  
Erik Cuevas, Universidad de Guadalajara, Mexico  
Prithviraj (Raj) Dasgupta, University of Nebraska, United States  
Kamil Dedecius, Institute of Information Theory and Automation, Academy of Sciences of the CZ, Czech Republic  
Kyriakos Deliparaschos, Cyprus University of Technology, Cyprus  
Mingcong Deng, Tokyo University of Agriculture and Technology, Japan  
Wenjie Dong, University of Texas Pan American, United States  
Xingye Dong, Beijing Jiaotong University, China  
António Dourado, University of Coimbra, Portugal  
Vasile Dragan, Romanian Academy, Romania  
Venky Dubey, Bournemouth University, UK  
Ioan Dumitrache, University “Polytechnica” of Bucharest, Romania  
Richard J. Duro, Universidade Da Coruña, Spain  
Vaclav Dvorak, Brno University of Technology, Czech Republic  
Marc Ebner, Ernst-Moritz-Arndt-Universität Greifswald, Germany  
Petr Ekel, Pontifical Catholic University of Minas Gerais, Brazil  
Mohammed El-Abd, American University of Kuwait, Kuwait  
Simon G. Fabri, University of Malta, Malta  
Christophe Farges, Bordeaux University, IMS—UMR 5218 CNRS, France  
David Feil-Seifer, University of Nevada Reno, United States  
David Fernández-Llorca, University of Alcalá, Spain  
Antoine Ferreira, INSA Centre Val de Loire, France  
Juan J. Flores, University of Michoacan, Mexico  
Mauro Franceschelli, Università di Cagliari, Italy  
Georg Frey, Saarland University, Germany  
Toyomi Fujita, Department of Electronics and Intelligent Systems, Japan  
Mauro Gaggero, National Research Council of Italy, Italy  
Maria I. Garcia-Planas, Universitat Politècnica de Catalunya, Spain  
Ryszard Gessing, Silesian University of Technology, Poland

Lazea Gheorghe, Technical University of Cluj-Napoca, Romania  
Paulo Gil, Universidade Nova de Lisboa, Portugal  
Giuseppina Gini, Politecnico di Milano, Italy  
Maria Gini, University of Minnesota, United States  
Arthur Gómez, Universidade do Vale do Rio dos Sinos, Brazil  
Amr Goneid, The American University in Cairo, Egypt  
Bhaskaran Gopalakrishnan, West Virginia University, United States  
Lucian Teodor Grigorie, University of Craiova, Romania  
Lech Grzesiak, Warsaw University of Technology, Poland  
Kevin Guelton, University of Reims Champagne-Ardenne, France  
Deepak Gupta, Wichita State University, United States  
Maki K. Habib, The American University in Cairo, Egypt  
Jörg Hähner, Universität Augsburg, Germany  
Wolfgang Halang, Fernuniversitaet, Germany  
Kensuke Harada, AIST, Japan  
Jennifer Harding, Loughborough University, UK  
Dominik Henrich, University of Bayreuth, Germany  
Suranga Hettiarachchi, Indiana University Southeast, United States  
Diana Hintea, Coventry University, United Kingdom  
Yasuhisa Hirata, Tohoku University, Japan  
Mathieu Hobon, ENSAM—LCFC, France  
Wladyslaw Homenda, Warsaw University of Technology, Poland  
Atsushi Imiya, IMIT Chiba University, Japan  
Gianluca Ippoliti, Università Politecnica delle Marche, Italy  
Mircea Ivanescu, University of Craiova, Romania  
Sarangapani Jagannathan, Missouri University of Science and Technology,  
United States  
Michael Jenkin, York University, Canada  
Wootae Jeong, Korea Railroad Research Institute, Korea, Republic of  
Michail Kalogiannakis, University of Crete, Greece  
Vikram Kapila, NYU Polytechnic School of Engineering, United States  
Tohru Kawabe, University of Tsukuba, Japan  
Graham Kendall, University of Nottingham Malaysia Campus, Malaysia  
Sarath Kodagoda, University of Technology Sydney, Australia  
Israel Koren, University of Massachusetts, United States  
Krzysztof Kozłowski, Poznan University of Technology, Poland  
Ondrej Krejcar, University of Hradec Kralove, Czech Republic  
Masao Kubo, National Defense Academy of Japan, Japan  
Barbara Kühnlenz, Coburg University of Applied Sciences, Germany  
Kolja Kühnlenz, University of Applied Sciences and Arts, Germany  
Nikolay Vladimirovich Kuznetsov, Saint-Petersburg State University, Russian  
Federation  
Sébastien Lahaye, Istia—LARIS, France  
Jimmy Lauber, University of Valenciennes, France  
Sangyoon Lee, Connecticut College, United States

Dimitri Lefebvre, GreaH Université Le Havre, France  
 Kauko Leiviskä, University of Oulu, Finland  
 Gennady Leonov, Saint-Petersburg State University, Russian Federation  
 Kang Li, Queen's University Belfast, UK  
 Tsai-Yen Li, National Chengchi University, Taiwan  
 Gordon Lightbody, University College Cork, Ireland  
 Veronique Limère, Ghent University, Belgium  
 Huei-Yung Lin, National Chung Cheng University, Taiwan  
 Guoping Liu, University of South Wales, UK  
 Paolo Lollini, Università degli Studi di Firenze, Italy  
 Antonio Mendes Lopes, University of Porto, Portugal  
 Luís Seabra Lopes, University of Aveiro, Portugal  
 Sérgio F. Lopes, University of Minho, Portugal  
 Gonzalo Lopez-Nicolas, Universidad de Zaragoza, Spain  
 Martin Lukac, Tohoku University, Japan  
 Danilo Maccio, National Research Council of Italy, Italy  
 José Tenreiro Machado, Institute of Engineering, Polytechnic of Porto, Portugal  
 Anthony Maciejewski, Colorado State University, United States  
 Kurosh Madani, University of Paris-EST Créteil (UPEC), France  
 Magdi S. Mahmoud, King Fahd University of Petroleum and Minerals, Saudi Arabia  
 Frederic Maire, Queensland University of Technology, Australia  
 Om Malik, University of Calgary, Canada  
 Stamatis Manesis, University of Patras, Greece  
 Fabio Marchese, Università degli Studi di Milano-Bicocca, Italy  
 Philippe Martinet, Ecole Centrale de Nantes, France  
 Alicja Mazur, Wroclaw University of Technology, Poland  
 Ross McAree, University of Queensland, Australia  
 Luis Merino, Pablo de Olavide University, Spain  
 Konstantinos Michail, Cyprus University of Technology, Cyprus  
 Sanya Mitaim, Thammasat University, Thailand  
 Sara Moein, Multimedia University, Malaysia  
 Francesco Carlo Morabito, University Mediterranea of Reggio Calabria, Italy  
 Cesar de Prada Moraga, Universidad de Valladolid, Spain  
 Vladimir Mostyn, VSB—Technical University of Ostrava, Czech Republic  
 Vassilis C. Moulitanitis, University of the Aegean, Greece  
 George Moustiris, National Technical University of Athens, Greece  
 Riccardo Muradore, University of Verona, Italy  
 Shimon Y. Nof, Purdue University, United States  
 Juan A. Nolazco-Flores, ITESM, Campus Monterrey, Mexico  
 Andrzej Obuchowicz, University of Zielona Góra, Poland  
 José Valente de Oliveira, Universidade do Algarve, Portugal  
 Lei Pan, Southeast University, China  
 Gordon Parker, Michigan Technological University, United States  
 Igor Paromtchik, INRIA, France

Ignacio Parra-Alonso, University of Alcalá, Spain  
Pushparaj Mani Pathak, Indian Institute of Technology, India  
Dimitri Peaucelle, LAAS-CNRS, France  
Carlos Eduardo Pereira, Federal University of Rio Grande Do Sul—UFRGS, Brazil  
D.T. Pham, The University of Birmingham, UK  
Angel P. Del Pobil, Universitat Jaume I, Spain  
Nirvana Popescu, University Politehnica of Bucharest, Romania  
David Powers, Flinders University, Australia  
Radu-Emil Precup, Politehnica University of Timisoara, Romania  
Libor Preucil, Czech Technical University in Prague, Czech Republic  
José Ragot, Centre de Recherche en Automatique de Nancy, France  
Oscar Reinoso, Miguel Hernandez University, Spain  
Mihailo Ristic, Imperial College London, UK  
Cameron Riviere, Carnegie Mellon University, United States  
Juha Röning, University of Oulu, Finland  
Mehdi Roopaei, University of Texas at San Antonio, United States  
Yassine Ruichek, Université de Technologie Belfort-Montbéliard, France  
Jocelyn Sabatier, IMS Laboratory -Bordeaux University, France  
Marco Sabatini, Università degli Studi di Roma La Sapienza, Italy  
Priti Srinivas Sajja, Sardar Patel University, India  
Antonio Sala, Universidad Politecnica de Valencia, Spain  
Abdel-Badeeh Mohamed Salem, Ain Shams University, Egypt  
Jurek Sasiadek, Carleton University, Canada  
Hideyuki Sawada, Kagawa University, Japan  
Daniel Schmidt, University of Kaiserslautern, Germany  
Gerald Seet, Nanyang Technological University, Singapore  
João Sequeira, Instituto Superior Técnico/Institute for Systems and Robotics, Portugal  
Roman Sergienko, Institute of Telecommunications Engineering, Ulm University, Germany  
Madhavan Shanmugavel, Monash University, Malaysia  
Jinhua She, Tokyo University of Technology, Japan  
Masahiro Shiomi, ATR—Agent Interaction Design Lab, Japan  
Michael Short, Teesside University, United Kingdom  
Vasile Sima, National Institute for Research & Development in Informatics, Romania  
Dan Simon, Cleveland State University, United States  
Daniel Simon, INRIA, France  
Olivier Simonin, INRIA—LORIA, France  
Andrzej Sluzek, Khalifa University, United Arab Emirates  
Jorge Solis, Karlstad University/Waseda University, Sweden  
Stefano Squartini, Università Politecnica delle Marche, Italy  
Burkhard Stadlmann, Upper Austria University of Applied Sciences, Wels, Austria  
Olaf Stursberg, University of Kassel, Germany  
Raúl Suárez, Universitat Politecnica de Catalunya (UPC), Spain

Abdelhamid Tayebi, Lakehead University, Canada  
 Hakan Temeltas, Istanbul Technical University, Turkey  
 Daniel Thalmann, Nanyang Technological University, Singapore  
 Simon Thompson, AIST, Japan  
 Gui Yun Tian, Newcastle University, UK  
 Dan Tiba, Institute of Mathematics Academia Romana, Romania  
 Sven Tomforde, Augsburg University, Germany  
 Wa-Muzemba Tshibangu, Morgan State University, United States  
 Avgoustos Tsinakos, University of Kavala Institute of Technology/Athabasca  
 University of Canada, Greece  
 Antonios Tsourdos, Cranfield University, UK  
 Ali Emre Turgut, Universite Libre de Bruxelles, Belgium  
 Spyros Tzafestas, National Technical University of Athens, Greece  
 Bartłomiej Ufnalski, Warsaw University of Technology, Poland  
 Angel Valera, Universidad Politécnica de Valencia, Spain  
 Bram Vanderborght, Vrije Universiteit Brussel, Belgium  
 Cesar Vargas-Rosales, Tecnológico de Monterrey, Campus Monterrey, Mexico  
 Ramiro Velazquez, Universidad Panamericana, Mexico  
 Federico Vicentini, National Research Council of Italy (CNR), Italy  
 Arnoud Visser, Universiteit van Amsterdam, The Netherlands  
 Ljubo Vlacic, Griffith University, Australia  
 Damir Vrancic, Jožef Stefan Institute, Slovenia  
 Bernardo Wagner, Leibniz Universität Hannover, Germany  
 Feng Wan, University of Macau, Macau  
 Jingyu Wang, Northwestern Polytechnical University (NPU), China  
 Long Wang, Peking University, China  
 Shengwei Wang, The Hong Kong Polytechnic University Kowloon, Hong Kong  
 H.M. Wee, Chung Yuan Christian University, Taiwan  
 James Whidborne, Cranfield University, United Kingdom  
 Warren White, Kansas State University, United States  
 Qinmin Yang, Zhejiang University, China  
 Gang Yao, Shanghai Maritime University, China  
 Weiwei Yu, Northwestern Polytechnical University, China  
 Ludek Zalud, Brno University of Technology, Czech Republic  
 Jie Zhang, University of Newcastle, United Kingdom  
 Primo Zingaretti, Università Politecnica delle Marche, Italy  
 Loredana Zollo, Università Campus Bio-Medico, Italy

### **Additional Reviewers**

Mihaela Barreau, University of Angers, France  
 Lukas Exel, AES, Germany  
 Saïd Moussaoui, Ecole Centrale de Nantes, France  
 Pietro Muraca, DEIMES, Italy

Hassen Riahi, ISTIA, France  
Maximilian Sand, Universität Bayreuth, Germany  
Johannes Schmölz, Coburg University of Applied Sciences and Arts, Germany  
Adrian Stoica, University “Polytechnica” of Bucharest, Romania  
Francesco Tedesco, University of Calabria, Italy  
Teodor Tiplica, LARIS/istia, France  
Tobias Werner, Universität Bayreuth, Germany

## **Invited Speakers**

Krzysztof Kozłowski, Poznan University of Technology, Poland  
Gerhard Schweitzer, ETH Zurich, Switzerland  
Oliver Brock, TU Berlin, Germany  
Faïz Ben Amar, Université Pierre et Marie Curie, Institut Systèmes Intelligents et de Robotique, France

# Contents

## Part I Invited Papers

<b>Formation Control and Vision Based Localization of a System of Mobile Robots</b> . . . . .	3
Krzysztof Kozłowski and Wojciech Kowalczyk	
<b>Path Tracking Controllers for Fast Skidding Rover</b> . . . . .	29
Mohamed Krid, Ziad Zamzami and Faiz Benamar	

## Part II Intelligent Control Systems and Optimization

<b>Gravitational Search Algorithm-Based Evolving Fuzzy Models of a Nonlinear Process.</b> . . . . .	51
Radu-Emil Precup, Emil-Ioan Voisan, Emil M. Petriu, Mircea-Bogdan Radac and Lucian-Ovidiu Fedorovici	
<b>Speech Phoneme Classification by Intelligent Decision-Level Fusion</b> . . .	63
Fréjus A.A. Laleye, Eugène C. Ezin and Cina Motamed	
<b>Genetic Algorithm for Automated X-Ray Diffraction Full-Profile Analysis of Electrolyte Composition on Aluminium Smelters.</b> . . . . .	79
Alexandr Zaloga, Shakhnaz Akhmedova, Igor Yakimov, Sergey Burakov, Eugene Semenkin, Petr Dubinin, Oksana Piksina and Eugene Andryushchenko	

## Part III Robotics and Automation

<b>Improving the 3D Positioning for Low Cost Mobile Robots.</b> . . . . .	97
Rafael Socas, Sebastian Dormido, Raquel Dormido and Ernesto Fabregas	

<b>Kinematics, Simulation, and Analysis of the Planar and Symmetric Postures of a Serial-Parallel Climbing Robot</b> . . . . .	115
Adrián Peidró, Arturo Gil, José María Marín, Yerai Berenguer and Óscar Reinoso	
<b>A Standing Assistance Scheme Using a Patient’s Physical Strength by a Load Estimation Considering the Muscle Arrangements of a Human Leg</b> . . . . .	137
Daisuke Chugo, Satoshi Muramatsu, Sho Yokota and Hiroshi Hashimoto	
<b>Multimodal Image Registration and Visual Servoing</b> . . . . .	157
M. Ourak, B. Tamadazte, N. Andreff and E. Marchand	
<b>Trajectory Tracking Control of an Excavator Arm Using Guaranteed Cost Control</b> . . . . .	177
Alexander Gurko, Oleg Sergiyenko, Juan Ivan Nieto Hipólito, Igor Kirichenko, Vera Tyrsa and Wilmar Hernandez	
<b>Computation of Curvature Skeleton to Measure Deformations in Surfaces</b> . . . . .	197
Carlos M. Mateo, Pablo Gil and Fernando Torres	
<b>On Structure and Distribution of Software for Mobile Manipulators</b> . . . . .	209
Andreas Schierl, Andreas Angerer, Alwin Hoffmann, Michael Vistein and Wolfgang Reif	
<b>Combination of HMM and DTW for 3D Dynamic Gesture Recognition Using Depth Only</b> . . . . .	229
Hajar Hiyadi, Fakhreddine Ababsa, Christophe Montagne, El Houssine Bouyakhf and Fakhita Regragui	
<b>Model-Free (Human) Tracking Based on Ground Truth with Time Delay: A 3D Camera Based Approach for Minimizing Tracking Latency and Increasing Tracking Quality</b> . . . . .	247
Philip Nicolai, Jörg Raczkowski and Heinz Wörn	
<b>Highly Parallelizable Algorithm for Keypoint Detection in 3-D Point Clouds</b> . . . . .	267
Jens Garstka and Gabriele Peters	
<b>3D Modelling of Biped Robot Locomotion with Walking Primitives Approach in Simulink Environment</b> . . . . .	287
Ramil Khusainov, Ilya Shimchik, Ilya Afanasyev and Evgeni Magid	

**Part IV Signal Processing, Sensors, Systems Modelling and Control**

**Freezing Method Approach to Stability Problems** . . . . . 307  
 Artur Babiarez, Adam Czornik and Michał Niezabitowski

**Scanning Techniques with Low Bandwidth Radar for Robotic Mapping and Localization** . . . . . 321  
 Paul Fritsche and Bernardo Wagner

**Pole Placement of Linear Time-Varying Discrete Systems and Its Application to Trajectory Tracking Control of Nonlinear Systems** . . . . . 337  
 Yasuhiko Mutoh, Masakatsu Kemmotsu and Lisa Awatsu

**Bayesian Quadrature Variance in Sigma-Point Filtering** . . . . . 355  
 Jakub Průher and Miroslav Šimandl

**Adaptive Nonlinear Information Filters for Multiple Sensor Fusion** . . . . 371  
 Aritro Dey, Smita Sadhu and Tapan Kumar Ghoshal

**Decentralized Control: Application to a Turboprop Engine** . . . . . 391  
 Christophe Le Brun, Emmanuel Godoy, Dominique Beauvois, Bogdan Liacu and Ricardo Noguera

**Author Index** . . . . . 411

**Part I**  
**Invited Papers**

# Formation Control and Vision Based Localization of a System of Mobile Robots

Krzysztof Kozłowski and Wojciech Kowalczyk

**Abstract** This paper presents trajectory tracking algorithms for a group of mobile robots. The dynamics of differentially driven mobile robots with nonholonomic constraints is taken into account. The paper describes simulation results for the formation motion in the environment free from obstacles and with circle-shaped static obstacles, the experimental testbed and experiments for the kinematic version of the Vector Field Orientation algorithm. The collision avoidance utilizes local artificial potential function that requires only the local information about the environment.

## 1 Introduction

Rapid development in the field of multirobot systems began in the 1980s. It was possible due to continuous progresses in electronics (wireless communication, efficient motor control, new sensors), computer sciences (reliable and efficient embedded systems, advances in operating systems and programming languages), mechanics (more effective motors, accurate and reliable gears) and electrotechnics (durable high-capacity batteries), to mention only some of them.

It is clear now that there is no common approach for all applications. Making the right choice about the control method should depend on the task for the robot group to perform, the environment (space dimension, structured/unstructured), the type of robot and its on board equipment which determines communication capabilities, computing power and ability to determine the posture and the surroundings. The possibility of autonomous operation depends on these features.

---

K. Kozłowski (✉) · W. Kowalczyk

Chair of Control and Systems Engineering, Faculty of Computer Sciences,  
Poznan University of Technology, Piotrowo 3A, Poznan, Poland  
e-mail: krzysztof.kozlowski@put.poznan.pl

W. Kowalczyk

e-mail: wojciech.kowalczyk@put.poznan.pl

© Springer International Publishing Switzerland 2016

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*,  
Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_1

The proper choice of the system architecture is another important problem. The centralization makes some aspects of the control much simpler, however, the price is that the system is not fault-tolerant. A large number of robots means that the central part of the system is heavily loaded. The architecture also affects the communication in the system. In the centralized approach some data can be broadcast to the robot-agents, while in the decentralized one it may be necessary to use peer-to-peer communication links. As all agents utilize the same wireless medium with limited capacity the extensive data exchange may lead to large delays.

In the tasks that require a very high accuracy and synchronization the centralized approach is usually used. Virtual rigid body methods are the main area of their applications [4, 8, 18]. Leader following methods [2, 17, 19] may be considered as partly centralized because the leader plays the critical role in the reliability of the system, while damage to the follower did not disturb the rest of the formation continue the task. Behavioral methods [1, 11, 12] are entirely decentralized and consequently they are fault-tolerant but ineffective if the precise and synchronized motion must be performed.

The alternative to the local artificial potential function used in this paper is navigation function [13, 14, 20]. In this method, however, the entire knowledge of the environment is required. They require much more computing power in comparison with local artificial potential function based methods.

Other methods that utilize an approach similar to that presented in this paper is published in [15, 16]. The second presents experimental results for the algorithm that takes into account the dynamics of robots and acceleration limits on robot wheels. In [7] input-output linearization of the mobile robot dynamics is used to simplify the formation control task.

In Sect. 2 a model of the multirobot system is introduced. Section 3 presents control methods. Section 4 describes components of the experimental testbed. Simulation results are given in Sect. 5. Section 6 presents experimental results. The last section of the paper contains concluding remarks.

## 2 Model of the System

The model of the kinematics of the  $i$ -th robot ( $i = 1, \dots, N$ ), where  $N$ —number of robots, is given as follows:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{vi} \\ u_{\omega i} \end{bmatrix}, \quad (1)$$

where  $x_i, y_i, \theta_i$  represent position and orientation coordinates of the center of the  $i$ -th robot, respectively, and  $\mathbf{u}_i = [u_{vi} \ u_{\omega i}]^T$  is the input vector of linear and angular velocity controls.

The model of the dynamics of the  $i$ -th differentially driven mobile platform is given by the following equation:

$$\mathbf{M}_i \dot{\boldsymbol{\omega}}_{wi} + \mathbf{C}_i(\dot{\mathbf{q}}_i) \boldsymbol{\omega}_{wi} + \mathbf{D}_i \boldsymbol{\omega}_{wi} + \mathbf{G}_i(\mathbf{q}_i) = \boldsymbol{\tau}_i, \quad (2)$$

where  $\mathbf{M}_i$  is mass matrix,  $\mathbf{C}_i(\dot{\mathbf{q}}_i)$ —Coriolis matrix,  $\mathbf{D}_i$ —damping coefficients matrix,  $\mathbf{G}_i$ —gravitational force vector,  $\boldsymbol{\omega}_{wi} = [\omega_{Ri} \ \omega_{Li}]^T$ —robot wheels velocity vector,  $\boldsymbol{\tau}_i = [\tau_{Ri} \ \tau_{Li}]^T$ —input torque vector for left and right wheel, respectively.

$\mathbf{M}_i$  is a symmetric positive definite, constant matrix:

$$\mathbf{M}_i = \begin{bmatrix} m_{11i} & m_{12i} \\ m_{12i} & m_{11i} \end{bmatrix}, \quad (3)$$

where

$$m_{11i} = \frac{1}{4b_i^2} r_i^2 (m_i b_i^2 + I_i) + I_{wi}, \quad (4)$$

$$m_{12i} = \frac{1}{4b_i^2} r_i^2 (m_i b_i^2 - I_i). \quad (5)$$

In the above equations  $m_i = m_{ci} + 2m_{wi}$ ,  $I_i = m_{ci} a_i^2 + 2m_{wi} b_i^2 + I_{ci} + 2I_{mi}$ ;  $m_{ci}$ —mass of the body of the  $i$ -th robot,  $m_{wi}$ —mass of the robots wheel,  $I_{ci}$ —moment of inertia of the  $i$ -th robot body about the vertical axis through the center of mass,  $I_{wi}$ —moment of inertia of the wheel with the rotor of motor about the wheel axis,  $I_{mi}$ —moment of inertia of the wheel with the rotor of the motor about the wheel diameter,  $r_i$ —robot wheel radius,  $a_i$ —distance between the center of mass of the robot  $[x_{si} \ y_{si}]^T$  and wheel axis,  $2b_i$ —distance between the robot's wheels (Fig. 1).

Coriolis matrix  $\mathbf{C}(\dot{\mathbf{q}}_i)$  can be written as:

$$\mathbf{C}_i(\dot{\mathbf{q}}_i) = \begin{bmatrix} 0 & -c_i \dot{\theta}_i \\ c_i \dot{\theta}_i & 0 \end{bmatrix}, \quad (6)$$

where  $c_i = \frac{1}{2b_i} r_i^2 m_{ci} a_i$ .

Damping coefficient matrix  $\mathbf{D}_i$  is as follows:

$$\mathbf{D}_i = \begin{bmatrix} d_{11i} & 0 \\ 0 & d_{22i} \end{bmatrix}, \quad (7)$$

where  $d_{11i}$  and  $d_{22i}$  are nonnegative damping coefficients.

Gravitational vector  $\mathbf{G}_i = [0 \ 0]^T$  when robots move on the horizontal plane.

Transformation between linear and angular velocity vector  $\boldsymbol{\varpi}_i$  and robot wheel velocity vector  $\boldsymbol{\omega}_{wi}$  is as follows:

$$\boldsymbol{\omega}_{wi} = \mathbf{B}_i \boldsymbol{\varpi}_i, \quad (8)$$

where

$$\mathbf{B}_i = \begin{bmatrix} \frac{1}{r_i} & \frac{b_i}{r_i} \\ \frac{1}{r_i} & -\frac{b_i}{r_i} \end{bmatrix}. \quad (9)$$

Transformation between force and torque for the platform  $\bar{\boldsymbol{\tau}}_i = [F_i \ \tau_i]^T$  and wheel torques  $\boldsymbol{\tau}_i$  is given by the following equation:

$$\bar{\boldsymbol{\tau}}_i = \bar{\mathbf{B}}_i \boldsymbol{\tau}_i, \quad (10)$$

where

$$\bar{\mathbf{B}}_i = \begin{bmatrix} \frac{1}{r_i} & \frac{1}{r_i} \\ \frac{b_i}{r_i} & -\frac{b_i}{r_i} \end{bmatrix}. \quad (11)$$

Notice that  $\bar{\mathbf{B}}_i = \mathbf{B}_i^T$ .

Substituting (8) and its derivative into (2) and left-multiplying by the  $\bar{\mathbf{B}}_i$  model of the robots dynamics can be rewritten as follows:

$$\bar{\mathbf{M}}_i \dot{\boldsymbol{\omega}}_i + \bar{\mathbf{C}}_i(\dot{q}_i) \boldsymbol{\omega}_i + \bar{\mathbf{D}}_i \boldsymbol{\omega}_i = \bar{\mathbf{B}}_i \boldsymbol{\tau}_i \quad (12)$$

where  $\bar{\mathbf{M}}_i = \bar{\mathbf{B}}_i \mathbf{M}_i \mathbf{B}_i$ ,  $\bar{\mathbf{C}}_i = \bar{\mathbf{B}}_i \mathbf{C}_i(q_i) \mathbf{B}_i$ , and  $\bar{\mathbf{D}}_i = \bar{\mathbf{B}}_i \mathbf{D}_i \mathbf{B}_i$ .

The right side of the Eq.(12) represents the force and torque controls applied to the mobile platform.

### 3 Tracking Algorithms

This section presents two control algorithms of trajectory tracking for a group of mobile robots. Collision avoidance is based on the artificial potential functions.

#### 3.1 Do Algorithm

In this section a brief description of the algorithm originally presented in [3] is given. The dynamics of the system is taken into account. Robots of the group track desired trajectories (defined separately for each of them) avoiding collisions during the motion. When the robots are close to each another repulsive component of the control is activated. The shape of the robot is approximated with the circle, which is a reasonable assumption in the case of the differentially driven mobile robot. The algorithm takes into account uncertainties of the parameters of robots model dynamics. It is assumed that the task space is free from obstacles other than robots.

The parameter  $s$  represents the common trajectory parameter. The reference position of the  $i$ -th robot is as follows:

$$\mathbf{q}_{di} = \begin{bmatrix} x_{di}(s) \\ y_{di}(s) \end{bmatrix} = \mathbf{q}_d + \begin{bmatrix} l_{xi} \\ l_{yi} \end{bmatrix}, \quad (13)$$

where  $\mathbf{q}_d$  is the common desired trajectory for the formation and  $[l_{xi} \ l_{yi}]^T$  is the constant vector of desired offset for the  $i$ -th robot. The reference orientation is as follows:

$$\theta_{di} = \arctan\left(\frac{y'_{di}}{x'_{di}}\right) \quad (14)$$

where  $x'_{di}$  i  $y'_{di}$  and partial derivatives of the desired position of the  $i$ -th robot with respect to the  $s$  parameter:  $x'_{di} = \frac{\partial x_{di}}{\partial s}$ ,  $y'_{di} = \frac{\partial y_{di}}{\partial s}$ .

The following auxiliary variables are introduced:

$$\theta_{ie} = \theta_i - \alpha_{\theta i} \quad (15)$$

$$v_{ie} = v_i - \alpha_{vi} \quad (16)$$

where  $\alpha_{\theta i}$  i  $\alpha_{vi}$  are virtual controls of the orientation  $\theta_i$  and linear velocity  $v_i$ , respectively. They are given by the equations:

$$\alpha_{\theta i} = \theta_{di} + \arctan\left(\frac{\zeta_1}{\zeta_2}\right), \quad (17)$$

where

$$\zeta_1 = -k_0 u_{di} (-\psi(\Omega_{xi}) \sin(\theta_{di}) + \psi(\Omega_{yi}) \cos(\theta_{di})) \quad (18)$$

$$\zeta_2 = -k_0 u_{di} (\psi(\Omega_{xi}) \cos(\theta_{di}) + \psi(\Omega_{yi}) \sin(\theta_{di})) + 1 \quad (19)$$

and

$$\alpha_{vi} = \cos(\alpha_{\theta i}) (-k_0 u_{di}^2 \psi(\Omega_{xi}) + \cos(\theta_{di}) u_{di}) \quad (20)$$

$$+ \sin(\alpha_{\theta i}) (-k_0 u_{di}^2 \psi(\Omega_{yi}) + \sin(\theta_{di}) u_{di}), \quad (21)$$

where  $k_0$  is positive constant,  $\psi(\cdot)$  is a scalar function (it can be assumed that  $\psi(\cdot) = \arctan(\cdot)$ ),  $u_{di}$  is given by the equation:

$$u_{di} = \dot{s} \sqrt{(x'_{di})^2 + (y'_{di})^2}, \quad (22)$$

and  $\Omega_{xi}$ ,  $\Omega_{yi}$  are components of the sums of  $x$  and  $y$  of the position errors and collision avoidance terms:

$$\boldsymbol{\Omega}_i = \begin{bmatrix} \Omega_{xi} \\ \Omega_{yi} \end{bmatrix} = \mathbf{q}_i - \mathbf{q}_{di} + \sum_{j=1, j \neq i}^N \beta'_{ij} \mathbf{q}_{ij}. \quad (23)$$

In the last equation  $\mathbf{q}_{ij} = \mathbf{q}_i - \mathbf{q}_j$  are partial derivatives of the potential functions  $\beta'_{ij} = \frac{\partial \beta_{ij}}{\partial \left(\frac{1}{2} \|\mathbf{q}_{ij}\|^2\right)}$ .

Potential function  $\beta_{ij} \in (0, \infty)$  is obtained as the result of mapping of the function  $h_{ij} \in (0, 1)$  using the formula:

$$\beta_{ij} = \frac{h_{ij}}{1 - h_{ij}}. \quad (24)$$

In the above equation  $h_{ij}$  is a bump function. One of the possible equations of function  $B_a$  is as follows:

$$B_a(l) = \begin{cases} 1 & \text{for } l < r \\ 1 - \frac{\int_r^l f(\tau-r)f(R-\tau)d\tau}{\int_r^R f(\tau-r)f(R-\tau)d\tau} & \text{for } r \leq l < R \\ 0 & \text{for } l \geq R \end{cases} \quad (25)$$

where  $f(\cdot)$  is a scalar function that fulfills the following conditions:

- $\beta'_{ij} = 0, \beta''_{ij} \geq 0$  for  $\|\mathbf{q}_{ij}\| = \|\mathbf{q}_{ijd}\|$ ,
- $\beta_{ij} > 0$  for  $0 < \|\mathbf{q}_{ij}\| < b_{ij}$ ,
- $\beta_{ij} = 0, \beta'_{ij} = 0, \beta''_{ip} = 0, \beta'''_{ij} = 0$  for  $\|\mathbf{q}_{ip}\| \geq b_{ij}$ ,
- $\beta_{ij} = \infty$  for  $\|\mathbf{q}_{ij}\| \leq (r_i + r_j)$ ,
- $\beta_{ij} = \mu_1, \left| \beta'_{ij} \right| \leq \mu_2$  and  $\left| \beta''_{ij} \mathbf{q}_{ij}^T \mathbf{q}_{ij} \right| \leq \mu_3 \forall (r_i + r_j) < \|\mathbf{q}_{ij}\| < b_{ip}$ ,
- $\beta_{ij}$  is at least triply differentiable with respect to  $\frac{\|\mathbf{q}_{ij}\|^2}{2}$  for  $\|\mathbf{q}_{ij}\| > (r_i + r_j)$

where  $\mathbf{q}_{ijd} = \mathbf{q}_{id} - \mathbf{q}_{jd}$ ,  $b_{ij}$  is strictly positive constant satisfying the condition  $(r_i + r_j) < b_{ij} \leq \min(R_i, R_j, \|\mathbf{q}_{ijd}\|)$ ,  $\mu_1, \mu_2, \mu_3$  are positive constants.

For  $\|\mathbf{q}_{ij}\| \leq (r_i + r_j)$  functions  $\beta'_{ij} = \infty, \beta''_{ij} = \infty, \beta'''_{ij} = \infty$  and  $\beta'_{ij} = \frac{\partial \beta_{ij}}{\partial \left(\frac{1}{2} \|\mathbf{q}_{ij}\|^2\right)}$ ,  $\beta''_{ij} = \frac{\partial^2 \beta_{ij}}{\partial \left(\frac{1}{2} \|\mathbf{q}_{ij}\|^2\right)^2}, \beta'''_{ij} = \frac{\partial^3 \beta_{ij}}{\partial \left(\frac{1}{2} \|\mathbf{q}_{ij}\|^2\right)^3}$  for  $\|\mathbf{q}_{ij}\| > (R_i + R_j)$ .

For the function  $f(c) = e^{-\frac{1}{c}}$  Eq. (24) above conditions are fulfilled.

Control law of the  $i$ -th robot is as follows:

$$\boldsymbol{\tau}_i = (\bar{\mathbf{B}}_i)^{-1} \left( -\mathbf{L}_i \boldsymbol{\varpi}_{ie} - \boldsymbol{\Phi}_i \boldsymbol{\Theta}_i - [\chi_i \theta_{ie}]^T \right) \quad (26)$$

where

$$\chi_i = \boldsymbol{\Omega}_i^T - \theta_{ie} \frac{\partial \alpha_{\theta i}}{\partial \mathbf{q}_i} - \sum_{j=1, j \neq i}^N \left( \frac{\partial \alpha_{\theta i}}{\partial \mathbf{q}_{ij}} \theta_{ie} - \frac{\partial \alpha_{\theta i}}{\partial \mathbf{q}_{ji}} \theta_{je} \right) \bar{\mathbf{A}}_{2i} \quad (27)$$

and  $L_i$  is symmetric, positive definite matrix of the coefficients. Model parameter vector is as follows:

$$\Theta_i = \left[ b_i c_i \quad \bar{d}_{11i} \quad \bar{d}_{12i} \quad \bar{m}_{11i} \quad \frac{c_i}{b_i} \quad \bar{d}_{21i} \quad \bar{d}_{22i} \quad \bar{m}_{22i} \right]^T, \quad (28)$$

and the matrix of variables:

$$\Phi_i = \begin{bmatrix} \omega_i^2 & -\alpha_{vi} & -\alpha_{\omega i} & -\dot{\alpha}_{vi} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -v_i \dot{\theta}_i & -\alpha_{vi} & -\alpha_{\omega i} & -\dot{\alpha}_{\omega i} \end{bmatrix}. \quad (29)$$

The product  $\Phi_i \Theta_i$  in Eq. (26) is the adaptive component of the control (the algorithm takes into account uncertainties of the robot dynamics parameters). Vector  $\Delta_{2i}$  in (26) is as follows:

$$\Delta_{2i} = \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix}. \quad (30)$$

Virtual control of the angular velocity  $\alpha_{\omega i}$  in Eq. (29) is given by the equation:

$$\alpha_{\omega i} = -k\theta_{ie} - \frac{\Omega_i^T \Delta_{1i}}{\theta_{ie}} + \frac{\partial \alpha_{\theta i}}{\partial \mathbf{q}_{di}} \dot{q}_{od} + \frac{\partial \alpha_{\theta i}}{\partial \theta_{di}} \dot{\theta}_{od} + \frac{\partial \alpha_{\theta i}}{\partial u_{di}} \dot{u}_{od} + \frac{\partial \alpha_{\theta i}}{\partial \mathbf{q}_i} (\mathbf{u}_i + \Delta_{1i}) \quad (31)$$

$$+ \sum_{j=1, j \neq i}^N \frac{\partial \alpha_{\theta i}}{\partial \mathbf{q}_{ij}} (\mathbf{u}_i + \Delta_{1i} - (\mathbf{u}_j + \Delta_{1j})), \quad (32)$$

where  $k$  is a positive constant and  $\Delta_{1i}$  in (31) is as follows:

$$\Delta_{1i} = \begin{bmatrix} (\cos(\theta_{ie}) - 1) \cos(\alpha_{\theta i}) - \sin(\theta_{ie}) \sin(\alpha_{\theta i}) \\ \sin(\theta_{ie}) \cos(\alpha_{\theta i}) + (\cos(\theta_{ie}) - 1) \sin(\alpha_{\theta i}) \end{bmatrix} \alpha_{vi}. \quad (33)$$

The bounded control  $\mathbf{u}_i$  in Eq. (31) is given by the equation:

$$\mathbf{u}_i = -k_0 u_{di}^2 \begin{bmatrix} \psi(\Omega_{ix}) \\ \psi(\Omega_{iy}) \end{bmatrix} + \dot{\mathbf{q}}_{di}. \quad (34)$$

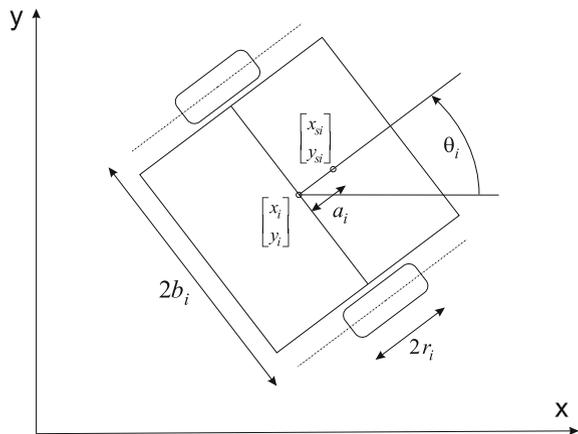
In [3] additional equations are available, including higher order derivatives needed for implementation. Stability proof and asymptotical convergence of the position and orientation to desired values is also presented in this paper.

### 3.2 VFO Algorithm

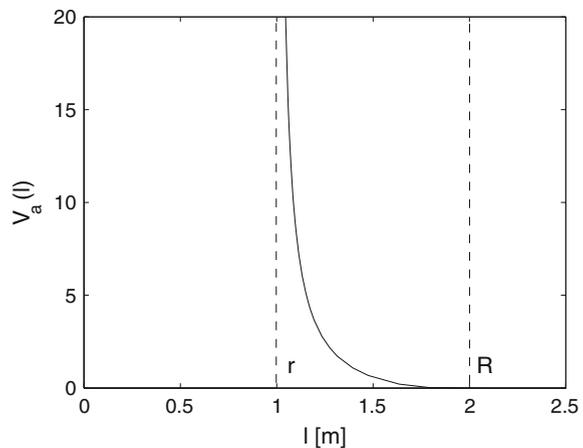
Vector Field Orientation method (VFO) for a single differentially driven mobile robot is proposed in [10]. Combining this approach with local artificial potential function method a new control algorithm for multiple mobile robots is proposed [6]. It is assumed that the model of the system dynamics is known and accurate. In the task space static obstacles can appear. They are modeled with circles as the robots are. Similar to the previous algorithm the range of the artificial potential function associated with the obstacle is limited, hence autonomous operation with only on board sensor usage is possible. In Fig. 2 APF as a function of distance to the center of the obstacle is shown. It disappears at  $l = R$ .

Reference path for the  $i$ -th robot is given by the vector  $q_{di} = [x_{di} \ y_{di}]^T$ . Reference orientation can be computed using reference linear velocity:  $\theta_{di} = \text{atan2}c$

**Fig. 1** Differentially driven mobile robot,  $r_i$ —radius of the wheel,  $2b_i$ —distance between wheels,  $a_i$ —distance between wheel axis and center of mass of the robot,  $[x_i \ y_i]^T$ —robots position,  $\theta_i$ —robots orientation,  $[x_{si} \ y_{si}]^T$ —position of the center of mass of the robot



**Fig. 2** APF as a function of distance to the center of the obstacle (indexes omitted for simplicity)



( $\dot{y}_{di}, \dot{x}_{di}$ ) where  $atan2c(\cdot, \cdot)$  is continuous version of the  $atan2(\cdot, \cdot)$  function described in [10].

Let's define  $\mathbf{q}_i = [x_i \ y_i]^T$ —position vector for the  $i$ -th robot. One can write the position tracking error as follows:

$$\mathbf{e}_i = \begin{bmatrix} e_{xi} \\ e_{yi} \end{bmatrix} = \mathbf{q}_{di} - \mathbf{q}_i. \quad (35)$$

The dynamics of the robot can be linearized by the following control feedback:

$$\boldsymbol{\tau}_i = \bar{\mathbf{B}}_i^{-1} (\bar{\mathbf{M}}_i \mathbf{v}_i + \bar{\mathbf{C}}_i \boldsymbol{\omega}_i + \bar{\mathbf{D}}_i \boldsymbol{\omega}_i), \quad (36)$$

where  $\mathbf{v}_i$  is a new control input.

Substituting (36) into (12) one obtains an equation of linearized robots dynamics in the form of integrator:

$$\mathbf{v}_i = \begin{bmatrix} v_{vi} \\ v_{\omega i} \end{bmatrix} = \dot{\boldsymbol{\omega}}_i. \quad (37)$$

For the linearized dynamics of the robot the following linear control law can be proposed:

$$\mathbf{v}_i = \dot{\mathbf{u}}_i + \mathbf{k}_{\boldsymbol{\omega}} (\mathbf{u}_i - \boldsymbol{\omega}_i), \quad (38)$$

where  $\mathbf{k}_{\boldsymbol{\omega}}$  is positive definite diagonal gain matrix. New control  $\mathbf{u}_i = [u_{vi} \ u_{\omega i}]^T$  is computed using VFO (Vector Field Orientation) method [10].

According to this method a convergence vector is introduced:

$$\mathbf{h}_i = \begin{bmatrix} h_{xi} \\ h_{yi} \\ h_{\theta i} \end{bmatrix} = \begin{bmatrix} k_p E_{xi} + \dot{x}_{di} \\ k_p E_{yi} + \dot{y}_{di} \\ k_{\theta} e_{ai} + \dot{\theta}_{ai} \end{bmatrix}, \quad (39)$$

where  $k_p, k_{\theta} > 0$  are control gains for position and orientation, respectively.

Modified position errors are defined as follows:

$$E_{xi} = e_{xi} + \bar{w}_{xi} \quad (40)$$

$$E_{yi} = e_{yi} + \bar{w}_{yi} \quad (41)$$

where  $\bar{w}_i = [\bar{w}_{xi} \ \bar{w}_{yi}]^T$  is the collision avoidance term that will be described further in this paper;  $e_{ai} = \theta_{ai} - \theta_i$  is the auxiliary orientation error. The auxiliary orientation variable  $\theta_{ai}$  is defined as  $\theta_{ai} = atan2c(h_{yi}, h_{xi})$ . Next the following control law  $\mathbf{u}_i = [u_{vi} \ u_{\omega i}]^T$  for the  $i$ -th mobile platform is proposed:

$$\begin{aligned} u_{vi} &= h_{xi} \cos \theta_i + h_{yi} \sin \theta_i \\ u_{\omega i} &= h_{\theta i} \end{aligned} \quad (42)$$

The VFO method requires the following assumptions be imposed.

**Assumption 3.1** Desired trajectories do not intersect artificial potential function areas of obstacles and robots do not interact when tracking is executed perfectly.

**Assumption 3.2** If the robot position is in the repel area then the reference trajectory is frozen:

$$\mathbf{q}_{di}(t) = \mathbf{q}_{di}(t^-), \quad (43)$$

where  $t^-$  is the time value before robot gets to the repel area. Higher derivatives of  $q_{di}(t)$  are kept zero until robot leaves the repel area [9].

**Assumption 3.3** When  $e_{ai} \in (\frac{\pi}{2} + \pi d - \delta, \frac{\pi}{2} + \pi d + \delta)$ , where  $\delta$  is a small positive value,  $d = 0, \pm 1, \pm 2, \dots$ , then auxiliary orientation variable  $\theta_{ai}$  is replaced by  $\tilde{\theta}_{ai} = \theta_{ai} + \text{sgn}(e_{ai} - (\frac{\pi}{2} + \pi d)) \varepsilon$ , where  $\varepsilon$  is a small value that fulfills condition  $\varepsilon > 0$  and  $\text{sgn}(\cdot)$  denotes the signum function.

**Assumption 3.4** When robot reaches a saddle point the reference trajectory is disturbed to drive robot out of the local equilibrium point. In the saddle point the following condition is fulfilled:

$$\|\mathbf{h}_i^*\| = 0, \quad (44)$$

where  $\mathbf{h}_i^* = [h_{xi} \ h_{yi}]^T$ . In this case  $\theta_{ai}(t)$  is frozen:  $\theta_{ai}(t) = \theta_{ai}(t^-)$ .

There is one unstable equilibrium point (saddle point) associated with each obstacle. It appears as a result of cancelling attraction vector (connected with the motion to the desired position) and repulsion vector (connected with the collision avoidance).

In [6] stability proof is presented. It is also shown that the position and orientation error can be reduced to the arbitrarily small neighborhood of desired values.

The presented algorithm can be applied to the simplified model of the system. By neglecting the dynamics of the mobile platforms the control algorithm can be simplified. The results of the Eq. (42) can be used as an input to the system (1). The Assumptions 3.1–3.4 still hold in this case. Stability proof and asymptotic convergence of the position and orientation to the desired values for kinematic version of the VFO algorithm with collision avoidance are presented in [5].

### 3.3 Artificial Potential Function

Collision avoidance behavior is based on the artificial potential functions (APF). All physical objects in the environment like robots and static obstacles are surrounded with APF's that rise to infinity near objects boundaries  $r_j$  ( $j$ —number of the robot/obstacle) and decrease to zero at some distance  $R_j > r_j$ .

One can introduce the following function:

$$B_{aij}(l_{ij}) = \begin{cases} 0 & \text{for } l_{ij} < r_j \\ e^{\frac{l_{ij}-r_j}{l_{ij}-R_j}} & \text{for } r_j \leq l_{ij} < R_j \\ 0 & \text{for } l_{ij} \geq R_j \end{cases}, \quad (45)$$

that gives output  $B_{aij}(l_{ij}) \in (0, 1)$ . The distance between the  $i$ -th robot and the  $j$ -th object is as follows:  $l_{ij} = \|\mathbf{q}_j - \mathbf{q}_i\|$ , where  $\mathbf{q}_i$ —position of the  $i$ -th robot,  $\mathbf{q}_j$ —position of the  $j$ -th object.

In [3] APF given by Eq. (25) is proposed. It is smooth at the bound  $R_j$  where it disappears as the function (45), however, it requires higher computing power because the integral must be computed in the each iteration of the algorithm. In [9] simpler APF is proposed but due to the limited degree of differentiability its usage (in the original form) is limited to the kinematic version of the algorithm.

By scaling function (45) to the range  $(0, \infty)$  using equation:

$$V_{aij}(l_{ij}) = \frac{B_{aij}(l_{ij})}{1 - B_{aij}(l_{ij})}, \quad (46)$$

APF is defined which is used to avoid collisions. Summing gradients of the APF's associated with robots and obstacles the term  $\bar{w}_i$  required to compute  $E_{x_i}$  and  $E_{y_i}$  is as follows:

$$\bar{w}_i = - \sum_{j=1, j \neq i}^N \left[ \frac{\partial V_{arij}(l_{ij})}{\partial \mathbf{q}_i} \right]^T - \sum_{k=1}^M \left[ \frac{\partial V_{aoik}(l_{ik})}{\partial \mathbf{q}_i} \right]^T, \quad (47)$$

where  $M$ —number of static obstacles;  $V_{arij}$  and  $V_{aoik}$  are the APF's given by (46) associated with the  $j$ -th robot and the  $k$ -th obstacle, respectively.

The first and second order time derivatives of the control algorithm equations are calculated as follows:

$$\dot{\theta}_{ai} = \frac{\dot{h}_{yi}h_{xi} - h_{yi}\dot{h}_{xi}}{h_{xi}^2 + h_{yi}^2} \quad (48)$$

$$\ddot{\theta}_{ai} = \frac{\ddot{h}_{yi}h_{xi}^3 + \ddot{h}_{yi}h_{xi}h_{yi}^2 - h_{yi}\ddot{h}_{xi}h_{xi}^2 - h_{yi}^3\ddot{h}_{xi}}{(h_{xi}^2 + h_{yi}^2)^2} \quad (49)$$

$$+ \frac{-2\dot{h}_{xi}h_{xi}^2\dot{h}_{yi} + 2\dot{h}_{xi}^2h_{xi}h_{yi} - 2\dot{h}_{yi}^2h_{yi}h_{xi} + 2\dot{h}_{yi}h_{yi}^2\dot{h}_{xi}}{(h_{xi}^2 + h_{yi}^2)^2} \quad (50)$$

$$\dot{\mathbf{E}}_i(t) = \dot{\mathbf{e}}_i - \sum_{j=1, j \neq i}^N \left[ \frac{\partial^2 V_{arij}(l_{ij})}{\partial \mathbf{q}_i^2} \right] \dot{\mathbf{q}}_j - \sum_{k=1}^M \left[ \frac{\partial^2 V_{aoik}(l_{ik})}{\partial \mathbf{q}_i^2} \right] \dot{\mathbf{q}}_k \quad (51)$$

$$\ddot{\mathbf{E}}_i(t) = \ddot{\mathbf{e}}_i - \sum_{j=1, j \neq i}^N \left( \left[ \frac{\partial^2 V_{arj}(l_{ij})}{\partial \mathbf{q}_i^2} \right] \ddot{\mathbf{q}}_j + \begin{bmatrix} \frac{\partial^3 V_{arj}(l_{ij})}{\partial x_i^3} & \frac{\partial^3 V_{arj}(l_{ij})}{\partial x_i^2 \partial y_i} & \frac{\partial^3 V_{arj}(l_{ij})}{\partial y_i^2 \partial x_i} \\ \frac{\partial^3 V_{arj}(l_{ij})}{\partial x_i^2 \partial y_i} & \frac{\partial^3 V_{arj}(l_{ij})}{\partial y_i^2 \partial x_i} & \frac{\partial^3 V_{arj}(l_{ij})}{\partial y_i^3} \end{bmatrix} \begin{bmatrix} \dot{x}_i^2 \\ \dot{x}_i \dot{y}_i \\ \dot{y}_i^2 \end{bmatrix} \right) \quad (52)$$

$$- \sum_{k=1}^M \left( \left[ \frac{\partial^2 V_{aok}(l_{ik})}{\partial \mathbf{q}_i^2} \right] \ddot{\mathbf{q}}_k + \begin{bmatrix} \frac{\partial^3 V_{aok}(l_{ik})}{\partial x_i^3} & \frac{\partial^3 V_{aok}(l_{ik})}{\partial x_i^2 \partial y_i} & \frac{\partial^3 V_{aok}(l_{ik})}{\partial y_i^2 \partial x_i} \\ \frac{\partial^3 V_{aok}(l_{ik})}{\partial x_i^2 \partial y_i} & \frac{\partial^3 V_{aok}(l_{ik})}{\partial y_i^2 \partial x_i} & \frac{\partial^3 V_{aok}(l_{ik})}{\partial y_i^3} \end{bmatrix} \begin{bmatrix} \dot{x}_i^2 \\ \dot{x}_i \dot{y}_i \\ \dot{y}_i^2 \end{bmatrix} \right) \quad (53)$$

$$\dot{u}_{vi} = \dot{h}_{xi} \cos(\theta_i) + \dot{h}_{yi} \sin(\theta_i) + \omega_i (-h_{xi} \sin(\theta_i) + h_{yi} \cos(\theta_i)) \quad (54)$$

$$\dot{u}_{\omega i} = k_{\theta} \dot{e}_{ai}(t) + \ddot{\theta}_{ai}(t) \quad (55)$$

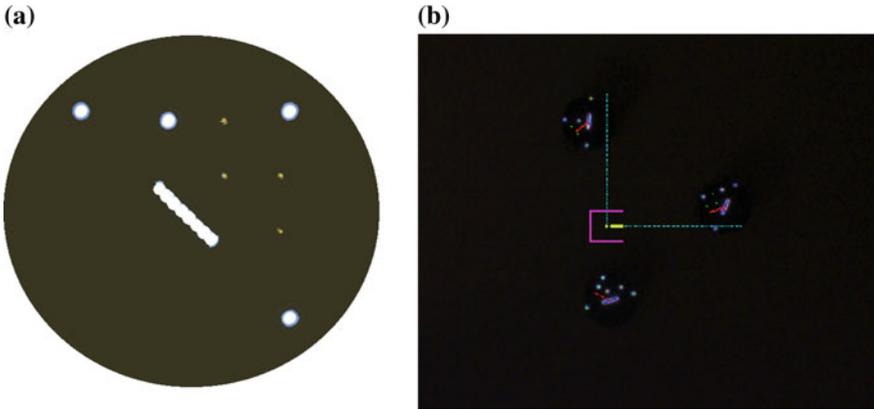
## 4 Multirobot Testbed

This section presents the system architecture of the multirobot testbed. Software and hardware modules of the system are described in detail.

The localization system is based on the digital camera uEye UI-1240SE-C combined with PENTAX C418DX lens (focal length 4.8 mm) (Fig. 3) connected to the PC computer via USB2.0 interface. The camera is fixed above the scene surface at 2.7 m and covers rectangular area  $4 \times 3.2$  m where the robots can operate. The

**Fig. 3** USB top-view camera





**Fig. 4** LED markers. **a** View of the LED marker used for robot localization. **b** Task space view in the marker recognition application

camera has a  $1/2''$  CMOS sensor with resolution  $1280 \times 1024$  pixels and ensures the possibility of image acquisition frequency 25 frames per second with full resolution.

Vision server processes the data from the camera and recognizes current robots posture using LED markers (Fig. 4a) which are mounted on the top of each robot. The LED marker consists of a line of 8 diodes which form a large area for rough marker detection, 3 'corner diodes' for accurate localization and 5 diodes for coding the number of the marker/robot. Figure 4b shows the screen-shot from the vision server application. It consists of the top view of three robots. In Fig. 5 MTracker robot without marker is shown. Fig. 6a presents three robots equipped with markers. In Fig. 6b MTracker robots in various hardware configurations are shown.

Single measurement requires an image from the camera which is processed in order to determine the position of the markers on the image frame. Calculated positions of the robots are transformed from the image coordinate space to the physical coordinates. Resolution of the vision localization based on the described algorithm is about 0.4mm for the position coordinates and  $0.3^\circ$  for the orientation angle. A processed image with calculated position is used for visualization on the operator console.

The high level controller is an optional component of the presented system. As the low level controller is equipped with a radio transceiver (cc2500 radio module) it is possible to replace WiFi subsystems used in localization and operator console systems. Such modularity means that the high level controller is not mandatory.

In the presented system a built-in computer with Intel Atom 1.6 GHz processor and SSD disk is used. It may run under Windows, Linux, QNX or some other operating system.

High-level task manager receives messages through the WiFi radio link and passes them to the high level motion controller. The outputs of this controller are velocities of robot wheels that are transmitted through a serial port to the low level controller.

**Fig. 5** MTracker robot without the marker mounted on the top



**(a)**



**(b)**



**Fig. 6** Formation of robots. **a** Formation of three MTrackers with active markers. **b** MTracker robots in various hardware configurations

Passing these signals through the task manager makes it possible to send their values to the remote operator console, present them on the screen and store in a database or file. Depending on the level of autonomy of the robot the motion planner robot can be used or not. The obstacle recognition subsystem analyzes the location data frame provided by the localization subsystem and classifies if a particular marker is mounted on the current robot or on the other robots/obstacles.

The robots are equipped with two controllers: a high-level PC controller and a DSP low-level controller.

The low level controller is responsible for the robot motor control and communication with the high-level controller. Other modules of the system can be used optionally. Because of the high computational capabilities (signal processor TMS 320F28335 150MHz) the low level controller takes the role of the platform motion controller as well as the formation controller when simple algorithms are implemented. It manages measurements using a variety of sensors: encoders, IR and proximity sensors, gyroscopes, two-axis accelerometers and compasses. Depending on the needs these values can be used locally or transmitted to other subsystems.

The platform is equipped with a cc2500 radio transceiver. If there is no need to use the high level control, this radio can be used to receive messages from the localization system, from the operator console or sending the robot state to visualize it and store in a remote system (in this case WiFi radios are replaced with cc2500 transceivers).

## 5 Simulation Results

In this section simulation results of the presented control algorithms are given. Both algorithms were tested with  $N = 10$  robots.

### 5.1 Do Algorithm

Figure 7a shows robots paths in  $(x, y)$  plane. Initially robots are located in random coordinates surrounding the origin of the coordination frame. Desired paths are marked with dashed lines and robots paths with solid ones. Figure 7b presents the initial part of the simulation. Figure 7c illustrates desired orientation (dashed line) and real robot orientations.

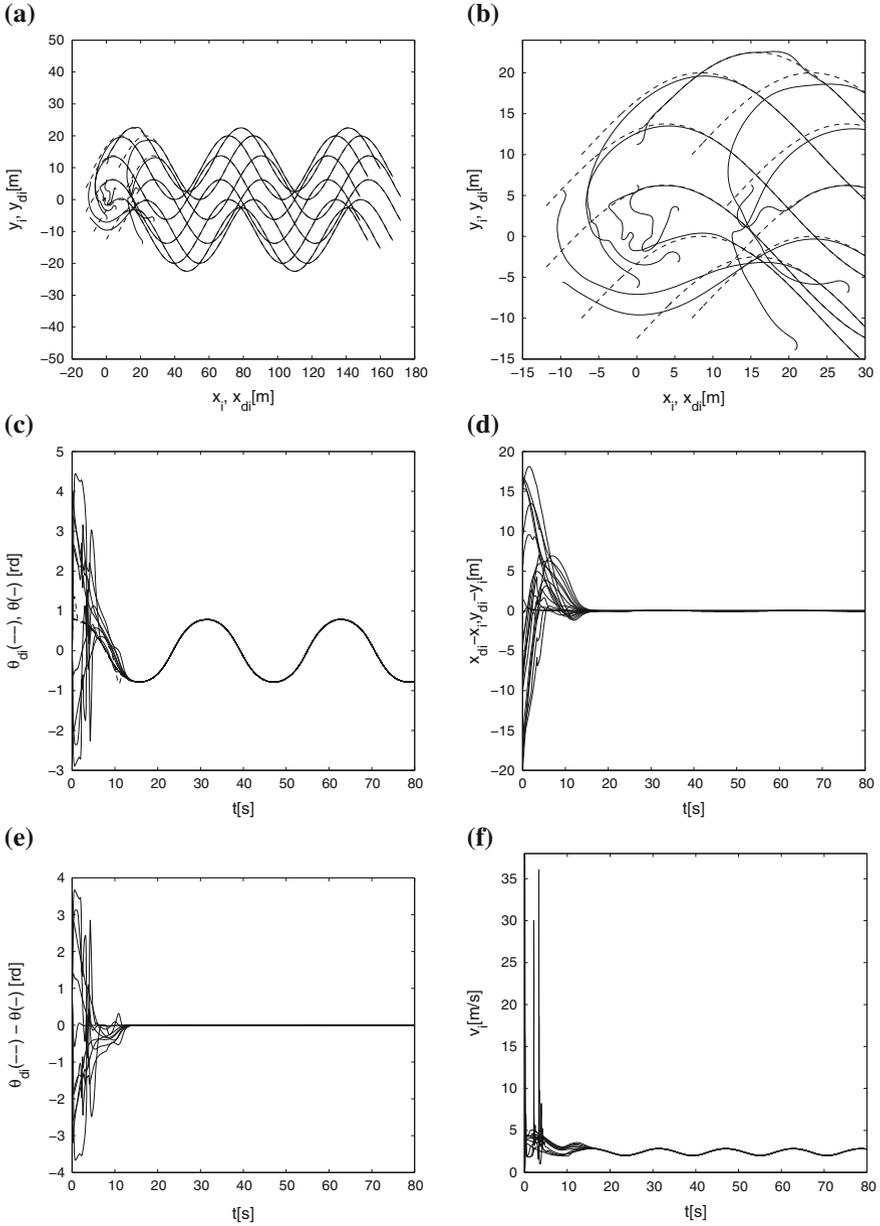
As can be observed in Fig. 7d, e, position and orientation errors converge to near zero values in about 15 s.

In Figs. 7f and 8a linear and angular velocities of the robot platform are shown. In the transient states large value peaks are observed, but when robots reach desired velocity values oscillate smoothly as the desired trajectories.

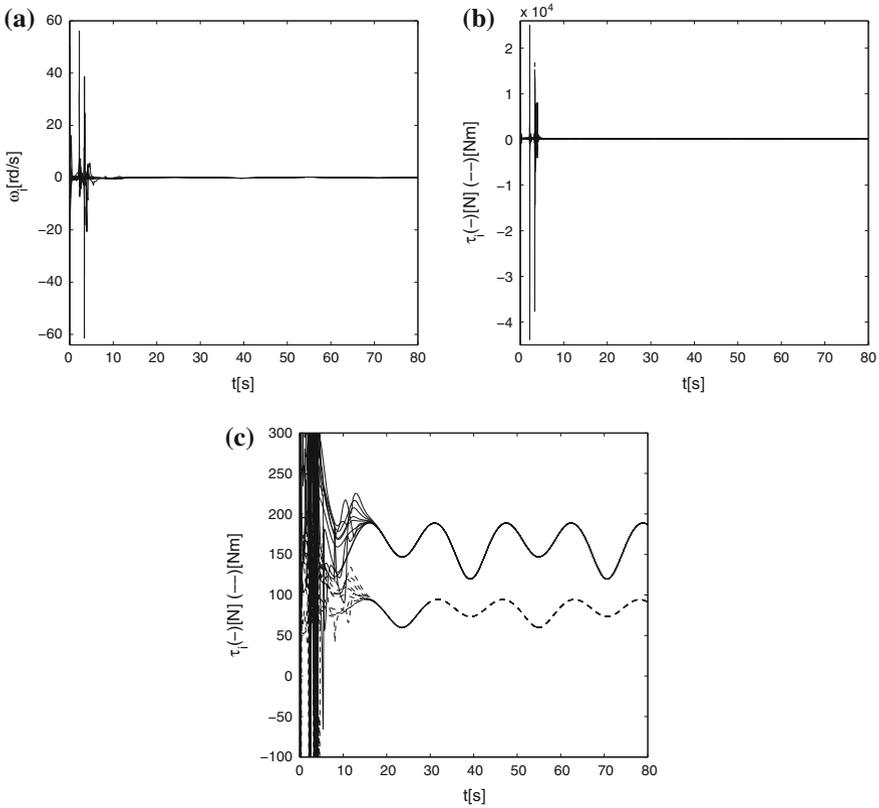
In Fig. 8b control force and torques are presented. Similar to the velocities signals large value peaks that are observed. Figure 8c presents a scaled graph to highlight the forces (solid lines) and torques (dashed lines) applied to the platform during a longer period.

### 5.2 VFO Algorithm

In this section VFO algorithm simulation results are presented. The scenario differs from that presented in Sect. 5.1 because a static obstacle can occur in the environment.



**Fig. 7** Do algorithm. **a** Robot paths on  $(x, y)$ -plane. **b** Robot paths on  $(x, y)$ -plane during initial transient state. **c** Robot orientations  $\theta_i$ . **d** Position errors. **e** Orientation errors. **f** Linear velocities



**Fig. 8** Do algorithm. **a** Angular velocities. **b** Control forces and torques. **c** Control forces and torques (graph with changed scale)

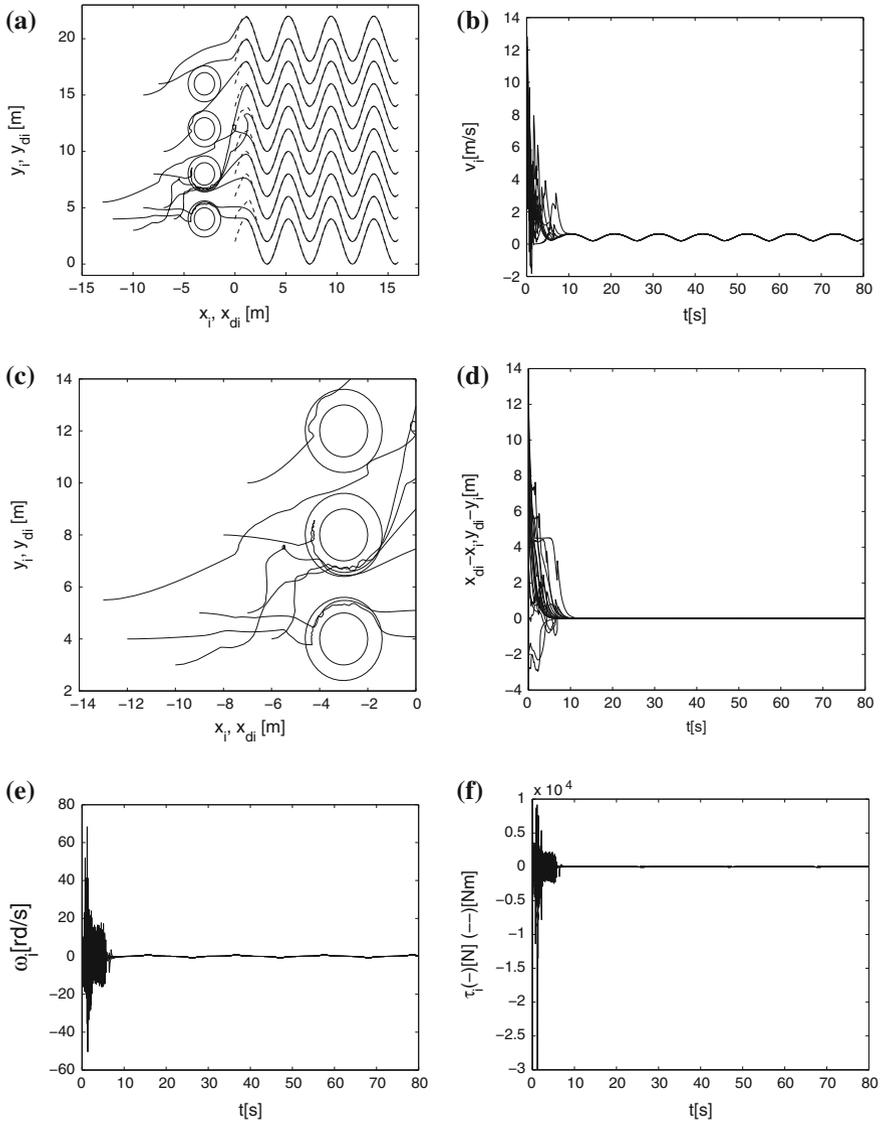
Figure 9a shows robots paths in  $(x, y)$  plane. Robots' initial positions are random points located in the left half of the coordination plane. The desired trajectories start from  $x = 0$  line. The robots have to move through the area where four static obstacles are located before they reach the desired positions. Figure 9c presents robot paths during the transient state.

In Fig. 9d position errors are illustrated. They reach desired trajectories in about 12 s.

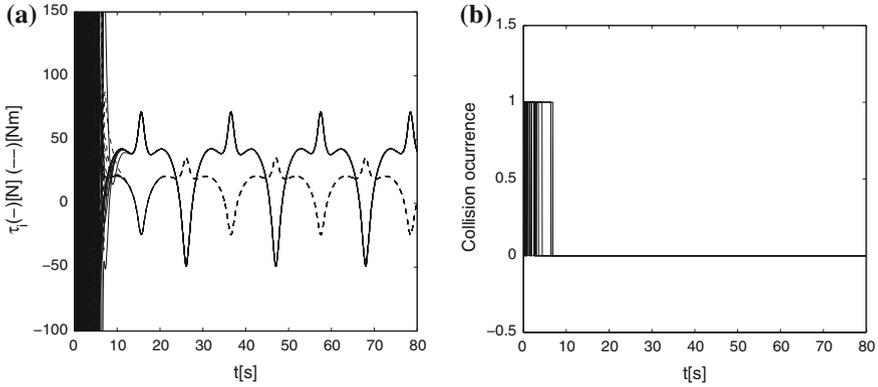
Figure 9b, e show linear and angular velocities of the platform.

Figure 9f presents control forces (solid line) and control torques (dashed lines). Figure 10a highlights the same signal in steady state.

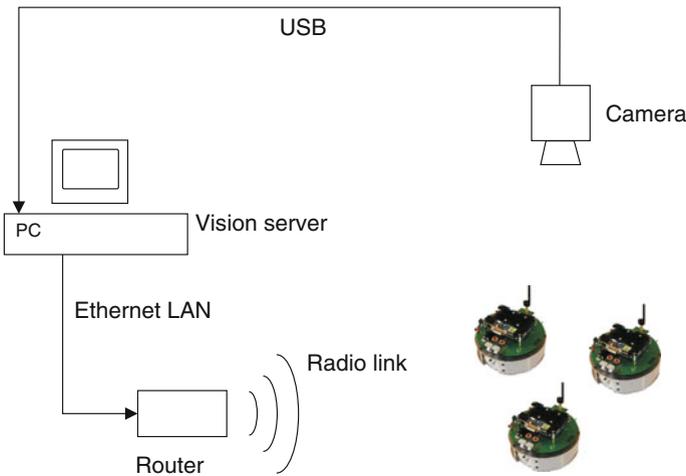
In Fig. 10b the signal representing collision avoidance module activity is shown (1—active, 0—inactive). As expected, the module is activated many times during initial motion and never activates during motion along the desired trajectories.



**Fig. 9** VFO algorithm. **a** Robot paths on  $(x, y)$ -plane. **b** Linear velocities. **c** Position errors. **d** Robot orientations  $\theta_i$ . **e** Angular velocities. **f** Control forces and torques



**Fig. 10** VFO algorithm. **a** Control forces and torques (graph with changed scale). **b** Collision occurrence



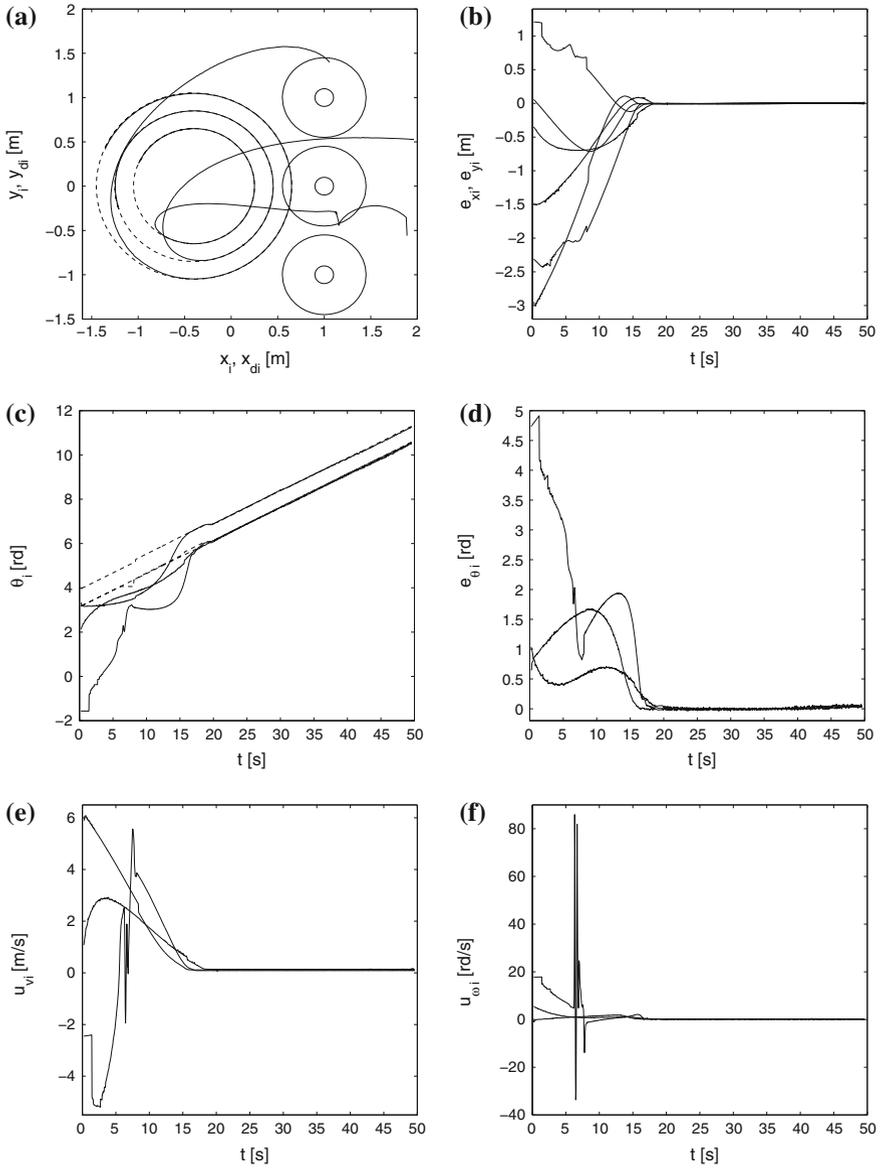
**Fig. 11** Physical implementation of the system

## 6 VFO Algorithm—Experimental Results

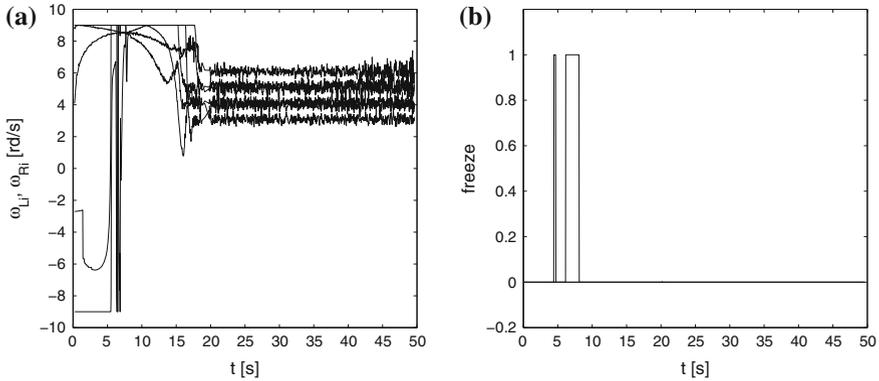
This section presents experimental results of the VFO algorithm for kinematic model of the robot. In Fig. 11 scheme of the experimental testbed is shown. In experiment 1 three robots move along circular desired trajectories. Initially, robots are positioned far from the desired trajectories (Fig. 12a).

In this figure robot paths on  $(x, y)$  plane are presented. Dashed lines represent the desired paths of robots.

In Fig. 12b position errors as a function of time are shown. They decrease to the near zero value in less then 20s.



**Fig. 12** VFO algorithm experiment 1. **a** Robot paths on  $(x, y)$ -plane. **b** Position errors. **c** Orientation of the robots. **d** Orientation errors. **e** Linear velocity controls. **f** Angular velocity controls



**Fig. 13** VFO algorithm experiment 1. **a** Wheel velocity controls. **b** Collision avoidance activation/desired trajectory freeze signal

Figure 12c shows a time graph of the orientations. Figure 12d presents orientation errors.

In Fig. 12d orientation errors are shown.

In Fig. 12e, f linear and angular controls are presented.

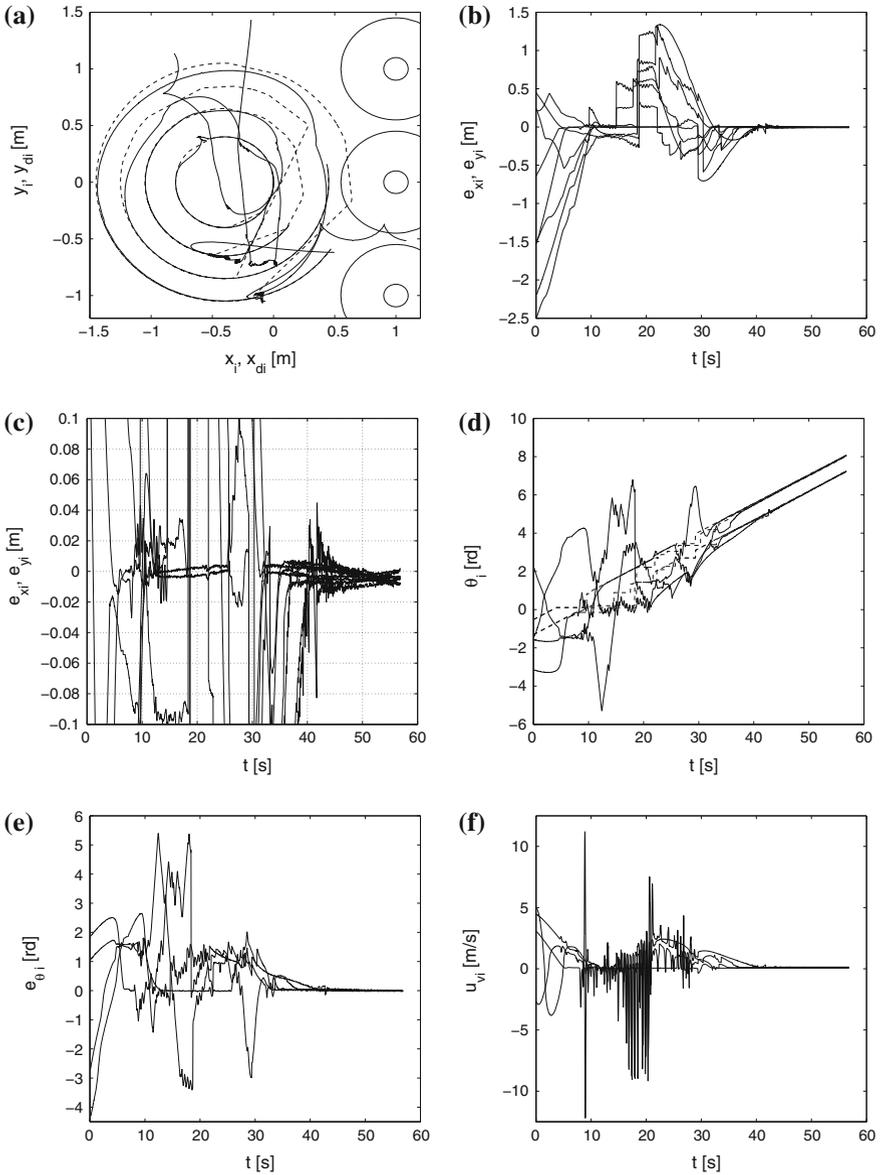
Figure 13a shows angular controls for the left and right wheel. Wheel velocities are scaled [10] with a procedure that keeps the direction of motion of the mobile platform. It limits the wheel angular velocities to  $\pm 9 \frac{rad}{s}$ .

Figure 13b shows the signal representing collision avoidance module activity (1—active, 0—inactive). It also represents the period when the desired trajectory is ‘frozen’.

In the second experiment the desired trajectories are of the same type, however, initial configuration causes the robots to come into hard interaction. Robots must mutually bypass each other and this process increases the transient state time. In this case four robots are used. Figure 14a presents robots paths in  $(x, y)$ -plane. The straight segments of the desired trajectory exist thanks to the process of the desired trajectory ‘freezing’ in the case of the occurrence of the collision. When the robot completes the collision avoidance procedure the desired trajectory ‘jumps’ to catch up with the task execution. Initially, robots have to avoid collision with the static obstacles centered at points  $(1, 1)$ ,  $(1, 0)$  and  $(1, -1)$ .

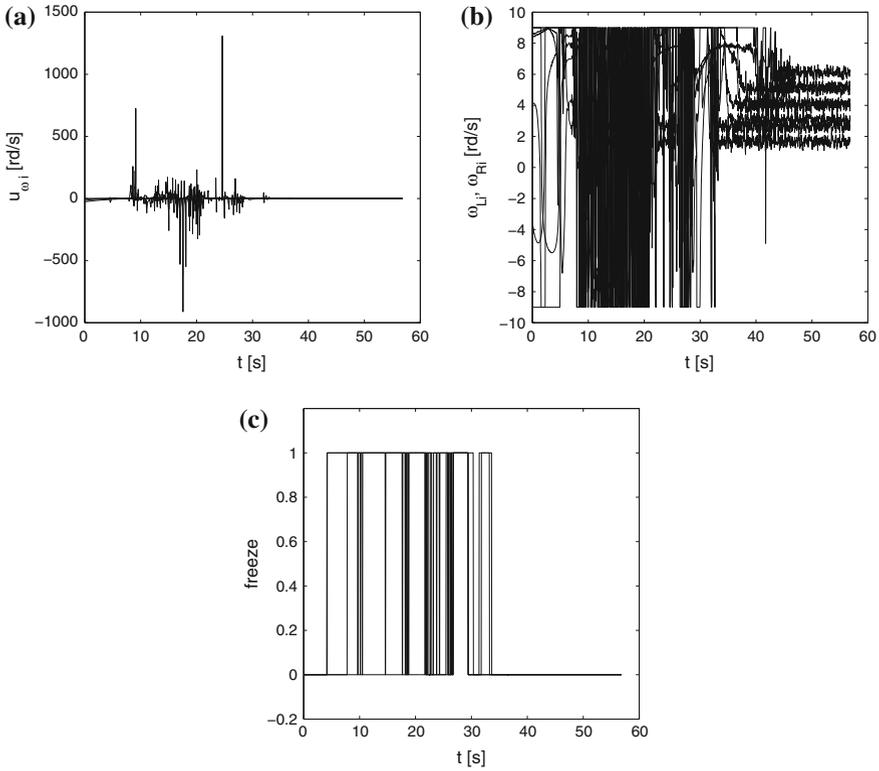
Figure 14b presents position errors. They are quickly reduced, however, when robots are near their desired positions they have to rearrange the formation bypassing each other. The position errors increase during this process (15–25 s of the experiment) and then fall to near zero value at about the 55th second. In Fig. 14c the same graph with changed y-axis scale is presented. As can be observed, the error in steady state is less than 1 cm in each coordinate.

In Fig. 14d orientations (solid lines) and their desired values (dashed lines) are presented. In Fig. 14e orientation errors are shown.



**Fig. 14** VFO algorithm experiment 2. **a** Robot paths on  $(x, y)$ -plane. **b** Position errors. **c** Orientation of the robots. **d** Position errors (orientation of the robots). **e** Orientation errors. **f** Linear velocity controls

Figures 14f and 15a present linear and angular controls for the mobile platforms. In Fig. 15b angular velocities of left and right wheel are shown. In Fig. 15c the



**Fig. 15** VFO algorithm experiment 2. **a** angular velocity controls. **b** wheel velocity controls. **c** collision avoidance activation/desired trajectory freeze signal

signal representing collision avoidance module activity is illustrated (1—active, 0—inactive).

### 6.1 Conclusions

This paper presents control algorithms for trajectory tracking by the multiple robots. Dynamic properties of the nonholonomic mobile platforms were taken into account. Robots track desired trajectories avoiding collisions. For both Do and VFO algorithms simulation results are presented. For the kinematic version of the VFO algorithm experimental results are shown.

The testbed for the multiple mobile robot system is described in detail including both hardware and software components. Experimental results for the kinematic version of the VFO algorithm is presented. Collision avoidance was achieved by the APF use.

All presented algorithms provide practical convergence as the position and orientation errors decrease to the small values. In all algorithms oscillations connected with collision avoidance were observed. They resulted from the sensitivity of the system near the obstacle or in the case the robots are near each other. In these cases the control signal is obtained as a sum of tracking term and collision avoidance term. The second one increases rapidly near the obstacle giving a highly variable direction of resulting control. This leads to oscillations that may be even increased by the inevitable delays in the real system.

The Do algorithm is characterized by the highest computational complexity. Simulation results were obtained using numerical differentiation of some higher order components of the control equations. It was difficult to calculate these derivatives manually. In VFO algorithm all the terms of the control equations were implemented in analytical form.

In both simulation and experimental results control signals were reaching practically unrealizable values. Usually, they were peaks connected with the collision avoidance processes. In the experiments their influence on motor control signals was removed by the use of the scaling procedure that keeps the direction of motion of the robot. It extends the transient state but avoids wheel slippage and unpredictable disturbances on motion direction because of the uncertainties in dynamic limits of motor drives, gear and wheel.

Future work includes experiments with a greater number of robots. The authors plan to improve robots behavior in transient states and conduct some experiments utilizing algorithms based on the navigation functions.

## References

1. Balch, T., Arkin, R.: Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **14**(6), 926–939 (1998)
2. Das, A., Fierro, R., Kumar, V., Ostrowski, J., Spletzer, J., Taylor, C.: A vision-based formation control framework. *IEEE Trans. Robot. Autom.* **18**(5), 813–825 (2002)
3. Do, K.: Formation tracking control of unicycle-type mobile robots with limited sensing ranges. *IEEE Trans. Control Syst. Technol.* **16**(3), 527–538 (2008)
4. Egerstedt, M., Xiaoming, H.: Formation constrained multi-agent control. *IEEE Trans. Robot. Autom.* **17**(6), 947–951 (2001)
5. Kowalczyk, W.: Control algorithms for the formation of mobile robots, PhD thesis (in Polish) (2008)
6. Kowalczyk, W., Kozłowski, K., Tar, J.: Trajectory tracking for multiple unicycles in the environment with obstacles. In: 2010 IEEE 19th International Workshop on Robotics in Alpine-Adria-Danube Region (RAAD), pp. 451–456 (2010)
7. Lawton, J., Beard, R., Young, B.: A decentralized approach to formation maneuvers. *IEEE Trans. Robot. Autom.* **19**(6), 933–941 (2003)
8. Lewis, M., Tan, K.-H.: High precision formation control of mobile robots using virtual structures. *Auton. Robots* **4**(4), 387–403 (1997)
9. Mastellone, S., Stipanowic, D., Graunke, C., Intlekofer, K., Spong, M.: Formation control and collision avoidance for multi-agent nonholonomic systems: theory and experiments. *Int. J. Robot. Res.* **27**(1), 107–126 (2008)

10. Michalek, M., Kozłowski, K.: Vector-field-orientation feedback control method for a differentially driven vehicle. *IEEE Trans. Control Syst. Technol.* **18**(1), 45–65 (2010)
11. Monteiro, S., Bicho, E.: A dynamical systems approach to behavior-based formation control. In: *International Conference on Robotics Automation*, pp. 2606–2611 (2002)
12. Reynolds, C.: Flocks, herds, and schools: a distributed behavioral model. In: *Computer Graphics*, pp. 25–34 (1987)
13. Rimon, E., Koditschek, D.: The construction of analytic diffeomorphisms for exact robot navigation on star worlds. In: *1989 IEEE International Conference on Robotics and Automation*, 1989. Proceedings, vol. 1, pp. 21–26 (1989)
14. Rimon, E., Koditschek, D.: Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom.* **8**(5), 501–518 (1992)
15. Rodriguez-Seda, E., Tang, C., Spong, M., Stipanovic, D.: Trajectory tracking with collision avoidance for nonholonomic vehicles with acceleration constraints and limited sensing. *Int. J. Robot. Res.* **33**(12), 1569–1592 (2014)
16. Stipanovic, D., Hokayem, P., Spong, M., Siljak, D.: Avoidance control for multi-agent systems. *ASME J. Dyn. Syst. Meas. Control Spec. Issue Multi-Agent Syst.* **129**, 699–707 (2007)
17. Takahashi, H., Nishi, H., Ohnishi, K.: Autonomous decentralized control for formation of multiple mobile robots considering ability of robot. *IEEE Trans. Ind. Electron.* **51**(6), 1272–1279 (2004)
18. Tan, K.-H., Lewis, M.: Virtual structures for high-precision cooperative mobile robotic control. In: *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, vol. 1, pp. 132–139 (1996)
19. Tanner, H., Pappas, G., Kumar, V.: Leader-to-formation stability. *IEEE Trans. Robot. Autom.* **20**(3), 443–455 (2004)
20. Urakubo, T., Tsuchiya, K., Tsujita, K.: Motion control of a two-wheeled mobile robot. *ASME J. Dyn. Syst. Meas. Control Spec. Issue Multi-Agent Syst.* **15**(7), 711–728 (2001)

# Path Tracking Controllers for Fast Skidding Rover

Mohamed Krid, Ziad Zamzami and Faiz Benamar

**Abstract** This article presents a comparative analysis between two new path tracking controllers, a linear and non-linear one. First, dynamic model of the vehicle moving on a horizontal plane is developed and validated with the experimental platform Spido. Then, to reduce the complexity, the model is linearized by assuming that the side slip angles remain small and a linear lateral slippage model introducing the lateral contact stiffness is used. The kinematics of the vehicle is also linearized by assuming that the rover remains close a reference path and with a small relative heading angle. The linearized model permits the design of a Linear Quadratic Regulator LQR that minimizing the angle deviation and the lateral error. Then, nonlinear predictive controller is presented. This approach combines the kinematic model of the path-tracking with the dynamic model of the vehicle to deduce a non-linear Continuous Generalized Predictive Controller NCGPC controller. The predictive approach is dedicated to achieve an accurate path tracking and minimizing the angle deviation and the lateral error during prediction horizon time. Finally, we compare performance results between the NCGPC and the LQR controller. The simulation results show that both controllers have equivalent path tracking performance only in the straight line sections. However, predictive controller clearly outperform LQR controller when cornering. NCGPC controller has a higher rate of convergence to the reference trajectory with minimal lateral position error, thanks to the anticipation of future reference path changes.

---

M. Krid

College of Engineering, Industrial Engineering, King Saud University,  
Riyadh, Saudi Arabia  
e-mail: mkrid@ksu.edu.sa

Z. Zamzami · F. Benamar (✉)

Institut des Systèmes Intelligents et Robotiques ISIR,  
Sorbonne Universités, UPMC University, Paris 06, UMR 7222, 75005 Paris, France  
e-mail: amar@isir.upmc.fr

Z. Zamzami

e-mail: zamzami@isir.upmc.fr

© Springer International Publishing Switzerland 2016

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*, Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_2

**Keywords** Mobile robot · Path tracking · Nonlinear continuous time generalized predictive control · LQR controller · Lateral slippage · Lateral dynamics

## 1 Introduction

Increasing adoption of outdoor mobile robots in challenging field conditions demand increasing capabilities in operating speed and distance range. Therefore we are interested in the design and control of fast rovers which are able to move autonomously in natural environment at high velocity (up to 10 m/s) following a reference path referred by RTK-GPS points. Generally, vehicle dynamics are neglected when designing the path tracking controller [1, 2], however purely kinematic based approaches are clearly incongruous for solving the problem of motion control of fast rovers.

Many researches focused on path following of autonomous vehicle in presence of sliding have been developed [1, 3]. This problem has an important nonlinearity in the dynamic and kinematic models. Usually a linearized model is used to resolve the control problems [4] and the vehicle dynamics are neglected. These approaches significantly reduce the control performance (accuracy, stability, robustness...).

On the other hand, when we observe animals or humans during their movement we understand that forward vision field is vital to anticipate the future state. Most of control laws do not explicitly take account the future state of the system and the implication of the current input on the future output (PID, LQR...) [5, 6]. This induces the idea of predictive control which depends not only on the actual state of the system but also on its predicted state. This capacity of fusion between actual dynamics, future predicted state and eventually constraints makes this control to be one of the most powerful [7]. In general, Model Predictive Control (MPC) uses linear models that makes model prediction solving sufficiently efficient to be runned in real time.

In off-road mobile robotics context and especially high speed, wheel's slippage and robot dynamics can not be neglected. Basic mobile robot control laws based on the ideal rolling assumption, as developed in [8] or [9], cannot be applied confidently. In order to improve the path tracking in presence of sideslip phenomena without reducing the robot speed, a nonlinear predictive approach will be used here and a closed-form control law will be defined.

In this paper, the approach of Nonlinear Continuous-time Generalized Predictive Control (NCGPC) is used for developing a robot path tracking controller and will be compared to the classical LQR approach. The model prediction considers the lateral dynamics, the wheel slippage and the nonlinear kinematics form of the path tracking problem. This article is organized as follow. We begins with a description of a 3-dof vehicle dynamic model, using a linear slippage model. The lateral dynamics model is then validated using the real robot. Afterward, the LQR approach is used in order to design a new controller for path tracking achievement. Finally, a non-linear model based predictive controller is then designed and generalized to

our multi-inputs multi-outputs dynamic system. Controller performances are then compared and validated using a Co-simulation ADAMS/Simulink model and the Spido experimental platform.

## 2 Lateral Dynamics Model

This section describes the vehicle dynamic model which will be used for control design. The model will be also discussed through experimental validation carried out on the Spido platform. This experimental platform is an off-road vehicle with four steered and driven wheels and with 4 independent suspensions shown in Fig. 1. The vehicle weight is about 620 kg and the front and rear half wheel base are respectively 0.62 m and 0.58 m. It is equipped with a Real Time Kinematic GPS (RTK-GPS) which supplies a localization with respect to a reference station accurate within  $\pm 2$  cm.

We consider in this section the double track model, well-known as the handling model, illustrated in Fig. 2. The vehicle model has 3 dof: longitudinal, lateral and yaw motions [10] of the chassis.

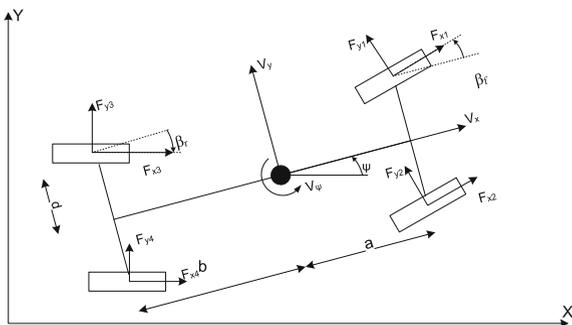
We denote by  $F_x$  and  $F_y$  the longitudinal and lateral tire forces,  $V_x$  and  $V_y$  the longitudinal and lateral velocities of the chassis center of mass,  $\dot{\psi}$  is its yaw rate,  $I_z$  is the yaw-inertia moment,  $M$  is the car mass,  $d$  is the half-width of the wheel-base,  $a$  and  $b$  are the distances from the center of gravity to the front and rear axles and,  $\beta_f$  and  $\beta_r$  are front and rear wheel steering angles.

Except for the gravity and aero dynamic forces, the main external forces applied on the vehicle are produced by the tires. The lateral forces acting on the front and rear tire are assumed to be a linear function of the side-slip angle [11], i.e.

**Fig. 1** The Spido experimental platform



**Fig. 2** Model of the four-wheeled mobile robot



$$F_{y(f,r)} = C_{(f,r)}\alpha_{(f,r)} \quad (1)$$

where  $C_f$  and  $C_r$  are, respectively, the tire cornering stiffness of the front and rear tires.

The side-slip angle  $\alpha_{f,r}$  is defined as the angle between the wheel velocity and the longitudinal axis of the wheel itself. Assuming that side-slip angles are quite small (less than  $10^\circ$  in practice), we can use a linear approximation, that is:

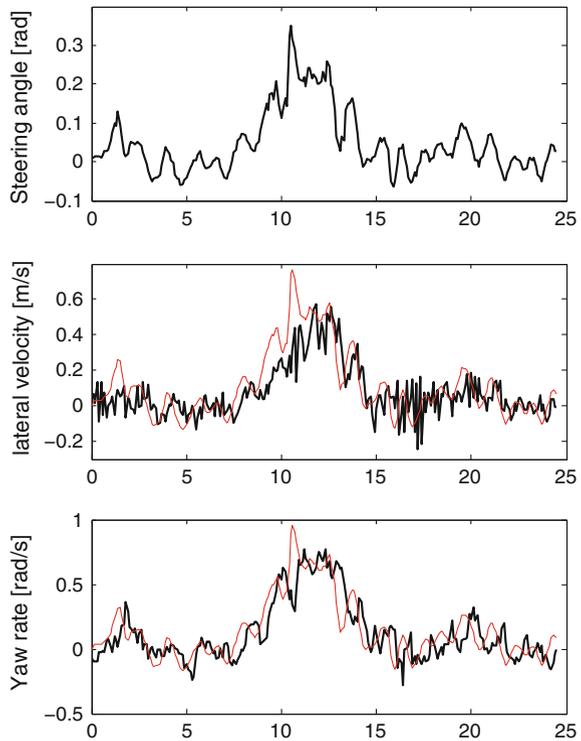
We focus in this paper on the lateral dynamics and the path tracking control. Henceforth, the longitudinal velocity  $V_x$  will be considered constant ( $\dot{V}_x = 0$ ). The dynamic model can be then given by the following equation of motion.

$$\begin{pmatrix} \dot{V}_y \\ \dot{\psi} \end{pmatrix} = A \begin{pmatrix} V_y \\ \psi \end{pmatrix} + B \begin{pmatrix} \beta_f \\ \beta_r \end{pmatrix} + d \begin{pmatrix} 0 \\ \Delta F_{xf} + \Delta F_{xr} \end{pmatrix} \quad (2)$$

where  $\Delta F_{xf}$ ,  $\Delta F_{xr}$  are front and rear traction force difference between right and left wheels,  $A$  and  $B$  are  $2 \times 2$  matrix of:

$$\begin{aligned} a_{11} &= 2 \frac{C_f + C_r}{M V_x} & a_{12} &= 2 \frac{a C_f - b C_r}{M V_x} - V_x \\ b_{11} &= \frac{F_{xf} - 2C_f}{M} & b_{12} &= \frac{F_{xr} - 2C_r}{M} \\ a_{21} &= 2 \frac{a C_f - b C_r}{V_x I_z} & a_{22} &= 2 \frac{a^2 C_f + b^2 C_r}{V_x I_z} \\ b_{21} &= \frac{a(F_{xf} - 2C_f)}{I_z} & b_{22} &= \frac{b(-F_{xr} + 2C_r)}{I_z} \\ \alpha_f &= \frac{V_y + a\dot{\psi}}{V_x} - \beta_f, & \alpha_r &= \frac{V_y - b\dot{\psi}}{V_x} - \beta_r \end{aligned} \quad (3)$$

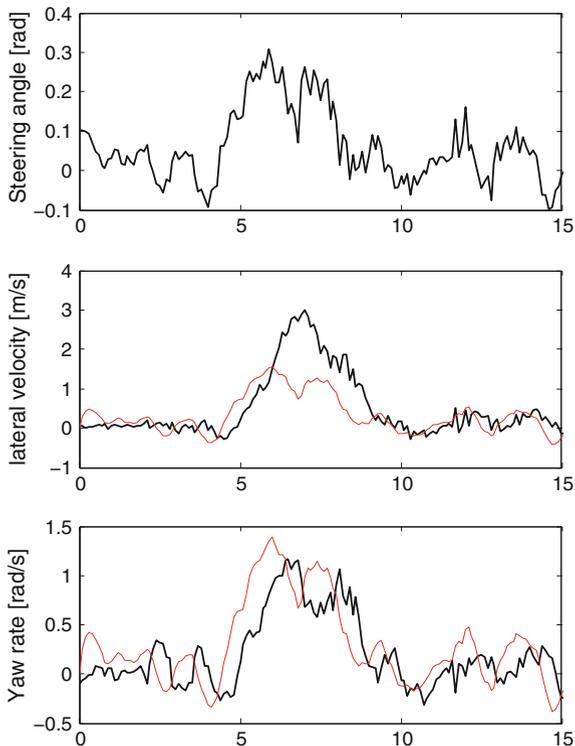
**Fig. 3** Predicted (*red*) and measured (*black*) lateral velocity and yaw rate at 4 m/s



The linear state model described in (2) can be validated using the Spido platform.

The dynamic model described in (2) is identified and validated in an open loop controller. We compare from the real robot and the mathematical model output in Figs. 3 and 4 at different speeds, respectively 4 m/s and 6 m/s. In this validation stage, we consider the same trajectory for the different velocities. We assume constant all of the parameter of the state model matrix  $A$  and  $B$  except the longitudinal velocity  $V_x$ . We emphasize two observations in these tests: (1) the error between the measured and estimated lateral velocity varies with the magnitude of the velocity, as shown in Figs. 2, 3 and 4 we observe a clear delay between the measured and estimated value. These observations were key to enhance the performance of the controller as described in the following section.

**Fig. 4** Predicted (*red*) and measured (*black*) lateral velocity and yaw rate at 6 m/s



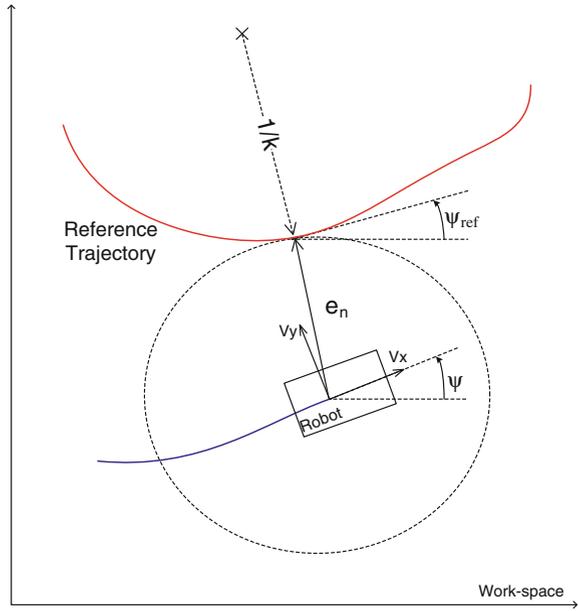
### 3 Linear Path Tracking Controller Design

#### 3.1 Robot-Path Kinematic Model

We will consider in this section the design of a linear controller for tracking a given prescribed path defined in the horizontal plane and at a constant forward velocity ( $\dot{V}_x = 0$ ).  $\psi$  and  $\psi_{ref}$  are the robot heading and the heading of the local tangent of the prescribed path with respect to a fixed arbitrary reference direction, as it can be illustrated in Fig. 5. The rate of change of  $\psi_{ref}$  with respect to the curvilinear distance measured along the path defines the path curvature  $k$ .  $e_\psi$  represents the heading error of the robot.  $e_n$  is the lateral deviation of the robot center of mass with respect to the prescribed path.

Sign conventions for both  $e_\psi$  and  $e_n$  are denoted in Fig. 5. With reference to the same figure and by assuming that the angular deviation is small, the following kinematical relationships are derived:

**Fig. 5** The reference and actual path



$$\begin{aligned}\dot{e}_n &= V_x \sin e_\psi + V_y \cos e_\psi \\ &= V_x e_\psi + V_y\end{aligned}\quad (4)$$

$$\begin{aligned}\dot{e}_\psi &= \dot{\psi} - \dot{\psi}_{ref} \\ &= \dot{\psi} - \frac{V_x \cos e_\psi - V_y \sin e_\psi}{1/k + e_n}\end{aligned}\quad (5)$$

In case of straight line tracking ( $k = 0$ ), this last equation is not singular and is equivalent to  $\dot{e}_\psi = \dot{\psi}$ , since  $\dot{\psi}_{ref} = 0$  for a straight line path. Assuming that  $e_n \ll 1/k$ , Eq. (5) can be linearized as follow:

$$\dot{e}_\psi = \dot{\psi} - V_x k + V_x k^2 e_n \quad (6)$$

Assuming a constant curvature, the derivative with respect to time of this latter equation gives

$$\ddot{e}_\psi = \ddot{\psi} + V_x^2 k^2 e_\psi + V_x k^2 V_y \quad (7)$$

### 3.2 Design Control

In this section, we define a path tracking control of the robot using linear state feedback. Combing the kinematic model of the path-tracking errors (4) and (7) with the dynamic model of the vehicle (2) results in a new state-space model:

$$\dot{X} = AX + BU$$

where  $X = [V_y \dot{\psi} e_\psi \dot{e}_\psi e_n]^T$  is the state vector and  $U = [\beta_f \beta_r \Delta F_{xf} \Delta F_{xr}]^T$  is the inputs vector,  $A$  is  $5 \times 5$  matrix and  $B$  is  $5 \times 4$  matrix. In this study, the main objective of the control design is to improve the steering behavior of the robot in the presence of lateral and yaw disturbances. Therefore, the controller should minimize the side-slip angle  $\beta$  of the robot center mass through minimizing lateral velocity  $V_y$  and its yaw rate (thereby decreasing the probability of lateral skidding). To achieve good tracking performance, the cost function must include a term that penalizes both the heading angle deviation  $e_\psi$  and the lateral offset  $e_n$ ,

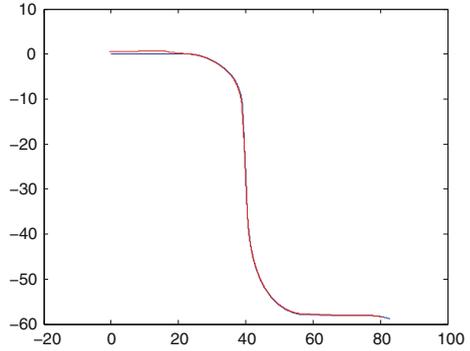
$$J = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t (X^T QX + U^T RU) dt \quad (8)$$

The second term on the right-hand penalizes the weighted-magnitude of the control inputs vector. This will prevent the use of too high control commands and the resultant control actions will be smooth.  $Q$  and  $R$  are the nominal state-weighting and the input-weighting matrices.

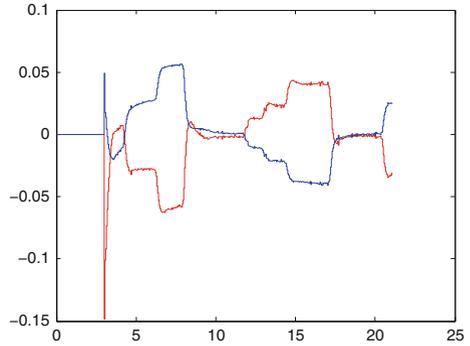
### 3.3 Tracking Performance of the Linear Controller

In this section, the LQR controller presented in the previous section has been tested using co-simulation between ADAMS for the mechanical model and Matlab for the control one. The ADAMS model consider a mechanical model with 16 DOF, including 4 independent double wishbone suspensions. The classical  $Z$  and  $O$  paths are used for testing the controller, they are plotted respectively on (Figs. 6 and 10). In the simulation, the LQR controller gains are calculated on-line, as they depend on the longitudinal velocity  $V_x$  and the reference path curvature  $k$ . The path curvature was filtered to remove the abrupt variations. During the first 3 s of simulation, the system is open-loop until reaching the reference longitudinal velocity. For the first path, we give the control inputs for the front and rear steering angles. The time evolution of lateral deviation and orientation errors are shown on Figs. 8 and 9. For the second path, we give on Fig. 11 the differential forces  $\Delta F_{xf}$   $\Delta F_{xr}$  for the front and the rear axles (Fig. 7).

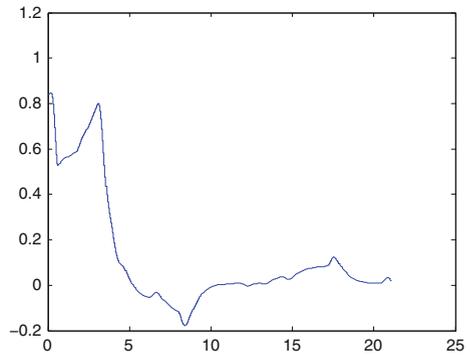
**Fig. 6** The reference (*blue*) and actual (*red*) paths for the 'Z' path



**Fig. 7** Steering angles: front (*red*) and rear (*blue*) for the 'Z' path (rad)

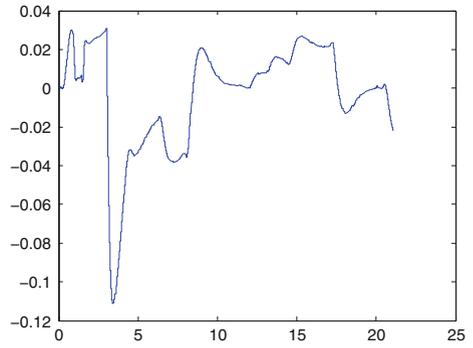


**Fig. 8** Lateral deviation for the 'Z' path (m)

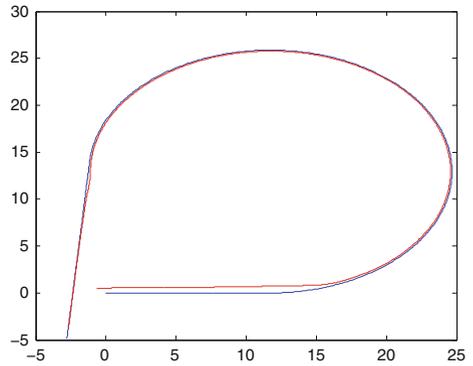


Results show that the proposed LQR control based on the dynamics model is able to track the reference path with quite good accuracy and quite small and smooth inputs. The new inputs controllers,  $\Delta F_{xf}$  and  $\Delta F_{xr}$ , seems to increase the rank of the controllability matrix what permits to reduce the steering angles.

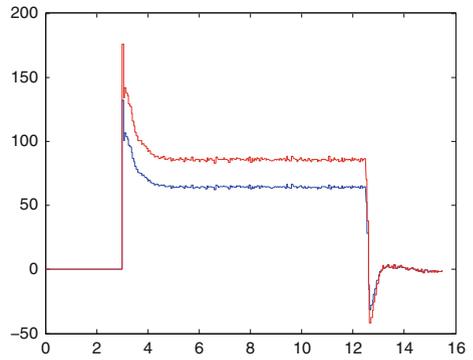
**Fig. 9** Yaw deviation for the 'Z' path (rad)



**Fig. 10** The reference (*blue*) and actual (*red*) paths for the 'O' path



**Fig. 11** The differential forces on front (*red*) and rear (*blue*) axle for 'O' path (N)



## 4 Nonlinear Predictive Approach

Using the dynamic model developed in the previous section, we will now define a predictive controller which have to minimize the tracking error along a receding time horizon. As with the linear approach, we assume a constant forward velocity ( $\dot{V}_x = 0$ ). We combine the dynamic model of the vehicle (2) and the non-linear kinematic model of path tracking composed by the projection of the actual global position of the vehicle ( $X, Y$ ) on the reference path. Using this model a nonlinear predictive control is computed, dedicated to achieve an accurate path tracking despite the nonlinear dynamical phenomena such as the bad grip conditions that leads to slipping.

Combing the dynamics model (2) and the kinematic one results in a new nonlinear state-space model:

$$\begin{aligned}\dot{V}_y &= a_{11}V_y + a_{12}V_\psi + b_{11}\beta_f + b_{12}\beta_r \\ \dot{V}_\psi &= a_{21}V_y + a_{22}V_\psi + b_{21}\beta_f + b_{22}\beta_r \\ \dot{\psi} &= V_\psi \\ \dot{X} &= V_x \cos \psi - V_y \sin \psi \\ \dot{Y} &= V_x \sin \psi + V_y \cos \psi\end{aligned}\tag{9}$$

We define the output  $y$  by:

$$y = \begin{pmatrix} \psi \\ X \\ Y \end{pmatrix}$$

Although a linear kinematic model is usually used [5] to simplify the controller form, in our case, we choose a nonlinear kinematic model for increased accuracy. The delay observed earlier when we compared the measured and estimated value of the state of the model (2) and the nonlinear form of the kinematic model motivate us to choose Nonlinear Continuous-time Generalized Predictive Control.

### 4.1 NCGPC Based Controller Design

To introduce the design of NCGPC controller, we can write the dynamic model (9) in the following form:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= (h_1(x), \dots, h_m(x))\end{aligned}$$

where, the state vector  $x \in \mathfrak{N}^n$ , the output  $y \in \mathfrak{N}^m$  and input  $u \in \mathfrak{N}^p$ . The NCGPC controller minimizes a quadratic cost criterion which is based on the difference

between the predicted state  $y$  and a reference signal  $w$ . We denote  $e_i(t)$  the error between the output  $h_i$  and the reference signal  $w_i(t)$  at the time  $t$  and  $w = (w_1(t) \dots w_m(t))$ .

$$e_i(t) = h_i(x(t)) - w_i(t) \quad (10)$$

The output prediction is based on the expansion in Taylor series. An approximation of the reference signal is done in the same way.

$$\hat{y}(t + \tau) = \sum_{k=0}^{\rho} y^{(k)}(t) \frac{\tau^k}{k!} + R(\tau^\rho) \quad (11)$$

We can define the cost function as following

$$J_i = \frac{1}{2} \int_0^{T_i} [\hat{e}_i(t + \tau)]^2 d\tau \quad (12)$$

where  $T_i$  is the prediction horizon time of the  $i$ th output  $h_i$  and  $\tau$  a given value belonging to the time interval  $[t, t + T_i]$ . We then deduce the global cost function  $J$ :

$$J = \sum_{i=1}^m J_i = \frac{1}{2} \sum_{i=1}^m \left( \int_0^{T_i} [\hat{e}_i(t + \tau)]^2 d\tau \right) \quad (13)$$

To derive the control law, we need to minimize the expression (13) of the criterion with respect to control  $u$ :

$$\frac{\partial J}{\partial u} = 0_{p \times 1} \quad (14)$$

The analytical expression of the control law  $u$  can be found in (14). The analytical expression of the control law is not detailed in this paper. To understand this issue, more information can be found in [12–14]. It was demonstrated that the stability of such NCGPC controller is guaranteed for system having a relative degree of each output less than or equal to four. That is particularly true for our system.

## 4.2 Simulation and Experimental Results

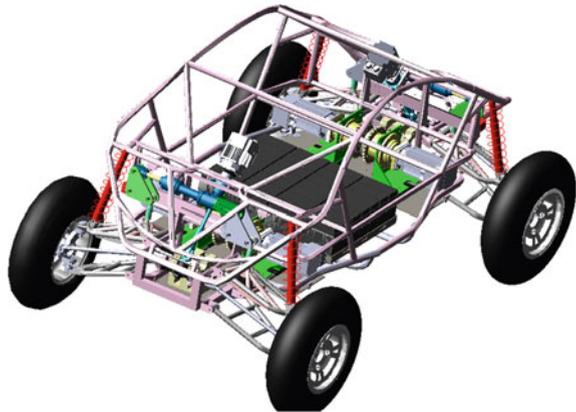
The NCGPC controller presented in the previous section has been tested using co-simulation between Adams for the plant model and Matlab/Simulink for the control one. The Adams model considers a multibody system with 16-dof, including 4 independent double wishbone suspensions. The wheel-ground contact model is the solid/solid native model composed by a series of non-linear spring-damper for

normal forces and a regularized Coulomb friction model for tangential forces. The slope at the origin of the latter will give contact stiffness  $C_f$  and  $C_r$  defined in Sect. 2.

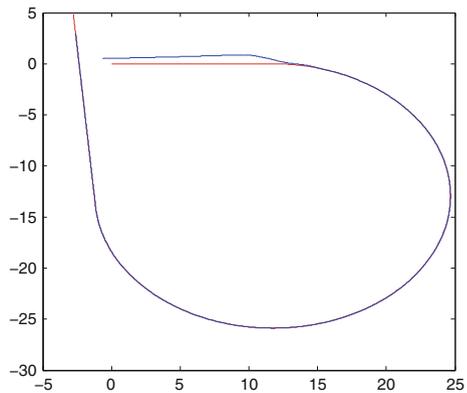
The classical  $O$ -shaped paths are used for testing the controller. Since the simulation should start preferably from equilibrium, during the first 3 s of simulation, the vehicle is open-loop driven till it reaches the reference longitudinal velocity.

Figures 13 and 14 illustrate controller performance with initial error at two velocity levels  $V_x = 4$  m/s and  $V_x = 6$  m/s. Each one compares the actual and the reference paths in the horizontal plane. We can appreciate the effect of vehicle velocity during the transition phase between the straight line and the curvilinear sections; the controller’s rate of convergence to the reference trajectory is inversely proportional to the robot’s velocity. We denote that the simulation started with an initial lateral deviation of 1.2 m and we use a time receding horizon of 0.6 s. Furthermore, we can deduce from the position and angular error at  $V_x = 4$  m/s in Figs. 15 and 16, respectively at  $V_x = 6$  m/s in Figs. 17 and 18, that the NCGPC controller provides equivalent performance. We notice that the controller stabilizes the initial position error during

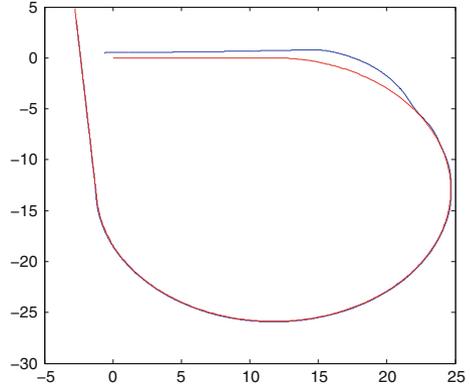
**Fig. 12** CAD view of the used 16-dof Adams model



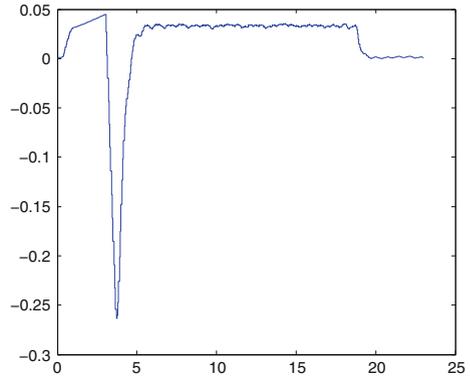
**Fig. 13** Reference (*red*) and actual (*blue*) robot position (m) at  $V_x = 4$  m/s



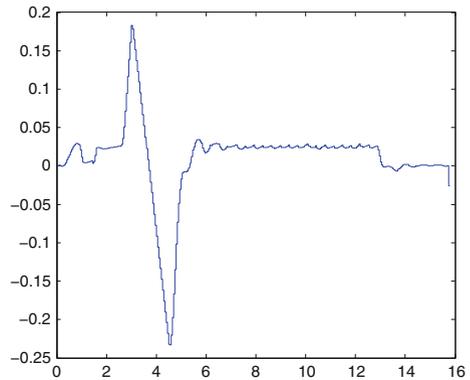
**Fig. 14** Reference (*red*) and actual (*blue*) robot position (m) at  $V_x = 6\text{ m/s}$



**Fig. 15** Angular deviation (rad) versus time (s) graph at  $V_x = 4\text{ m/s}$



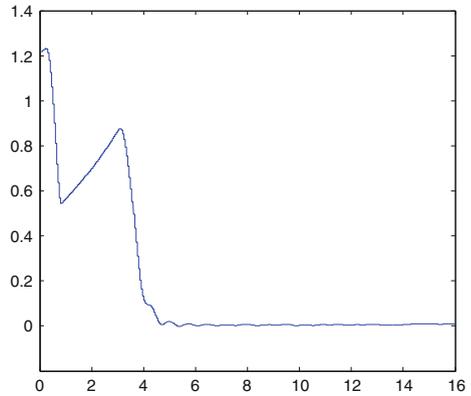
**Fig. 16** Angular deviation (rad) versus time (s) graph at  $V_x = 6\text{ m/s}$



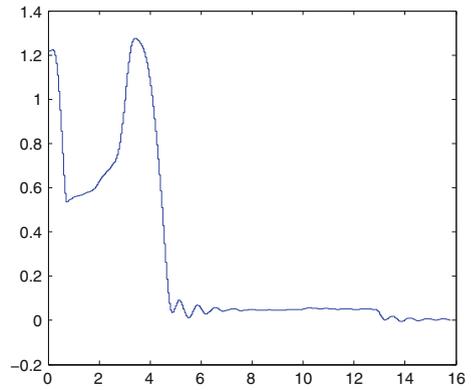
the first second and start anticipation the path curvature change at approximately 1 s (Fig. 12).

We compare performance results of the *NCGPC* controller developed in this paper to the *LQR*-based controller developed in Sect. 3. Figure 19 shows the reference path

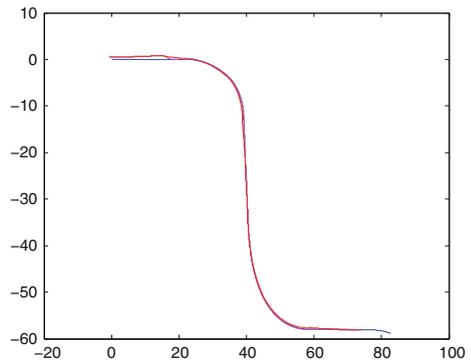
**Fig. 17** Lateral deviation (m) versus time (s) graph at  $V_x = 4$  m/s



**Fig. 18** Lateral deviation (m) versus time (s) graph at  $V_x = 6$  m/s

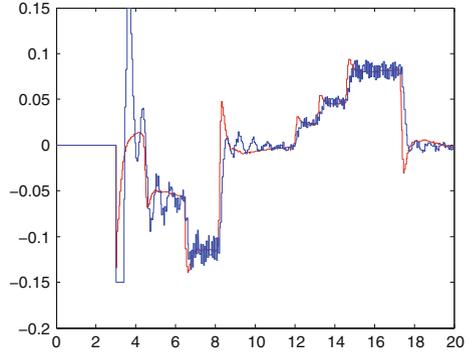


**Fig. 19** Comparing 2 different controllers: LQR (red) versus NCGPC (magenta) robot position

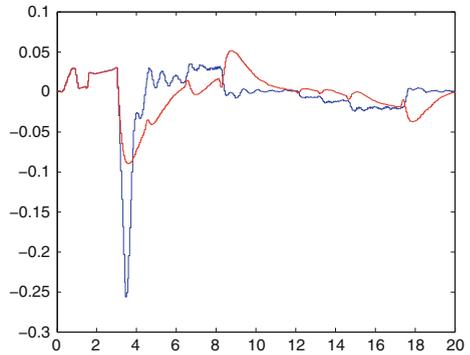


in blue and the performed paths in red and magenta colors using the LQR and NCGPC controllers respectively. We can deduce that both controllers have equivalent path tracking performance only in the straight line sections. However, predictive controller clearly outperform LQR controller when cornering as shown in Fig. 22. NCGPC

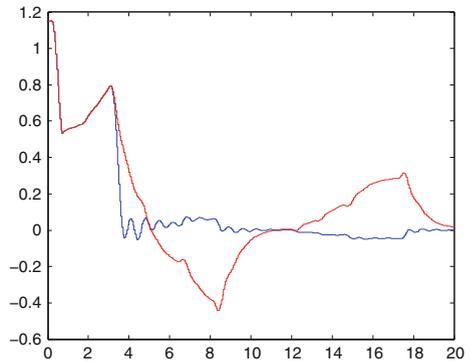
**Fig. 20** Steering angle (rad) versus time (s) graph: LQR (red) and NCGPC (blue)



**Fig. 21** Angular error (rad) versus time (s) graph: LQR (red) and NCGPC (blue)

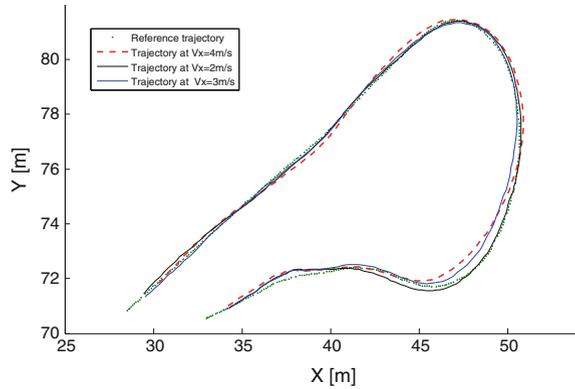


**Fig. 22** Lateral deviation (m) versus time (s) graph: LQR (red) and NCGPC (blue)



controller has a higher rate of convergence to the reference trajectory with minimal lateral position error, thanks to the anticipation of future reference path changes. The predictive controller can track the path with a small lateral error comparing to the pure feedback-based LQR controller. It is clearly shown that this latter can not control accurately the lateral dynamics and the sideslip angle (Figs. 20 and 21).

**Fig. 23** Experimental results: performed paths at different velocities using the predictive NCGPC controller



We observe that there is no difference between both controllers, during the straight line phase and at the end of the path when the robot reduces its velocity, this is mainly due to the low presence of the sideslip phenomena in these operating conditions (low steering angle and velocity).

The NCGPC controller presented has been also tested using the Spido experimental platform. The classical  $O$  paths (Fig. 23) was used for testing the controller at different forward velocities,  $V_x = 2, 3, 4$  m/s. The path to be followed is recorded by a preliminary run achieved in manual driving. The testing conditions induced perturbations due to the terrain irregularities and friction conditions changes. The ground is composed by concrete area and wet nonuniform grass area where the vehicle is inevitably prone to slide. We must notice that our controller model is based on constant front and rear cornering stiffness. But thanks to controller robustness, the control law ensure a trajectory tracking with low tracking errors. The errors increase with velocities because of lateral slippage are increasing and probably roll angle as well which leads to stability limit and to the limitation of the model. We must note that the lateral acceleration reaches  $0.7g$  which is the maximal value of the availability of the friction linear model and dynamic friction coefficient of  $0.7$ .

## 5 Conclusion

In this work, we present two controllers for mobile robot path tracking with high speed and dynamics. The mobile robot is a four wheel steered rover with suspension and capable to perform high speed cornering on natural ground. By combining the dynamic model of the vehicle, a slip based tyre model and a classical kinematic model of path tracking model, a linear and nonlinear model are deduced and validated using the real robot. Based on minimization of tracking error cost function along a receding time horizon, then nonlinear continuous-time generalized predictive control (NCGPC) has been designed and its performance is compared to that of the LQR

controller. Simulation and experimental results show a clear advantage of predictive controller when it comes to efficient and stable path tracking. This superiority is more pronounced for high speed motion and high curvature cornering, simply because such controller predict the robot behaviour over a receding time horizon and choose the optimal control input which drive the plant along the prescribed trajectory. An important advantage of the proposed solution is its easy integration because this controller supply an analytical expression of the control law  $u$  that reduce the real time calculation which simplify the implementation of the controller algorithm. This work will be continued to extend the NCGPC approach to redundant steering system, that includes front and rear steering and differential torque due to traction forces. Also, in order to increase the controller performance, another challenge for this study concerns the definition of the optimal prediction horizon time, probably as function of path curvature.

**Acknowledgments** This work was supported by the French ANR project EquipEx RobotEx (ANR-10- EQPX-44).

## References

1. Fang, H., Lenain, R., Thuilot, B., Martinet, P.: Trajectory tracking control of farm vehicles in presence of sliding. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005), pp. 58–63 (2005)
2. Tan, S.L., Gu, J.: Investigation of trajectory tracking control algorithms for autonomous mobile platforms: theory and simulation. In: 2005 IEEE International Conference Mechatronics and Automation, vol. 2, pp. 934–939 (2005)
3. Bouton, N., Lenain, R., Thuilot, B., Fauroux, J.-C.: A rollover indicator based on the prediction of the load transfer in presence of sliding: application to an all terrain vehicle. In: 2007 IEEE International Conference on Robotics and Automation, pp. 1158–163 (2007)
4. Yang, J.-M., Kim J.-H.: Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Trans. Robot. Autom.* **15**(3), 578–587 (1999)
5. Krid, M., Ben Amar, F.: A dynamic based path tracking controller for a fast rover with independent steering and drive. In: International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Sept 2011
6. Padhy, P.K., Majhi, S.: On-line mobile robot path tracking using PI controller. In: India Conference, 2006 Annual IEEE, pp. 1–4, Sept 2006
7. Rossiter, J.A.: Model Based Predictive Control: A Practical Approach, Control Series. CRC PRESS (2003)
8. Cariou, C., Lenain, R., Thuilot, B., Martinet, P.: Adaptive control of four-wheel-steering off-road mobile robots: application to path tracking and heading control in presence of sliding. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, pp. 1759–1764, Sept 2008
9. Micaelli, A., Samson, C.: Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Research report rr-2097, INRIA (1993)
10. Liaw, C.C., Chung, W.C., Song, C.C., Hsieh, C.-S.: A feedback linearization design for the control of vehicle's lateral dynamics. In: SICE-ICASE, 2006. International Joint Conference, pp. 4937–4942, Oct 2006
11. Jung, S., Hsia, T.C.: Explicit lateral force control of an autonomous mobile robot with slip. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005), pp. 388–393, Aug 2005

12. Chen, W.H.: Predictive control of general nonlinear systems using approximation. *IEE Proc. Control Theory Appl.* **151**(2), 137–144 (2004)
13. Gawthrop, P.J., Ballance, D.J., Chen, W.H.: Optimal control of nonlinear systems: a predictive control approach. *Automatica*, **39**(4), 633–641 (2003). Final version as accepted by *Automatica* supplied by the author
14. Dabo, M., Chafouk, H., Langlois, N.: Unconstrained NCGPC with a guaranteed closed-loop stability: case of nonlinear siso systems with the relative degree greater than four. In: *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009*, pp. 1980–1985, Dec 2009

**Part II**  
**Intelligent Control Systems**  
**and Optimization**

# Gravitational Search Algorithm-Based Evolving Fuzzy Models of a Nonlinear Process

Radu-Emil Precup, Emil-Ioan Voisan, Emil M. Petriu,  
Mircea-Bogdan Radac and Lucian-Ovidiu Fedorovici

**Abstract** Implementation issues related to evolving Takagi-Sugeno-Kang (TSK) fuzzy models of a nonlinear process are offered. The nonlinear process is the pendulum dynamics in the framework of the representative pendulum-crane systems, where the pendulum angle is the output variable of the TSK fuzzy models. An online identification algorithm (OIA) is given, which continuously evolves the rule bases and the parameters of the TSK fuzzy models, adds new rules with more summarization power and modifies the existing rules and parameters. The OIA includes an input selection algorithm and a Gravitational Search Algorithm that updates the parameters in the rule consequents. The evolving TSK fuzzy models are validated by experiments conducted on pendulum-crane laboratory equipment.

**Keywords** Evolving Takagi-Sugeno-Kang fuzzy models · Gravitational search algorithm · Implementation issues · Pendulum Dynamics

---

R.-E. Precup (✉) · E.-I. Voisan · M.-B. Radac · L.-O. Fedorovici  
Politehnica University of Timisoara, Timisoara, Romania  
e-mail: radu.precup@upt.ro; radu.precup@aut.upt.ro

E.-I. Voisan  
e-mail: emil.voisan@upt.ro

M.-B. Radac  
e-mail: mircea.radac@upt.ro

L.-O. Fedorovici  
e-mail: lucian.fedorovici@aut.upt.ro

E.M. Petriu  
University of Ottawa, Ottawa, Canada  
e-mail: petriu@uottawa.ca

© Springer International Publishing Switzerland 2016

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*, Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_3

## 1 Introduction

The evolving Takagi-Sugeno-Kang (TSK) fuzzy models are characterized by the continuous online learning for rule base learning according to [1–4]. An online identification algorithm (OIA) continuously evolves the rule bases and the parameters of the TSK fuzzy models, and the models are built online by the adding mechanism (adding new or removing old local models). As shown in [5], the OIAs dedicated to evolving TSK fuzzy models are divided in three categories. The OIA proposed as follows belongs to the second category, namely the incremental algorithms, which implement only adding mechanisms. Some representative incremental algorithms are RAN, SONFIN, SCFNN, NeuroFAST, DENFIS, eTS, FLEXFIS and PANFIS.

Using the recent results related to evolving TSK fuzzy models reported in [6, 7], a new OIA will be suggested in the sequel. This OIA is inspired from [8], and it computes rule bases and parameters that continuously evolve by adding new rules with more summarization power, the existing rules and parameters are modified in terms of using the potentials of new data points. The new OIA is a modified version of that proposed in [9], which includes an input selection, and it is characterized by inserting a Gravitational Search Algorithm (GSA) [10, 11], that updates the parameters in the rule consequents.

The proposed OIA is advantageous with respect to other OIAs by the fact that the GSA replaces the recursive least squares algorithm, so there is no need to compute the covariance matrices. Therefore, the computational effort is reduced.

The presentation is dedicated to the fuzzy modeling of the pendulum dynamics in the framework of pendulum-crane systems. Some details concerning the implementation of evolving TSK fuzzy models of this process are given in the next sections. Other fuzzy models dedicated to this nonlinear process have been recently discussed in [12–17].

The following topics are treated as follows: the GSA-based OIA is presented in the next section. The case study that leads to new TSK fuzzy models for the pendulum dynamics and implementation issues are treated in Sect. 3. The conclusions are pointed out in Sect. 4.

## 2 Gravitational Search Algorithm-Based Online Identification Algorithm

The OIA is formulated using the details on the algorithms given in [7–9] and the GSA described in [10, 11]. The OIA consists of the following steps:

*Step 1* The rule base structure is initialized by initializing the parameters in the rule antecedents. The initialization means that the initial TSK fuzzy model has a single rule, i.e.,  $n_R = 1$ , where  $n_R$  is the number of rules. The subtractive clustering [18] is implemented to compute the parameters of the TSK fuzzy models using the first data point  $\mathbf{p}_1$ , where the data point  $\mathbf{p}$  at the discrete time step  $k$  is

$$\mathbf{p}_k = [p_k^1 \ p_k^2 \ \dots \ p_k^{n+1}]^T, \quad (1)$$

where  $T$  indicates the matrix transposition, the data point in the input-output data space  $\mathbf{R}^{n+1}$  is

$$\mathbf{p} = [\mathbf{z}^T \ y]^T = [z_1 \ z_2 \ \dots \ z_n \ y]^T = [p^1 \ p^2 \ \dots \ p^n \ p^{n+1}]^T \in \mathbf{R}^{n+1}, \quad (2)$$

the rule base of the affine-type TSK fuzzy models that are identified is

$$\begin{aligned} \text{Rule } i : & \text{ IF } z_1 \text{ IS } LT_{i1} \text{ AND } \dots \text{ AND } z_n \text{ IS } LT_{in} \\ & \text{ THEN } y_i = a_{i0} + a_{i1}z_1 + \dots + a_{in}z_n, \quad i = 1 \dots n_R, \end{aligned} \quad (3)$$

where  $z_j, j = 1 \dots n$ , are the input variables,  $n$  is the number of input variables,  $LT_{ij}, i = 1 \dots n_R, j = 1 \dots n$ , are the input linguistic terms,  $y_i$  is the output of the local affine model in the rule consequent of the rule  $i, i = 1 \dots n_R$ , and  $a_{il}, i = 1 \dots n_R, l = 0 \dots n$ , are the parameters in the rule consequents.

The algebraic product t-norm to model the AND operator and the weighted average defuzzification method in the TSK fuzzy model structure lead to the output  $y$  of the TSK fuzzy model

$$y = \left[ \sum_{i=1}^{n_R} \tau_i y_i \right] / \left[ \sum_{i=1}^{n_R} \tau_i \right] = \sum_{i=1}^{n_R} \lambda_i y_i = \sum_{i=1}^{n_R} \lambda_i [1 \ \mathbf{z}^T]^T \boldsymbol{\pi}_i, \quad (4)$$

where the firing degree and the normalized degree of the rule  $i$  are  $\tau_i(\mathbf{z})$  and  $\lambda_i(\mathbf{z})$ , respectively:

$$\begin{aligned} \tau_i(\mathbf{z}) &= \text{AND}(\mu_{i1}(z_1), \mu_{i2}(z_2), \dots, \mu_{in}(z_n)) = \mu_{i1}(z_1) \cdot \mu_{i2}(z_2) \cdot \dots \cdot \mu_{in}(z_n), \quad i = 1 \dots n_R, \\ \lambda_i(\mathbf{z}) &= \tau_i(\mathbf{z}) / \left[ \sum_{i=1}^{n_R} \tau_i(\mathbf{z}) \right], \quad i = 1 \dots n_R, \end{aligned} \quad (5)$$

and the vector  $\boldsymbol{\pi}_i, i = 1 \dots n_R$ , in (4) is the parameter vector of the rule  $i$

$$\boldsymbol{\pi}_i = [a_{i0} \ a_{i1} \ a_{i2} \ \dots \ a_{in}]^T, \quad i = 1 \dots n_R. \quad (6)$$

The parameters are initialized as a part of the parameters specific to the OIAs given in [7–9]

$$\begin{aligned} \hat{\boldsymbol{\theta}}_1 &= [(\boldsymbol{\pi}_1^T)_1 \ (\boldsymbol{\pi}_2^T)_1 \ \dots \ (\boldsymbol{\pi}_{n_R}^T)_1]^T = [0 \ 0 \ \dots \ 0]^T, \quad r_s = 0.4, \quad (7) \\ k &= 1, \quad n_R = 1, \quad \mathbf{z}_1^* = \mathbf{z}_k, \quad P_1(\mathbf{p}_1^*) = 1, \end{aligned}$$

and the parameters of the GSA [10, 11], related to the generation of the initial population of agents, namely the  $n_R(n+1)$ -dimensional search space of the parameters in the rule consequents, the number of agents  $N$ , and initialize randomly the agents' velocity vector  $\mathbf{V}_{i,0} \in \mathbf{R}^{n_R(n+1)}$  of  $i$ th agent,  $i = 1 \dots N$ .  $\hat{\boldsymbol{\theta}}_k$  in (7) is an estimation of

the parameter vector in the rule consequents at the discrete time step  $k$ ,  $r_s, r_s > 0$ , is the spread of all Gaussian input m.f.s  $\mu_{ij}, i = 1 \dots n_R, j = 1 \dots n$ , of the fuzzy sets of the input linguistic terms  $LT_{ij}$

$$\mu_{ij}(z_j) = \exp[-(4/r_s^2)(z_j - z_{ij}^*)^2], i = 1 \dots n_R, j = 1 \dots n, \quad (8)$$

and  $z_{ij}^*, i = 1 \dots n_R, j = 1 \dots n$ , are the centers of these m.f.s.  $\mathbf{p}_1^*$  in (8) is the first cluster centre,  $\mathbf{z}_1^*$  is the centre of the rule 1 and also a projection of  $\mathbf{p}_1^*$  on the axis  $z$  defined in (2).  $P_1(\mathbf{p}_1^*)$  in (7) is the potential of  $\mathbf{p}_1^*$ .

The input selection algorithm proposed in [7] is next applied in order to select the important input variables from all possible input variables. This algorithm ranks the inputs according to their importance factors and is described in [9].

*Step 2* At the next time step,  $k$  is set to  $k = k + 1$ , and the next data sample  $\mathbf{p}_k$  is read.

*Step 3* The potential of each new data sample is calculated as

$$P_k(\mathbf{p}_k) = (k-1)/[(k-1)(\vartheta_k + 1) + \sigma_k - 2\nu_k], \vartheta_k = \sum_{j=1}^{n+1} (p_k^j)^2, \quad (9)$$

$$\sigma_k = \sum_{j=1}^{n+1} \sum_{l=1}^{k-1} (p_l^j)^2, \nu_k = \sum_{j=1}^{n+1} (p_k^j \sum_{l=1}^{k-1} p_l^j).$$

*Step 4* The potentials of the centers of existing rules (clusters) are recursively updated in terms of

$$P_k(\mathbf{p}_l^*) = (k-1)P_{k-1}(\mathbf{p}_l^*) / \left[ k-2 + P_{k-1}(\mathbf{p}_l^*) + P_{k-1}(\mathbf{p}_l^*) \sum_{j=1}^{n+1} (d_{k(k-1)}^j)^2 \right], \quad (10)$$

where  $P_k(\mathbf{p}_l^*)$  is the potential at the discrete time step  $k$  of the cluster centre, which is a prototype of the rule  $l$ .

*Step 5* The possible modification or upgrade of the rule base structure is carried out using the potential of the new data compared to the potential of existing rules' centers. The rule base structure is modified if certain conditions specified in [7, 8] are fulfilled.

*Step 6* The parameters in the rule consequents are updated using the velocity and position update equations specific to the GSA

$$\begin{aligned} \mathbf{V}_{i,k} &= \rho_i \mathbf{V}_{i,k-1} + \mathbf{A}_{i,k-1}, \\ \hat{\boldsymbol{\theta}}_{i,k} &= \hat{\boldsymbol{\theta}}_{i,k-1} + \mathbf{V}_{i,k}, k = 2 \dots D, i = 1 \dots N, \end{aligned} \quad (11)$$

where  $\rho_i, 0 \leq \rho_i \leq 1$ , is a uniform random variable and  $\mathbf{A}_{i,k-1} \in \mathbf{R}^{n_R(n+1)}$  is the acceleration vector of  $i$ th agent. The fitness function used in the GSA is

$$f_k = y_k - \boldsymbol{\Psi}_{k-1}^T \hat{\boldsymbol{\theta}}_{k-1}, k = 2 \dots D. \quad (12)$$

The parameter vector  $\hat{\theta}_k$  is obtained as

$$\hat{\theta}_k = \min_{i=1 \dots N} \hat{\theta}_{i,k}. \quad (13)$$

The output of the TSK fuzzy model given in (4) is expressed in the vector form

$$\begin{aligned} y &= \Psi^T \theta, \theta = [\pi_1^T \quad \pi_2^T \quad \dots \quad \pi_{n_R}^T]^T, \\ \Psi^T &= [\lambda_1[1 \quad \mathbf{z}^T] \quad \lambda_2[1 \quad \mathbf{z}^T] \quad \dots \quad \lambda_{n_R}[1 \quad \mathbf{z}^T]]. \end{aligned} \quad (14)$$

*Step 7* Using (14), the output of the evolving TSK fuzzy model at the next discrete time step  $k + 1$  is predicted

$$\hat{y}_{k+1} = \Psi_k^T \hat{\theta}_k. \quad (15)$$

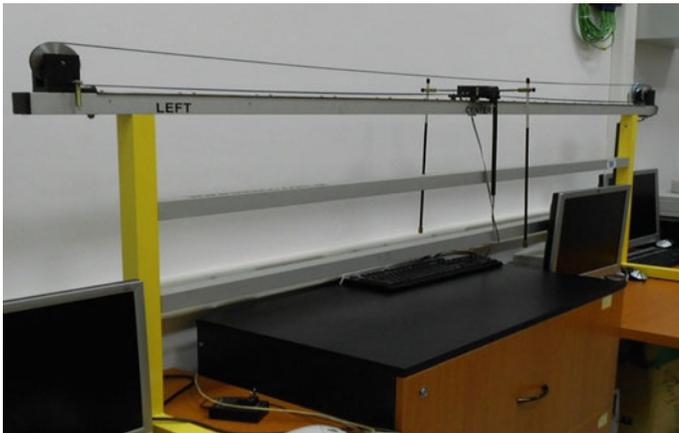
The algorithm continues with the step 2 until all data points from the set of input-output data are used.

$$\{\mathbf{p}_k | k = 1 \dots D\} \quad (16)$$

The step 1 of the OIA is conducted offline. The steps 2–7 are conducted online.

### 3 Case Study and Experimental Results

The laboratory setup built around the pendulum-cart system [19] has been used to exemplify and validate the OIA and the evolving TSK fuzzy models. The laboratory setup is illustrated in Fig. 1.



**Fig. 1** Laboratory setup in the Intelligent Control Systems Laboratory of the Politehnica University of Timisoara

The expression of the state-space model of the process in the pendulum-cart system is

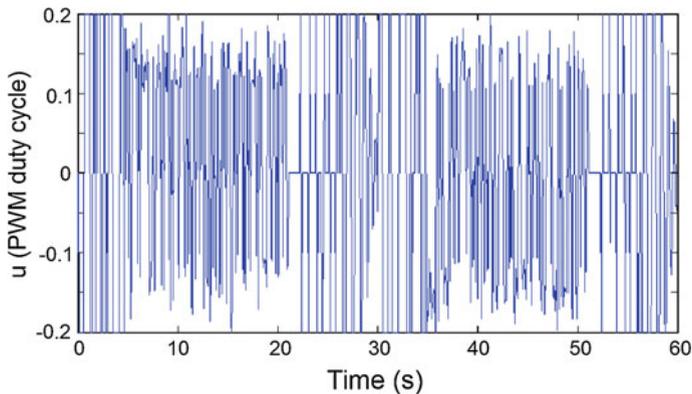
$$\begin{aligned}
 \dot{x}_1 &= x_3, \\
 \dot{x}_2 &= x_4, \\
 \dot{x}_3 &= \left\{ \frac{J_p}{(m_c + m_p)l_d} \left[ \frac{p_1 u}{(m_c + m_p)l_d} - x_4^2 \sin x_2 - \frac{(f_c - p_2)x_3}{(m_c + m_p)l_d} \right] + [g \sin x_2 \right. \\
 &\quad \left. - \frac{f_p x_4}{(m_c + m_p)l_d} \right] \cos x_2 \right\} / \left[ \frac{J_p}{(m_c + m_p)l_d^2} - \cos^2 x_2 \right], \\
 \dot{x}_4 &= \left\{ \left[ \frac{p_1 u}{(m_c + m_p)l_d} - x_4^2 \sin x_2 - \frac{(f_c - p_2)x_3}{(m_c + m_p)l_d} \right] \cos x_2 + \frac{1}{l} [g \sin x_2 \right. \\
 &\quad \left. - \frac{f_p x_4}{(m_c + m_p)l_d} \right] \right\} / \left[ \frac{J_p}{(m_c + m_p)l_d^2} - \cos^2 x_2 \right], \\
 y &= x_2,
 \end{aligned} \tag{17}$$

where the variables are:  $x_1$ —the cart position (the distance between the cart and the centre of the rail),  $x_2$ —the angle between the upward vertical and the ray pointing at the centre of mass cart,  $x_3$ —the cart velocity,  $x_4$ —the pendulum angular velocity,  $u$ —the control signal represented by a constrained PWM voltage signal,  $|u| \leq u_{\max} > 0$ ,  $m_c$ —the equivalent mass of the cart,  $m_p$ —the mass of the pole and load, and  $l_d$ —the distance from the axis of rotation to the centre of mass. The parameters in (17) are:  $J_p$ —the moment of inertia of the pendulum-cart system with respect to the axis of rotation,  $p_1$ —the ratio between the control force and the control signal,  $p_2$ —the ratio between the control force and  $x_3$ ,  $f_c$ —the dynamic cart coefficient, and  $f_p$ —the rotational friction coefficient. The parameter values of the laboratory setup are [7, 9, 19]

$$\begin{aligned}
 u_{\max} &= 0.5, \quad m_c = 0.76 \text{ kg}, \quad m_p = 0.052 \text{ kg}, \quad l_d = 0.011 \text{ m}, \quad J_p = 0.00292 \text{ kg} \cdot \text{m}^2, \\
 p_1 &= 9.4 \text{ N}, \quad p_2 = -0.548 \text{ N s/m}, \quad f_c = 0.5 \text{ N s/m}, \quad f_p = 6.65 \cdot 10^{-5} \text{ N m s/rad}.
 \end{aligned} \tag{18}$$

The OIA presented in the previous sections has been applied in order to obtain evolving TSK fuzzy models of the process that can be characterized by the nonlinear crisp model given in (17). The OIA has been implemented starting with the eFS Lab code given in [20, 21] and adding the functionalities taken from [7–11].

The sampling period has been set to 0.01 s. The control signal  $u$  has been generated as two weighted sums of pseudo-random binary signals (Fig. 2) to cover different ranges of magnitudes and frequencies. This process input has been applied to the laboratory setup to generate the input-output data points  $(\mathbf{z}_k, y_k)$ ,  $k = 1 \dots D$ , and a total number of 6000 data points has been used in the tests. The data points are separated in training data and validation data. The first  $D = 2500$  data points (the time domain from 0 to 25 s) in Fig. 2 belong to the validation data, and the rest of  $D = 3500$  data points (the time domain from 25 to 60 s) in Fig. 2 belong to the



**Fig. 2** Control signal versus time, corresponding to training data (0–25 s) and testing data (25–60 s)

testing (validation) data. The process output  $y$  is not presented in Fig. 2, but it will be illustrated as follows.

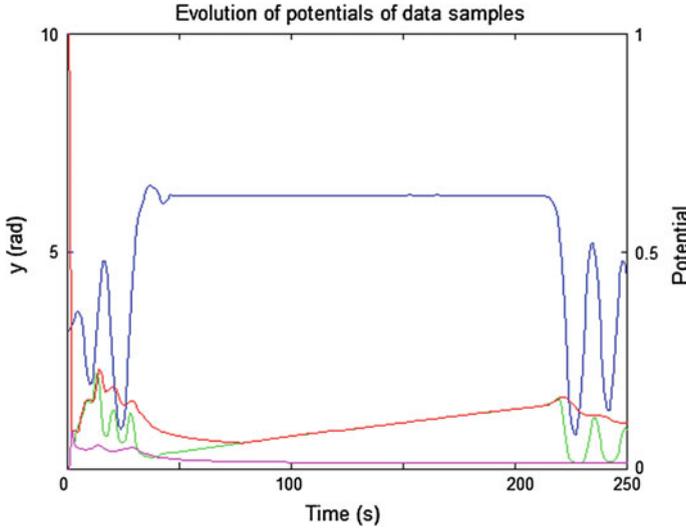
The input selection algorithm, which belongs to the step 1 of the OIA has been applied for three values of the importance threshold, as in [9],  $\lambda = 0.4, 0.3$  and  $0.2$ , and one value of the significance threshold,  $\tau = 0.5$ . This has led to three final TSK fuzzy models with the following inputs: the TSK fuzzy model 1, with the input  $u_k$ , the TSK fuzzy model 2 with the inputs  $u_k$  and  $y_{k-1}$ , and the TSK fuzzy model 3 with the inputs  $u_k, y_{k-1}$  and  $y_{k-2}$ . The output of all these three TSK fuzzy models is  $y_k$ .

The number of inputs of the TSK fuzzy models is variable during the iterations of the OIA, and that is the reason why the considered fuzzy models are the final ones. The inputs of the three TSK fuzzy models have been obtained from delayed system inputs and/or outputs, which have been extracted from the training and validation data sets. The values of the parameters of the GSA included in the step 6 of the GSA have been set to: number of agents  $N = 100$ , zero initial velocity vectors, exponential type depreciation law of the gravitational constant with the initial gravitational constant  $G_0 = 5$ .

The final results are similar to those obtained for the OIA given in [9], but with the recursive least squares algorithm employed in the step 6. Therefore, the TSK fuzzy model 1 has evolved to  $n_R = 2$  rules, the TSK fuzzy model 2 has evolved to  $n_R = 7$  rules, and the TSK fuzzy model 3 has evolved to  $n_R = 9$  rules.

The OIA and the TSK fuzzy model performance have been compared with other OIAs that result in evolving TSK fuzzy models, ANFIS [22], DENFIS [23] and FLEXFIS [24]. The comparison of all fuzzy models has been carried out using the root mean square error (RMSE) as performance index:

$$RMSE = \sqrt{\frac{1}{D} \sum_{k=1}^D (y_k - x_{2,k})^2}, \quad (19)$$



**Fig. 3** Evolution of output (*blue*) and potential of data points of TSK fuzzy model 3 on training data

the same inputs, numbers and shapes of m.f.s as those of the three TSK fuzzy models, and the numbers of rules  $n_R$  have been set such that to be very close. The variable  $y_k$  in (19) is the output (the pendulum angle) of the TSK fuzzy models and  $x_{2,k}$  is the output (the pendulum angle) of the laboratory setup at the discrete time moment  $k$ . The RMSE has been computed and measured for the training data and for the testing (validation) data as well.

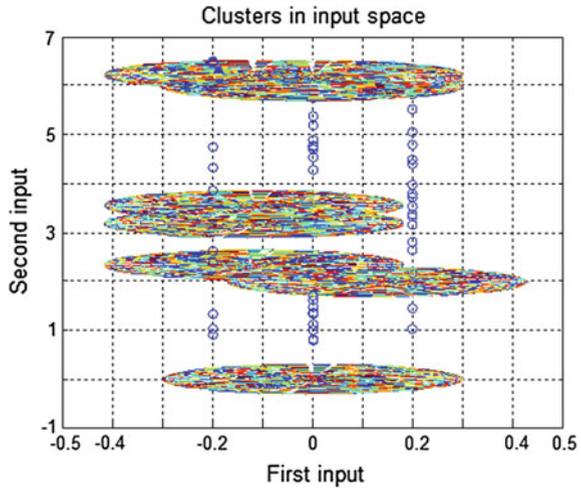
Some of the results for the TSK fuzzy model 3 on the training data are presented in Fig. 3 as clusters in the input space and in Fig. 4 as clusters in the input space. The time responses of the system output versus time of the TSK fuzzy model 3 and of the real-world process on the validation data are shown in Fig. 5.

The RMSE can be used as an objective function in appropriately defined optimization problems solved by several classical and nature-inspired optimization algorithms [24–33]. These optimization problems can be inserted in the OIA, and they must be accompanied by real-world constraints.

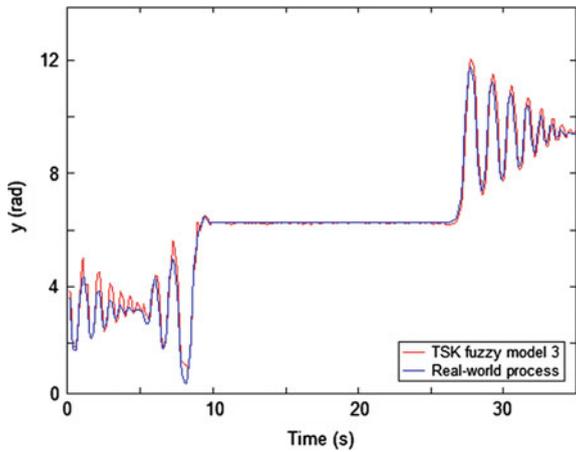
The comparison of results reads to the conclusion that the best performance on the validation (testing) data is exhibited by the TSK fuzzy model 3 obtained by the OIA presented in the previous section. In addition, the performance is very close to that achieved by the TSK fuzzy model obtained by the OIA proposed in [9].

As shown in [9], the evolving TSK fuzzy models obtained by the OIA outperform the evolving fuzzy models obtained by ANFIS, DENFIS and FLEXFIS. However, the performance depends on the parameters of both the input selection algorithm and the GSA.

**Fig. 4** Clusters in input space of TSK fuzzy model 3 on training data



**Fig. 5** Pendulum angle versus time of TSK fuzzy model 3 and real-world process for validation data



## 4 Conclusions

An OIA for evolving TSK fuzzy models has been proposed. The new features of this OIA are the inclusion of an input selection algorithm and of a nature-inspired optimization algorithm represented by the GSA. This offers not only a more systematic approach but also the alleviation of the computational effort. But the random parameters specific to the GSA and the parameters of the input selection algorithm affect the results and the models are sensitive with respect to the choice of these parameters.

The real-time experimental results related to the fuzzy modeling of the pendulum dynamics in pendulum-crane systems validate the OIA and the evolving TSK fuzzy models as well. Future research will be focused on considering other fuzzy models for several applications [34–44] accounting for the further simplification of the OIA.

**Acknowledgments** This work was supported by a grant from the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109, by a grant from the Partnerships in priority areas – PN II program of the Romanian National Authority for Scientific Research ANCS, CNDI – UEFISCDI, project number PN-II-PT-PCCA-2011-3.2-0732, by grants from the Partnerships in priority areas – PN II program of the Romanian Ministry of Education and Research (MEDC) – the Executive Agency for Higher Education, Research, Development and Innovation Funding (UEFISCDI), project numbers PN-II-PT-PCCA-2013-4-0544 and PN-II-PT-PCCA-2013-4-0070, and by a grant from the NSERC of Canada.

## References

1. Angelov, P.: *Evolving Rule based Models: A Tool for Design of Flexible Adaptive Systems*. Springer, Berlin (2002)
2. Mouchaweh, M.S., Devillez, A., Lecolier, G.V., Billaudel, P.: Incremental learning in fuzzy pattern matching. *Fuzzy Sets Syst.* **132**, 49–62 (2002)
3. Lughofer, E.: *Evolving Fuzzy Systems—Methodologies*. Advanced Concepts and Applications. Springer, Berlin (2011)
4. Precup, R.-E., Angelov, P., Costa, B.S.J., Sayed-Mouchaweh, M.: An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Comput. Ind.* (2015). doi:[10.1016/j.compind.2015.03.001](https://doi.org/10.1016/j.compind.2015.03.001)
5. Dovžan, D., Logar, V., Škrjanc, I.: Implementation of an Evolving fuzzy model (eFuMo) in a monitoring system for a waste-water treatment process. *IEEE Trans. Fuzzy Syst.* (2014). doi:[10.1109/TFUZZ.2014.2379252](https://doi.org/10.1109/TFUZZ.2014.2379252)
6. Precup, R.-E., Filip, H.-I., Radac, M.-B., Pozna, C., Dragos, C.-A., Preitl, S.: Experimental results of evolving Takagi-Sugeno fuzzy models for a nonlinear benchmark. In: *2012 IEEE 3<sup>rd</sup> International Conference on Cognitive Infocommunications*, pp. 567–572. Kosice, Slovakia (2012)
7. Precup, R.-E., Filip, H.-I., Radac, M.-B., Petriu, E.M., Preitl, S., Dragos, C.-A.: Online identification of evolving Takagi-Sugeno-Kang fuzzy models for crane systems. *Appl. Soft Comput.* **24**, 1155–1163 (2014)
8. Angelov, P., Filev, D.: An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Trans. Syst. Man Cybern. B Cybern.* **34**, 484–498 (2004)
9. Precup, R.-E., Voisan, E.-I., Petriu, E.M., Radac, M.-B., Fedorovici, L.-O.: Implementation of evolving fuzzy models of a nonlinear process. In: *12<sup>th</sup> International Conference on Informatics in Control, Automation and Robotics*, vol.1, pp. 5–14. Colmar, Alsace, France (2015)
10. Precup, R.-E., David, R.-C., Petriu, E.M., Preitl, S., Radac, M.-B.: Gravitational search algorithms in fuzzy control systems tuning. In: *18<sup>th</sup> IFAC World Congress*, pp. 13624–13629. Milano, Italy (2011)
11. David, R.-C., Precup, R.-E., Petriu, E.M., Radac, M.-B., Preitl, S.: Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity. *Inf. Sci.* **247**, 154–173 (2013)
12. Chang, W.-J., Huang, B.-J.: Robust fuzzy control subject to state variance and passivity constraints for perturbed nonlinear systems with multiplicative noises. *ISA Trans.* **53**, 1787–1795 (2014)

13. Allouche, B., Vermeiren, L., Dequidt, A., Dambrine, M.: Step-crossing feasibility of two-wheeled transporter: analysis based on Takagi-Sugeno descriptor approach. In: IEEE 17<sup>th</sup> International Conference on Intelligent Transportation Systems, pp. 2675–2680. Qingdao, China (2014)
14. Chiu, C.-S.: A dynamic decoupling approach to robust T-S fuzzy model-based control. IEEE Trans. Fuzzy Syst. **22**, 1088–1100 (2014)
15. Tsai, P.-W., Chen, C.-W.: Novel criterion for nonlinear time-delay systems using LMI fuzzy Lyapunov method. Appl. Soft Comput. **25**, 461–472 (2014)
16. Gao, Q., Feng, G., Dong, D., Liu, L.: Universal fuzzy models and universal fuzzy controllers for discrete-time nonlinear systems. IEEE Trans. Cybern. **54**, 880–887 (2015)
17. Li, H., Sun, X., Shi, P., Lam, H.-K.: Control design of interval type-2 fuzzy systems with actuator fault: sampled-data control approach. Inf. Sci. **302**, 1–13 (2015)
18. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Syst. Man Cybern. **15**, 116–132 (1985)
19. Turnau, A., Pilat, A., Hajduk, K., Korytowski, A., Grega, W., Gorczyca, P., Kolek, K., Rosól, M.: Pendulum–Cart System User’s Manual. INTECO Ltd., Krakow (2008)
20. Ramos, J.V., Dourado, A.: On-line interpretability by rule base simplification and reduction. In: European Symposium on Intelligent Technologies, Hybrid Systems and Their Implementation on Smart Adaptive Systems, pp. 1–6. Aachen, Germany (2004)
21. Aires, L., Araújo, J., Dourado, A.: Industrial monitoring by evolving fuzzy systems. In: Joint 2009 IFSA World Congress and 2009 EUSFLAT Conference, pp. 1358–1363. Lisbon, Portugal (2009)
22. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**, 665–685 (1993)
23. Kasabov, N.K., Song, Q.: DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans. Fuzzy Syst. **10**, 144–154 (2002)
24. Lughofer, E., Klement, E. P.: FLEXFIS: a variant for incremental learning of Takagi-Sugeno fuzzy systems. In: 14<sup>th</sup> IEEE International Conference on Fuzzy Systems, pp. 915–920. Reno, NV, USA (2005)
25. Danković, B., Nikolić, S., Milojković, M., Jovanović, Z.: A class of almost orthogonal filters. J. Circuits Syst. Comput. **18**, 923–931 (2009)
26. Filip, F.-G., Leiviskä, K.: Large-scale complex systems. In: Nof, S.Y. (ed.) Springer Handbook of Automation, pp. 619–638. Springer, Heidelberg (2009)
27. Blažič, S., Matko, D., Škrjanc, I.: Adaptive law with a new leakage term. IET Control Theory Appl. **4**, 1533–1542 (2010)
28. Precup, R.-R., David, R.-C., Petriu, E.M., Preitl, S., Paul, A.S.: Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity. In: Gaspar-Cunha, A., Takahashi, R., Schaefer, G., Costa, L. (eds.) Soft Computing in Industrial Applications. Advances in Intelligent and Soft Computing, vol. 96, pp. 141–150. Springer, Heidelberg (2011)
29. Horváth, L., Rudas, I.J.: Active knowledge for the situation-driven control of product definition. Acta Polytech. Hung. **10**(2), 217–234 (2013)
30. Osaba, E., Diaz, F., Onieva, E., Carballedo, R., Perallos, A.: AMCPA: a population metaheuristic with adaptive crossover probability and multi-crossover mechanism for solving combinatorial optimization problems. Int. J. Artif. Intell. **12**(2), 1–23 (2014)
31. Tang, L., Zhao, Y., Liu, J.: An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production. IEEE Trans. Evol. Comput. **18**, 209–225 (2014)
32. Savio, M.R.D., Sankar, A., Vijayarajan, N.R.: A novel enumeration strategy of maximal bicliques from 3-dimensional symmetric adjacency matrix. Int. J. Artif. Intell. **12**(2), 42–56 (2014)
33. Zhao, X., Yin, S., Li, H., Niu, B.: Switching stabilization for a class of slowly switched systems. IEEE Trans. Autom. Control **60**, 221–226 (2015)
34. Preitl, S., Precup, R.-E.: Introducere in Conducerea Fuzzy a Proceselor. Editura Tehnica, Bucharest (1997)

35. Baranyi, P., Tikk, D., Yam, Y., Patton, R.J.: From differential equations to PDC controller design via numerical transformation. *Comput. Ind.* **51**, 281–297 (2003)
36. Precup, R.-E., Tomescu, M.L., Preitl, S.: Fuzzy logic control system stability analysis based on Lyapunov's direct method. *Int. J. Comput. Commun. Control* **4**, 415–426 (2009)
37. Boza, Sánchez, A.S., Haber-Guerra, R., Gajate, A.: Artificial cognitive control system based on the shared circuits model of sociocognitive capacities. A first approach. *Eng. Appl. Artif. Intell.* **24**, 209–219 (2011)
38. Precup, R.-E., Dragos, C.-A., Preitl, S., Radac, M.-B., Petriu, E.M.: Novel tensor product models for automatic transmission system control. *IEEE Syst. J.* **6**, 488–498 (2012)
39. Vaščák, J.: Adaptation of fuzzy cognitive maps by migration algorithms. *Kybernetes* **41**, 429–443 (2012)
40. Johanyák, Z.C.: Fuzzy modeling of thermoplastic composites' melt volume rate. *Comput. Inf.* **32**, 845–857 (2013)
41. Husek, P.: Robust PI controller design with respect to fuzzy sensitivity margins. *Appl. Soft Comput.* **13**, 2037–2044 (2013)
42. Mishra, J., Gho, S.: Uncertain query processing using vague set or fuzzy set: which one is better? *Int. J. Comput. Commun. Control* **9**, 730–740 (2014)
43. Kaur, G., Dhar, J., Guha, R.K.: Stock market forecasting using ANFIS with OWA operator. *Int. J. Artif. Intell.* **12**(2), 102–114 (2014)
44. Liu, Z., Chen, C., Zhang, Y.: Decentralized robust fuzzy adaptive control of humanoid robot manipulation with unknown actuator backlash. *IEEE Trans. Fuzzy Syst.* **23**, 605–616 (2015)

# Speech Phoneme Classification by Intelligent Decision-Level Fusion

Fréjus A.A. Laleye, Eugène C. Ezin and Cina Motamed

**Abstract** This paper explores the decision fusion for the phoneme recognition problem through intelligent combination of Naive Bayes and Learning Vector Quantization (LVQ) classifiers and feature fusion using Mel-frequency Cepstral Coefficients (MFCC), Relative Spectral Transform—Perceptual Linear Prediction (Rasta-PLP) and Perceptual Linear Prediction (PLP). This work emphasizes optimal decision making from decisions of classifiers which are trained on different features. The proposed architecture consists of three decision fusion approaches which are weighted mean, deep belief networks (DBN) and fuzzy logic. We proposed a performance comparison on a dataset of an African language phoneme, Fongbe, for experiments. The latter produced the overall decision fusion performance with the proposed approach using fuzzy logic whose classification accuracies are 95.54 % for consonants and 83.97 % for vowels despite the lower execution time of Deep Belief Networks.

## 1 Introduction

Phoneme classification is an integrated process to phoneme recognition and an important step in automatic speech recognition. Since the 60s, very significant research progress related to the development of statistical methods and artificial intelligence techniques, have tried to overcome the problems of analysis and characterization of the speech signal. Among the problems, there is still the acoustic and linguistic

---

F.A.A. Laleye · E.C. Ezin

Unité de Recherche en Informatique et Sciences Appliquées,  
Institut de Mathématiques et de Sciences Physiques, Université d'Abomey-Calavi,  
BP 613 Porto-Novo, Abomey Calavi, Benin  
e-mail: frejus.laleye@imsp-uac.org

F.A.A. Laleye · C. Motamed (✉)

Laboratoire d'Informatique Signal et Image de la Côte d'Opale,  
Université du Littoral Côte d'Opale, 50 rue F. Buisson, BP 719,  
62228 Calais Cedex, France  
e-mail: motamed@lisisic.univ-littoral.fr

© Springer International Publishing Switzerland 2016

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*,  
Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_4

specificity of each language. Considering the number of languages that exists, there were some good reasons for addressing phoneme recognition problems.

The aim of speech recognition is to convert acoustic signal to generate a set of words from a phonemic or syllabic segmentation of the sentence contained in the signal. Phoneme classification is the process of finding the phonetic identity of a short section of a spoken signal [11]. To obtain good recognition performance, the phoneme classification step must be well achieved in order to provide phoneme acoustic knowledge of a given language. Phoneme classification is applied in various applications such as speech and speaker recognition, speaker indexing, synthesis etc. and it is a difficult and challenging problem.

In this paper, we placed the phoneme recognition problems in a classification context from multiple classifiers. We dealt with the decision-level fusion from two different classifiers namely Naive Bayes and Learning Vector Quantization (LVQ). Since the 90s, the combining classifiers has been one of the most sustained research directions in the area of pattern recognition. Methods of decision-level fusion have been successfully applied in various areas such as the recognition and verification of signatures, the identification and face recognition or the medical image analysis. In automatic speech recognition, decision-level fusion was introduced to recognize phoneme, speech, speaker age and gender and to identify language with the best performance. The work we present in this paper deals with the phoneme recognition of Fongbe language which is an unresourced language. Fongbe is an African language spoken especially in Benin, Togo and Nigeria countries. It is a poorly endowed language which is characterized by a series of vowels (oral and nasal) and consonants (oral and nasal). Its recent written form consists of a number of Latin characters and the International Phoneoutic Alphabet. Scientific studies on the Fongbe started in 1963. In 2010, there was the first publication of Fongbe-French dictionary [3]. Since 1976, several linguists have worked on the language and many papers have been published on the linguistic aspects of Fongbe. Until today, these works have been aimed at the linguistic description of Fongbe, but very few works have addressed automatic processing with a computing perspective.

The idea behind this work is to propose a robust discriminatory system of consonants and vowels thanks to intelligent classifier combination based on decision-level fusion. To achieve this goal, we investigated on both methods of decision fusion namely the non-parametric method using weighted combination and parametric method using deep neural networks and a proposed adaptive approach based on fuzzy logic. The intelligent decision-level fusion used in this work to perform classification is carried out after the feature-level fusion of MFCC, Rasta-PLP, PLP coefficients applied to classifiers to represent the phonetic identity of each phoneme of the chosen language. In other words, the features were initially combined to produce coefficients as input variables to the classifiers. Experiments were performed on our Fongbe phoneme dataset and showed better performance with the proposed fuzzy logic approach. The rest of the paper is organized as follows. In Sect. 2, we briefly present the related works on phoneme recognition and decision fusion. Section 2.1 presents an overview on the proposed classification system. In Sect. 3, we describe the classifier methods and their algorithms. In Sect. 4, the proposed Fongbe phoneme

classification is detailed and explained. Experimental results are reported in Sect. 5. In the same section we present a detailed analysis of the used performance parameters to evaluate the decision fusion methods. We finally conclude this paper in Sect. 6.

## 2 Related Works

This work deals with two different issues namely decision-level fusion from multiple classifiers and phoneme classification of a West Africa local language (Fongbe).

### 2.1 Overview on Phoneme Classification

Some of the recent research works related to phoneme classification applied to the world's languages are discussed as follows.

In [22], the authors proposed an approach of phoneme classification which performed better on TIMIT speech corpus, with warp factor value greater than 1. They have worked on compensating inter-speaker variability through Vocal tract length normalization multi-speaker frequency warping alternative approach. Finally, they compared each phoneme recognition results from warping factor between 0.74 and 1.54 with 0.02 increments on nine different ranges of frequency warping boundary. Their obtained results showed that performance in phoneme recognition and spoken word recognition was respectively improved by 0.7 % and 0.5% using warp factor of 1.40 on frequency range of 300–5000 Hz.

Phoneme classification is investigated for linear feature domains with the aim of improving robustness to additive noise [1]. In this paper, the authors performed their experiments on all phonemes from the TIMIT database in order to study some of the potential benefits of phoneme classification in linear feature domains directly related to acoustic waveform, with the aim of implementing exact noise adaptation of the resulting density model. Their conclusion was that they obtained the best practical classifiers paper by using the combination of acoustic waveforms with  $PLP + \Delta + \Delta\Delta$ .

In [11], the authors integrated into phoneme classification a non-linear manifold learning technique, namely “Diffusion maps” that is to build a graph from the feature vectors and maps the connections in the graph to Euclidean distances, so using Euclidean distances for classification after the non-linear mapping is optimal. The experiments performed on more than 1100 isolated phonemes, excerpted from the TIMIT speech database, of both male and female speakers show that Diffusion maps allows dimensionality reduction and improves the classification results.

The work presented in [30] successfully investigates a convolutional neural network approach for raw speech signal with the experiments performed on the TIMIT and Wall Street Journal corpus datasets. Still on the TIMIT datasets, the authors in [37] focused their work on the robustness of phoneme classification to additive

noise in the acoustic waveform domain using support vector machines (SVMs). The authors in [9] used a preprocessing technique based on a modified Rasta-PLP algorithm and a classification algorithm based on a simplified Time Delay Neural Network (TDNN) architecture to propose an automatic system for classifying the English stops [b, d, g, p, t, k]. And in [8], they proposed an artificial Neural Network architecture to detect and classify correctly the acoustic features in speech signals.

Several works have been achieved on the TIMIT dataset which is the reference speech dataset, but other works were performed on other languages than those included in the TIMIT dataset. We can cite, for example the following papers [19, 26, 28, 34], where the authors worked respectively on Vietnamese, Afrikaans, English, Xhosa, Hausa language and all American English phonemes.

A state of the art on the works related to Fongbe language stands out the works in the linguistic area. In [2], the authors studied how six Fon enunciative particles work: the six emphatic particles h...n “hence”, sin “but”, m “in”, l “insist”, lo “I am warning you”, and n “there”. Their work aimed at showing the variety and specificity of these enunciative particles. In these works [3, 20] listed in the Fongbe language processing, the authors introduced and studied grammar, syntax and lexicology of Fongbe.

In [18], the authors addressed the Fongbe automatic processing by proposing a classification system based on a weighted combination of two different classifiers. Because of the uncertainty of obtained opinions of each classifier due to the imbalance per class of training data, the authors used the weighted voting to recognize the consonants and vowels.

## 2.2 *Decision-Level Fusion Methods*

The second issue dealt with in this work is the decision fusion for optimal Fongbe phoneme classification. Combining decisions from classifiers to achieve an optimal decision and higher accuracy became an important research topic. In the literature, there are researchers who decided to combine multiple classifiers [6, 16, 33]. Other researchers worked on mixture of experts [14, 15].

In decision fusion methods, there are so-called non-parametric methods (classifiers outputs are combined in a scheme whose parameters are invariant) and the learning methods that seek to learn and adapt on the available data, the necessary parameters to the fusion. In speech recognition, several researchers successfully adopted the decision level fusion to recognize phoneme, speech, speaker age and gender and to identify language. For example, the authors in [24] performed decision level combination of multiple modalities for the recognition and the analysis of emotional expression. Some authors adopted non-parametric methods as weighted mean [13, 21, 27] and majority voting [7, 31]. Others adopted parametric methods as Bayesian inference [25, 32, 36] and Dempster-Shafer method [10].

In this work we adopted both methods to compare their performance in decision fusion of classifiers for an optimal phoneme classification of Fongbe language. First,

we performed a weighted mean, which is a non-parametric method, to combine decisions. This method needs a threshold value chosen judiciously by experiment in the training stage. The second method we used is a parametric method with learning based on deep belief networks. Deep Belief Networks (DBNs) have recently shown impressive performance in decision fusion and classification problems [29]. In addition to these two methods we also used an adaptive approach based on fuzzy logic. Fuzzy logic is often used for classification problems and has recently shown a good performance in speech recognition [23]. Indeed, the limitations of the use of threshold value that requires weighted mean is that the value is fixed and does not provide flexibility to counter any variations in the input data. In order to overcome the limitations of the threshold based weighted mean which gives a hard output decision of which either “True” or “false” and the time that can be taken a training process of deep belief networks, we proposed a third approach based on fuzzy logic which can imitate the decision of humans by encoding their knowledge in the form of linguistic rules. Fuzzy logic requires the use of expert knowledge and is able to emulate human thinking capabilities in dealing with uncertainties.

### 3 Classification Methods and Algorithms

We detail in this section the algorithm implemented in the classification methods used for the discriminating system of consonants and vowels phonemes.

#### 3.1 Naive Bayes Classifier

Naive Bayes is a probabilistic learning method based on the Bayes theorem of Thomas Bayes with independence assumptions between predictors. It appears in the speech recognition to solve the multi-class classification problems. It calculates explicitly the probabilities for hypothesis and it is robust to noise in input data. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. The Bayes classifier decides the class  $c(x)$  of the input data  $x$  based on the Bayes rule:

$$p(c|x) = \frac{p(c, x)}{p(x)} \quad (1)$$

$$= \frac{p(c)p(x|c)}{\sum_{c'} p(c')p(x|c')} \quad (2)$$

where  $p(c)$  is the prior probability of class  $c$ , and  $p(x|c)$  is the class  $c$ -conditional probability of  $x$ .

Consider an example  $X = \{x_1, x_2, \dots, x_n\}$   
 $X$  is classified as the class  $C = +$  if and only if,

$$F(X) = \frac{p(C = +|X)}{p(C = -|X)} \geq 1 \quad (3)$$

$F(X)$  is a Bayesian classifier.

Naive Bayes is the simplest form of Bayesian network, in which we assume that all attributes are independent given the class [38].

$$p(X|c) = p(x_1, x_2, \dots, x_n|c) = \prod_{i=1}^n p(x_i|c) \quad (4)$$

The naive Bayesian classifier is obtained by:

$$F_{nb}(X) = \frac{p(C = +|X)}{p(C = -|X)} \prod_{i=1}^n \frac{p(x_i|C = +)}{p(x_i|C = -)} \quad (5)$$

### 3.2 Learning Vector Quantization Classifier

Learning Vector Quantization (LVQ) is a supervised version of vector quantization. Networks LVQ were proposed by Kohonen [17] and are hybrid networks which use a partially supervised learning [5].

**Algorithm** LVQ method algorithm can be summarized as follows:

1. Initialize the weights  $w_{ij}^{(1)}$  to random values between 0 and 1.
2. Adjust the learning coefficient  $\eta(t)$
3. For each prototype  $p_i$ , find the neuron of the index  $i^*$  which has the weight vector  $w_{i^*}^{(1)}$  closest to the  $p_i$ .
4. If the specified class at the network output for the neuron of the index  $i^*$  corresponds to the prototype of the index  $i$ , then do:

$$w_{i^*}^{(1)}(t+1) = w_{i^*}^{(1)}(t) + \eta(t)(p(t) - w_{i^*}^{(1)}(t)) \quad (6)$$

else

$$w_{i^*}^{(1)}(t+1) = w_{i^*}^{(1)}(t) - \eta(t)(p(t) - w_{i^*}^{(1)}(t)) \quad (7)$$

5. If the algorithm has converged with the desired accuracy, then stop otherwise go to step 2 by changing the prototype.

## 4 Proposed Phoneme Classification System

### 4.1 Overview of Classification System

Our intelligent fusion system is summarized in two modules which are each subdivided into submodules:

1. feature-level fusion and classification: the first module performs classification with Naive Bayes and LVQ classifier and produces outputs with the coefficients obtained after features fusion and which are applied as input. This module contains the submodules which are (i) signal denoising, (ii) feature extraction (MFCC, PLP, and Rasta-PLP), (iii) features fusion and classification with Naive Bayes and LVQ.
2. decision-level fusion and optimal decision making: the second module performs in parallel the decisions fusion with fuzzy approach that we proposed and the method with learning based on Deep Belief Networks.

Both modules are separated by an intermediate module which performs weighted mean calculation of classifiers outputs and contains the submodule which is (iv) standardization for classifiers decisions database. In this submodule, the outputs of the first module are combined to produce a single decision that is applied to the decision-level fusion module. The various steps are shown in Fig. 1.

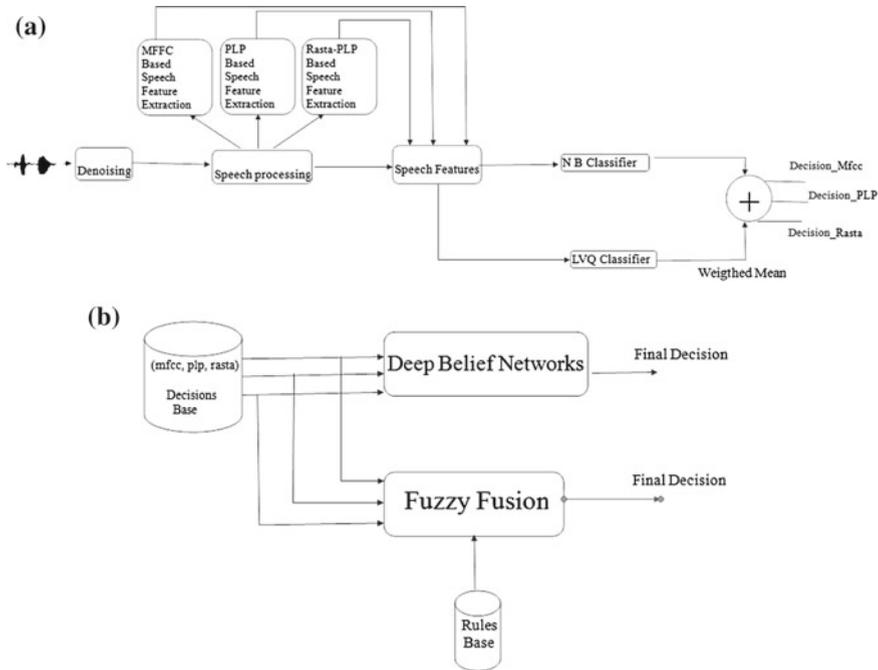
### 4.2 Speech Feature Extraction

From phoneme signals we extracted MFCC, PLP and Rasta-PLP coefficients to perform the proposed adaptive decision fusion using Fuzzy approach and deep belief networks. The benefit of using these three types of coefficients is to expand the variation scale from input data of classification system. This enabled our system to learn more acoustic information of Fongbe phonemes. These three speech analysis techniques were initially allowed to train two classifiers and then put together to build the set of input variables to the decision fusion. Phoneme signals were split into frame segments of length 32 ms and the first 13 cepstral values were taken.

### 4.3 Decision Fusion Using Simple Weighted Mean

An intermediate step between the two steps was the normalization of output data of the first step. First, we calculated the weighted mean value of the two classifier outputs for each coefficient using the expression (8).

$$input_1 = \frac{S^{naivebayes} \times \tau^{naivebayes} + S^{lvq} \times \tau^{lvq}}{\tau^{naivebayes} + \tau^{lvq}} \quad (8)$$



**Fig. 1** Paradigm of our classification system. **a** Classification and standardization. **b** Decision fusion using fuzzy logic and deep belief networks

$S^A$  represents the output of classifier A whereas  $\tau^A$  represents the recognition rate of classifier A. Before applying fuzzy logic and neuronal technique to fuse the decisions of each classifier, we performed the output combination based on the simple weighted sums method using the threshold value obtained and given by Eq.9.

$$\tau = -1.2 \sum_i C_i + 2.75 \left( \sum_k w_k^1 \lambda_1 + \sum_k w_k^2 \lambda_2 \right) \quad (9)$$

$C_i$ : is the number of class i,  $w_k^1$ : weight of classifier k related to class 1,  $w_k^2$ : weight of classifier k related to the class 2,  $\lambda_1$  and  $\lambda_2$  are values that are 0 or 1 depending on the class. For example, for the consonant class:  $\lambda_1 = 1$  and  $\lambda_2 = 0$ . The results are compared with fuzzy logic method and neuronal method to evaluate the performance of our phoneme classification system.

#### 4.4 Fuzzy Logic Based Fusion

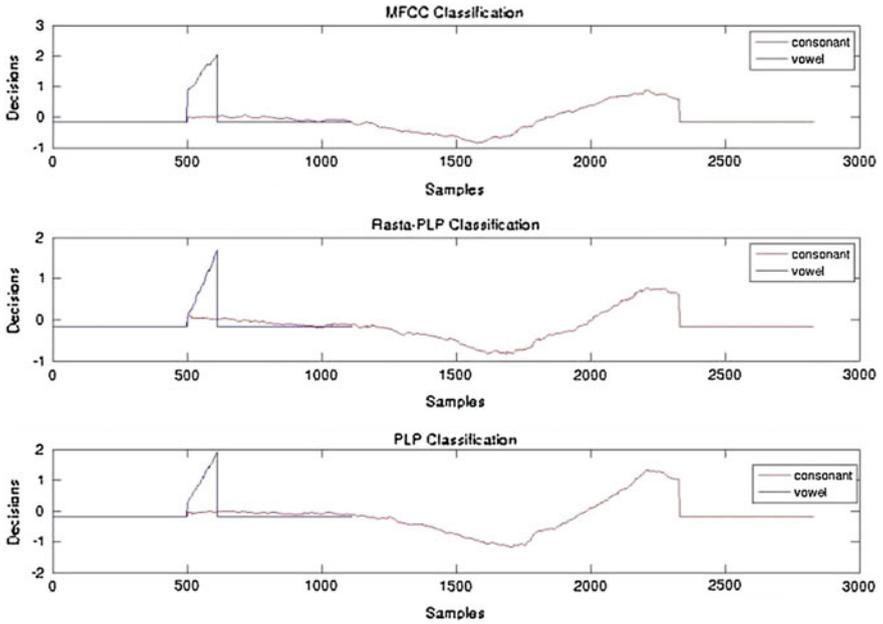
The Nature of the results obtained in the first step allows us to apply fuzzy logic on four membership functions. The inputs to our fuzzy logic system are MFCC, PLP

**Table 1** Generated fuzzy rules

Rules no	Input			Output
	MFCC	Rasta	PLP	
1	Low	Low	Low	Consonant
2	Low	Low	Medium	Vowel
3	Low	Low	High	Consonant
4	Low	Medium	Low	Vowel
5	Low	High	Low	Consonant
6	Low	High	High	Consonant
7	Low	Very high	Low	Vowel
8	Low	Very high	Very high	Vowel
9	Medium	Low	Low	Vowel
10	Medium	Low	Very high	Vowel
11	Medium	Very high	Low	Vowel
12	Medium	Very high	Very high	Vowel
13	High	Low	Low	Consonant
14	High	Low	High	Consonant
15	High	High	Low	Consonant
16	High	High	High	Consonant
17	Very high	Low	Low	Vowel
18	Very high	Low	Medium	Vowel
19	Very high	Low	High	Consonant
20	Very high	Low	Very high	Vowel
21	Very high	Medium	Low	Vowel
22	Very high	Medium	Very high	Vowel
23	Very high	High	High	Consonant
24	Very high	Very high	Low	Vowel
25	Very high	Very high	Medium	Vowel
26	Very high	Very high	Very high	Vowel

and Rasta-PLP and the output obtained is the membership degree of a phoneme to a consonant or vowel class. The input variables are fuzzified into four complementary sets namely: *low*, *medium*, *high* and *very high* and the output variable is fuzzified into two sets namely: consonant and vowel. Table 1 shows the fuzzy rules which were generated after fuzzification.

First, the input data is arranged in an interval as  $[X_{\min} \dots X_{\max}]$ . The different membership functions were obtained by examining the local distribution of samples of both classes (see Fig. 2). Local distribution has induced four subsets according to the variation of the input data and the output is obtained depending on the nature of the data. For example, if we give MFCC, PLP and Rasta as input to the system, the consonant or vowel output is obtained according to the subsets of the input data.



**Fig. 2** *Top* Local distribution of decisions from MFCC coefficients classification, *Middle* local distribution of decisions from Rasta-PLP coefficients, *Bottom* local distribution of decisions from PLP coefficients

Because of the linearity of values in the subsets, a simple triangle curve (*trimf*) is used for low and medium membership functions and a trapeze curve (*trapmf*) is used for high and very high membership functions.

#### 4.5 DBN Based Fusion

This method based on the use of deep belief networks (DBNs) requires a learning step for a good adaptation of the decisions to the system input. DBNs are multilayered probabilistic generative models which are constructed as hierarchies of recurrently connected simpler probabilistic graphical models, so called Restricted Boltzmann Machines (RBMs) [4, 12]. Every RBM consists of two layers of neurons, a hidden and a visible layer. Using unsupervised learning, each RBM is trained to encode in its weight matrix a probability distribution that predicts the activity of the visible layer from the activity of the hidden layer [29].

To perform the classifier for making of decision, we used the DBN parameters showed in Table 2

**Table 2** DBN parameters

RBM layer 1	200 units
RBM layer 2	200 units
Learning rate	0.01
Training epochs	100
Batch size	8

Algorithms 1 and 2 summarize the different parts of our classifier implemented with Matlab. Function names give the idea about the operation they perform and sentences beginning with // represent comments. For example, `final_decision_2 ← dbnfusion(all_input)` means that the optimal decision given by DBN fusion is stored in `final_decision_2`.

---

**Algorithm 1:** Classification with Naive Bayes and LVQ
 

---

**Data:** Phoneme signals

**Result:** Decision of each classifier for each extraction technique.

*signal denoising;*

**for** *signal* ∈ *phoneme\_database* **do**

*signal* ← *denoising*(*signal*);

*base* ← *put*(*signal*)

**end**

*Feature extraction;*

**for** *signal* ∈ *base* **do**

*m* ← *mfcc\_calculation*(*signal*);

*p* ← *plp\_calculation*(*signal*);

*r* ← *rasta\_calculation*(*signal*);

*base\_mfcc* ← *put*(*m*);

*base\_plp* ← *put*(*p*);

*base\_rasta* ← *put*(*r*);

**end**

*training* ← *put*(*m,p,r*);

*//Classification with Naive Bayes and LVQ;*

**for** *i* ← 1 to *size*(*training*) **do**

**if** *i* ≤ *size*(*base\_mfcc*) **then**

*bayes\_mfcc\_decision* ← *bayes*(*training*(*i*));

*lvq\_mfcc\_decision* ← *lvq*(*training*(*i*));

**end**

**if** *i* > *size*(*base\_mfcc*) **and** *i* ≤ *size*(*base\_mfcc*) + *size*(*base\_plp*) **then**

*bayes\_plp\_decision* ← *bayes*(*training*(*i*));

*lvq\_plp\_decision* ← *lvq*(*training*(*i*));

**end**

**if** *i* > *size*(*base\_mfcc*) + *size*(*base\_plp*) **and**

*i* ≤ *size*(*base\_mfcc*) + *size*(*base\_plp*) + *size*(*base\_rasta*) **then**

*bayes\_rasta\_decision* ← *bayes*(*training*(*i*));

*lvq\_rasta\_decision* ← *lvq*(*training*(*i*));

**end**

**end**

---

## 5 Experimental Results and Analysis

We present the different results obtained after training and testing with two classifiers and the results of decision fusion with fuzzy logic approach and deep belief networks. Experiments were performed on phonemes of the Fongbe language that we describe in the next section. Programming was done with Matlab in an environment which is Intel Core i7 CPU L 640 @ 2.13 GHz  $\times$  4 processor with 4 GB memory.

---

### Algorithm 2: Decision fusion with Fuzzy logic and Deep belief networks

---

**Data:** Decision of each classifier for each extraction technique.

**Result:** Final Decision

```

//calculation of recognition rate;
for j ← 1 to size(classes) and k ← 1 to size(classifiers) do
  | τ ← -1, 2 ∑i Ci + 2, 75(∑k wk1λ1 + ∑k wk2λ2);
end
//calculation of weighted mean values as input of fuzzy system;
for l ← 1 to 3 do
  | inputi ←  $\frac{\tau^{naivebayes} * \tau^{naivebayes} + \tau^{lvq} * \tau^{lvq}}{\tau^{naivebayes} + \tau^{lvq}}$ ;
  | all_input ← put(inputi);
end
final_decision_1 ← fuzzylogicsystem(all_input);
final_decision_2 ← dbnfusion(all_input);

```

---

### 5.1 Speech Data Structure

The used speech dataset was obtained by recording different phonemes pronounced by foreigners and natives speakers with a recorder in various environments of real life. It contains 174 speakers whose ages are between 9 and 45 years, including 53 women (children and adults) and 119 men (children and adults). It is an audio corpus of around 4 h of pronounced phonemes which includes 4929 speech signals for all 32 phonemes. 80 % of speech signals in dataset is used to construct the training data and 20 % for the testing data.

### 5.2 Classification Results

LVQ parameters:

- number of hidden neurons: 60
- first class and second class percentage: 0.6 and 0.4
- learning rate: 0.005
- number of epochs: 750

**Table 3** Training and testing results

Classifier	MFCC		Rasta-PLP		PLP	
	$C_1$	$C_2$	$C_1$	$C_2$	$C_1$	$C_2$
Training results						
Naive Bayes	88.66	51.53	90.43	59.17	88.2	68.25
LVQ	98.09	47.44	97.32	40.65	97.35	51.53
Testing results						
Naive Bayes	92.29	38.34	91.48	46.04	93.10	60.24
LVQ	98.78	24.95	98.58	21.70	97.97	20.89

Values are estimated in percentage

Normal distribution is used for Naive Bayes classification. Table 3 shows the training results and the testing recognition rate.

### 5.3 Decision Fusion Results of Classifiers

Table 4 presents the fusion results of the methods we used.

### 5.4 Performance Analysis

Several measures were developed to deal with the classification problem [35]. The values of True Positive (TP), True Negative (TN), False Positive and False Negative were calculated after decision fusion with the different used methods. These values were used to compute performance parameters like sensitivity (SE), specificity (SP), Likelihood Ratio Positive (LRP), Accuracy (Ac) and Precision (Pr). Three other important measures were used as evaluation metrics: *F*-measure, *G*-measure and execution time. *F*-measure considers both the precision *Pr* and the sensitivity *SE* to compute the score which represents the weighted harmonic mean (precision and sensitivity). *G*-mean is defined by sensitivity and specificity and measures the balanced performance of learning between the positive class and the negative class.

**Table 4** Results of decision fusion using fuzzy logic

Fusion methods	Consonant (%)	Vowel (%)
Weighted mean	99.73	54.02
Fuzzy logic	95.54	83.97
Deep belief networks	88.84	84.79

**Table 5** Performance analysis

Parameters	Naive Bayes	LVQ	Using weighted mean	Using fuzzy logic	Using deep belief nets
SE	0.93	0.99	0.99	0.95	0.88
SP	0.60	0.25	0.38	0.84	0.86
LRP	2.36	1.32	1.60	5.94	6.28
LRN	0.12	0.04	0.03	0.06	0.14
Ac	0.77	0.62	<b>0.69</b>	<b>0.90</b>	<b>0.87</b>
Pr	0.70	0.57	0.62	0.86	0.88
F-measure	0.80	0.72	<b>0.76</b>	<b>0.90</b>	<b>0.88</b>
G-measure	0.75	0.50	<b>0.61</b>	<b>0.89</b>	<b>0.87</b>
Execution time (s)	–	–	<b>0.10</b>	<b>0.7</b>	<b>0.04</b>

Values in bold are emphasized for the performance comparison

Execution time measures the computation time of each fusion methods in the testing step.

We used the same dataset to evaluate the performance of Naive Bayes, LVQ and the decision fusion methods on consonants and vowels of Fongbe phonemes. Table 4 shows that by considering the balance of phoneme classes, decision fusion of classifiers based on fuzzy logic achieved better performance even if the approaches based on the weighted mean and deep belief networks classified respectively consonants and vowels better than fuzzy logic. We noticed that fuzzy logic approach combined efficiently the decisions and got the optimal decision, but with an execution time increased by sixty percent compared to DBN. The results in Table 5 show the highest performances of Fuzzy logic approach on Accuracy, F-measure and G-measure parameters which were the chosen metrics to evaluate the performance of the compared methods. The best performances obtained with fuzzy logic confirmed that adding extra expert knowledge improves decision making after decision combination made by multiple classifiers.

## 6 Conclusion

This paper evaluates the performance of three decision-level fusion methods by intelligent classifier combination in a speech phoneme classification problem. The performance evaluation was achieved with methods such as weighted mean, deep belief networks and fuzzy logic after combination of Naive Bayes and LVQ and feature-level fusion. The main idea is to make an optimal decision compared with the decisions obtained with each classifier. The results of the accuracy, F-measure and G-measure parameters achieved in Table 5, show the best performance with the proposed decision fusion using fuzzy logic which uses human reasoning. So, this paper highlights two main results: (i) the performance comparison of three decisions

fusion methods in a phoneme classification problem with multiple classifiers and (ii) the proposal of a robust Fongbe phoneme classification system which incorporates a fusion of Naive Bayes and LVQ classifiers using fuzzy logic approach. This proposal builds on the performance achieved by our fuzzy logic-based approach compared to DBN-based approach and especially because of the limitations of the fixed threshold value in weighted combination. Future works include automatic speech segmentation into syllable units and an automatic continuous speech recognition based on speech phoneme classification.

## References

1. Ager, M., Cvetkovic, Z., Sollich, P.: Phoneme classification in high-dimensional linear feature domains. *Comput. Res. Repository* (2013)
2. Agoli-Agbo, E.O., Bernard, C.: *Les particules nonciatives du fon*. Institut national des langues et civilisations orientales, Paris, 1st edition (2009)
3. Akoha, A.B.: *Syntaxe et lexicologie du fon-gbe: Bénin*. Ed. L'harmattan, p. 368 (2010)
4. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: *Advances in Neural Information Processing Systems* (2006)
5. Borne, P., Benrejeb, M., Haggege, J.: *Les rseaux de neurones, présentation et applications*. TECHNIP Editions, p. 90 (2007)
6. Cho, S.-B., Kim, J.: Combining multiple neural networks by fuzzy integral and robust classification. *IEEE Trans. Syst. Man Cybern.* 380–384 (1995)
7. Corradini, A., Mehta, M., Bernsen, N., Martin, J., Abrilian, S.: Multimodal input fusion in humancomputer interaction. In: *NATO-ASI Conference on Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management* (2003)
8. Esposito, A., Ezin, E., Ceccarelli, M.: Preprocessing and neural classification of english stop consonants [b, d, g, p, t, k]. In: *The 4th International Conference on Spoken Language Processing*, pp. 1249–1252. Philadelphia (1996)
9. Esposito, A., Ezin, E., Ceccarelli, M.: Phoneme classification using a rasta-PLP preprocessing algorithm and a time delay neural network: performance studies. In: *Proceedings of the 10th Italian Workshop on Neural Nets*, pp. 207–217. Salerno (1998)
10. Foucher, S., Laliberte, F., Boulianne, G., Gagnon, L.: A dempster-shafer based fusion approach for audio-visual speech recognition with application to large vocabulary french speech. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1 (2006)
11. Genussov, M., Lavner, Y., Cohen, I.: Classification of unvoiced fricative phonemes using geometric methods. In: *12th International Workshop on Acoustic Echo and Noise Control*. Tel-Aviv, Israel (2010)
12. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. *Neural Comput.* **18**, 1527–1554 (2006)
13. Iyengar, G., Nock, H., Neti., C.: Audio-visual synchrony for detection of monologue in video archives. In: *IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 329–332 (2003)
14. Jacobs, R.: Methods for combining experts's probability assessments. *Neural Comput.* 867–888 (1995)
15. Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive mixture of local experts. *Neural Comput.* 79–87 (1991)
16. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. *IEEE Trans. Patt. Anal. Mach. Intell.* 226–239 (1998)
17. Kohonen, T.: An introduction to neural computing. *Neural Netw.* **1**, 3–16 (1988)

18. Laleye, F.A.A., Ezin, E.C., Motamed, C.: Weighted combination of naive bayes and lvq classifier for fonbe phoneme classification. In: Tenth International Conference on Signal-Image Technology and Internet-Based Systems, pp. 7–13, Marrakech. IEEE (2014)
19. Le, V.-B., Besacier, L.: Automatic speech recognition for under-resourced languages: Application to vietnamese language. In: IEEE Transactions on Audio, Speech, and Language Processing, pp. 1471–1482. IEEE (2009)
20. Lefebvre, C., Brousseau, A.: A grammar of fonbe, de gruyter mouton, p. 608 (2001)
21. Lewis, T.W., Powers, D.M.: Improved speech recognition using adaptive audio-visual fusion via a stochastic secondary classifier. *Int. Symp. Intell. Multimedia Video Speech Process.* **1**, 551–554 (2001)
22. Lung, J.W.J., Salam, M.S.H., Rehman, A., Rahim, M.S.M., Saba, T.: Fuzzy phoneme classification using multi-speaker vocal tract length normalization. *IETE Technical Review*, London, 2nd edn (2014)
23. Malcangi, M., Ouazzane, K., Patel, K.: Audio-visual fuzzy fusion for robust speech recognition. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE, Dallas (2013)
24. Metallinou, A., Lee, S., Narayanan, S.: Decision level combination of multiple modalities for recognition and analysis of emotional expression. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 2462–24665 (2010)
25. Meyer, G., Mulligan, J., Wuergler, S.: Continuous audio-visual digit recognition using n-best decision fusion. *Inf. Fusion* **5**, 91–101 (2004)
26. Mugler, E.M., Patton, J.L., Flint, R.D., Wright, Z.A., Schuele, S.U., Rosenow, J., Shih, J.J., Krusienski, D.J., Slutzky, M.W.: Direct classification of all american english phonemes using signals from functional speech motor cortex. *J. Neural Eng.* (2014)
27. Neti, C., Maison, B., Senior, A., Iyengar, G., Decuetos, P., Basu, S., Verma, A.: Joint processing of audio and visual information for multimedia indexing and human-computer interaction. In: Sixth International Conference RIAO, pp. 294–301. France, Paris (2000)
28. Niesler, T., Louw, P.H.: Comparative phonetic analysis and phoneme recognition for afrikaans, english and xhosa using the african speech technology telephone speech database. *S. Afr. Comput. J.* 3–12 (2004)
29. O'Connor, P., Neil, D., SC, L., Delbruck, T., Pfeiffer, M.: Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* (2013)
30. Palaz, D., Collobert, R., Magimai-Doss, M.: End-to-end phoneme sequence recognition using convolutional neural networks. *Idiap-RR* (2013)
31. Pflieger, N.: Context based multimodal fusion. In: ACM International Conference on Multimodal Interfaces, pp. 265–272 (2004)
32. Pitsikalis, V., Katsamanis, A., Papandreou, G., Maragos, P.: Adaptive multimodal fusion by uncertainty compensation. In: Ninth International Conference on Spoken Language Processing, vol. 7, pp. 423–435. Pittsburgh (2006)
33. Rogova, G.: Combining the results of several neural networks classifiers. *Neural Netw.* 777–781 (1994)
34. Schlippe, T., Djomgang, E.G.K., Vu, N.T., Ochs, S., Schultz, T.: Hausa large vocabulary continuous speech recognition. In: The third International Workshop on Spoken Languages Technologies for Under-resourced Languages. Cape-Town (2012)
35. Wang, S., Yao, X.: Diversity analysis on imbalanced data sets by using ensemble models. *IEEE Symp. Comput. Intell. Data Min.* 324–331 (2009)
36. Xu, H., Chua, T.: Fusion of av features and external information sources for event detection in team sports video. *ACM Trans. Multimedia Comput. Commun. Appl.* **2**, 44–67 (2006)
37. Yousafzai, J., Cvetkovic, Z., Sollich, P.: Tuning support vector machines for robust phoneme classification with acoustic waveforms. In: 10th Annual conference of the International Speech Communication Association, pp. 2359–2362. England (2009). ISCA-INST SPEECH COMMUNICATION ASSOC
38. Zhang, H.: Exploring conditions for the optimality of nave bayes. *IJPRAI* **19**, 183–198 (2005)

# Genetic Algorithm for Automated X-Ray Diffraction Full-Profile Analysis of Electrolyte Composition on Aluminium Smelters

Alexandr Zaloga, Shakhnaz Akhmedova, Igor Yakimov,  
Sergey Burakov, Eugene Semenkin, Petr Dubinin, Oksana Piksina  
and Eugene Andryushchenko

**Abstract** Aluminium is produced by means of the electrolysis of alumina in molten fluoride salts. A certain proportion of the fluoride compounds continuously evaporates, and this negatively impacts on the optimal composition of the electrolyte in the electrolytic baths. It means that a regular adjustment of the electrolyte composition is required by the addition of fluorides based on the results of the automatic express analysis of the electrolyte. The XRD phase analysis of crystallized electrolyte samples automatically performs the control of the main composition characteristics. This method, most frequently implemented in conjunction with aluminium smelters, necessitates periodic calibration with reference samples, whose phase composition is known exactly. The preparation of such samples is relatively complex since samples include 5–6 different phases with variable microcrystalline structure. One further

---

S. Akhmedova · S. Burakov · E. Semenkin (✉)  
Siberian State Aerospace University, 31 “Krasnoyarsky Rabochy” av.,  
Krasnoyarsk 660014, Russia  
e-mail: eugenesemenkin@yandex.ru

S. Akhmedova  
e-mail: shahnaz@inbox.ru

S. Burakov  
e-mail: burakov\_krasu@mail.ru

A. Zaloga · I. Yakimov · P. Dubinin · O. Piksina · E. Andryushchenko  
Siberian Federal University, 79 Svobodny av., Krasnoyarsk 660041, Russia  
e-mail: zaloga@yandex.ru

I. Yakimov  
e-mail: i-s-yakimov@yandex.ru

P. Dubinin  
e-mail: dubinin-2005@yandex.ru

O. Piksina  
e-mail: opiksina@gmail.com

E. Andryushchenko  
e-mail: john25041992@mail.ru

© Springer International Publishing Switzerland 2016

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*, Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_5

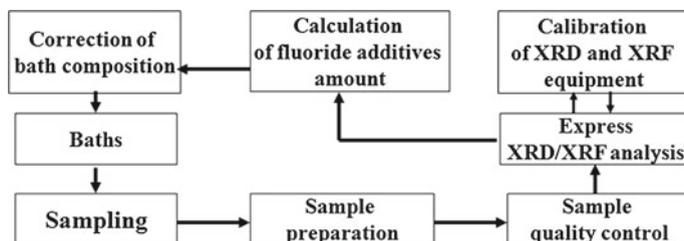
diffraction method is the Rietveld method, which can be implemented without the use of reference samples. The method is based on the modelling of the experimental powder patterns of crystalline samples as the sum of the powder patterns of comprised phases, calculated from their atomic crystal structure. Included in the simulation is a refinement of the profile parameters and crystal structure of phases using the non-linear least squares method. The difficulty associated with the automation of this approach is that a set of initial values for the parameters must be inputted that must be automatically refined by LSM to exact values. In order to resolve this problem, an optimization method was put forward by the article based on an evolutionary choice of initial values of profile and structural parameters using a genetic algorithm. The criterion of the evolution is the minimization of the profile R-factor, which represents the weighted discrepancy between the experimental and model powder patterns of the electrolyte sample. It is established that this approach achieves the required level of accuracy and complete automation of the electrolyte composition control.

**Keywords** Aluminium industry · Aluminium electrolyte · Cryolite ratio · Bath ratio · X-ray diffraction analysis · Rietveld method · Genetic algorithms · Chemical control

## 1 Introduction

Aluminium production is based on the electrolysis of alumina consisting of various forms of  $\text{Al}_2\text{O}_3$ . Electrolytes of aluminium electrolysis baths in their composition comprise melts of the senary system Na-Al-Ca-Mg-F-O at a temperature of 930–950 °C. Part of the components continuously evaporates from the baths, which shifts the composition of the electrolyte from the optimum point. It causes a technological need for the regular adjustment of the electrolyte composition by fluoride salts. The amount of fluoride salt for addition into the electrolysis bath is calculated from the results of operational analysis of the chemical composition of crystallized electrolyte samples.

The bath chemistry composition or more exactly the bath ratio (BR is wt.  $\text{NaF}/\text{AlF}_3$ ) or cryolite ratio (CR is mol.  $\text{NaF}/\text{AlF}_3$ ,  $\text{CR} = 2\text{BR}$ ) is a principal and fundamental cell parameter to achieve the best performance of electrolytic baths. The CR has a direct impact on the temperature control, alumina solubility, ledge formation and current efficiency. The industrial characterization of the bath electrolyte for process control is generally carried out using two automatic calibrating methods: a combined X-ray diffraction (XRD) analysis of CR and X-ray fluorescence (XRF) analysis of  $\text{CaF}_2$  and  $\text{MgF}_2$ . The technologically required accuracy of the CR analysis is characterized by standard deviation in 0.02 CR units. The XRD method performs quantitative phase analysis (QPA) of the mineral composition of the electrolyte samples, and then the CR is calculated from the found mineral phase concentrations. Appropriate electrolyte reference samples are required in the XRD and



**Fig. 1** Shows a scheme of the system of X-ray process control and correction of electrolyte composition, acting on the aluminium smelters

XRF instrument calibration on crystalline phases and on calcium and magnesium, respectively.

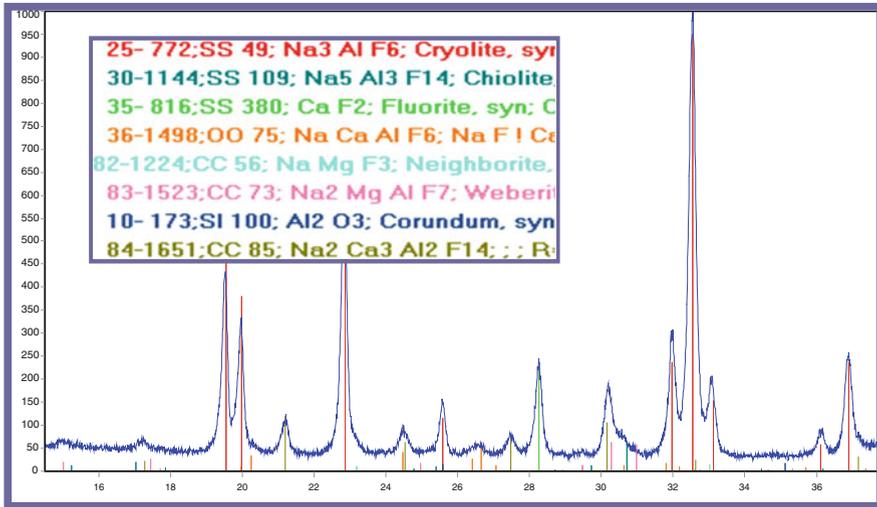
Figure 1 shows a scheme of the system of X-ray process control and correction of electrolyte composition, acting on the aluminium smelters.

Sampling from the cells is performed manually by certain regulations. The preparation of samples for analysis (grinding and fitting of powder in the cell) is performed automatically, along the sample preparation lines. Pressing quality control is carried out visually, and X-ray phase analysis on an automatic diffractometer. As a result of analysis the mass of fluoride compounds to be added in the bath to adjust the composition of the electrolyte to the optimum is calculated. Reproducibility and accuracy of analysis is periodically randomly checked on routine and reference samples, respectively. In a case of unsatisfactory results, the identification of causes and, if necessary, re-calibration of the X-ray equipment is carried out.

Figure 2 shows the XRD full powder pattern and the mineral composition of one of a number electrolyte samples. The colour sticks are XRD reference standards of identified minerals from the PDF ICDD database.

An alternative standardless XRD method is quantitative phase full-profile analysis by the Rietveld method [2]. The estimation of the CR from XRD data by the Rietveld method is possible with the technologically desired accuracy. In addition, the Rietveld method universally applies to baths from various plants that are sampled with different methods and that are characterized by different crystallinity and very diverse chemical compositions. However, the Rietveld method is interactive and time-consuming, and the industrial electrolyte samples analysis from many hundreds of baths should be fully automatic. Even the best attempts to automate the full-profile CR analysis by the Rietveld method has not yet provided satisfactory accuracy [3]. This is due to the need for automatic refinement not only of the profile parameters of X-ray diffraction patterns, but also of the crystal structure parameters of some fluoride electrolyte phases, which can vary in different samples and critically affect the accuracy of the CR determination.

An automatic method for XRD CR electrolyte analysis is proposed in this article. The method is based on an evolutionary hybrid genetic algorithm, which controls the full-profile analysis by the Rietveld method.



**Fig. 2** Shows a XRD full powder pattern and a mineral composition one of a number electrolyte samples. *Colour sticks* are XRD reference standards of identified minerals from data base PDF ICDD [1]

## 2 The Method and Program of Evolutionary Full-Profile Phase Analysis

Evolutionary genetic algorithms simulate the biological processes of natural selection in wildlife and are successfully used in various fields of science and technology [4]. Genetic algorithms are also used in diffraction structural analysis [5, 6] to determine the approximate models of the atomic crystal structure of materials by their X-ray powder diffraction patterns. Crystal structure includes the coordinates of atoms in the symmetrically independent part of the unit cell of crystal material and some additional parameters. This information is accumulated in the Crystal Structure Databases (ICSD, CSD) [7, 8].

The essence of the approach is to generate a random set (population) of trial crystal structures and evolve it using probabilistic formal genetic operations: selection, crossover, mutation, etc.

Evolutionary selection is based on the offspring structure evaluation by fitness function, which represents here the weighted difference between the model profile (calculated from the trial structure) and the experimental diffraction pattern, i.e. one must minimize the profile  $R_{wp}$ -factor of the Rietveld method. The crystal structure model found is refined by the Rietveld method based on the nonlinear least squares. The Rietveld method is also used for the quantitative X-ray analysis of multiphase materials. The possibility of refinement of the actual crystal structure of multiphase material phases makes QPA “structurally sensitive” and thereby greatly increases its accuracy.

A two-level hybrid genetic algorithm (GA) of structural analysis [9] is used mainly to analyse the crystal structure of inorganic substances. This GA performs the evolution of profile and structural parameters of the Rietveld method and controls its refinement by the derivative difference minimizing method (DDM) [10] (an analogue of the Rietveld method).

The DDM method is based on the minimization of difference curve derivatives:

$$MF = \sum w_1 \left( \frac{\partial}{\partial \theta} (Y_o - Y_c) \right)^2 + \sum w_2 \left( \frac{\partial^2}{\partial \theta^2} (Y_o - Y_c) \right)^2 \rightarrow \min, \quad (1)$$

where  $Y_o$  and  $Y_c$  are observed and calculated profile intensities, correspondingly,  $\theta$  is the diffraction angle,  $w$  is the weight coefficient and the summation is fulfilled over the entire XRD powder profile.

The calculated profile is

$$Y_c(\bar{P}, \theta_j) = K \sum_j S_i \sum_i I_{ih}(\bar{P}_{str}) \cdot \Omega_i(\bar{P}_{prof}, \theta_h, \theta_j), \quad (2)$$

where  $K$  is constant;  $S_i$  are scale factors of the calculated diffraction profile for phase  $i$ ;  $I_{ih}$  is a function of the crystal structure parameters for the phase  $i$ ;  $\Omega_i$  is the profile function of diffraction reflexes;  $\bar{P} = (\bar{P}_{prof}, \bar{P}_{str})$  is the vector of the profile and crystal structure parameters.

The DDM method includes a refinement of the profile and crystal structure parameters  $\bar{P}$  of phases by the nonlinear least squares method (LSM). The initial values of the parameters are determined by the hybrid GA.

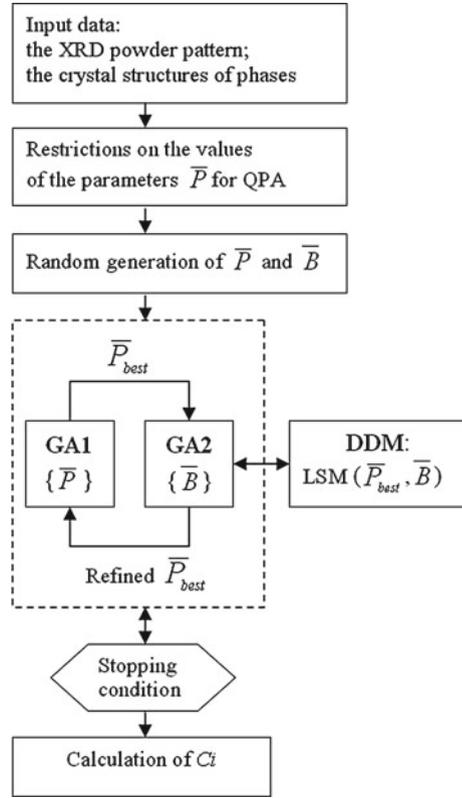
The GA fitness function is the R-factor of the DDM, which represents a numerical derivative of the relative difference between the calculated and experimental powder pattern and is computed in a similar way to the usual Rwp-factor of the Rietveld method.

The authors of [11] have shown that it is possible to perform the automated standardless full-profile quantitative X-ray diffraction analysis on the basis of a two-level hybrid GA with the DDM.

The concept of evolutionary XRD QPA is the searching on the first level of the GA for the initial approximation of a profile and refinable structural parameters within given value ranges and then its refinement by the DDM on the second level of the GA. The QPA feature is that the crude initial values of the parameters can be determined in advance. For example, the atomic coordinates of the crystal structures are taken from the Crystal Structure Databases. Therefore, the search for more accurate initial values by the GA can be performed within narrow ranges of parameter values. The flowchart of the GA is shown in Fig. 3.

The profile parameters include the width of the diffraction reflexes, their shape, etc. Refinable structural parameters include the coordinates of atoms in the common positions of phase crystal lattices. Along with them, the dimensions of crystalline cell

**Fig. 3** The flowchart of the two-level hybrid GA



axes and texture parameters (preferred orientation of particles) are refined, as well as the scale factors  $S_j$  of calculated diffraction profiles of phases in the powder patterns of the material. The listed parameters are binarized and encapsulated in a string, the GA chromosome. Objects of the evolution in GA2 are bit strings  $\bar{B}$ . Each bit set in '1' specifies a corresponding parameter of  $\bar{P}$  to be refined by the DDM on the current generation. The better the refining has been, the higher  $B$ -type fitness is assigned. Thus genetic operations over  $B$ -individuals generate strategies of  $P$ -individuals refinement.

The evolution of the parameters in the iterative execution process on both GA levels provides a selection of good initial approximations for the DDM. Periodic refinement of the best parametric strings by DDM leads to a convergence of any of them to low  $R_{wp}$ -factor values (less than 10%). Then, the optimized scaling factors  $S_j$  of the calculated diffraction profiles of phases are used to calculate the phase concentrations  $C_j$  in the material:

$$C_j = S_j Z_j M_j V_j / \sum_{j=1}^N S_j Z_j M_j V_j \quad (3)$$

where  $V_j$ ,  $Z_j$ ,  $M_j$  are, respectively, the cell volume, the number of formula units per cell and the molecular weight of phase  $j$ ;  $N$  is the number of crystalline phases in the samples.

The XRD QPA procedure by the GA can be divided into three stages.

- Search and refinement of profile parameters, optionally together with the parameters of the anisotropic broadening of lines; decomposition of the diffraction pattern and an initial refinement of profile parameters are performed by the Le Bail algorithm built into DDM.
- Search and refinement of structural parameters together with texture parameters. Search intervals of refinable atomic coordinates are given in the neighbourhood of their positions in the structures, taken from the database.
- Joint refinement of all parameters by DDM under the control of the second level of the GA.

The precision of the method, estimated on test data from the Round Robin on QPA [12], was shown to be 0.45 wt.% per phase.

This method is implemented in a special computer program with a user-friendly interface. Figure 4 shows the form to specify settings for the genetic algorithm and their optimal empirical values. Figure 5 shows the form to specify the profile and the structural parameters of the mineral phases. The refinable parameters are indicated by the green buttons.

The convergence of the genetic algorithm depends on the magnitude of the range of parameter variation. The ranges are determined automatically by the iterative analysis of a set of electrolyte reference samples [13]. Figures 6 and 7 show the

Fig. 4 The form for settings for the genetic algorithm

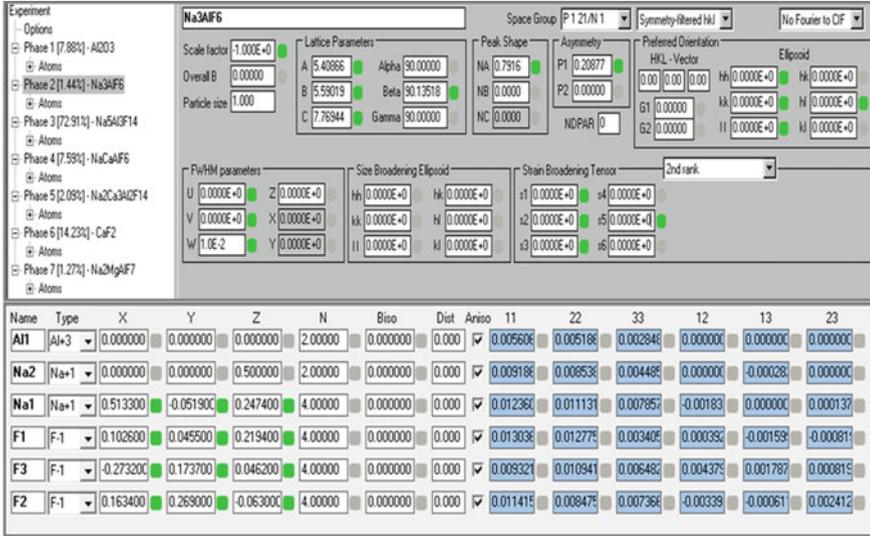


Fig. 5 The form for the profile (top) and structural (bottom) parameters with settings for Cryolite

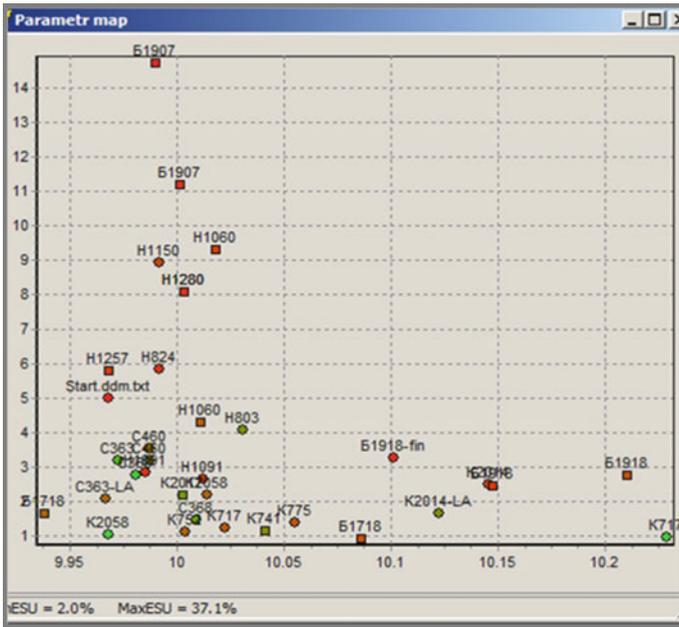


Fig. 6 The map of the distribution of Weberite in the electrolyte reference samples; the horizontal axis is the R-factor, the vertical axis is the concentration

Parameter	Min Value	Max Value	Max - Min	Mean Value
Concentration	0.930	14.740	13.81	6.621
ESU (34)	0.100	0.970	0.87	0.526
ScaleFactor (34)	-1.000E+0	1.683E-6	1.00E+0	-3.736E-2
A (34)	6.957	7.246	0.29	7.059
B (34)	9.938	10.229	0.29	10.013
C (34)	7.181	7.335	0.15	7.284
Alpha (34)	90.000			
Beta (34)	90.000			
Gamma (34)	90.000			
Shape NA (34)	0.32370	1.65800	1.3343	1.14422
Shape NB (3)	7.728	45.160	37.43	18.907
Assym P1 (34)	0.02000	0.38592	0.3659	0.22340
Assym P2 (3)	-0.083	0.057	0.14	-0.026
U (24)	0.01968	1.20600	1.1863	0.29114
V (0)				
W (34)	-0.00577	0.17640	0.1822	0.02931

Fig. 7 The computed interval values of Weberite profile and lattice parameters

forms of the results of the iterative analysis for one of the mineral phases in Table 1 (Weberite).

Figure 6 shows a map of the distribution of Weberite concentrations in the samples. The concentrations vary from 0.93 to 14.74 % mass in difference samples. The point colour shows the accuracy of parameter values (e.s.u.): red for high accuracy and green for low accuracy. The accuracy significantly depends on the concentration. Figure 7 shows the computed interval values of profile and lattice parameters for this phase. The intervals can be reduced by eliminating some samples with low phase concentrations. Intervals of varying atomic coordinates and other structural parameters are calculated similarly. The calculated intervals may be used by the GA1 to analyse bath electrolyte samples.

### 3 Accuracy Evaluation of the CR Analysis and Discussion

The method of the evolutionary full-profile QPA can be configured for an automatic analysis of multiple samples with the same type of phase composition as a result of

**Table 1** The phase composition of the electrolyte samples

#	Phases	Chem. formula	Fraction (% mass)	CR area
1.	Cryolite	Na <sub>3</sub> AlF <sub>6</sub>	0–90	>1.67
2.	Chiolite	Na <sub>5</sub> Al <sub>3</sub> F <sub>14</sub>	0–85	<3.0
3.	Fluorite	CaF <sub>2</sub>	0–9	>2.45
4.	Ca-cryolite 1	NaCaAlF <sub>6</sub>	0–15	<3.0
5.	Ca-cryolite 2	Na <sub>2</sub> Ca <sub>3</sub> Al <sub>2</sub> F <sub>14</sub>	0–20	<2.95
6.	Weberite	Na <sub>2</sub> MgAlF <sub>7</sub>	0–15	<2.85
7.	Neiborite	NaMgF <sub>3</sub>	0–6	>2.5
8.	Sodium fluoride	NaF	0–5	>3.0
9.	$\alpha$ -, $\beta$ -, $\gamma$ -alumina	Al <sub>2</sub> O <sub>3</sub>	2–5	

the selection of relevant search intervals and settings of genetic operations. Industrial electrolytic samples of aluminium production are suitable for this purpose. The phase composition of the electrolyte samples is shown in Table 1, and the crystal structure of these phases are taken from the database ICSD (Inorganic Crystal Structure Database).

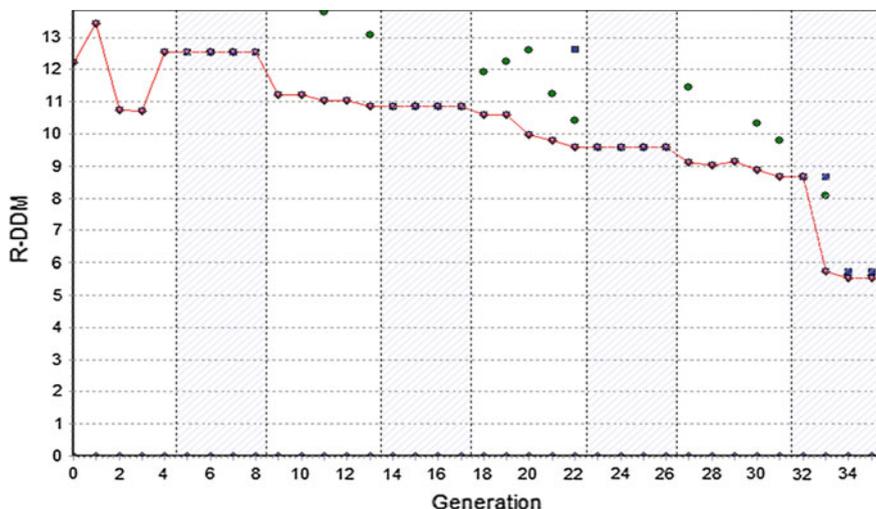
The high-quality industry reference samples (IRS) of chemical and phase composition [13] prepared directly from the industrial electrolyte samples of four large aluminium smelters, containing calcium and magnesium, were used for testing and accuracy evaluation of the method. The search intervals and settings of genetic operations for the automatic analysis of the profile, which were universal for this task, as well as the structural parameters of fluoride phases have been identified in special experiments of preliminary studies. CR values calculation from the found phase concentrations were fulfilled according to the formula:

$$CR = \frac{\text{total NaF, mol}}{\text{total AlF}_3, \text{ mol}} = 2 \cdot \frac{\sum_i \alpha_i C_i}{\sum_i \beta_i C_i} \quad (4)$$

where  $C_i$  is the concentration of the  $i$ th phase (% mass), calculated according to (1);  $\alpha_i$ ,  $\beta_i$  are the mass fractions of NaF and AlF<sub>3</sub>, respectively.

An analysis of 26 IRS was conducted. Figure 8 shows an example of a plot of the convergence in the evolutionary full-profile QPA for one of them.

On the 1st level of the GA, the profile and some structural parameter values are searched for in the specified search range including atomic coordinates, allowing variation in a given neighbourhood of its positions in the structures of fluoride phases taken from the structural data base ICSD, as well as texture parameters of the phases. Level 2 of the GA controls the gradual full-profile DDM-based refinement of the best current solutions found on the 1st level of the GA. The sizes of test solution



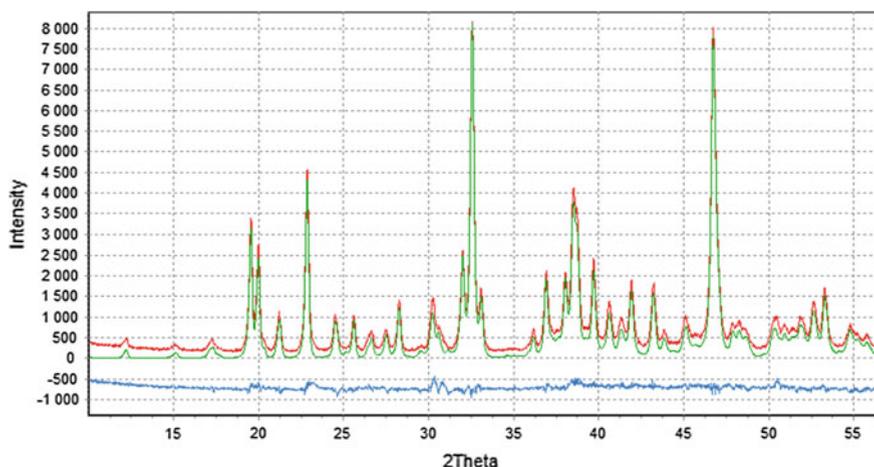
**Fig. 8** The plot of convergence in the evolutionary XRD QPA from analysis of sample H803 from IRS; the x-axis depicts the evolution generation number, the y-axis depicts the profile  $R_{wp}$ -factor; *dotted vertical lines* show the time of the shift from the level 1 GA to the level 2 GA and back; the *red line* is the current best fitness value, *blue dots* show the current worst fitness value, *green dots* show an average fitness value for the current population

populations were: 30 for the 1st level of the GA and 10 for the 2nd level of the GA. The alternation of GA levels brings the iterative search process to the true solution with  $R_{wp} = 5.6\%$  after 37 generations of evolution (in this example).

On the first three GA1 cycles, the best solution in the population of the parametric strings gradually improved and the R-factor decreased from 13.7 to 9.7%. However, a good approximation for the local search has not yet been found and the refinement with DDM cannot reduce the  $R_{wp}$  value (even-numbered bands in Fig. 8). Finally, on the 4th GA1 cycle, the R-factor dropped less (from 9.7 to 8.8% in generations 27–31) and the approximation was found which could be refined with the local search. This approximation was refined with DDM through the choosing of various parameter combinations by the GA2 within the last 4 generations. After refinement, the R-factor decreased from 8.8 to 5.6% and then was stabilized.

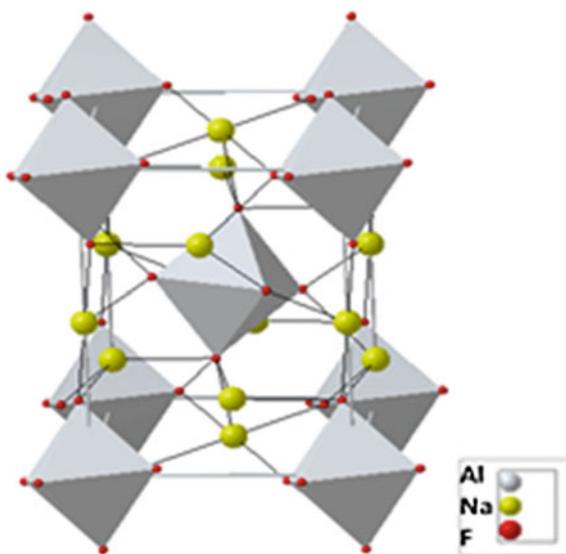
After the final refinement of all parameters by DDM, the  $R_{wp}$  was equal to 5.3% and the phase concentrations calculated by (1) were used for the CR calculation according to (2). The comparison of the experimental and calculated final powder patterns is shown in Fig. 9.

A feature of the method is the refinement of the crystal structure of the electrolyte main phases. The crystal structure of cryolite, chiolite and calcium cryolite may vary slightly from sample to sample. This is caused by the nonequilibrium crystallization of sample dew, its sampling from the bath and variable sampling factors such as the electrolyte composition and temperature, mold mass and temperature, weight of the sample, time since loading the bath with alumina, etc. Figure 10 shows the



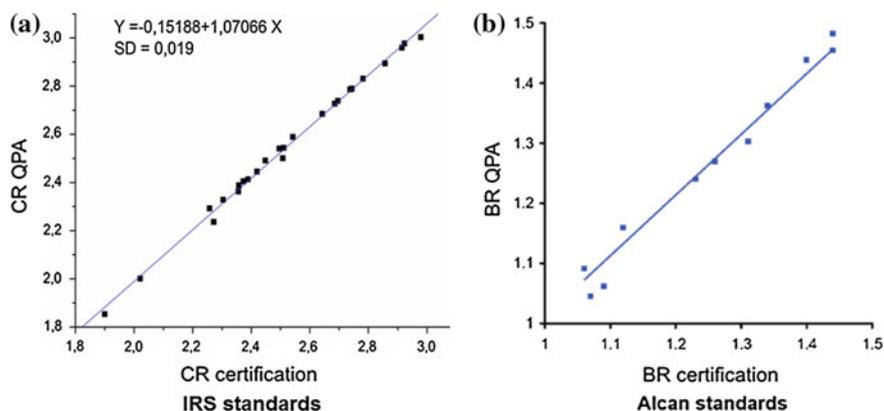
**Fig. 9** The experimental (*red*) and calculated (*green*) diffraction patterns for reference electrolyte sample H803 and the difference between them (*blue*, contains background from an amorphous phase of alumina); radiation with CuK $\alpha$  wavelength;  $R_{wp}$ -factor = 5.3 %

**Fig. 10** Structure of cryolite and the main differences in the samples C363 and C460, respectively: inclination angle of octahedras axis relative to the *c*-axis: 18.69° and 19.51°; edges and the angle of parallelogram in the base of the octahedras: 2.484 Å, 2.727 Å, 88.50° and 2.513 Å, 2.569 Å, 89.04°; Na2–Na1 distance: 3.249 and 3.188 Å



atomic crystal structure of cryolite and identifies its main differences in two reference samples (C363 and C460).

The main structural variations are in the change of the inclination angle and the geometric dimensions of  $[AlF_6]^{-3}$  octahedras, and in the corresponding shifts of Na atoms, located in the common position for (Na2). This is reflected in the variation of the lattice parameters, in particular, with the *c*-axis size and the value of monoclinic



**Fig. 11** Regression charts for the estimation of the full-profile analysis accuracy: **a** for CR values, calculated by the data of evolutionary XRD QPA, to the certified CR values for electrolyte IRS, **b** for BR values ( $BR = CR/2$ ), calculated by the automatic Rietveld analysis, to the certified BR values for electrolyte reference samples (Alcan International Ltd, 2005) [14]

angle  $\beta$ , same as its intensity value. The difference in lattice parameters leads to a perceptible shift of the analytical diffraction reflexes of cryolite to about  $2\Theta \sim 0.02^\circ - 0.03^\circ$ , and the difference in the coordinates of the atoms leads to a relative change in the integrated intensity of the reflexes of up to 4%. Neglect of these distortions of the crystal structure significantly affects the accuracy of the electrolyte analysis and the corresponding errors of the CR determination are about 0.01–0.02.

The left side of Fig. 11 shows the regression graph of the corresponding calculated and certified CR values for a set with 26 IRS of industrial electrolyte. The calculated CR values correspond to the certified CR values with precision, characterized by a standard deviation (SD) in 0.019 CR units. The achieved accuracy completely satisfies the technological requirements of the production.

The advantages of determining the CR by the evolutionary XRD QPA method of relatively automatic full-profile CR analysis by the Rietveld method can be estimated by comparing the results with the data of [15]. It should be remembered that many of the aluminium factories, instead of the cryolite ratio, apply its counterpart—the bath ratio (BR), which is related as  $BR = CR/2$ .

The right side of Fig. 11 shows the regression graph of the corresponding BR values, calculated according to the automatic full-profile BR analysis by the Rietveld method, to the certified values for the standard electrolyte samples of the company ALCAN International Ltd [14], taken from [15]. Unfortunately, the standard deviation value for the BR graph in Fig. 11 in [15] was not provided. To compare the graphs for BR and CR on the same scale, all values along each of the axes on the right graph should be doubled. After performing this procedure it becomes apparent that the standard deviation of the evolutionary XRD QPA is at least twice as good.

## 4 Conclusions

The increased accuracy of the analysis by the evolutionary QPA method can be explained through the fact that the genetic algorithm searches various variants of the approximate values of profile and structural parameters (i.e. adapts them to a particular sample) as well as periodically refining the best ones on the 2nd level of the GA by the DDM—full-profile analysis method. However, a variant of full-profile analysis by the Rietveld method automatically specifies the pre-selected initial values (i.e. does not adapt to a particular sample) of the profile and structural parameters.

The disadvantage of the current version of the evolutionary QPA method is that the amount of computational time is significantly higher than for the existing calibration XRD methods of CR process control. However, the development and optimization of the evolutionary method certainly will increase the processing speed of CR full-profile analysis and bring it to a comfortable level. Thus, this approach opens up new prospects for electrolyte CR automatic standardless determination by XRD data for chemical control on aluminium smelting baths.

## References

1. PDF: Powder Diffraction File, Inorganic Database. ICDD. <http://www.icdd.com>
2. Young, R.A.: The Rietveld Method. Oxford University Press, New York (1993)
3. Feret, F.R.: Breakthrough in analysis of electrolytic bath using Rietveld-XRD method. In: Light Metals, pp. 343–346 (2008)
4. Paszkowicz, W.: Genetic Algorithms, a nature-inspired tool: a survey of applications in materials science and related fields: part II. In: Materials and Manufacturing Processes, vol. 28, Issue 7 (Genetic Algorithms), pp. 708–725 (2013)
5. Feng, Z.J., Dong, C.: GEST: a program for structure determination from powder diffraction data using a genetic algorithm. *J. Appl. Cryst.* **40**, 583 (2007)
6. Kenneth, D., Harris, M.: Structure solution from powder X-Ray diffraction data by genetic algorithm techniques, applied to organic materials generated as polycrystalline products from solid state processes. *Mater. Manuf. Process.* **24**, 293–302 (2009)
7. ICSD: Inorganic Crystal Structure Database. FIZ Karlsruhe. <http://www.fiz-karlsruhe.de/icسد.html>
8. CSD: Cambridge Structural Database. Cambridge Crystallographic Data Centre. <http://www.ccdc.cam.ac.uk/products/csd/>
9. Yakimov, Y.I., Semenkin, E.S., Yakimov, I.S.: Two-level genetic algorithm for a full profile fitting of X-ray powder patterns. *Z. Kristallogr. Suppl.* **30**, 21–26 (2009)
10. Solovoyov, L.A.: The Derivative Difference Minimization Method. Chapter 10. *Powder Diffraction Theory and Practice*. In: Dinnebier, R.E. Billinge, S.J.L. (eds.) Royal Society of Chemistry, p. 507(2008)
11. Yakimov, I.S., Zaloga, A.N., Solov'ev, L.A., Yakimov, Y.I.: Method of evolutionary structure sensitive quantitative X-ray phase analysis of multiphase polycrystalline materials. *Inorg. Mater.* **48**(14), 1285–1290 (2012)
12. Nicola, V.Y., Scarlett et al.: Round robin on quantitative phase analysis: samples 2. *J. Appl. Cryst.* **35**, 383–400 (2002)
13. Yakimov, I.S., et al.: Developing industry standard samples of electrolyte aluminum electrolytic cells. *Stand. Samples* **4**, 34–42 (2008)

14. Electrolytic Bath Standards. Alcan International Ltd., Quebec, Canada (2005)
15. Karsten Knorr Present progress in fast XRD analysis applying the Rietveld method for bath control. In: The 19th International Symposium and Exhibition of ICSOBA (2012)

**Part III**  
**Robotics and Automation**

# Improving the 3D Positioning for Low Cost Mobile Robots

Rafael Socas, Sebastian Dormido, Raquel Dormido  
and Ernesto Fabregas

**Abstract** A new algorithm to improve the 3D positioning for low cost mobile robots is presented. The core of the algorithm is based on a Finite State Machine (FSM) which estimates the 3D position and orientation of the robots, also a low pass filter and a threshold calculator are used in the system to filter and to estimate the noise in the sensors. The system sets dynamically the parameters of the algorithm and makes them independent of the noise. The algorithm has been tested with differential wheel drive robots, however it can be used with other different types of robots in a simple way. To improve the accuracy of the estimations, a new reference system based on the accelerometer of the robot is presented which reduces the accumulative error that the odometry produces.

**Keywords** 3D positioning · Mobile robots · Sensors

## 1 Introduction

In mobile robots applications a good positioning and orientation estimation are crucial [1]. Different methods can be used to solve this problem. The most common techniques can be divided in seven categories [2, 3]: 1. Odometry; 2. Inertial navigation; 3. Magnetic compasses; 4. Active beacon; 5. Global Positioning Systems; 6. Landmark navigation; and 7. Model matching. At the same time, they can be

---

R. Socas (✉) · S. Dormido · R. Dormido · E. Fabregas  
Department of Computer Science and Automatic Control, UNED University,  
28040 Madrid, Spain  
e-mail: rsocas@telefonica.net

S. Dormido  
e-mail: sdormido@dia.uned.es

R. Dormido  
e-mail: raquel@dia.uned.es

E. Fabregas  
e-mail: efabregas@bec.uned.es

categorized into two groups based on the position measurements: relative, (also called dead-reckoning which includes the categories 1 and 2) and absolute (which includes the rest of the categories). Many applications usually combine two of them, one of each group to compensate the lacks of a single method.

The low cost robots frequently have few sensors. They typically have wheel encoders, accelerometers and obstacle detectors [4–6]. For this reason, in this kind of platforms, the techniques that can only be applied are Odometry and Inertial Navigation. Although both techniques produce accumulative errors they provide a good short-term accuracy. The odometry measures the distance that each wheel of the robot has travelled over the time. With this information, the position and the orientation of the robot can be obtained. On the other hand, inertial navigation uses accelerometers and gyroscopes to measure the acceleration and the rate of rotation of the robot. These measurements have to be integrated to obtain the position and the orientation by inertial navigation techniques. However, the cost of the gyroscopes constrains on the environments in which they are practical for use. In low cost robot the techniques that can be used to estimate the position are odometry and inertial navigation with accelerometers.

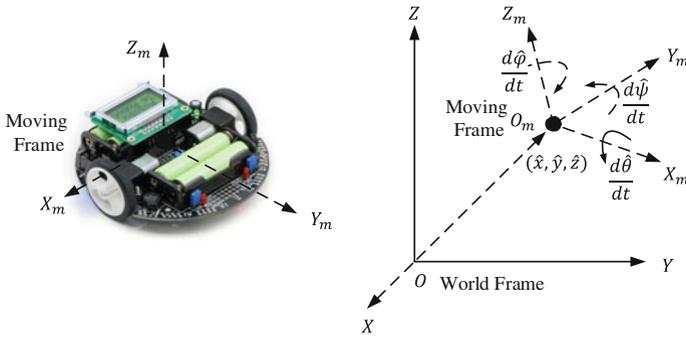
In general way, low cost accelerometers have poor signal to noise ratio when the robot has small accelerations (common situation in many applications, for example in robots with constant speed), for this reason, the positioning and orientation estimation by accelerometers in low cost platforms is a bad solution [7]. On the other hand, the accelerometers have a good performance in tilt angles estimation [8, 9].

In this paper a new 3D positioning algorithm for low cost robots is proposed, it combines odometry and tilt estimation to obtain the position and the orientation in 3D. Also, the tilt estimation is used as a reference system to reduce the orientation error that the classical odometry produces. The paper is organized as follows. Section 2 presents the 3D positioning in mobile robots applications. The proposed algorithm is presented in Sect. 3. The experimental results are analysed in Sect. 4. And finally, the conclusions and future work are presented in Sect. 5.

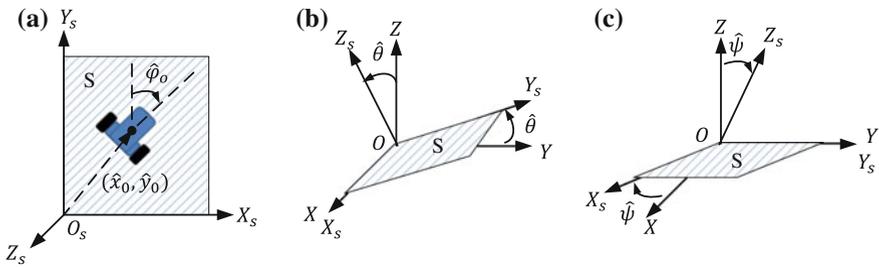
## 2 3D Positioning and Orientation

The estimation of the position and orientation of mobile robots is one of the basic preconditions for their autonomy. The aim of this work is to estimate the position and orientation of the robot in 3D. Both estimations can be defined as a vector with six components  $\mathbf{p}o = \{\hat{x}, \hat{y}, \hat{z}, \hat{\varphi}, \hat{\theta}, \hat{\psi}\}$  [10]. The position  $(\hat{x}, \hat{y}, \hat{z})$  and the orientation  $(\hat{\varphi}, \hat{\theta}, \hat{\psi})$  are defined with respect to the world frame ( $O$ ) as it is shown in Fig. 1. The elevation  $\hat{\theta}$ , the bank  $\hat{\psi}$  and the heading  $\hat{\varphi}$  are the Tait-Bryan angles. Finally, on the robot a moving frame ( $O_m$ ) is defined.

When the robot is travelling on a flat surface  $S$ , a new reference frame can be defined  $O_s$  (surface frame). Via odometry the position  $(\hat{x}_o, \hat{y}_o)$  and the orientation  $\hat{\varphi}_o$  with respect to the surface frame can be calculated. A tilt estimation based on an accelerometer is used to obtain the elevation  $\hat{\theta}$  and the bank  $\hat{\psi}$  angles of the surface  $S$  with respect to the world frame (see Fig. 2).



**Fig. 1** Position  $(\hat{x}, \hat{y}, \hat{z})$  and orientation  $(\hat{\varphi}, \hat{\theta}, \hat{\psi})$  with respect to the world frame ( $O$ )



**Fig. 2** **a** Position  $(\hat{x}_o, \hat{y}_o)$  and orientation  $(\hat{\varphi}_o)$  with respect to the surface frame ( $O_s$ ). **b** Elevation angle  $(\hat{\theta})$ . **c** Bank angle  $(\hat{\psi})$

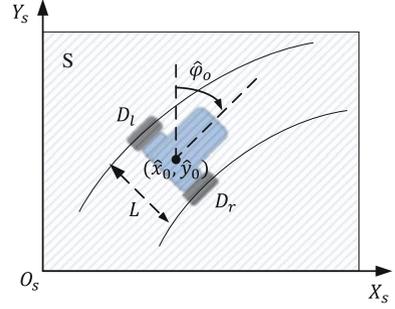
With the parameters  $\hat{x}_o, \hat{y}_o, \hat{\varphi}_o, \hat{\theta}$  and  $\hat{\psi}$  the components of the vector  $\mathbf{p}_o$  can be calculated. The methodology to estimate it will be explained in the following sections.

### 2.1 Position and Orientation Estimation via Odometry

Two differential wheels robots are the most common low cost mobile robots. These systems usually have wheel encoders, an accelerometer and other sensors to avoid obstacles. When the robot is travelling on a flat surface its position and orientation can be estimated using odometry [11, 12]. Wheel encoders allow to measure the distances that each wheel has travelled ( $D_l$  for the left wheel and  $D_r$  for the right wheel). With  $D_l, D_r$  and the distance between the two wheels  $L$  (see Fig. 3), the position and the orientation of the robot can be calculated in discrete-time domain using the Eqs. (1)–(3).

$$\hat{\varphi}_o[n] = \hat{\varphi}_o[n - 1] + \frac{D_l[n] - D_r[n]}{L} . \tag{1}$$

**Fig. 3** Position  $(\hat{x}_o, \hat{y}_o)$  and orientation  $(\hat{\varphi}_o)$  with respect to the surface frame  $(O_s)$



$$\hat{x}_o[n] = \hat{x}_o[n-1] + \left( \frac{D_l[n] + D_r[n]}{2} \right) \sin(\hat{\varphi}_o[n]) . \quad (2)$$

$$\hat{y}_o[n] = \hat{y}_o[n-1] + \left( \frac{D_l[n] + D_r[n]}{2} \right) \cos(\hat{\varphi}_o[n]) . \quad (3)$$

where

$D_l[n]$ : Distance travelled by left wheel.

$D_r[n]$ : Distance travelled by right wheel.

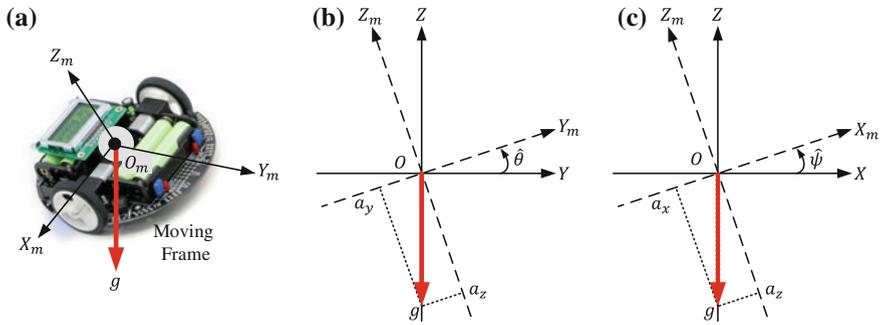
$L$ : Distance between the two wheels.

## 2.2 Tilt Estimation

The accelerometer is sensitive to the total acceleration of the mobile robot. It is composed of the inertial acceleration, the gravity field  $g$ , the centripetal and the tangential acceleration. This sensor measures the three components  $(a_x, a_y, a_z)$  of the total acceleration with respect to the moving frame  $(O_m)$ . In the two differential wheel drive robots considered in this paper, the gravity field is the most important component of the acceleration, for this reason, the rest of them are not considered.

On the other hand, the robot is made to drive three possible rotations: the elevation  $\hat{\theta}$ , the bank  $\hat{\psi}$  and the heading  $\hat{\varphi}$  angles with respect to the world frame  $(O)$ . With this assumption, the sensor in the robot is measuring the gravity field  $g$  (because the movements of the robot do not produce other accelerations or they are irrelevant) with respect to the moving frame on the robot (see Fig. 4). In the world frame the acceleration is always the same  $(0, 0, -g)$ , on the other hand, the relationship between the accelerations in the two frames and the rotation angles is defined by (4).

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \mathbf{R}(\hat{\theta}, \hat{\psi}, \hat{\varphi}) \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} . \quad (4)$$



**Fig. 4** **a** Gravity field with respect to the moving frame, the components  $(a_x, a_y, a_z)$  are the measurements of the accelerometer. **b** Robot with elevation angle  $\hat{\theta}$ . **c** Robot with a bank angle  $\hat{\psi}$

The rotation matrix  $\mathbf{R}(\hat{\theta}, \hat{\psi}, \hat{\phi})$  depends on the order of rotations. Therefore to calculate the angles  $\hat{\theta}$ ,  $\hat{\psi}$  and  $\hat{\phi}$  it is necessary to know the order of rotation. Considering a robot that rotates first an angle  $\hat{\theta}$  and then an angle  $\hat{\psi}$ , the measurements in the accelerometer are described by (5).

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} -g \cos(\hat{\theta}) \sin(\hat{\psi}) \\ -g \sin(\hat{\theta}) \\ -g \cos(\hat{\theta}) \cos(\hat{\psi}) \end{pmatrix}. \quad (5)$$

If the robot rotates first an angle  $\hat{\psi}$  and then an angle  $\hat{\theta}$ , the accelerometer measurements are given by (6).

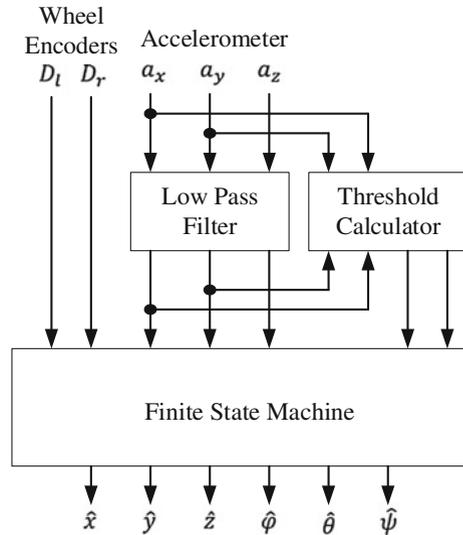
$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} -g \sin(\hat{\psi}) \\ -g \cos(\hat{\psi}) \sin(\hat{\theta}) \\ -g \cos(\hat{\psi}) \cos(\hat{\theta}) \end{pmatrix}. \quad (6)$$

Comparing (5) and (6) the components  $a_x$  and  $a_y$  are different in each equation, and therefore, we can conclude that the problem is undetermined. If the robot has gyroscopes, the order of rotations can be estimated and consequently the rotation angles can be calculated. As we mentioned before, low cost robots frequently have an accelerometer but they do not have gyroscopes. For this reason, in a general way in this kind of robots the elevation  $\hat{\theta}$  and the bank  $\hat{\psi}$  angle can not be estimated.

### 3 Algorithm Architecture

The limitations of the low cost robots to estimate the elevation and the bank angles has been analysed in Sect. 2. To avoid these problems in this paper we assume that:

**Fig. 5** Algorithm architecture



- (a) The two-wheel differential robot can travel only on a flat surface and,
- (b) this surface might have an elevation angle or a bank angle, but not both angles at the same time.

With the previous assumptions, to estimate the positioning in 3D and the orientation of the robot, the discrete-time architecture depicted in Fig. 5 has been proposed.

As it is shown in the Fig. 5, the algorithm is composed of the following signals and blocks:

- (a) Input signals,  $D_l[n]$  and  $D_r[n]$  directly reading from wheel encoders, and  $a_x[n]$ ,  $a_y[n]$  and  $a_z[n]$  from the accelerometer.
- (b) A low pass filter to remove the noise in the sensor's signal. The outputs of this blocks are  $a_{xf}[n]$ ,  $a_{yf}[n]$  and  $a_{zf}[n]$ .
- (c) A threshold calculator which obtains the threshold values ( $t_{hx}[n]$  and  $t_{hy}[n]$ ), these values will be used in the Finite State Machine (FSM) to define the changes between states.
- (d) The FSM which estimates the position and orientation parameters.
- (e) And finally, the output signals ( $\hat{x}[n]$ ,  $\hat{y}[n]$ ,  $\hat{z}[n]$ ,  $\hat{\phi}[n]$ ,  $\hat{\theta}[n]$ ,  $\hat{\psi}[n]$ ).

### 3.1 Low Pass Filter

Low pass filtering of the signals from the accelerometer is a good way to remove the noise (both mechanical and electrical) [13]. In this work, an IIR filter has been proposed (7).

$$a_{if}[n] = (1 - \alpha) a_i[n] + \alpha a_{if}[n - 1] \quad \text{for } i = \{x, y, z\} . \quad (7)$$

The inputs are the accelerometer's signal  $(a_x[n], a_y[n], a_z[n])$  and the outputs are the filtered signals  $(a_{xf}[n], a_{yf}[n], a_{zf}[n])$ . The parameter  $\alpha$  sets the cut off frequency and the group delay of the filter. In the design process,  $\alpha$  has to be selected to obtain a good balance between the noise reduction and the delay in the acceleration measurements.

### 3.2 Threshold Calculator

As we mention before, the positioning algorithm is based on a Finite State Machine. The states, as they will be explained in the next section, depend on the values of the elevation angle  $\hat{\theta}[n]$  and the bank angle  $\hat{\psi}[n]$ . When the robot is on a surface with  $\hat{\theta}[n] \approx 0$  or  $\hat{\psi}[n] \approx 0$ , the noise in the accelerometer sensor could produce changes between states and the system could be unstable. For this reason, the algorithm needs a threshold system to control the changes between the states taking in account the noise in the sensors.

As it will be explained later, when the elevation and the bank angle are close to zero ( $\hat{\theta} \approx 0$ ,  $\hat{\psi} \approx 0$ ), the values of the accelerations  $a_{xf}[n]$  and  $a_{yf}[n]$  should be zero except by the noise in the sensor. For this reason, the noise in these measurements will be considered in the threshold calculator.

The noise  $n_x[n]$  and  $n_y[n]$  in the acceleration signals  $a_x[n]$  and  $a_y[n]$  can be estimated by the difference between the acceleration signals and their filtered signals as it is shown in (8) and (9).

$$n_x[n] = a_x[n] - a_{xf}[n] . \quad (8)$$

$$n_y[n] = a_y[n] - a_{yf}[n] . \quad (9)$$

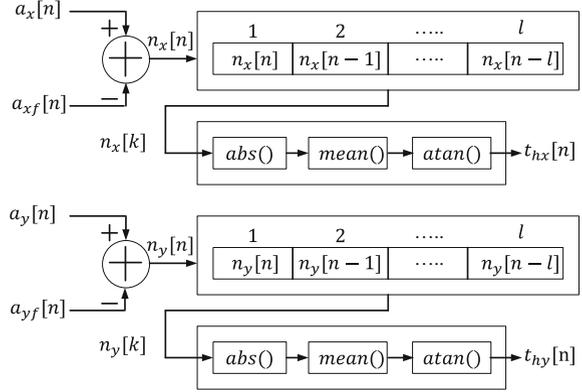
Then a simple moving average has been used to obtain a stable value of the noise, then the function  $atan()$  is applied to calculate the threshold angles by (10) and (11).

$$t_{hx}[n] = atan\left(\frac{1}{l} \sum_{k=1}^l abs(n_x[k])\right) . \quad (10)$$

$$t_{hy}[n] = atan\left(\frac{1}{l} \sum_{k=1}^l abs(n_y[k])\right) . \quad (11)$$

In this way, the proposed system obtains two dynamic threshold values ( $t_{hx}[n]$  and  $t_{hy}[n]$ ) one for each acceleration signal. Moreover, it has a degree of freedom, the parameter  $l$ , which will be set depending on how the level of noise changes over the time. The proposed threshold calculator is depicted in Fig. 6.

**Fig. 6** Threshold calculator. The threshold values are obtained as a moving average of the noise in the sensor



### 3.3 Finite State Machine

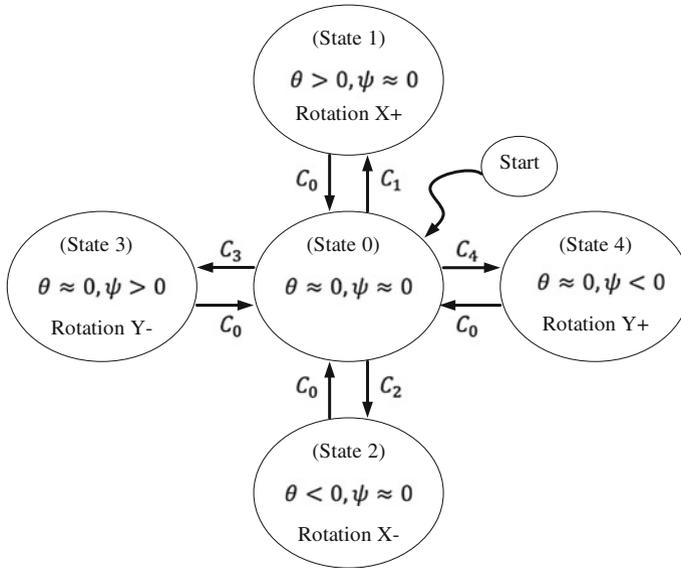
A Finite State Machine (FSM) has been included in the architecture to estimate the position in 3D of the robot (see Fig. 5). The inputs are the wheel encoders, the filtered accelerations and the threshold values. The outputs are the estimation of the position ( $\hat{x}[n]$ ,  $\hat{y}[n]$ ,  $\hat{z}[n]$ ) and the orientation ( $\hat{\varphi}[n]$ ,  $\hat{\theta}[n]$ ,  $\hat{\psi}[n]$ ) of the robot. As we mentioned before, the robot only can travel on a flat surface and this surface could have an elevation angle or a bank angle with respect to the world frame, but it can not have two rotations at the same time. Taking into account these considerations, in this section the behaviour of the FSM have been developed.

As it is shown in the Fig. 7, the FSM has five states and five conditions to change between the states. The meaning of each state is presented in the following definitions:

- (a) State 0: The robot is on a surface without elevation angle ( $\hat{\theta} \approx 0$ ) and without bank angle ( $\hat{\psi} \approx 0$ ) with respect to the world frame.
- (b) State 1: The robot is on a surface with a positive elevation angle ( $\hat{\theta} > 0$ ) and without bank angle ( $\hat{\psi} \approx 0$ ).
- (c) State 2: The robot is on a surface with a negative elevation angle ( $\hat{\theta} < 0$ ) and without bank angle ( $\hat{\psi} \approx 0$ ).
- (d) State 3: The robot is on a surface with a positive bank angle ( $\hat{\psi} > 0$ ) and without elevation angle ( $\hat{\theta} \approx 0$ ).
- (e) State 4: The robot is on a surface with a negative bank angle ( $\hat{\psi} < 0$ ) and without elevation angle ( $\hat{\theta} \approx 0$ ).

On the other hand, we assume that the system works in discrete-time, in each sample time the measurements of the sensor ( $D_l[n]$ ,  $D_r[n]$ ,  $a_{xf}[n]$ ,  $a_{yf}[n]$ ,  $a_{zf}[n]$ ) are available.

The conditions ( $C_0$ ,  $C_1$ ,  $C_2$ ,  $C_3$ ,  $C_4$ ) which control the changes between states are defined as function of the components of the acceleration ( $a_{xf}[n]$ ,  $a_{yf}[n]$ ,  $a_{zf}[n]$ ), the threshold values ( $t_{hx}[n]$  and  $t_{hy}[n]$ ) and two positive guard parameters ( $H$  and  $h$ ) as it is presented in the following Eqs. (12)–(18):



**Fig. 7** The FSM has five states (State 0, 1, 2, 3 and 4) and five transition conditions ( $C_0$ ,  $C_1$ ,  $C_2$ ,  $C_3$  and  $C_4$ )

$$\theta_{ilt} = \text{atan} (a_{yf}[n]/a_{zf}[n]) . \quad (12)$$

$$\psi_{ilt} = \text{atan} (a_{xf}[n]/a_{zf}[n]) . \quad (13)$$

$$C_0 : (\text{abs}(\theta_{ilt}) \leq t_{hy}[n] + h) \& (\text{abs}(\psi_{ilt}) \leq t_{hx}[n] + h) . \quad (14)$$

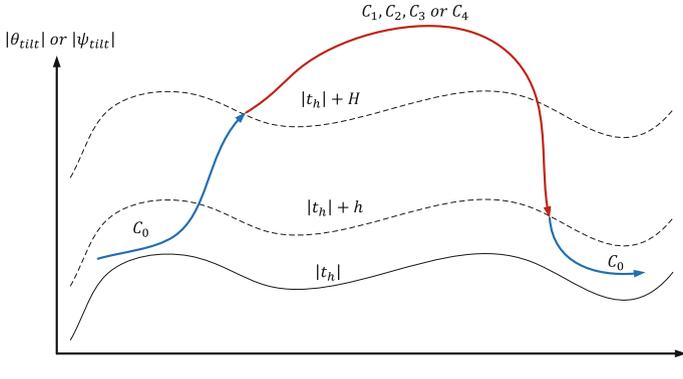
$$C_1 : \theta_{ilt} > t_{hy}[n] + H . \quad (15)$$

$$C_2 : \theta_{ilt} < -t_{hy}[n] - H . \quad (16)$$

$$C_3 : \psi_{ilt} > t_{hx}[n] + H . \quad (17)$$

$$C_4 : \psi_{ilt} < -t_{hx}[n] - H . \quad (18)$$

The parameters  $H$  and  $h$  create two guard bands around the threshold values. These guard bands make a new threshold value ( $t_{hy}[n] + H$  or  $t_{hx}[n] + H$ ) when the elevation and bank angles increase and another one ( $t_{hy}[n] + h$  or  $t_{hx}[n] + h$ ) when they decrease. This method avoids instabilities when  $\theta_{ilt} \approx t_{hy}[n]$  or  $\psi_{ilt} \approx t_{hx}[n]$ . In the Fig. 8 the meaning of the guard bands are presented. The parameters  $H$  and  $h$  have to be set taking in account the level of noise in the sensor.



**Fig. 8** Relationship between the guard bands ( $H$ ,  $h$ ), the angles ( $\theta_{tilt}$ ,  $\psi_{tilt}$ ) and the conditions ( $C_0$ ,  $C_1$ ,  $C_2$ ,  $C_3$ ,  $C_4$ ) in the FSM

### 3.4 Position and Orientation Estimation

In each state of the FSM an estimation of the 3D positioning has to be calculated. In the state 0 the classical odometry has been applied. For the rest of the states a combination of tilt estimation and odometry have been used. The aim of the 3D positioning estimation is to calculate the position vector of the robot ( $\mathbf{po}[n] = \{\hat{x}[n], \hat{y}[n], \hat{z}[n], \hat{\phi}[n], \hat{\theta}[n], \hat{\psi}[n]\}$ ). Depending on the state where the robot is, different equations have to be applied. These equations are defined as follow.

**Estimation in State 0.** In state 0, due to the tilt is zero, the classical odometry has to applied as it is presented in (19)–(24).

$$\hat{\phi}[n] = \hat{\phi}[n - 1] + \hat{\phi}_o[n] . \quad (19)$$

$$\hat{x}[n] = \hat{x}[n - 1] + D_c[n] \sin(\hat{\phi}[n]) . \quad (20)$$

$$\hat{y}[n] = \hat{y}[n - 1] + D_c[n] \cos(\hat{\phi}[n]) . \quad (21)$$

$$\hat{z}[n] = \hat{z}[n - 1] . \quad (22)$$

$$\hat{\theta}[n] = 0 . \quad (23)$$

$$\hat{\psi}[n] = 0 . \quad (24)$$

where

$\hat{\phi}_o[n]$ : Heading estimation by odometry  $\frac{D_l[n] - D_r[n]}{L}$  .

$D_c[n]$ : Mean distance  $D_c[n] = \frac{D_l[n] + D_r[n]}{2}$  .

$L$ : Distance between the two wheels.

**Estimation in State 1 and 2.** The elevation angle  $\hat{\theta}[n]$  can be calculated in state 1 by (25) and in state 2 by (26).

$$\hat{\theta}[n] = \text{atan} \left( \frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{-a_{zf}[n]} \right). \quad (25)$$

$$\hat{\theta}[n] = \text{atan} \left( \frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{a_{zf}[n]} \right). \quad (26)$$

In both states the components  $\hat{\varphi}[n]$ ,  $\hat{x}[n]$ ,  $\hat{y}[n]$  and  $\hat{\psi}[n]$  have the same expressions (27)–(30)

$$\hat{\varphi}[n] = \hat{\varphi}[n - 1] + \hat{\varphi}_o[n]. \quad (27)$$

$$\hat{x}[n] = \hat{x}[n - 1] + D_c[n] \sin(\hat{\varphi}[n]). \quad (28)$$

$$\hat{y}[n] = \hat{y}[n - 1] + D_c[n] \cos(\hat{\varphi}[n]) \cos(\theta_a). \quad (29)$$

$$\hat{\psi}[n] = 0. \quad (30)$$

where  $\theta_a = \text{abs}(\hat{\theta}[n])$ .

The component  $\hat{z}[n]$  can be calculated in state 1 by (31) and in state 2 by (32).

$$\hat{z}[n] = \hat{z}[n - 1] + D_c[n] \sin(\hat{\varphi}[n]) \sin(\theta_a). \quad (31)$$

$$\hat{z}[n] = \hat{z}[n - 1] - D_c[n] \sin(\hat{\varphi}[n]) \sin(\theta_a). \quad (32)$$

**Estimation in State 3 and 4.** On the other hand, the bank angle  $\hat{\psi}[n]$  can be calculated in state 3 by (33) and in state 4 by (34).

$$\hat{\psi}[n] = \text{atan} \left( \frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{-a_{zf}[n]} \right). \quad (33)$$

$$\hat{\psi}[n] = \text{atan} \left( \frac{\sqrt{a_{xf}[n]^2 + a_{yf}[n]^2}}{a_{zf}[n]} \right). \quad (34)$$

In both states the components  $\hat{\varphi}[n]$ ,  $\hat{x}[n]$ ,  $\hat{y}[n]$  and  $\hat{\theta}[n]$  have the same expressions (35)–(38)

$$\hat{\varphi}[n] = \hat{\varphi}[n - 1] + \hat{\varphi}_o[n]. \quad (35)$$

$$\hat{x}[n] = \hat{x}[n - 1] + D_c[n] \sin(\hat{\varphi}[n]) \cos(\psi_a). \quad (36)$$

$$\hat{y}[n] = \hat{y}[n - 1] + D_c[n] \cos(\hat{\varphi}[n]). \quad (37)$$

$$\hat{\theta}[n] = 0 . \quad (38)$$

where  $\psi_a = \text{abs}(\hat{\psi}[n])$ .

Finally, the component  $\hat{z}[n]$  can be calculated in state 3 by (39) and in state 4 by (40).

$$\hat{z}[n] = \hat{z}[n-1] + D_c[n] \sin(\hat{\phi}[n]) \sin(\psi_a) . \quad (39)$$

$$\hat{z}[n] = \hat{z}[n-1] - D_c[n] \sin(\hat{\phi}[n]) \sin(\psi_a) . \quad (40)$$

### 3.5 Heading Reference System Based on Tilt Angle

The heading angle  $\hat{\phi}[n]$  is the most important of the navigation parameters in terms of its influence on accumulated errors. In the proposed system, when the robot is in the state 0 the only way to calculate this parameter is by odometry ( $\hat{\phi}[n] = \hat{\phi}[n-1] + \hat{\phi}_o[n]$  where  $\hat{\phi}_o[n] = \frac{D_l[n] - D_r[n]}{L}$ ). But, when the robot is moving on a surface with an elevation angle  $\hat{\theta}[n] \neq 0$  or with a bank angle  $\hat{\psi}[n] \neq 0$  (states 1, 2, 3 or 4), the values of  $a_{xf}[n]$  and  $a_{yf}[n]$  can also be used to estimate the heading angle of the robot.

When the FSM is in state 1, 2, 3 or 4, two estimations of the heading can be obtained, one of them via odometry  $\hat{\phi}_o[n]$  and another one from the accelerometer  $\hat{\phi}_a[n]$  by the following Eqs. (41)–(44).

In state 1,

$$\hat{\phi}_a[n] = \begin{cases} \text{atan} \left( \frac{|a_{xf}|}{|a_{yf}|} \right) & \text{if } (a_{xf} > 0) \& (a_{yf} \leq 0) . \\ \frac{\pi}{2} + \text{atan} \left( \frac{|a_{yf}|}{|a_{xf}|} \right) & \text{if } (a_{xf} > 0) \& (a_{yf} > 0) . \\ \pi + \text{atan} \left( \frac{|a_{xf}|}{|a_{yf}|} \right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} > 0) . \\ \frac{3\pi}{2} + \text{atan} \left( \frac{|a_{yf}|}{|a_{xf}|} \right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} \leq 0) . \end{cases} \quad (41)$$

in state 2,

$$\hat{\phi}_a[n] = \begin{cases} \pi + \text{atan} \left( \frac{|a_{xf}|}{|a_{yf}|} \right) & \text{if } (a_{xf} > 0) \& (a_{yf} \leq 0) . \\ \frac{3\pi}{2} + \text{atan} \left( \frac{|a_{yf}|}{|a_{xf}|} \right) & \text{if } (a_{xf} > 0) \& (a_{yf} > 0) . \\ \text{atan} \left( \frac{|a_{xf}|}{|a_{yf}|} \right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} > 0) . \\ \frac{\pi}{2} + \text{atan} \left( \frac{|a_{yf}|}{|a_{xf}|} \right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} \leq 0) . \end{cases} \quad (42)$$

in state 3,

$$\hat{\varphi}_a[n] = \begin{cases} \frac{\pi}{2} + \operatorname{atan}\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf} > 0) \& (a_{yf} \leq 0) . \\ \pi + \operatorname{atan}\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf} > 0) \& (a_{yf} > 0) . \\ \frac{3\pi}{2} + \operatorname{atan}\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} > 0) . \\ \operatorname{atan}\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} \leq 0) . \end{cases} \quad (43)$$

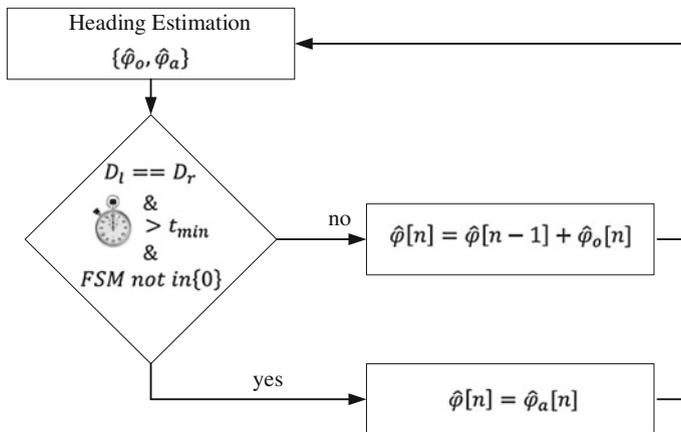
and finally, in state 4

$$\hat{\varphi}_a[n] = \begin{cases} \frac{3\pi}{2} + \operatorname{atan}\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf} > 0) \& (a_{yf} \leq 0) . \\ \operatorname{atan}\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf} > 0) \& (a_{yf} > 0) . \\ \frac{\pi}{2} + \operatorname{atan}\left(\frac{|a_{xf}|}{|a_{yf}|}\right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} > 0) . \\ \pi + \operatorname{atan}\left(\frac{|a_{yf}|}{|a_{xf}|}\right) & \text{if } (a_{xf} \leq 0) \& (a_{yf} \leq 0) . \end{cases} \quad (44)$$

where  $a_{xf} = a_{xf}[n]$  and  $a_{yf} = a_{yf}[n]$ .

Here, it is important to note that the heading estimation from the accelerometer  $\hat{\varphi}_a[n]$  is an absolute measure of the heading angle and it does not depend on the previous values. For this reason, this measure is a good way to avoid the accumulated errors that the classical odometry produces.

On the other hand, due to the acceleration signals have to be filtered to eliminate the noise in the sensor, the measurements of the acceleration have a delay which depends on the parameter  $\alpha$  of the filter (see Eq. 7). In this case, if the heading estimation is calculated as  $\hat{\varphi}[n] = \hat{\varphi}_a[n]$  it would be a bad approximation, for example, when the robot is making a circular path. Nevertheless, a little time after the robot starts to travel in straight line, the accelerations are stable (because the values of  $a_{xf}[n]$ ,  $a_{yf}[n]$  and  $a_{zf}[n]$  do not change) and the delay does not affect.



**Fig. 9** Proposed heading reference system

Taking in account the previous considerations, when the robot is travelling in straight line ( $D_l[n] = D_r[n]$ ) after certain time ( $time > t_{min}$ ) and the FSM is in state 1, 2, 3 or 4, the heading estimation from the accelerometer  $\hat{\varphi}_a[n]$  can be used as a reference value for the heading angle parameter. In this way, the accumulated error that the classical odometry produces in the heading angle can be avoided. The  $t_{min}$  value has to be defined as a function of the parameter  $\alpha$  of the filter. The reference system proposed in this paper is depicted in Fig. 9.

## 4 Experimental Results

To check the proposed algorithm in this paper, the low cost mOway differential wheel drive robot has been used. This robot has wheel encoders and a three-axis MMA7455L accelerometer from Freescale Semiconductor. The accelerometer has a dynamic range from  $-2g$  to  $2g$  (where  $g$  value is  $9.81 \text{ m/s}^2$ ) in the three axis. The positioning algorithm has been implemented in C++ language and it runs in a laptop with Windows OS. A radio frequency link of 2.4 GHz has been used to communicate the robot and the laptop. A flat surface without tilt angle has been used as the world frame and a moving surface as surface frame (see Sect. 2). Finally, a video camera has been used to capture the real movements of the robot and with the Tracker video analysis and modeling tool the real path of the robot has been captured and analysed.

Before checking the algorithm, the sensor and other parameters of the robot were calibrated. The results of these procedures are presented in the Table 1. On the other hand, the parameters of the algorithm were set as they are shown in Table 2.

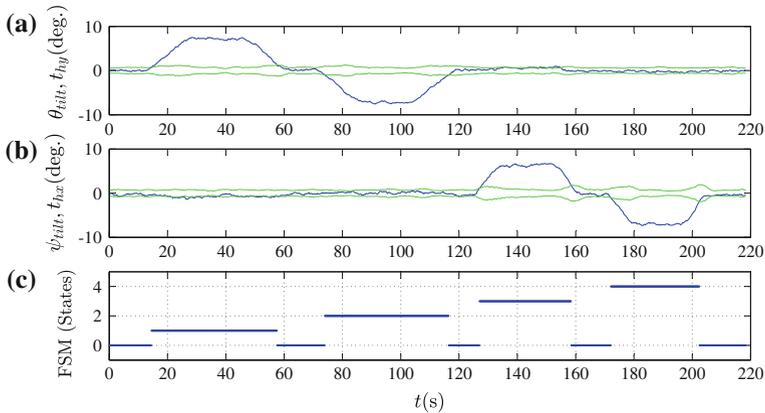
Next sections, show some experiments to analyse the response of the algorithm.

**Table 1** Robot parameters

Sampling period	$t_s = 0.2 \text{ s}$
Distance between two wheels	$L = 6.6 \text{ cm}$
Bias X acceleration	$a_{xbias} = -0.1059 \text{ g}$
Bias Y acceleration	$a_{ybias} = -0.3480 \text{ g}$
Bias Z acceleration	$a_{zbias} = 0.0927 \text{ g}$

**Table 2** Algorithm parameters

Filter constant	$\alpha = 0.9$
Length of threshold calculator	$l = 20$
Guard value $H$	$H = 0.2^\circ$
Guard value $h$	$h = 0.1^\circ$
Minimum time in reference system	$t_{min} = 1 \text{ s}$



**Fig. 10** FSM's behaviour. **a** Elevation angle in blue and  $t_{hy}$  in green. **b** Bank angle in blue and  $t_{hx}$  in green. **c** States of the FSM

#### 4.1 Behaviour of the FSM

To check the behaviour of the FSM, the robot was put on the moving surface (surface frame) with a constant speed of 10.3 cm/s in straight line. Then different tilts ( $\hat{\theta}$  and  $\hat{\psi}$ ) were applied and the states of the FSM were analysed. The elevation angle was modified from  $-8^\circ$  to  $8^\circ$ , in the same way, the bank angle was varied from  $-7.5^\circ$  to  $7.5^\circ$ . The result of the experiment is depicted in the Fig. 10.

As Fig. 10 shows, the behaviour of the system is according to the tilt angle. When tilt angles are close to zero, the system is in the state 0. The FSM is in the state 1 or 2 depending on the value of  $\theta_{tilt}$  as it is presented in Fig. 10a, c. And finally, states 3 and 4 depend on the value of  $\psi_{tilt}$  as it is shown in Fig. 10b, c.

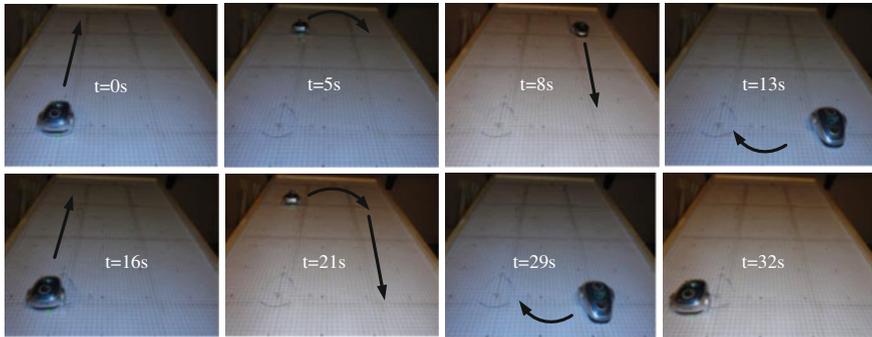
It is important to note that the system can change between states with an error smaller than  $1^\circ$  in the tilt estimation as it is presented in Fig. 10.

#### 4.2 3D Positioning Estimation

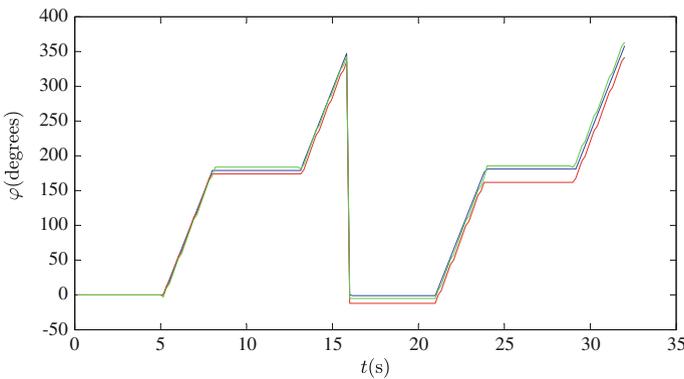
To analyse how the heading reference system improves the 3D positioning estimation, a new experiment has been set up. In this case, the robot has been programmed with a trajectory as it is shown in the Fig. 11.

The real path of the robot has been recorded by the video camera and two estimations have been calculated in the same experiment. One of them without the heading reference system and the another one using this system.

The real heading and its estimations by the algorithm are presented in the Fig. 12. As it can be noticed from this figure, the error in the heading estimation without



**Fig. 11** Snapshots of the experiment. In the *straight line* path the robot has a speed of 10.8 cm/s. In the *circular path* the speed of the *right wheel* is 10.3 cm/s and the *left wheel* is 17.5 cm/s

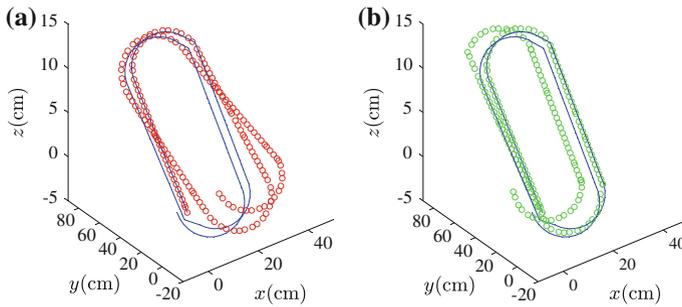


**Fig. 12** Heading estimation. *Solid blue line* is the real heading, *solid red line* is the heading estimation without reference system and *solid green line* is the heading estimation with the reference system

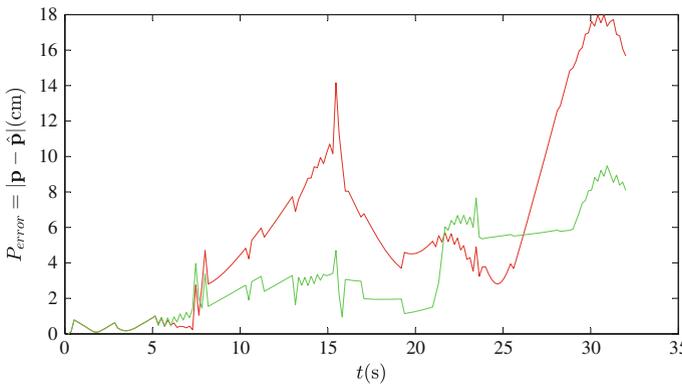
the reference system increases over time. However when the reference system is activated, the error in the heading estimation does not increase.

Considering the position parameters, the results of the experiment are depicted in the Fig. 13.

As it is shown in the figure, when the position is obtained with the reference system proposed in this work, the estimation error is reduced considerably. If the real position vector is defined as  $\mathbf{p} = \{x, y, z\}$  and the estimation position vector as  $\hat{\mathbf{p}} = \{\hat{x}, \hat{y}, \hat{z}\}$ , the positioning error can be defined as  $\mathbf{P}_{error} = |\mathbf{p} - \hat{\mathbf{p}}|$ . In Fig. 14, the positioning error is presented for this experiment. If these results are compared, although both methodologies have accumulative errors over time, the proposed reference system reduce significantly the positioning error.



**Fig. 13** 3D position parameters, *solid blue line* is the real path. **a** *Red circles* is the position estimation without reference system. **b** *Green circles* is the position estimation with reference system



**Fig. 14** *Red solid line* is the positioning error when the reference system has not been used. The *solid green line* is the positioning error when the reference system is activated

## 5 Conclusions

In this work we have proposed a new 3D positioning algorithm for low cost mobile robots. The architecture presented can be applied in robots which have wheel encoders and a three axis accelerometer. The problem formulation have been developed for two wheel differential robots, however this algorithm is easy to apply in other kind of robots in a simple way. The methodology presented is based on a Finite State Machine. A threshold calculator allows to set the system dynamically as a function of the noise in the accelerometer. Also a new reference system is proposed, this new system improves considerably the estimation of the algorithm respect to the classical odometry.

The proposed algorithm was checked in the low cost robot mOway. In these experiments, in a simple way, the algorithm has been set and precise estimations of the 3D position were obtained. The threshold calculator worked in a correct way to estimate the states of the FSM and it sets the systems dynamically. Finally with the new reference system the positioning error has been considerably reduced.

As a future work, now we are considering to apply this algorithm in robots which have also gyroscopes to allow estimations on surface with elevation and bank angles at the same time. On the other hand, new methods for reference systems in low cost robots are being explored.

## References

1. Borenstein, J., Feng, L.: Correction of systematic odometry errors in mobile robots. In: 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings, vol. 3, pp. 569–574. IEEE (1995)
2. Borenstein, J., Everett, H.R., Feng, L., Wehe, D.: Mobile robot positioning-sensors and techniques. Naval Command Control and Ocean Surveillance Center RDT and E Div San Diego CA (1997)
3. Lee, Y.C., Park, S.: Localization method for mobile robots moving on stairs in multi-floor environments. In: 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 4014–4020. IEEE (2014)
4. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots. MIT press (2011)
5. Everett, H.R.: Sensors for Mobile Robots: Theory and Application. AK Peters, Ltd. (1995)
6. Faisal, M., Hedjar, R., Alsulaiman, M., Al-Mutabe, K., Mathkour, H.: Robot localization using extended Kalman filter with infrared sensor. In: 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), pp. 356–360. IEEE (2014)
7. Liu, H.H., Pang, G.K.: Accelerometer for mobile robot positioning. IEEE Trans. Ind. Appl. **37**(3), 812–819 (2001)
8. Trimpe, S., D'Andrea, R.: Accelerometer-based tilt estimation of a rigid body with only rotational degrees of freedom. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2630–2636. IEEE (2010)
9. Luczak, S., Oleksiuk, W., Bodnicki, M.: Sensing tilt with MEMS accelerometers. Sens. J. IEEE **6**(6), 1669–1675 (2006)
10. Roberson, R.E., Schwertassek, R.: Dynamics of Multibody Systems, vol. 18. Springer, Berlin (1988)
11. Abbas, T., Arif, M., Ahmed, W.: Measurement and correction of systematic odometry errors caused by kinematics imperfections in mobile robots. In: International Joint Conference SICE-ICASE, pp. 2073–2078. IEEE (2006)
12. Jha, A., Kumar, M.: Two wheels differential type odometry for mobile robots. In: 2014 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), pp. 1–5. IEEE (2014)
13. Seifert, K., Camacho, O.: Implementing positioning algorithms using accelerometers. Freescale Semiconductor (2007)

# Kinematics, Simulation, and Analysis of the Planar and Symmetric Postures of a Serial-Parallel Climbing Robot

Adrián Peidro, Arturo Gil, José María Marín, Yerai Berenguer and Óscar Reinoso

**Abstract** This chapter presents the kinematic analysis of a novel 10-degrees-of-freedom serial-parallel redundant robot. The robot is designed for climbing and exploring 3D truss structures to execute maintenance and inspection tasks. First, the forward kinematic problem of the robot is solved. Then, since the general inverse kinematic problem is complex due to the kinematic redundancy, a simplified problem which assumes planar and symmetric postures is solved. Using a developed simulator, it is shown that these planar and symmetric postures are sufficient to execute many typical movements necessary to explore 3D structures, such as transitions between different beams or between different faces of a beam. Finally, the workspace of the robot when it adopts the planar and symmetric postures is analyzed for different designs, demonstrating the flexibility of these postures.

**Keywords** Biped robots · Climbing robots · Hybrid serial-parallel robots · Kinematics · Redundant robots · Simulation · Workspace

---

A. Peidro (✉) · A. Gil · J.M. Marín · Y. Berenguer · Ó. Reinoso  
Systems Engineering and Automation Department, Miguel Hernández University,  
Av. de la Universidad s/n, 03202 Elche, Alicante, Spain  
e-mail: [adrian.peidro@goumh.umh.es](mailto:adrian.peidro@goumh.umh.es)  
URL: <http://arvc.umh.es>

A. Gil  
e-mail: [arturo.gil@goumh.umh.es](mailto:arturo.gil@goumh.umh.es)

J.M. Marín  
e-mail: [jmarin@goumh.umh.es](mailto:jmarin@goumh.umh.es)

Y. Berenguer  
e-mail: [yberenguer@goumh.umh.es](mailto:yberenguer@goumh.umh.es)

Ó. Reinoso  
e-mail: [o.reinoso@goumh.umh.es](mailto:o.reinoso@goumh.umh.es)

# 1 Introduction

Many human-made vertical structures require periodic maintenance and inspection tasks. For example, the glass facades of skyscrapers must be cleaned, and the welded unions in the metallic skeletons of the buildings must be checked. Tasks like these are dangerous for human operators, who must work in environments difficult to access and are exposed to risks such as falling from height, contamination (e.g. inspections in nuclear or chemical facilities) or electrocution (e.g. maintenance of power transmission lines). To eliminate these risks, during the last two decades many researchers have been investigating the possibility of automating the execution of these tasks using climbing robots. Reference [11] presents an exhaustive analysis of the applications and design of climbing robots, as well as a comprehensive review of the locomotion and adhesion technologies.

Three-dimensional (3D) truss structures are present in many vertical structures such as bridges, towers and skeletons of buildings. These structures are typically constituted by a network of beams connected at structural nodes, and a high degree of mobility is required to explore them. Climbing robots for 3D trusses can be classified into two main types [13]: continuous-motion and step-by-step robots. Continuous-motion robots are faster, use wheels, and employ magnetism or friction to adhere to the structure [3, 14]. However, they usually have more difficulties to negotiate obstacles and their wheels may slip. Step-by-step robots have two grippers connected by a kinematic chain which has some degrees of freedom (DOF). Their name reflects their locomotion method: in each motion cycle, one gripper is fixed to the structure, whereas the kinematic chain moves the other gripper to the next attachment point of the structure, where it will be fixed. Then, the previously fixed gripper is released and a new motion cycle begins. During each motion cycle, these robots are equivalent to typical robot manipulators. Hence, they have a higher mobility that facilitates the avoidance of obstacles, but they are heavier, slower, and more complex.

The architecture of the kinematic chain of step-by-step robots can be serial, parallel, or hybrid. Serial architectures have larger workspaces than parallel ones, but they are less rigid and have a limited payload. The serial architectures have been the most explored ones in step-by-step climbing robots, with many different designs proposed by different authors. For example, [5] presents a 6-DOF robot to explore 3D metallic structures. Since the robot is powered by a battery, the movements are optimized to reduce the energy consumption and increase its autonomy. Another 4-DOF serial climbing robot is presented in [13]. Other authors propose robots inspired by inchworms, with 5 and 8 DOF [8, 12]. Reference [10] presents a similar modular robot whose number of DOF can be increased connecting more modules in series. Finally, [16] presents 3-DOF robots that can individually explore 3D trusses or can be combined with other robots to form more complex kinematic chains with increased maneuverability.

Parallel climbing robots have also been studied. These architectures offer a higher payload-to-weight ratio than serial robots, but their workspace is more limited. In [2],

the authors propose using a Gough-Stewart platform for climbing truss structures, pipelines and palm trees. The robot adheres to the structure using grippers or embracing it with its annular platforms.

Finally, hybrid climbing robots are composed of some serially connected parallel mechanisms, and they have the advantages of both architectures: high payload, maneuverability, and stiffness. A hybrid robot for climbing 3D structures is proposed in [15], where the authors combine a 3-RPR parallel robot with a rotation module connected in series. Another hybrid robot is proposed in [7]. In this case, the robot is biped and each leg is the serial combination of two 3-RPS parallel robots. Hence, the complete robot has 12 DOF.

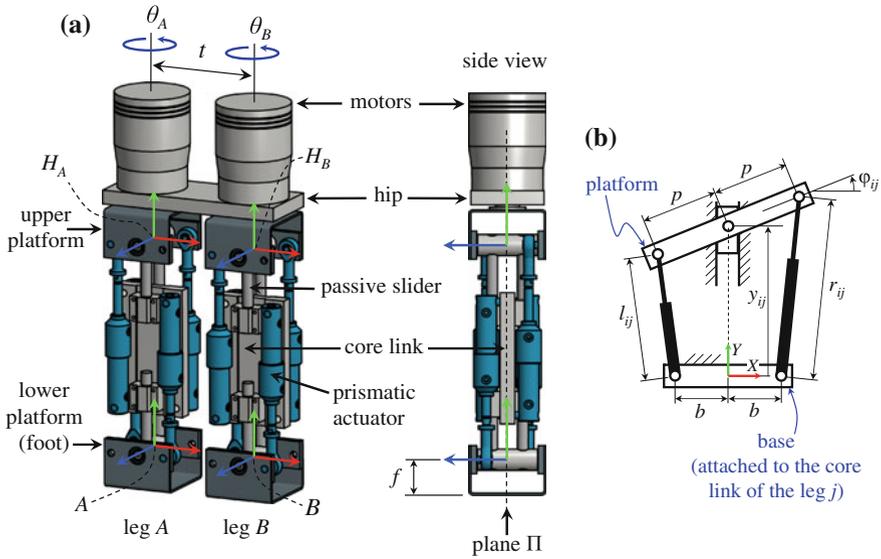
This chapter presents a novel 10-DOF hybrid robot for climbing 3D truss structures. The robot is biped and each leg is the serial combination of two parallel mechanisms. The design of the robot makes it especially suitable to maneuver in 3D truss structures and perform transitions between planes with different orientations. In this chapter, we solve the forward kinematic problem of this robot. Also, we analyze a simplified case of the inverse kinematic problem in which the robot adopts planar and symmetric postures. As the simulations show, these postures suffice for exploring 3D structures. Finally, we study the workspace of the robot to assess the feasibility of exploring 3D structures using the planar and symmetric postures when the conditions of the problem (design parameters of the robot, initial position, etc.) change.

This chapter is organized as follows. The architecture of the robot is described in Sect. 2. Next, the forward kinematic problem is solved in Sect. 3. In Sect. 4, a simplified yet useful version of the inverse kinematic problem is solved. Then, Sect. 5 presents a tool that simulates the forward kinematics of the robot. This tool is used to demonstrate the execution of some example trajectories in a 3D structure. Section 6 analyzes the workspace of the robot to further study the usefulness of the aforementioned simplified inverse kinematic problem. Finally, the conclusions are exposed in Sect. 7.

## 2 Description of the Robot

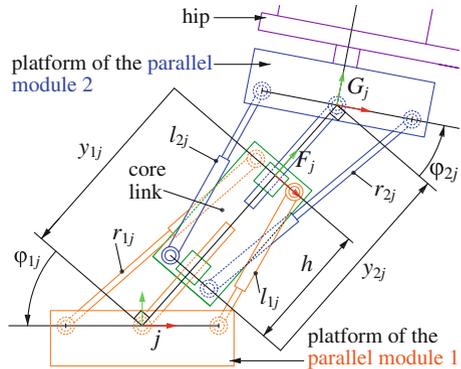
Figure 1a shows a 3D model of the biped climbing robot. The robot has two identical legs ( $A$  and  $B$ ) connected to the hip through revolute joints driven by motors (angles  $\theta_A$  and  $\theta_B$ ). Each leg has three links: a core link and two platforms. The lower platform is the foot of the leg and carries the gripper that fixes the robot to the structure (the grippers are not considered in the kinematic analysis presented in this chapter). The upper platform is connected to the hip through the aforementioned revolute joint. Each platform is connected to the core link by means of two prismatic actuators in parallel and a passive slider.

The mechanism composed of the core link, one platform, and the two prismatic actuators that connect these two elements, is a closed-loop linkage that will be called hereafter “parallel module”. The parallel modules are planar mechanisms that can



**Fig. 1** a 3D model of the climbing robot. b A parallel module

**Fig. 2** Kinematics of a generic leg  $j \in \{A, B\}$



be represented schematically as shown in Fig. 1b. Hence, each leg is the serial combination of the parallel modules 1 (which is connected to the foot) and 2 (which is connected to the hip). The prismatic actuators of each parallel module lie in opposite sides of the plane  $\Pi_j$ , which is one of the planes of symmetry of the core link of the leg  $j$  (see the side view in Fig. 1a). This is indicated with dashed lines in Fig. 2.

Figure 1a also shows some reference frames attached to different parts of the robot. In this chapter, the  $X$ ,  $Y$ , and  $Z$  axes of reference frames will be represented in red, green, and blue colors, respectively. The frames  $H_A$  and  $H_B$  are fixed to the hip of the robot, whereas the frames  $A$  and  $B$  are respectively attached to the feet of the legs  $A$  and  $B$ .

The robot has 10 DOF: the rotation angles  $\theta_A$  and  $\theta_B$ , and the four prismatic actuators of each leg. In the next sections, the forward and inverse kinematic problems of the robot will be analyzed. After that, we will simulate the forward kinematics to demonstrate its ability to explore 3D structures.

### 3 Forward Kinematics

In this section, the forward kinematic problem (FKP) of the robot is solved. The problem considered here consists in calculating the position and orientation of one foot with respect to the other foot when the joint coordinates are known: the angles  $\theta_A$  and  $\theta_B$  and the lengths  $(l_{ij}, r_{ij})$  of the linear actuators of the parallel modules ( $i \in \{1, 2\}$ ,  $j \in \{A, B\}$ ). First, the forward kinematics of the parallel modules is analyzed.

#### 3.1 FKP of the Parallel Modules

Figure 1b shows the  $i$ -th parallel module of the leg  $j$  ( $i \in \{1, 2\}$ ,  $j \in \{A, B\}$ ). A parallel module is a closed-loop planar mechanism composed of a mobile platform connected to a base through two prismatic actuators with lengths  $l_{ij}$  and  $r_{ij}$ . The platform is constrained to only translate vertically and rotate. The forward kinematics consists in calculating the position  $y_{ij}$  and the orientation  $\varphi_{ij}$  of the mobile platform in terms of  $l_{ij}$  and  $r_{ij}$ . According to Fig. 1b, the relationship between  $(l_{ij}, r_{ij})$  and  $(y_{ij}, \varphi_{ij})$  is:

$$(p \cos \varphi_{ij} - b)^2 + (y_{ij} + p \sin \varphi_{ij})^2 = r_{ij}^2 \quad (1)$$

$$(p \cos \varphi_{ij} - b)^2 + (y_{ij} - p \sin \varphi_{ij})^2 = l_{ij}^2 \quad (2)$$

These equations can be combined to obtain an equivalent system. Adding together Eqs. (1) and (2) yields Eq. (3), whereas subtracting Eqs. (2) from (1) results in Eq. (4):

$$4bp \cos \varphi_{ij} = 2y_{ij}^2 + 2b^2 + 2p^2 - l_{ij}^2 - r_{ij}^2 \quad (3)$$

$$4y_{ij}p \sin \varphi_{ij} = r_{ij}^2 - l_{ij}^2 \quad (4)$$

Solving  $\cos \varphi_{ij}$  from Eq. (3) gives:

$$\cos \varphi_{ij} = \frac{2y_{ij}^2 + 2b^2 + 2p^2 - l_{ij}^2 - r_{ij}^2}{4bp} \quad (5)$$

Squaring Eq. (4):

$$16y_{ij}^2 p^2 (1 - \cos^2 \varphi_{ij}) = (r_{ij}^2 - l_{ij}^2)^2 \quad (6)$$

Finally, substituting Eqs. (5) into (6) yields a cubic equation in  $\Upsilon_{ij} = y_{ij}^2$ :

$$\Upsilon_{ij}^3 + k_2^{ij}\Upsilon_{ij}^2 + k_1^{ij}\Upsilon_{ij} + k_0^{ij} = 0 \quad (7)$$

where:

$$k_2^{ij} = 2b^2 + 2p^2 - l_{ij}^2 - r_{ij}^2 \quad (8)$$

$$k_1^{ij} = \left[ (b+p)^2 - \frac{l_{ij}^2 + r_{ij}^2}{2} \right] \left[ (b-p)^2 - \frac{l_{ij}^2 + r_{ij}^2}{2} \right] \quad (9)$$

$$k_0^{ij} = b^2(l_{ij} + r_{ij})^2(l_{ij} - r_{ij})^2/4 \quad (10)$$

Equation (7) always has three roots, two of which may be complex. For a given strictly positive root  $\Upsilon_{ij}$  of Eq. (7), two solutions are obtained for  $y_{ij} = \pm\sqrt{\Upsilon_{ij}}$ . For each of these two values of  $y_{ij}$ ,  $\cos \varphi_{ij}$  is calculated from Eq. (5), whereas  $\sin \varphi_{ij}$  is obtained from Eq. (4):

$$\sin \varphi_{ij} = \frac{r_{ij}^2 - l_{ij}^2}{4y_{ij}p} \quad (11)$$

Once  $\cos \varphi_{ij}$  and  $\sin \varphi_{ij}$  are known,  $\varphi_{ij}$  is unequivocally determined in  $(-\pi, \pi]$ . If  $\Upsilon_{ij} = 0$ , then  $y_{ij} = 0$  and  $\cos \varphi_{ij}$  is calculated using Eq. (5). However,  $\sin \varphi_{ij}$  cannot be calculated from Eq. (11) since  $y_{ij} = 0$ . Instead,  $\sin \varphi_{ij}$  is calculated as follows:

$$\sin \varphi_{ij} = \pm\sqrt{1 - \cos^2 \varphi_{ij}} \quad (12)$$

obtaining two solutions. It is shown in [9], using Sturm's Theorem, that Eq. (7) cannot have more than two non-negative roots. Since each non-negative root of Eq. (7) yields two different pairs  $(y_{ij}, \varphi_{ij})$ , the FKP of each parallel module has four solutions at most.

Note that swapping the values of  $r_{ij}$  and  $l_{ij}$  neither affects Eq. (7) nor Eq. (5), but it changes the sign of  $\sin \varphi_{ij}$  in Eq. (11). Hence, swapping  $r_{ij}$  and  $l_{ij}$  changes the sign of  $\varphi_{ij}$ , leaving  $y_{ij}$  unchanged. This can be observed in Fig. 1b, where swapping  $r_{ij}$  and  $l_{ij}$  is equivalent to rotating the figure  $\pi$  rad about the vertical  $Y$  axis. This fact will be exploited in Sect. 4 to analyze the inverse kinematics.

### 3.2 FKP of the Complete Robot

The forward kinematics of the complete robot consists in calculating the position and orientation of one foot with respect to the other foot when the ten joint coordinates are known. The problem will be solved using Homogeneous Transformation Matrices (HTMs). An HTM has the following form [4]:

$$\mathbf{T}_{m/n} = \begin{bmatrix} \mathbf{R}_{m/n} & \mathbf{t}_{m/n} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (13)$$

where  $\mathbf{0}_{1 \times 3} = [0, 0, 0]$ . The matrix  $\mathbf{T}_{m/n}$  encodes the position and orientation of a frame  $m$  with respect to another frame  $n$ . Indeed,  $\mathbf{R}_{m/n} \in \mathbb{R}^{3 \times 3}$  is a rotation matrix whose columns are the vectors of the frame  $m$  expressed in the basis formed by the vectors of the frame  $n$ , whereas  $\mathbf{t}_{m/n} \in \mathbb{R}^{3 \times 1}$  is the position of the origin of the frame  $m$  in coordinates of the frame  $n$ .

The forward kinematics of one leg can be easily solved using HTMs. Figure 2 represents a generic leg  $j \in \{A, B\}$ . Each leg has two parallel modules whose bases are attached to the core link. The platform of the parallel module 1 is the foot of the leg, whereas the platform of the parallel module 2 is connected to the hip of the robot by means of a revolute joint. The variables  $(y_{1j}, \varphi_{1j}, y_{2j}, \varphi_{2j})$  are obtained from  $(l_{1j}, r_{1j}, l_{2j}, r_{2j})$  as explained in Sect. 3.1. All the reference frames of Fig. 2 are contained in the plane  $\Pi_j$ , which is one of the planes of symmetry of the core link of the leg  $j$  (see Fig. 1a). The transformation between the frame  $j$  (fixed to the foot) and the frame  $F_j$  (fixed to the core link) is:

$$\mathbf{T}_{F_j/j} = \begin{bmatrix} \cos \varphi_{1j} & \sin \varphi_{1j} & 0 & y_{1j} \sin \varphi_{1j} \\ -\sin \varphi_{1j} & \cos \varphi_{1j} & 0 & y_{1j} \cos \varphi_{1j} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Similarly, the transformation between the frame  $G_j$  (attached to the platform of the parallel module 2) and the frame  $F_j$  is:

$$\mathbf{T}_{G_j/F_j} = \begin{bmatrix} \cos \varphi_{2j} & -\sin \varphi_{2j} & 0 & 0 \\ \sin \varphi_{2j} & \cos \varphi_{2j} & 0 & y_{2j} - h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

where  $h$  is a geometric constant. Finally, a rotation  $\theta_j$  about the  $Y$  axis of the frame  $G_j$  transforms it into the frame  $H_j$ , which is attached to the hip:

$$\mathbf{T}_{H_j/G_j} = \begin{bmatrix} \cos \theta_j & 0 & \sin \theta_j & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_j & 0 & \cos \theta_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

The position and orientation of the frame  $H_j$  with respect to the frame  $j$  is obtained as follows:

$$\mathbf{T}_{H_j/j} = \mathbf{T}_{F_j/j} \mathbf{T}_{G_j/F_j} \mathbf{T}_{H_j/G_j} \quad (17)$$

which completes the FKP of any generic leg  $j$ . Once the forward kinematics of each leg is solved, it is straightforward to calculate the position and orientation of the foot

of one leg  $k \in \{A, B\} \setminus \{j\}$  with respect to the foot of the other leg  $j$ :

$$\mathbf{T}_{k/j} = \mathbf{T}_{H_j/j} \mathbf{T}_{H_k/H_j} \mathbf{T}_{k/H_k} \quad (18)$$

where  $\mathbf{T}_{k/H_k} = (\mathbf{T}_{H_k/k})^{-1}$  and  $\mathbf{T}_{H_k/H_j}$  is the HTM that encodes the position and orientation of the frame  $H_k$  with respect to the frame  $H_j$ :

$$\mathbf{T}_{H_k/H_j} = \begin{bmatrix} \mathbf{I} & \mathbf{t}_{H_k/H_j} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (19)$$

which is constant because both frames are attached to the same rigid body (the hip).  $\mathbf{I}$  is the  $3 \times 3$  identity matrix. Moreover, according to Fig. 1a:  $\mathbf{t}_{H_B/H_A} = [t, 0, 0]^T = -\mathbf{t}_{H_A/H_B}$ , where  $t$  is the distance between the parallel axes of the revolute actuators.

Note that, in theory, there are  $4^4 = 256$  different solutions to the FKP of the complete robot. This is because the kinematic chain between the feet has four parallel modules connected in series and the FKP of each module has four real solutions at most.

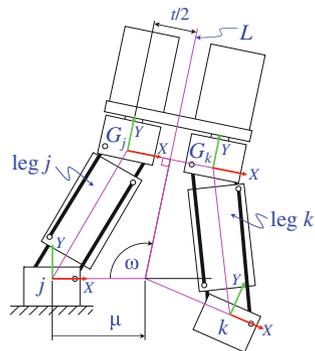
## 4 Inverse Kinematics

The inverse kinematic problem (IKP) consists in finding the values of the joint coordinates necessary to attain a desired relative position and orientation between the feet of the robot, and it is necessary for planning trajectories. In this robot, ten joint coordinates are used to place and orient one foot with respect to the other foot, which makes it redundant. Hence, the IKP is underconstrained and one should expect infinitely many solutions. This redundancy makes it difficult to solve the general IKP of this robot. Fortunately, many important movements necessary to explore a 3D structure (e.g., walking in one dimension, changing between planes, etc.) can be executed using the configuration analyzed in this section, which reduces the number of variables and simplifies the IKP.

The configuration considered in this section is depicted in Fig. 3, where the foot  $j$  is fixed to the structure and the foot  $k$  is mobile ( $j, k \in \{A, B\}$ ,  $j \neq k$ ). It is assumed that the  $Z$  axes of the frames attached to the feet are parallel and point in the same direction. Moreover, the origin of the frame attached to the foot  $k$  is contained in the  $XY$  plane of the frame attached to the foot  $j$ . In this situation, any variation in the length of the prismatic actuators of the parallel modules only produces planar motions of the frame  $k$  in the  $XY$  plane of the frame  $j$ . In this case, the position and orientation of the frame  $k$  relative to the frame  $j$  can be calculated as follows:

$$\mathbf{T}_{k/j} = \mathbf{T}_{G_j/j} \begin{bmatrix} \mathbf{I} & [t, 0, 0]^T \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} (\mathbf{T}_{G_k/k})^{-1} \quad (20)$$

**Fig. 3** The Planar Symmetric Inverse Kinematic (PSIK) problem



where  $\mathbf{T}_{G_j/j} = \mathbf{T}_{F_j/j} \mathbf{T}_{G_j/F_j}$ . Moreover, it is assumed that the joint coordinates of the parallel modules of the two legs are related as follows:

$$l_{ik} = r_{ij}, \quad r_{ik} = l_{ij} \quad (i = 1, 2) \quad (21)$$

This means that the joint coordinates of the parallel module  $i$  of the legs  $k$  and  $j$  are swapped. According to Sect. 3.1, this translates into:

$$y_{ik} = y_{ij}, \quad \varphi_{ik} = -\varphi_{ij} \quad (i = 1, 2) \quad (22)$$

It can be graphically checked that Eq. (22) implies that the legs  $k$  and  $j$  are symmetric with respect to the line  $L$ , which is the axis of symmetry of the hip of the robot. Substituting Eqs. (22) into (20), the matrix  $\mathbf{T}_{k/j}$  can be written only in terms of the variables of the leg  $j$  and has the following expression:

$$\mathbf{T}_{k/j} = \begin{bmatrix} -c(2\omega) & -s(2\omega) & 0 & \mu(1 - c(2\omega)) \\ s(2\omega) & -c(2\omega) & 0 & \mu \cdot s(2\omega) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

where  $s(x) = \sin x$ ,  $c(x) = \cos x$  and:

$$\mu = [t - 2(h - y_{1j} - y_{2j}) \sin \varphi_{2j}] / [2 \cos(\varphi_{1j} - \varphi_{2j})] \quad (24)$$

$$\omega = \varphi_{1j} - \varphi_{2j} + \pi/2 \quad (25)$$

Thus, under the condition of planar and symmetric motion, the position and orientation of the foot  $k$  relative to the foot  $j$  can be defined by two parameters  $(\mu, \omega)$ , which are indicated in Fig. 3. We define the Planar Symmetric Inverse Kinematic (PSIK) problem, which consists in calculating the joint coordinates  $(l_{1j}, r_{1j}, l_{2j}, r_{2j})$  needed to achieve a desired position and orientation  $(\mu, \omega)$ . Since the joint coordinates do not appear explicitly in Eqs. (24) and (25), the kinematic equations of the parallel

modules of the leg  $j$  must be included:

$$(p \cos \varphi_{1j} - b)^2 + (y_{1j} + p \sin \varphi_{1j})^2 = r_{1j}^2 \quad (26)$$

$$(p \cos \varphi_{1j} - b)^2 + (y_{1j} - p \sin \varphi_{1j})^2 = l_{1j}^2 \quad (27)$$

$$(p \cos \varphi_{2j} - b)^2 + (y_{2j} + p \sin \varphi_{2j})^2 = r_{2j}^2 \quad (28)$$

$$(p \cos \varphi_{2j} - b)^2 + (y_{2j} - p \sin \varphi_{2j})^2 = l_{2j}^2 \quad (29)$$

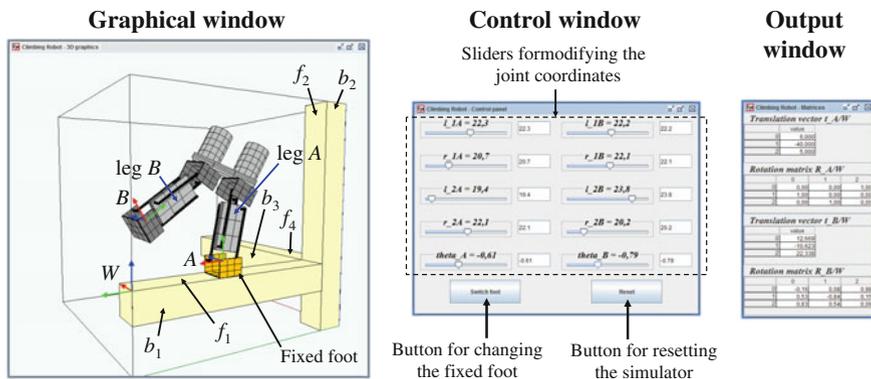
Hence, the PSIK problem requires calculating  $(l_{1j}, r_{1j}, l_{2j}, r_{2j}, y_{1j}, \varphi_{1j}, y_{2j}, \varphi_{2j})$  from Eqs. (24)–(29). Like the general inverse kinematic problem, the PSIK problem is underconstrained since eight unknowns must be obtained from six equations. However, the PSIK problem involves less variables and simpler equations. In the following section, we will show that some postures necessary to negotiate obstacles in a 3D structure can be analyzed solving the PSIK problem. Also, we will describe a method to choose appropriate solutions to the PSIK problem assuming that the lengths of the prismatic actuators of the parallel modules have upper and lower limits:  $l_{ij}$  and  $r_{ij}$  must be in  $[\rho_0, \rho_0 + \Delta\rho]$ , where  $\rho_0 > 0$  is the minimum length of the actuators and  $\Delta\rho > 0$  is their stroke.

## 5 Simulation

In this section, we will simulate the movements of the complete robot in an example 3D structure to validate the kinematic analyses of Sects. 3 and 4, and demonstrate the ability of the robot to explore the structure. More specifically, we will show how the robot can walk on a beam, perform transitions between different faces of the beams, and negotiate structural nodes.

To demonstrate these movements, we have developed a Java simulation tool that can be downloaded from <http://arvc.umh.es/parola/climber.html> (the latest version of Java may be required). The simulator implements the equations derived in Sect. 3 to solve the forward kinematics. As shown in Fig. 4, the simulator has a graphical window that shows the robot in the 3D test structure. The tool also has a window with a control panel where the user can modify the values of the ten joint coordinates, change the foot that is attached to the structure, or reset the simulation. It is important to remark that the simulation tool only implements the kinematic equations, without considering the dynamics of the robot (gravity is neglected) or the collisions between the robot and the structure. These advanced topics will be analyzed in the future.

Three reference frames are shown in the graphical window of the simulator: the world frame  $W$  (which is attached to one of the corners of the beam  $b_1$  of the structure) and the frames  $A$  and  $B$  (which are attached to the feet of the legs). The fixed foot is indicated in orange color. When the user modifies the value of a joint coordinate, the forward kinematics is solved and the position and orientation of the free foot with respect to the frame  $W$  is calculated as follows:



**Fig. 4** Interface of the tool developed to simulate the forward kinematics of the robot

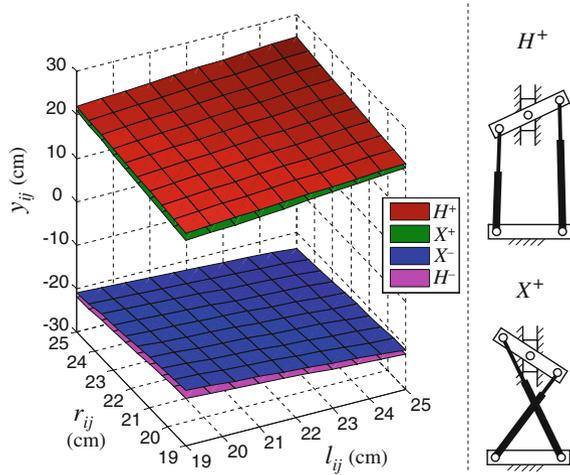
$$\mathbf{T}_{k/j} = \mathbf{T}_{j/W} \mathbf{T}_{k/j} \quad (30)$$

where  $\mathbf{T}_{k/j}$  is defined in Sect. 3.2,  $j$  denotes the fixed leg, and  $k$  denotes the mobile leg ( $j, k \in \{A, B\}$ ,  $j \neq k$ ). As shown in Fig. 4, the translation and rotation submatrices of  $\mathbf{T}_{A/W}$  and  $\mathbf{T}_{B/W}$  are indicated to the user in an output window of the tool. According to Sect. 3.2, there are 256 solutions to the forward kinematics of the complete robot since each parallel module can have up to four different solutions. However, it will be shown next that only one solution is valid.

For the following simulations, we will assume that  $b = p = 4$  cm,  $\rho_0 = 19$  cm, and  $\Delta\rho = 6$  cm, which means that the prismatic actuators must satisfy:  $r_{ij}, l_{ij} \in [19, 25]$  cm. Solving the forward kinematics of a parallel module for these ranges of the joint coordinates, and plotting the solution  $y_{ij}$  versus  $r_{ij}$  and  $l_{ij}$ , results in the four surfaces shown in Fig. 5. Each surface is associated with one of the configurations labeled as follows:  $H^+$ ,  $X^+$ ,  $H^-$ , and  $X^-$ . The solutions  $H^+$  and  $X^+$  are indicated in Fig. 5; the solutions  $H^-$  and  $X^-$  are their respective mirror images with respect to the base of the parallel module. According to the design of the robot (see Sect. 2), the only valid solution is  $H^+$ , since the other solutions are impossible due to mechanical interferences between different links of the legs. Moreover, Fig. 5 also provides a criterion for selecting the valid solution: the solution  $H^+$  always has the highest  $y_{ij}$  coordinate.

Once the only valid solution to forward kinematics has been characterized, we will simulate the execution of an example trajectory in the structure, which is composed of the three beams  $b_1$ ,  $b_2$ , and  $b_3$  indicated in Fig. 4. At the beginning of the trajectory, the robot lies on the face  $f_1$  of the beam  $b_1$ , and the objective is to move the robot to the face  $f_4$  of the beam  $b_3$ , negotiating the structural node where the three beams intersect. Next, we will show that such a trajectory can be executed by a sequence of basic movements that can be used to reach any other point of the structure. The values of the remaining geometric parameters of the robot are:  $t = 15.6$  cm,  $h = 16$  cm, which correspond to the design parameters of the prototype currently under development.

**Fig. 5** Solution surfaces of the FKP of a parallel module for  $b = p = 4$  cm. The surfaces  $H^+$  and  $H^-$  are almost parallel to the surfaces  $X^+$  and  $X^-$ , respectively



Moreover, the side of the square cross section of the beams measures 12 cm, and the distance between the face  $f_2$  of the beam  $b_2$  and the origin of the frame  $W$  is 88 cm.

### 5.1 Phase 1: Walking Along a Beam

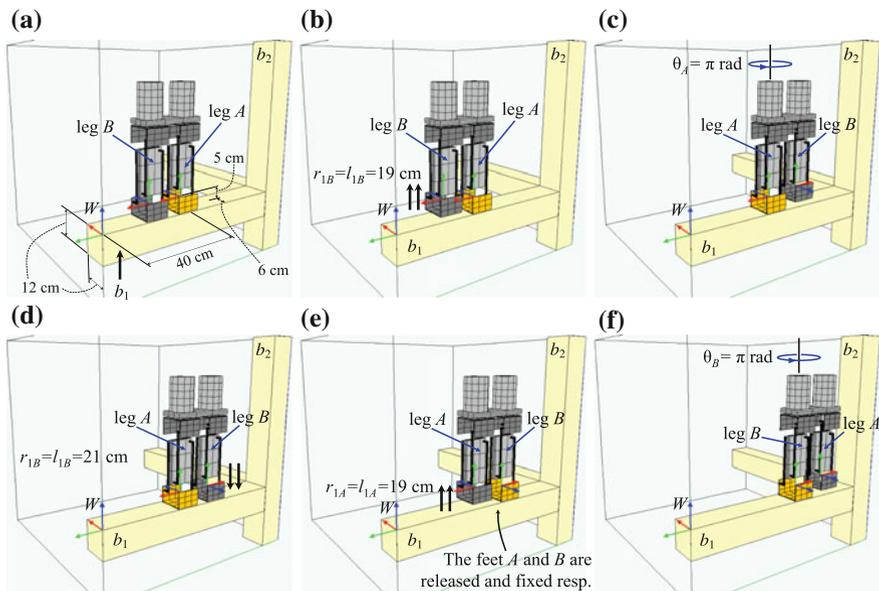
At the beginning of the trajectory (see Fig. 6a), the foot  $A$  is attached to the face  $f_1$  of the beam  $b_1$ , and the frame  $A$  has the following position and orientation:

$$\mathbf{t}_{A/W} = \begin{bmatrix} 6 \\ -40 \\ 5 \end{bmatrix} \text{ cm}, \mathbf{R}_{A/W} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (31)$$

The number “6” in  $\mathbf{t}_{A/W}$  means that the frame  $A$  is centered in the beam, whereas the number “5” is the size  $f$  of the feet, indicated in Fig. 1a. Initially, the joint coordinates have the following values:  $\theta_A = \theta_B = 0$ ,  $r_{ij} = l_{ij} = 21$  cm ( $i \in \{1, 2\}$ ,  $j \in \{A, B\}$ ). Starting from this configuration, Table 1 describes a simple sequence of movements that allows the robot to reach the vertical beam  $b_2$ . In each step of the given sequence, we indicate only the joint coordinates that change with respect to the previous step.

### 5.2 Phase 2: Concave Change of Plane

Once the beam  $b_2$  has been reached, it can be climbed to negotiate the structural node defined by the intersection of the three beams. The next objective is to perform



**Fig. 6** Example trajectory where the robot moves along a beam of the structure

**Table 1** Sequence of movements in the first phase of the simulated trajectory

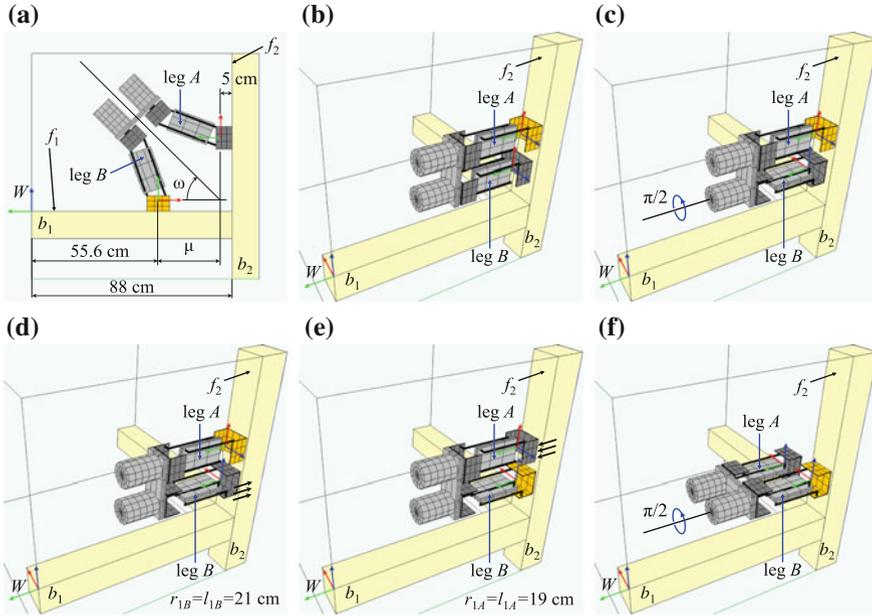
Step	Joint coordinates	Description of the movements in each step
1	$r_{1B} = l_{1B} = 19 \text{ cm}$	Retract the actuators connected to the foot $B$ to lift it (Fig. 6b)
2	$\theta_A = \pi \text{ rad}$	Rotate the robot about the leg $A$ (Fig. 6c)
3	$r_{1B} = l_{1B} = 21 \text{ cm}$	Extend the actuators connected to the foot $B$ until it touches the beam $b_1$ (Fig. 6d)
4	$r_{1A} = l_{1A} = 19 \text{ cm}$	Attach the foot $B$ to the face $f_1$ . Release and lift the foot $A$ retracting the actuators connected to it (Fig. 6e)
5	$\theta_B = \pi \text{ rad}$	Rotate the robot about the leg $B$ (Fig. 6f)

a concave transition between the faces  $f_1$  and  $f_2$ . Note that at the end of the previous phase (Fig. 6f), the  $Z$  axes of the frames attached to the two feet point in the same direction. Hence, the postures needed to change between these faces can be obtained solving the PSIK problem defined in Sect. 4.

Figure 7a indicates the input parameters needed to solve the PSIK problem:  $\mu = 27.4 \text{ cm}$ ,  $\omega = \pi/4 \text{ rad}$ , and  $j = B$ . Substituting these values and the geometric parameters of the robot into Eqs. (24)–(29) yields:

$$\frac{15.6 - 2(16 - y_{1B} - y_{2B}) \sin \varphi_{2B}}{2 \cos(\varphi_{1B} - \varphi_{2B})} = 27.4 \quad (32)$$

$$\varphi_{2B} - \pi/4 = \varphi_{1B} \quad (33)$$



**Fig. 7** A trajectory that includes a concave transition between different planes

$$(4 \cos \varphi_{1B} - 4)^2 + (y_{1B} + 4 \sin \varphi_{1B})^2 = r_{1B}^2 \quad (34)$$

$$(4 \cos \varphi_{1B} - 4)^2 + (y_{1B} - 4 \sin \varphi_{1B})^2 = l_{1B}^2 \quad (35)$$

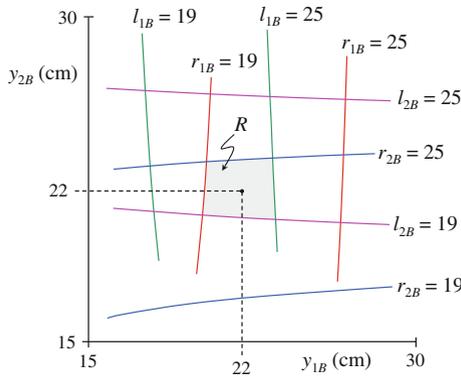
$$(4 \cos \varphi_{2B} - 4)^2 + (y_{2B} + 4 \sin \varphi_{2B})^2 = r_{2B}^2 \quad (36)$$

$$(4 \cos \varphi_{2B} - 4)^2 + (y_{2B} - 4 \sin \varphi_{2B})^2 = l_{2B}^2 \quad (37)$$

As discussed in Sect. 4, infinitely many solutions exist since there are eight variables to be solved from six equations. Next, we describe a way of choosing a proper solution to this underconstrained problem. First, Eq. (33) is used to eliminate  $\varphi_{1B}$  from Eq. (32). Then,  $\varphi_{2B}$  is solved from the resulting equation:

$$\varphi_{2B} = \sin^{-1} \left( \frac{13.7\sqrt{2} - 7.8}{y_{1B} + y_{2B} - 16} \right) \quad (38)$$

This solution can be substituted into Eqs. (33)–(37) to express the joint coordinates  $\{l_{1B}, r_{1B}, l_{2B}, r_{2B}\}$  in terms of  $\{y_{1B}, y_{2B}\}$ , which can be chosen so that  $l_{iB}, r_{iB} \in [19, 25]$  ( $i = 1, 2$ ). Figure 8 represents the curves of the  $(y_{1B}, y_{2B})$  plane in which each joint coordinate equals 19 or 25; any point inside the shaded region  $R$  enclosed by these curves is a valid solution to the PSIK problem. For example, the solution  $y_{1B} = y_{2B} = 22$  cm yields:  $r_{1B} \approx 20.59536194$ ,  $l_{1B} \approx 23.40761347$ ,  $r_{2B} \approx 23.65623783$ , and  $l_{2B} \approx 20.34961301$ , all in cm (these accurate values are



**Fig. 8** Region of valid solutions to the PSIK problem

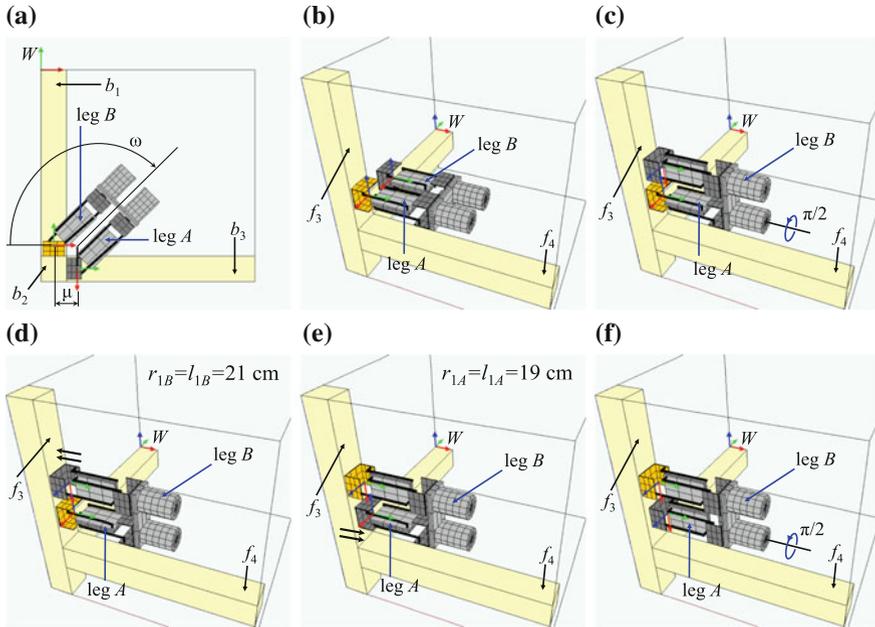
**Table 2** Sequence of movements in the second phase of the simulated trajectory

Step	Joint coordinates	Description of the movements in each step
1	$l_{iA} = r_{iA} = 21$ cm $l_{2B} = r_{2B} = 21$ cm $l_{1B} = r_{1B} = 19$ cm	Lift the foot $B$ and place both legs perpendicular to the face $f_2$ , leaving some distance between the foot $B$ and the face $f_2$ (Fig. 7b)
2	$\theta_B = \pi/2$ rad	Rotate the leg $B$ about its own axis (Fig. 7c)
3	$r_{1B} = l_{1B} = 21$ cm	Extend the actuators connected to the foot $B$ until it touches the face $f_2$ (Fig. 7d)
4	$r_{1A} = l_{1A} = 19$ cm	Attach the foot $B$ to the face $f_2$ . Release and lift the foot $A$ retracting the actuators connected to it (Fig. 7e)
5	$\theta_B = \pi$ rad	Rotate the robot about the leg $B$ (Fig. 7f)

valid only for the simulation; in a real implementation we will have to deal with the finite precision of the sensors). This solution is used to perform a transition between the faces  $f_1$  and  $f_2$  (see Fig. 7a). After performing this transition, the foot  $A$  is attached to the beam  $b_2$ , and the sequence of movements described in Table 2 is used to complete this phase.

### 5.3 Phase 3: Convex Change of Plane

At the end of phase 2, the  $Z$  axes of the frames attached to the feet are parallel to the beam  $b_2$  and point in the same direction. Hence, the PSIK problem can be solved to determine the joint coordinates that permit performing a convex transition from the face  $f_2$  to the face  $f_3$  (the face  $f_3$  is defined in Fig. 9). Substituting  $\mu = 11$  cm,  $\omega = 3\pi/4$  rad, and  $j = B$  in Eqs. (24)–(29), and following the procedure detailed in Sect. 5.2, we can obtain the region of the  $(y_{1B}, y_{2B})$  plane where  $l_{iB}, r_{iB} \in [19, 25]$  cm ( $i = 1, 2$ ). It can be checked that the solution



**Fig. 9** A trajectory that includes a convex transition between different planes

**Table 3** Sequence of movements in the third phase of the simulated trajectory

Step	Joint coordinates	Description of the movements in each step
1	$l_{iA} = r_{iA} = 21$ cm $l_{2B} = r_{2B} = 21$ cm $l_{1B} = r_{1B} = 19$ cm	Place both legs perpendicular to the face $f_3$ , leaving some distance between the foot B and the face $f_3$ (Fig. 9b)
2	$\theta_A = 3\pi/2$ rad	Rotate the robot about the leg A (Fig. 9c)
3	$r_{1B} = l_{1B} = 21$ cm	Extend the actuators connected to the foot B until it touches the face $f_3$ (Fig. 9d)
4	$r_{1A} = l_{1A} = 19$ cm	Attach the foot B to the face $f_3$ . Release and lift the foot A retracting the actuators connected to it (Fig. 9e)
5	$\theta_A = \pi$ rad	Rotate the leg A about its own axis (Fig. 9f)

adopted in the previous section ( $y_{1B} = y_{2B} = 22$  cm) is also valid here, obtaining in this case:  $r_{1B} \approx 24.85374622$ ,  $l_{1B} \approx 19.20940403$ ,  $r_{2B} \approx 21.99688208$ , and  $l_{2B} \approx 22.00311791$  (all in cm). For these values of the joint coordinates, the robot can perform a transition between the faces  $f_2$  and  $f_3$  (see Fig. 9a). After that, the foot A can be attached to the face  $f_3$ .

After attaching the foot A to the face  $f_3$ , the sequence of movements described in Table 3 is executed. After executing this sequence, solving exactly the same PSIK problem as in Sect. 5.2 permits the foot A of the robot to be attached to the face  $f_4$  of the beam  $b_3$ , which completes the trajectory.

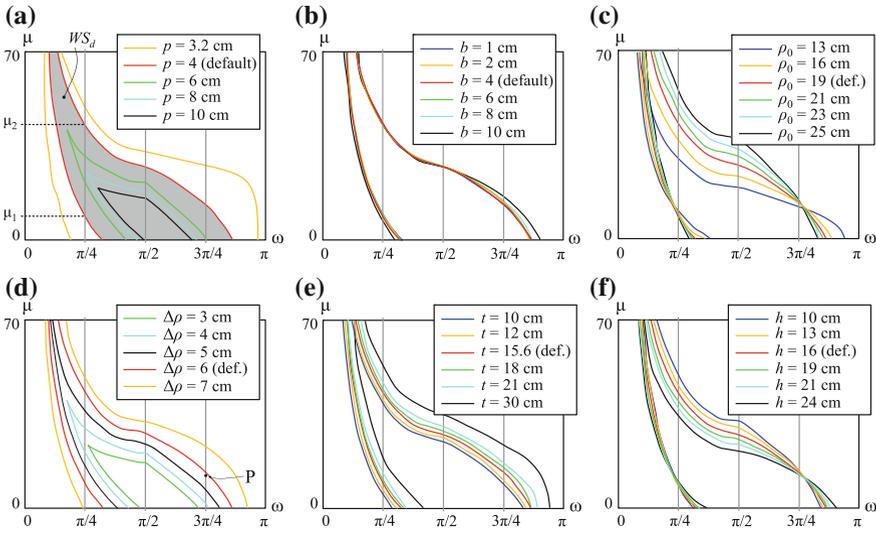
## 6 Workspace of the PSIK Problem

In the previous section, it has been shown that the robot can explore orthogonal 3D structures combining the following three basic movements: (1) moving along a beam using the rotations  $\{\theta_A, \theta_B\}$  of the hip (Fig. 6), (2) performing a concave transition between two beams (Fig. 7), and (3) performing a convex transition between two faces of the same beam (Fig. 9). Moreover, the robot can perform the last two movements adopting planar and symmetric postures, which simplifies the inverse kinematic problem (PSIK problem defined in Sect. 4). However, this has been shown using an example where the geometry of the robot and of the structure, as well as the initial position of the robot, have particular values that guarantee the existence of solutions to the problem of exploring the structure.

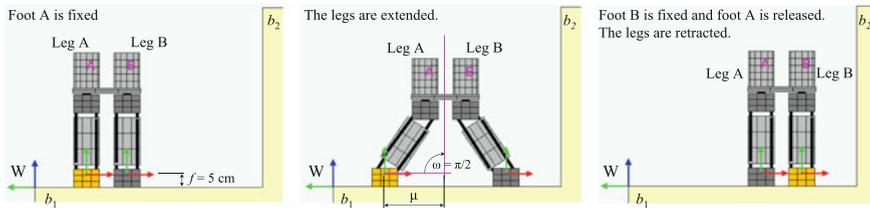
The objective of this section is to determine if the robot will be able to explore 3D structures adopting planar and symmetric postures if its geometry deviates from the design used in Sect. 5, if the robot starts at a different position, or if the geometry of the structure is different. Regarding the geometry of the structure, we will analyze what happens when the width of the cross section of the beams varies, assuming that the structure is still orthogonal. To this end, we will study the workspace of the PSIK problem, which is the set of pairs  $(\omega, \mu)$  (variables defined in Fig. 3) that can be attained satisfying the joint limits. This workspace can be used to assess the ability of the robot to execute the three basic movements defined in the previous paragraph using the planar and symmetric postures of the PSIK problem, for different designs of the robot.

Figure 10a–f show the PSIK workspaces for different choices of the geometric design parameters of the robot, obtained using a Monte Carlo algorithm [1]. Each subfigure of Fig. 10 shows the different PSIK workspaces that are obtained when the corresponding geometric parameter of the robot is varied, keeping the remaining parameters at the default values used in Sect. 5, which are:  $b = p = 4$ ,  $t = 15.6$ ,  $h = 16$ ,  $\Delta\rho = 6$ , and  $\rho_0 = 19$  (all in cm). The curves that have the same color in each subfigure of Fig. 10 enclose the PSIK workspace corresponding to each geometry, i.e., these curves are the boundaries of the workspaces. This is illustrated in Fig. 10a, where the PSIK workspace for the default geometry, denoted by  $WS_d$  and enclosed by the red curves, has been shaded. It is evident from Fig. 10 that the shape of the PSIK workspace is most sensitive to the parameters  $p$  and  $\Delta\rho$ , whereas the shape changes little with  $b$ .

Figures 10a–f also show three vertical lines at  $\omega = \pi/4$ ,  $\omega = \pi/2$ , and  $\omega = 3\pi/4$ . The lines  $\omega = \pi/4$  and  $\omega = 3\pi/4$  represent the concave (Fig. 7a) and convex (Fig. 9a) transitions, respectively. The line  $\omega = \pi/2$  represents a posture that can be used to move the robot along a beam like an inchworm, extending and retracting the legs as shown in Fig. 11. This is an alternative gait to the one shown in Fig. 6, which uses the rotations of the hip to move along a beam. The segment of these vertical lines that lies within the PSIK workspace of a given geometry is the set of lengths  $\mu$  that can be achieved for the corresponding value of  $\omega$ . For example, for the default workspace  $WS_d$  in Fig. 10a and  $\omega = \pi/4$ ,  $\mu$  must be between  $\mu_1$  and  $\mu_2$ . Next, we will analyze



**Fig. 10** PSIK workspaces for different values of the geometric design parameters of the robot. In the axes,  $\mu$  is in cm and  $\omega$  is in rad. **a** Varying  $p$ . **b** Varying  $b$ . **c** Varying  $\rho_0$ . **d** Varying  $\Delta\rho$ . **e** Varying  $t$ . **f** Varying  $h$



**Fig. 11** An inchworm-like gait to move along a beam

Fig. 10 to discuss if the robot can perform convex and concave transitions using the planar and symmetric postures of the PSIK problem in three cases.

**Case 1: The Geometry of the Robot is Modified.** Can the robot perform convex and concave transitions if its geometry is different from the default one? As Fig. 10 shows, the vertical lines  $\omega = \pi/4$  and  $\omega = 3\pi/4$  (which represent the concave and convex transitions) intersect the PSIK workspace for most of the geometries, which means that both transitions are possible. However, for some designs obtained by increasing  $p$  or decreasing  $\Delta\rho$ , the workspace is too small and does not intersect these vertical lines. Thus, these designs should be avoided.

**Case 2: The Initial Position of the Robot is Modified.** Even if the PSIK workspace intersects the vertical line  $\omega = \pi/4$ , the robot may be unable to perform a concave transition between two beams (e.g. to climb to the beam  $b_2$  in the example of Sect. 5.2) if the distance between the robot and the new beam is not adequate. This is because,

according to Fig. 10, for most of the geometries the feasible concave transitions (those obtained intersecting the PSIK workspace and the line  $\omega = \pi/4$ ) require  $\mu$  to be between a minimum value  $\mu_1$  and a maximum value  $\mu_2$  (e.g. see the default PSIK workspace  $WS_d$  in Fig. 10a). As a result, if the robot is too close ( $\mu < \mu_1$ ) or too far ( $\mu > \mu_2$ ) from the beam  $b_2$ , the concave transition will be impossible.

This can be a problem if the distance between the robot and the new beam cannot be adjusted continuously, but only by means of discrete increments, like when using the gait shown in Fig. 6. With this gait, which uses the rotations  $\{\theta_A, \theta_B\}$  of the hip, the robot can only travel a distance equal to an integer multiple of  $t$ . Thus, depending on the initial position of the robot, approaching the beam  $b_2$  using this gait may eventually place the robot at a point too close or too far from the beam, without the possibility to finely adjust its position to make possible the execution of a concave transition.

Nevertheless, this can be avoided if the inchworm-like gait of Fig. 11 is used to approach the beam  $b_2$ . With  $\omega = \pi/2$ , the robot travels a distance equal to  $2\mu$ , where  $\mu$  can take any value from the line  $\omega = \pi/2$  which lies inside the PSIK workspace for the considered geometry. Thus, using this gait permits the robot to continuously adjust its position along the beam  $b_1$  to position itself at a proper distance from the beam  $b_2$  to climb it performing a concave transition.

**Case 3: The Width of the Beams is Modified.** Even if the PSIK workspace intersects the vertical line  $\omega = 3\pi/4$ , the robot may be unable to perform a convex transition between different faces of a beam if the distance  $\mu$  necessary to execute this transition (shown in Fig. 9a) is too large, since the point representing this transition in the  $(\omega, \mu)$  plane may lie outside the PSIK workspace of the considered geometry. For example, the convex transition of Fig. 9a, which is represented by the point P in Fig. 10d, can be performed using the default geometry because P lies inside the default PSIK workspace, but if the stroke of the linear actuators is decreased to  $\Delta\rho = 5$  cm, P will lie outside the workspace and the transition will be infeasible.

As Fig. 9a shows, the distance  $\mu$  necessary to perform a convex transition equals the sum of the size  $f$  of the foot (indicated in Fig. 1a) and 0.5 times the width of the cross section of the beam. This assumes that the fixed foot of the robot is placed at the middle of the cross section of the beam, which will be the most typical case, especially if grippers are used to adhere to the structure. Thus, if  $f$  is too large, and/or the beam is sufficiently wide, the robot may be unable to perform a transition between different faces of a beam.

In case the beams of a given structure are too wide to allow the robot to execute convex transitions between different faces of the beams, the design of the robot should be modified. One possibility is to reduce the size  $f$  of the feet. Another possibility is to modify some of the design parameters of Fig. 10 to increase the maximum value of  $\mu$  that can be attained with  $\omega = 3\pi/4$ . Note that this maximum value changes little with the parameters  $\{b, \rho_0, h\}$ , whereas it increases when increasing  $\{t, \Delta\rho\}$  or decreasing  $p$ .

To sum up, the analysis presented in this section reveals that by choosing an appropriate design for the robot, the planar and symmetric postures of the PSIK

problem provide a relatively high flexibility to explore 3D structures using the basic movements defined in this section. This is partly thanks to the possibility of precisely adjusting the position of the robot along a beam by means of the inchworm-like gait of Fig. 11. This result suggests that the architecture of the robot studied in this chapter may be modified to design a simpler robot based on the PSIK problem, with an actuation scheme in which a single actuator may simultaneously drive various joints, producing symmetric movements. This symmetry in the movements was previously identified by Balaguer et al. [6] as one of the key criteria for designing climbing robots, since this symmetry reduces the number of actuators and hence the overall weight of the robot.

## 7 Conclusion

This chapter has presented the kinematic analysis of a novel biped climbing robot with a hybrid serial-parallel architecture. The forward kinematic problem has been solved, obtaining the relative position and orientation between the feet in terms of the joint coordinates. The inverse problem is more difficult due to the redundancy of the robot. Hence, a simplified inverse problem which considers planar and symmetric postures has been analyzed. Using a developed simulation tool, it has been shown that the simplified problem is sufficient to perform some important movements which are necessary to explore 3D structures. Also, the workspace of this simplified inverse kinematic problem has been analyzed in depth for different designs of the robot. This analysis has shown that the planar and symmetric postures of the simplified problem are also useful when the robot starts at other positions or when the width of the beams varies, provided that the geometric parameters of the robot are chosen properly.

To exploit all the possibilities offered by the proposed kinematic architecture, the general inverse kinematic problem of the robot will be solved in the future. Other problems that will need to be addressed include the determination of the workspace and singularities of the complete robot (not just the PSIK workspace), the dynamic modeling, and the planning of trajectories avoiding collisions in more complex structures. Also, considering the flexibility of the planar and symmetric postures of the PSIK problem, we will optimize the architecture of the robot to obtain a more lightweight robot able to perform these planar and symmetric movements using less actuators, as discussed at the end of Sect. 6.

**Acknowledgments** This work was supported by the Spanish Government through the Ministerio de Educación, Cultura y Deporte under an FPU grant (Ref: FPU13/00413) and through the Ministerio de Economía y Competitividad under Project DPI2013-41557-P.

## References

1. Alciatore, D., Ng, C.: Determining manipulator workspace boundaries using the Monte Carlo method and least squares segmentation. *ASME Robot. Kinemat. Dyn. Controls* **72**, 141–146 (1994)
2. Aracil, R., Saltaren, R.J., Reinoso, O.: A climbing parallel robot: a robot to climb along tubular and metallic structures. *IEEE Robot. Autom. Mag.* **13**(1), 16–22 (2006)
3. Baghani, A., Ahmadabadi, M.N., Harati, A.: Kinematics modeling of a wheel-based pole climbing robot (UT-PCR). In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2099–2104 (2005)
4. Bajd, T., Mihelj, M., Munič, M.: *Introduction to Robotics*. Springer, Netherlands (2013)
5. Balaguer, C., Giménez, A., Pastor, J.M., Padrón, V.M., Abderrahim, M.: A climbing autonomous robot for inspection applications in 3D complex environments. *Robotica* **18**(3), 287–297 (2000)
6. Balaguer, C., Giménez, A., Jardon, A.: Climbing robots' mobility for inspection and maintenance of 3D complex environments. *Auton. Robots* **18**(2), 157–169 (2005)
7. Figliolini, G., Rea, P., Conte, M.: Mechanical design of a novel biped climbing and walking robot. In: Parenti Castelli, V., Schiehlen, W. (ed.) *ROMANSY 18 Robot Design, Dynamics and Control*, CISM International Centre for Mechanical Sciences, vol. 524, pp. 199–206. Springer Vienna (2010)
8. Guan, Y., Jiang, L., Zhu, H., Zhou, X., Cai, C., Wu, W., Li, Z., Zhang, H., Zhang, X.: Climbot: a modular bio-inspired biped climbing robot. In: *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1473–1478 (2011)
9. Kong, X., Gosselin, C.M.: Generation and forward displacement analysis of RPR-PR-RPR analytic planar parallel manipulators. *ASME J. Mech. Design* **124**(2), 294–300 (2002)
10. Mampel, J., Gerlach, K., Schilling, C., Witte, H.: A modular robot climbing on pipe-like structures. In: *Proceedings of the 4th International Conference on Autonomous Robots and Agents*, pp. 87–91 (2009)
11. Schmidt, D., Berns, K.: Climbing robots for maintenance and inspections of vertical structures—a survey of design aspects and technologies. *Robot. Autom. Syst.* **61**(12), 1288–1305 (2013)
12. Shvalb, N., Moshe, B.B., Medina, O.: A real-time motion planning algorithm for a hyper-redundant set of mechanisms. *Robotica* **31**(8), 1327–1335 (2013)
13. Tavakoli, M., Marques, L., De Almeida, A.T.: 3DCLIMBER: climbing and manipulation over 3D structures. *Mechatronics* **21**(1), 48–62 (2011)
14. Tavakoli, M., Viegas, C., Marques, L., Norberto Pires, J., De Almeida, A.T.: OmniClimber-II: an omnidirectional climbing robot with high maneuverability and flexibility to adapt to non-flat surfaces. In: *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, pp. 1349–1354 (2013)
15. Tavakoli, M., Zakerzadeh, M.R., Vossoughi, G.R., Bagheri, S.: A hybrid pole climbing and manipulating robot with minimum DOFs for construction and service applications. *Ind. Robot Inter. J.* **32**(2), 171–178 (2005)
16. Yoon, Y., Rus, D.: Shady3D: a robot that climbs 3D trusses. In: *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 4071–4076 (2007)

# A Standing Assistance Scheme Using a Patient's Physical Strength by a Load Estimation Considering the Muscle Arrangements of a Human Leg

Daisuke Chugo, Satoshi Muramatsu, Sho Yokota and Hiroshi Hashimoto

**Abstract** In this paper, we propose a standing assistance scheme that uses a patient's own physical strength, as evaluated by physical activity estimates. In general, conventional assistive robots do not require patients to use their own physical strength to stand, which leads to decreased strength in the elderly. Therefore, an assistive robot that allows a patient to maximally use their remaining physical strength is desired. Assistive robots can achieve this objective by estimating the physical activity of a patient when they stand. Therefore, the activity estimate proposed here is based on a human musculoskeletal model of a lower limb, which exhibits a biarticular muscle function. The patient generates a natural standing motion using the biarticular muscle function, and the proposed model enables the assistive robot to estimate the patient's physical activity without using biosensors such as electromyographs. Using the proposed estimated results, our prototype assistive robot can assist elderly patients to use their remaining physical strength maximally by selecting a suitable assistive control method.

---

D. Chugo (✉)

School of Science and Technology, Kwansai Gakuin University,  
2-1 Gakuen, Sanda, Hyogo 6691337, Japan  
e-mail: chugo@kwansai.ac.jp

S. Muramatsu

School of Information Science and Technology, Tokai University,  
4-1-1, Kita-Kaname, Hiratsuka, Kanagawa 2591292, Japan  
e-mail: muramatsu@tokai.ac.jp

S. Yokota

Faculty of Science and Engineering, Toyo University,  
2100, Kujirai, Kawagoe, Saitama 3508585, Japan  
e-mail: s-yokota@toyo.jp

H. Hashimoto

Master Program of Innovation for Design and Engineering,  
Advanced Institute of Industrial Technology, 1-10-40, Higashi-Ooi,  
Shinagawa, Tokyo 1400011, Japan  
e-mail: hashimoto@aiit.ac.jp

© Springer International Publishing Switzerland 2016

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*,  
Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_8

**Keywords** Standing assistance · Musculoskeletal model · Joint traction · Physical activity estimate

## 1 Introduction

The act of standing may be the most serious and important activity in the daily life of an elderly person lacking physical strength [1, 2]. However, assisting elderly individuals in standing is a difficult task for caregivers and can be a primary source of the lumbago that many experience [3]. Therefore, creating a care service robot capable of assisting the elderly when they stand is important, and many such assistive devices have been developed and presented in previous works [4, 5].

In Japan, elderly people requiring assistance in daily life are classified into five different care levels [3], where care level 1 is minor and care level 5 represents a serious condition. Generally, the elderly people whose care level is 1 or 2 have difficulty in standing on their own but are able to perform normal daily life activities if standing assistance is provided. However, in many cases, standing assistance devices provide all the power necessary for the patient to stand and do not allow the patient to use their remaining physical strength. Thus, the patient's physical strength decreases [6]. In fact, between 2002 and 2003, more than 10% of care level 1 patients were subsequently assigned to higher care levels within the next year [3]. Thus, to improve the quality of life of elderly patients with low care levels, assistive robots should use the patient's remaining physical strength. However, no studies have been conducted toward this end.

Therefore, we have developed a novel assistive robot designed to aid patients in using their own physical strength to stand [7]. The robot is based on a walker (a popular assistance device for aged people in normal daily life) and uses a support pad actuated by manipulators with three degrees of freedom (Fig. 1) to assist patients in standing.

To maximally utilize the remaining physical strength of a patient while providing standing assistance, the robot is required to accurately estimate the physical activity of the patient so that it may coordinate its assistive force accordingly. However, generally, it is difficult to conduct such estimates without biosensors such as electromyographs (EMGs); furthermore, physical activity estimates with biosensors, which must be attached to the patient, is impractical because assistance robots should be low-cost and easy to use.

Previous works have proposed physical activity estimates using human models comprising linkages and joints without the use of biosensors [8]. These schemes evaluate the patient's physical activity using joint traction, which is calculated using the kinematic model as an index. However, many muscles generate human body movements, and the maximum amount of traction that muscles can generate changes according to the relative positions of the bones and muscles. Therefore, maximum joint traction is not constant, but changes according to the patient's posture. During

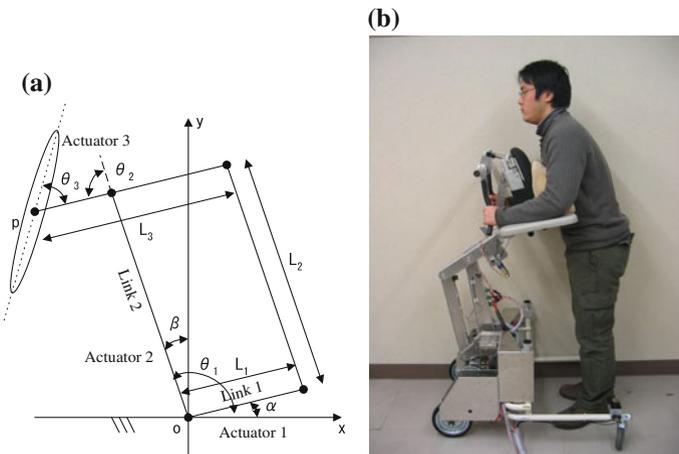
a standing motion, a patient’s posture changes considerably, which should be taken into consideration when evaluating a patient’s physical activity.

Therefore, in this paper, we propose a standing assistance scheme using a patient’s physical strength, which is evaluated by means of real-time physical activity estimates without additional biosensors. The paper is organized as follows: in Sect. 2, we propose the estimate scheme for a patient’s activity according to their posture during standing motion using a human musculoskeletal model of a lower limb, which expresses a biarticular muscle function; in Sect. 3, we propose a standing assistance control scheme on our robot, which uses a patient’s strength based on estimated results; in Sect. 4, we provide experimental results obtained using our prototype; and Sect. 5 concludes this paper.

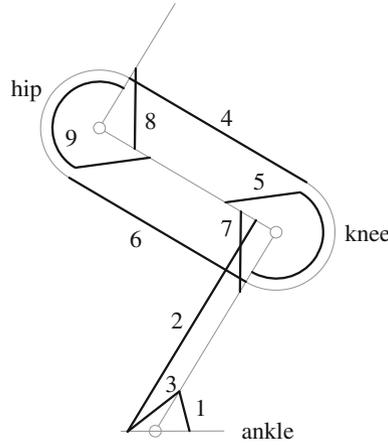
## 2 Physical Activity Estimate

### 2.1 Overview of the Proposed Estimate Scheme

In the linkage model of a human body [9, 10], joint traction is used as an index of a patient’s load. However, this index does not consider the posture of the patient, and in some cases, this index diverges from the experience of nursing specialists, especially when the patient is in a half-sitting posture. When the patient stands, the muscles shown in Fig. 2 generate a lifting motion [11]. Many muscles (shown in Table 1) are used to accomplish the standing movement, and the traction, which muscles can



**Fig. 1** Our developed robot for standing assistance. **a** Frame kinematic model. **b** Overview of our robot



**Fig. 2** Muscle arrangements in the human leg

**Table 1** Human leg muscles

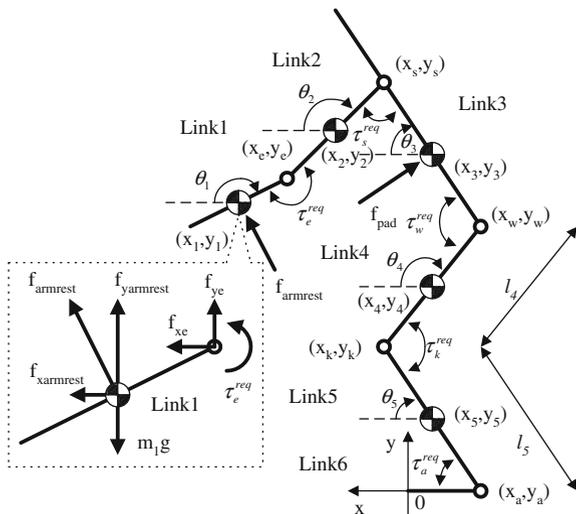
No.	Muscle actuator	Physical areas (cm <sup>2</sup> )
1	Tibialis anterior (TA)	19.7
2	Gastrocnemius (GAS)	99.1
3	Soleus (SOL)	247.6
4	Rectus femoris (RF)	43.5
5	Vastus lateralis (VAS)	248.1
6	Semimembranosus (SM)	60.2
7	Biceps femoris and short head (BFSH)	8.7
8	Iliopsoas (IL)	23.0
9	Gluteus maximus (GMAX)	20.0

maximally generate, changes according to the relative position between frames and muscles.

Thus, we propose a novel physical activity estimate scheme that takes all this into consideration. In this paper, we focus on the traction of the knee and waist joints, which are the main forces propelling patients to stand. Our proposed algorithm is as follows:

- First, we derive the required traction (knee joint  $\tau_k^{req}$  and waist joint  $\tau_k^{req}$ ) to accomplish a standing motion with our assistive robot.
- Second, we derive the maximum traction (knee joint  $\tau_k^{max}$  and waist joint  $\tau_k^{max}$ ) that the muscles can generate for the posture at the present time.
- Comparing the two derived tractions, we evaluate the physical activity of the patient,  $\mu_i$ , which demonstrates how much the patient is required their own phys-

**Fig. 3** Linkage model of a human body



ical strength as compared with their maximum power (1).  $i$  is the identification character (for example, in the case of the knee joint,  $i$  is  $k$ ):

$$\mu_i = \frac{\tau_i^{req}}{\tau_i^{max}}. \tag{1}$$

## 2.2 Derivation of the Required Traction

To estimate the applied load to each joint, we approximate human motion based on the movement of the linkage model on a two-dimensional (2D) plane [9]. Using this model, we can derive the traction of each joint and estimate the patient's load.

The assistance system is designed in such a way that patients lean on a pad and grasp an armrest while standing with our assistance (we will explain our prototype more closely in the next section), which means that our system uses the pad to apply force to the patient's chest and the armrest to apply force to their forearm. These forces move vertically (at the pad) and horizontally (at the armrest). Considering these conditions, we propose a linkage model that approximates the human body with our assistance device (see Fig. 3).

This model consists of six linkages. The armrest applies the assistance force ( $f_{armrest}$ ) to the center position of Link 1 and the support pad applies the force ( $f_{pad}$ ) to the center position of Link 3.  $m_i$  is the mass of the link ( $i = 1, \dots, 6$ ) and  $I_i$  is the moment of inertia.  $(x_i, y_i)$  is the position of the center of gravity on each link, and  $(x_i, y_i)$  ( $i = a, k, w, s,$  and  $e$ ) is the position of each joint. We assume that each linkage is in pillar form with its mass distributed uniformly:

**Table 2** Human body parameters

No.	Name	M (%)	C.G (%)	K (%)	Length (m)
1	Forearm	3.2	41.5	27.9	0.35
2	Humerus	5.4	52.9	26.2	0.39
3	Trunk	57	49.3	34.6	0.48
4	Femur	22	47.5	27.8	0.61
5	Leg	10.2	40.6	27.4	0.56
6	Foot	2.2	59.5	20.4	0.26

M The ratio of the mass of the body segment to that of the body

C.G. The ratio of segmental length, which shows the location of the center of gravity on the longitudinal axis

K The ratio of the gyration radius of the body segment to the length of its segment

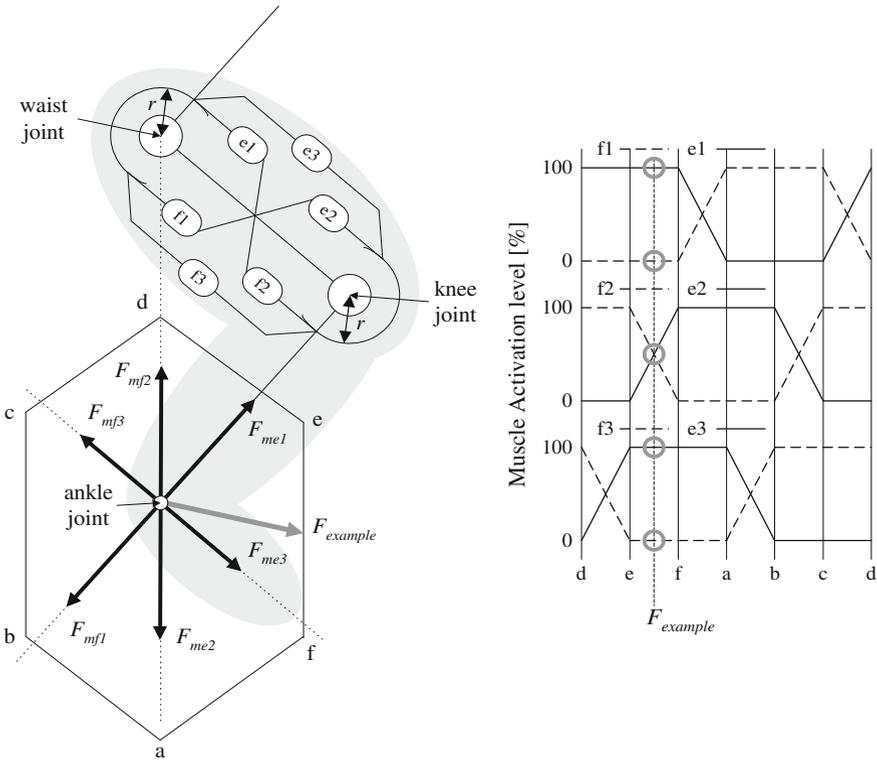
$$\begin{aligned}
\tau_w^{req} = & -(y_s - y_t) f_{xarmrest} - (x_s - x_t) f_{yarmrest} \\
& + (y_w - y_3) f_{xpad} + (x_w - x_3) f_{ypad} \\
& + m_3 \{(y_w - y_3) \ddot{x}_3 - (x_w - x_3) (\ddot{y}_3 - g)\} \\
& + m_2 \{(y_s - y_w) \ddot{x}_2 - (x_s - x_w) (\ddot{y}_2 - g)\} \\
& + m_1 \{(y_s - y_w) \ddot{x}_1 - (x_s - x_w) (\ddot{y}_1 - g)\} + I_3 \ddot{\theta}_3 - \tau_s^{req}
\end{aligned} \quad , \quad (2)$$

$$\begin{aligned}
\tau_k^{req} = & -(y_w - y_k) (f_{xarmrest} + f_{xpad}) \\
& + (x_w - x_k) (f_{yarmrest} + f_{ypad}) \\
& + m_4 \{(y_k - y_4) \ddot{x}_4 - (x_k - x_4) (\ddot{y}_4 - g)\} \\
& + m_3 \{(y_k - y_w) \ddot{x}_3 - (x_k - x_w) (\ddot{y}_3 - g)\} \\
& + m_2 \{(y_k - y_w) \ddot{x}_2 - (x_k - x_w) (\ddot{y}_2 - g)\} \\
& + m_1 \{(y_k - y_w) \ddot{x}_1 - (x_k - x_w) (\ddot{y}_1 - g)\} + I_4 \ddot{\theta}_4 - \tau_w^{req}
\end{aligned} \quad . \quad (3)$$

Here, we use body parameters chosen from a standard body of data from adult Japanese males [10]; see Table 2. To derive the required body parameters for calculating the moment force, we measure the length of each body segment and the mass of the entire body of each individual patient.

We know from previous research [12] that the maximum force that each muscle can realize at the ankle joint is  $F_{me1}$ ,  $F_{me2}$ ,  $F_{me3}$ ,  $F_{mf1}$ ,  $F_{mf2}$ , and  $F_{mf3}$ , and the output distribution of the force at the ankle joint is expressed kinematically as a hexagon (see Fig. 5).

The directions of  $F_{me1}$  and  $F_{mf1}$  are parallel to the leg, the directions of  $F_{me2}$  and  $F_{mf2}$  are parallel to the straight line that connects the waist and ankle joints, and the directions of  $F_{me3}$  and  $F_{mf3}$  are perpendicular to the leg. Furthermore, Oshima et al.'s previous research [12] demonstrates that there is a relationship between the force output vector and the activation level,  $\eta_i$ , of the muscle working in the force output direction. This relationship is shown in Fig. 4, and our system can estimate the



**Fig. 4** Musculoskeletal model considering the role of the antagonistic and biarticular muscles

activation level of each muscle using the output force at the ankle joint. For example, when the output force is  $F_{example}$ , as in Fig. 4, the direction of the force vector is between e and f.

Therefore, the activation levels of each muscle are  $\eta_{e1} = \eta_{e3} = 100(\%)$ ,  $\eta_{f1} = \eta_{f3} = 0(\%)$ , and  $\eta_{e2} = \eta_{f2} = 50(\%)$ , as shown in Fig. 4.

Using this model, we propose a physical activity estimate scheme for a patient according to their posture. First, our system calculates the required traction of the waist joint,  $\tau_w^{req}$ , and of the knee joint,  $\tau_k^{req}$ , using Eqs. (2) and (3), respectively. From the kinematic relationship shown in Fig. 4, the force output vector  $(f_x, f_y)$  at the ankle joint is derived as

$$\begin{aligned} & \begin{vmatrix} \tau_w^{req} \\ \tau_k^{req} \end{vmatrix} \\ &= \begin{vmatrix} l_5 \sin \theta_5 + l_4 \sin \theta_4 & -(l_5 \cos \theta_5 + l_4 \cos \theta_4) \\ l_5 \sin \theta_5 & -l_5 \cos \theta_5 \end{vmatrix} \cdot \begin{vmatrix} f_x \\ f_y \end{vmatrix}. \end{aligned} \tag{4}$$

Second, our system derives the distribution of the output force at the ankle joint from the patient's posture, and then adapts the force output vector  $(f_x, f_y)$  derived from (4) to the hexagon from Fig. 4, which expresses the distribution of the output force, and derives the muscle activation level,  $\eta_i$ , at this time.

We know from previous research [13] that the maximum force,  $F_i^{\max}$ , that a muscle can generate is

$$F_i^{\max} = A_i \sigma, \quad (5)$$

where  $A_i$  is the cross-sectional area of each muscle and  $\sigma$  is the maximum force that the muscle can generate per unit area. In this study, we set  $\sigma = 50(\text{N}/\text{cm}^2)$  [12] and use the values shown in Table 1 for the cross-sectional area of each muscle [11].  $i$  is the identification number of the muscle.

When the muscle activation level is  $\eta_i$ , the maximum traction outputs of the waist joint,  $\tau_w^{\max}$ , and the knee joint,  $\tau_k^{\max}$ , that the muscle can generate with the posture at a given time is derived as

$$\begin{aligned} \tau_w^{\max} = & (\eta_{e1} F_{e1}^{\max} - \eta_{f1} F_{f1}^{\max}) r \\ & + (\eta_{e3} F_{e3}^{\max} - \eta_{f3} F_{f3}^{\max}) r, \end{aligned} \quad (6)$$

$$\begin{aligned} \tau_k^{\max} = & (\eta_{e2} F_{e2}^{\max} - \eta_{f2} F_{f2}^{\max}) r \\ & + (\eta_{e3} F_{e3}^{\max} - \eta_{f3} F_{f3}^{\max}) r, \end{aligned} \quad (7)$$

where  $r$  is the moment arm of each joint [14].  $\tau_w^{\max}$  and  $\tau_k^{\max}$  change according to the relative position between muscles and frames, which means that they reflect the posture of the patient.

Using (2), (3), (6), and (7), we can derive the physical activity of the patient,  $\mu_i$ , as (1). If the physical activity (1) is a large value compared with the maximum activity that the muscles can generate, then the load is evaluated as being heavy. Usually, the patient does not use their maximum power, and in this study, we set the threshold showing the capability of the patient as  $\mu^{\max} = 40(\%)$ , which is based on the opinions of the nursing specialists [12].

### 3 Assistance Control

#### 3.1 System Overview

Figure 5a shows our proposed assistance robot. The system consists of a support pad with three DOFs and a walker. The support pad is activated by our new assistance manipulator, which has four parallel linkages [7]. The patient leans on the support pad and grasps the armrest while standing with assistance (see Fig. 1b). In general,

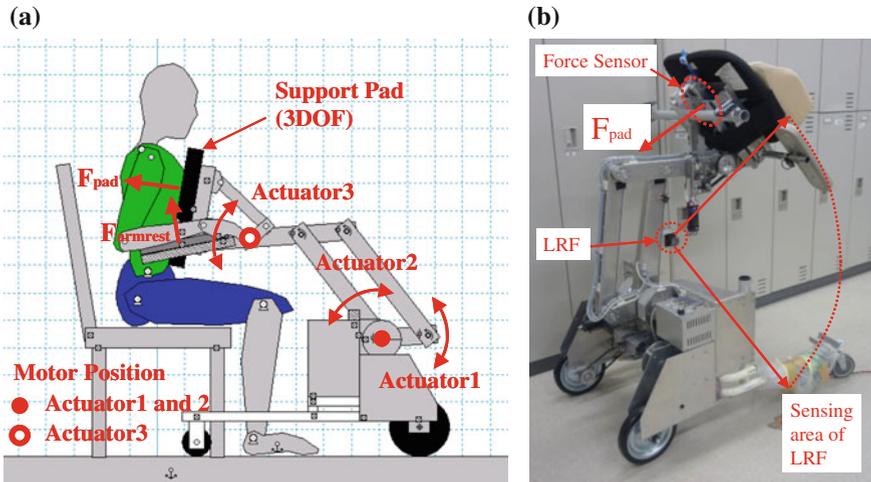


Fig. 5 Prototype of the assistive robot. a The robot’s actuators. b The robot’s sensors

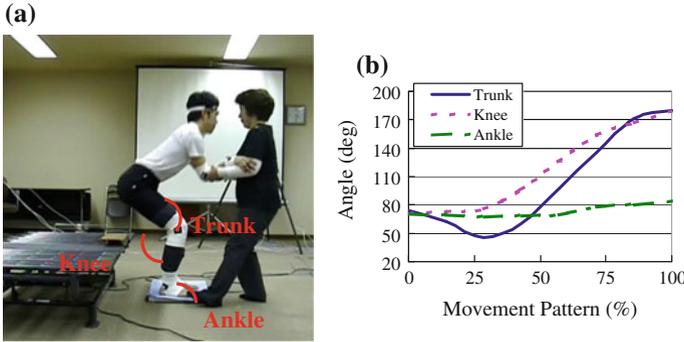
fear of falling forward during the standing motion reduces elderly patients’ standing ability [15]. With the proposed scheme, patients can easily maintain their posture during a standing motion without the fear of falling forward.

Figure 5b shows the prototype of the proposed robot. The prototype is able to lift patients up to 180-cm tall and weighing up to 150 kg. Furthermore, because of its actuated wheels, the prototype can assist patients in walking. To measure a patient’s posture, the prototype has a force sensor and a laser range finder in its body (see Fig. 5b).

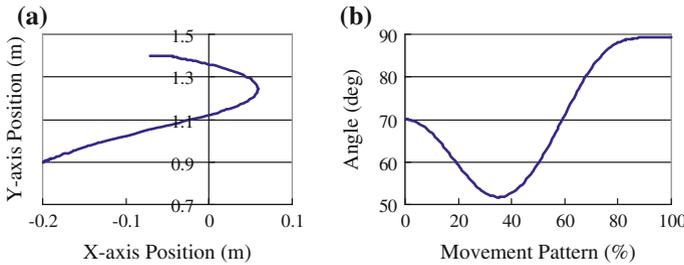
Our physical activity estimate scheme, which is proposed in the previous section, requires real-time data regarding its assistance force and the patient’s posture. To measure its assistance force, our support pad has two force sensors on its body that measure  $F_{pad}$  and  $F_{armrest}$  (see Fig. 5b). To measure the patient’s posture, we use a laser range finder; thus, special markers do not need to be stuck onto the patient for a motion capture systems.

### 3.2 Standing Motion as Recommended by Nursing Specialists

Previous studies have proposed many types of assisted standing. Based on Kamiya’s experience as a nursing specialist, she proposed using the patient’s maximum strength to stand, as shown in Fig. 6. For effective standing assistance, we use a control reference, as shown in Fig. 7 [16]. Figure 7a shows the support pad’s position tracks, and Fig. 7b shows its angle tracks. The movement pattern in Fig. 7b refers to the ratio of the standing motion as determined by (8);



**Fig. 6** Standing motion recommended by nursing specialists. **a** Standing motion. **b** Angular value of each joint



**Fig. 7** Derived control references. The coordination of mechanisms is defined in Fig. 1a. **a** Position of  $P(x_p, y_p)$ . **b** Inclination  $\theta_p$

$$\hat{s} = \frac{t}{t_s}. \tag{8}$$

Here,  $t_s$  is the time required to complete the standing operation, and  $t$  is the present time.

### 3.3 Assistance Control Scheme Based on Physical Activity

In order to use the remaining physical strength of a patient, our assistance system uses a novel combination of damping and position control [17]. Damping control is suitable for controlling objects with contact. From (2) and (3), the assistance force,  $F_y (= f_{yarmrest} + f_{ypad})$ , in the lifting direction will reduce the required traction of each joint ( $\tau_w^{req}$  and  $\tau_k^{req}$ ) because the coefficients of  $F_y$ ,  $-(x_s - x_w)$ , and  $(x_w - x_3)$  in (2) and  $(x_w - x_k)$  in (3) will be negative in the usual standing posture. Therefore, we can expect that the damping control that increases  $F_y$  will reduce the required load of a patient during standing motion.

In our proposed control algorithm, if the physical activity of the patient is great, our system uses the damping control to reduce the patient's load. On the other hand, if the activity of the patient is small, our system uses the position control, which does not assist the force, but uses the remaining physical strength of the patient. In our previous works, our system used joint traction as an index of the patient's load for this algorithm [17]. In this paper, we extended our assistance algorithm using a proposed index of the patient's physical activity defined in (1).

Furthermore, for practical use, our system equips two assistance modes. One is a load-reducing mode. In this mode, our system assists the patient's body to reduce the fixed rate of the physical strength of a knee joint. The patient can set this fixed rate according to their bodily situation and in this paper, and we call this fixed rate the assistance force ratio. This mode is suitable for patients whose required care levels are serious. The other is a rehabilitation mode. In this mode, our system assists the patient's body only when the required physical strength of the patient exceeds that of which they are capable. The patient can set any value as the threshold and this mode is suitable for patients who have enough dexterity to stand up and require limited assistance.

### 3.3.1 Deriving the Reference

Before using the robot for assistance, we measure the height and mass of each patient individually. The length of each body segment is derived based on Table 2 and used by the reference generator as it derives the velocity control reference (9) of each actuator (Nos. 1, 2, and 3) from the motion reference (shown in Fig. 6) using the following equation:

$$\mathbf{v}_i^{ref} = \left[ v_i^{ref}(0), \dots, v_i^{ref}(\hat{s}), \dots, v_i^{ref}(1) \right]^T. \quad (9)$$

Here,  $v_i^{ref}$  is the velocity control reference ( $i = 1, 2, 3$ ), which is a function of the movement pattern  $\hat{s}$  defined in (8). For more details regarding the calculation process, please refer to our previous work [17].

### 3.3.2 Control Algorithm

Our system estimates the physical activity of the patient using the proposed scheme (1) while assisting patients as they stand. Based on this estimate, the system selects a suitable control scheme for damping and position controls. For this to happen, the output of each actuator is derived using

$$v_i = v_i^{ref} - B(F_y - F_{y0}) - K(x_i - x_i^{ref}), \quad (10)$$

where  $F_y (= f_{yarmrest} + f_{ypad})$  is the force applied to the vertical direction on the support pad and armrest.  $x_i^{ref}$  is the angular position reference derived from (9), and  $x_i$  is the actual angular position.  $v_i$  is the updated reference value that our system inputs to the motor controller during the assisted standing motion.  $F_{y0}$  is the coefficient and force that the patient applies to the support pad while he or she stands. Using (10), our system can switch between the position control mode and the damping control mode.

### 3.3.3 Controller's Parameter Coordination

$B$  and  $K$  are constants used to coordinate the ratio between the damping and position controls. Our system applies the damping control mode when the estimated physical activity of each joint  $\mu_i$ , which is defined in (1), exceeds the threshold,  $\mu^{max}$ .  $\mu^{max}$  is derived in (11) in the case a in the load-reducing mode:

$$\mu^{max} = \max \{ \mu_k, \dots, \mu_j \}. \quad (11)$$

In the case of the rehabilitation mode, the patient may set the coefficient according to their own body situation:

$$\mu_i^{max} = (1 - r) \mu_i; \quad (12)$$

where  $r$  ( $0 \leq r \leq 1$ ) is the assistance force ratio, which the patient can set.

To apply the damping control mode when the estimated physical activation,  $\mu_i$ , exceeds  $\mu^{max}$ , the coefficient  $B$  that validates the damping control mode is derived in (13):

$$\begin{cases} B = b (\mu_i - \mu^{max}) & \text{if } (\mu_i \geq \mu^{max}) \\ B = 0 & \text{if } (\mu_i < \mu^{max}) \end{cases} \quad (13)$$

On the other hand, the position control mode is always useful because it helps the patient to maintain stable posture during motion. Therefore, we set the coefficient,  $K$ , which validates the position control mode to a constant. Please note that the values of  $b$  and  $K$  are derived experimentally.

## 4 Experiments

### 4.1 Experimental Setup

To verify the effectiveness of our proposed scheme, eight subjects test the prototype robot based on the proposed estimate scheme. Two subjects (Subjects A and B) are young students and four subjects (Subjects C–F) are 54–72 years old with care levels of 1 or 2. Two subjects (Subjects G and H) are hemiplegics aged 32 and 64 years.

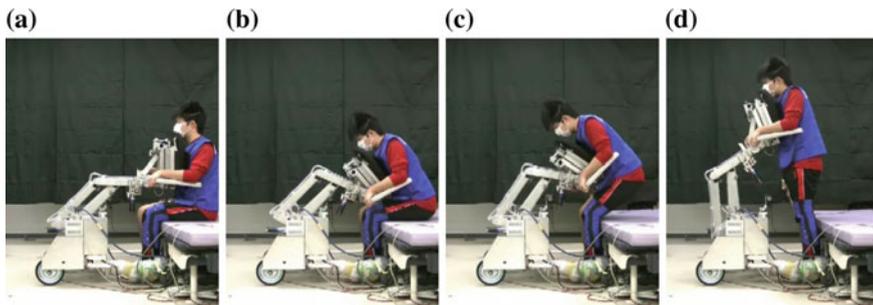
The young subjects (Subjects A and B) wear special clothing designed to limit their motion in order to simulate an elderly person’s limited mobility [18].

Unless otherwise noted, each subject tests the following three cases five times. In Case 1, the robot assists with the standing motion using only the position control mode. Only subjects A and B test this case because the robot does not assist with force and the subject is required to stand using only their own physical strength. In Case 2, the robot assists the subject using our proposed scheme. In this case, the robot uses the force control mode when the subject’s physical activity exceeds their capability threshold. In case 2A (rehabilitation mode), we set the assistance ratio to 30(%), and in case 2B (rehabilitation mode) we set the threshold of the subject’s capability to  $\mu^{\max} = 40\%$  based on the opinion of nursing specialists [12]. In Case 3, the robot assists the subject with the force control mode as necessary, similar to Case 2. The difference between Cases 2 and 3 is that in Case 3, the robot estimates the physical activity of the subject using joint traction, as in our previous work [17]. In this case, we set the threshold of the subject’s capability as  $\tau_{prev}^{\max} = 0.5(\text{Nm/kg})$  based on previous research [19].

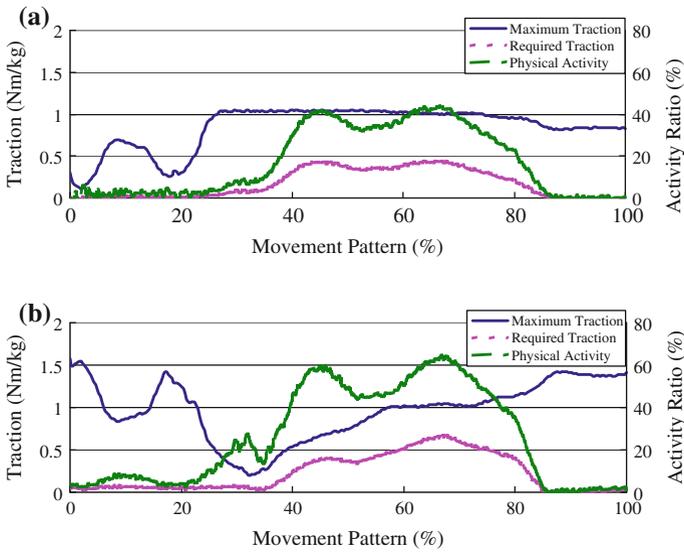
In all cases, we use the standing motion recommended by nursing specialists [16], as specified in Sect. 3.2.

## 4.2 Experimental Results

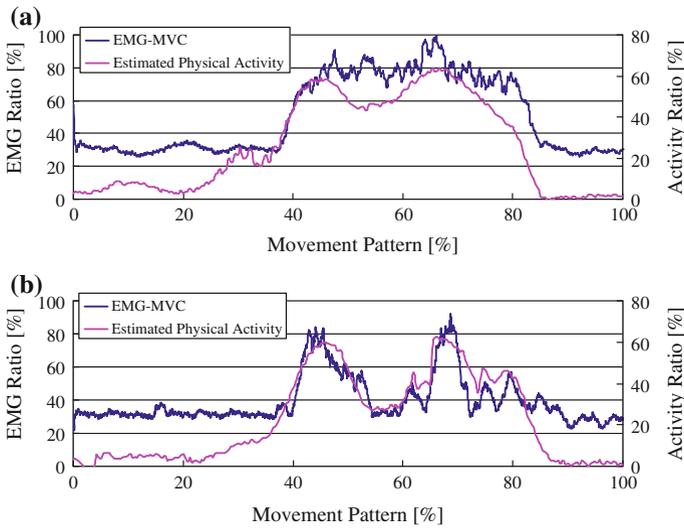
The subject stands up as shown in Fig. 8. Figure 9 shows the required traction,  $\tau_i^{req}$ , the maximum traction,  $\tau_i^{\max}$  (defined in (2), (3), (6), and (7)), and the estimated physical activity of the subject,  $\mu_i$  (defined in (1)) for each joint. As Fig. 8 shows, there are different tendencies between  $\tau_i^{\max}$  and  $\mu_i$ . The estimated load  $\mu_i$  increases—especially at 40–75% movement in a knee joint, around which time the subject lifts their upper body and their load tends to be heavy. This result is similar to the experiences of nursing specialists [8].



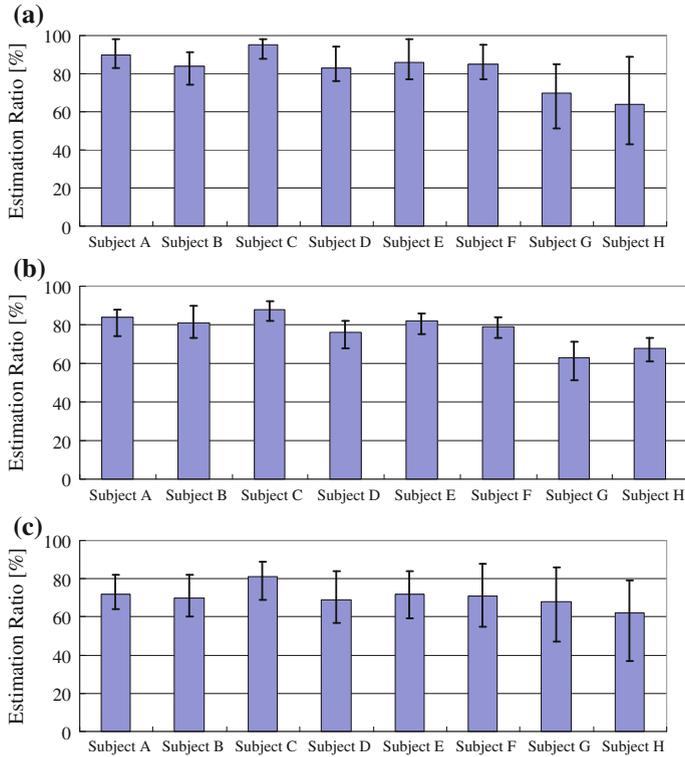
**Fig. 8** Standing motion with our assistance robot (Case 1, Subject A). **a** 0(%). **b** 30(%). **c** 60(%). **d** 100(%)



**Fig. 9** A required traction, a maximum traction, and the estimated physical activity. (Case 1, Subject A). **a** A waist joint. **b** A knee joint



**Fig. 10** The estimated physical activity and the measured muscle activity during a standing motion (Subject A). **a** Case 1 (without force assistance). **b** Case 2 (with force assistance)



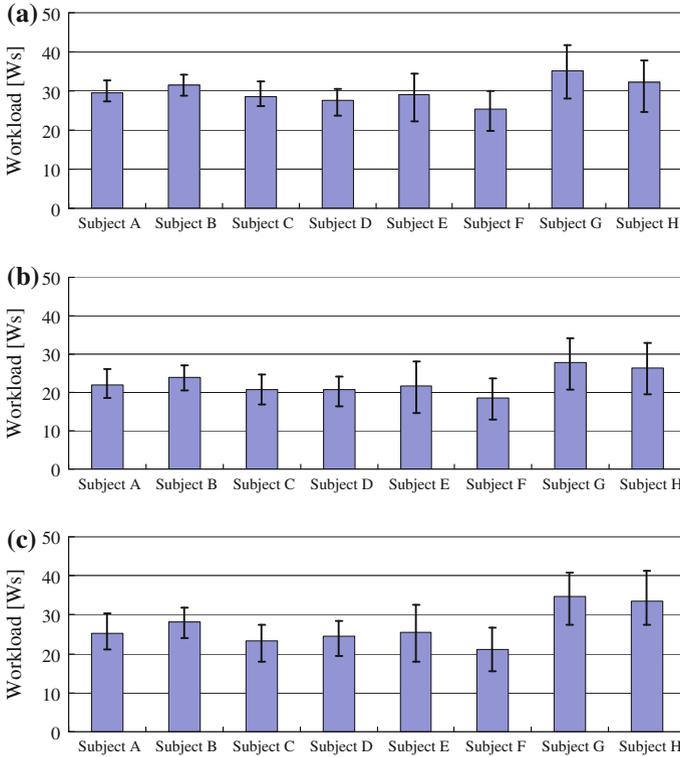
**Fig. 11** Correct estimate ratio of the physical activity. **a** Case 2A (with proposed estimate, rehabilitation mode). **b** Case 2B (with proposed estimate, load-reducing mode). **c** Case 3 (with previous scheme)

Furthermore, Fig. 10 shows the EMG data of a vastus lateralis (VAS) muscle that is normalized by maximum voluntary contraction. This data reflects the activity of the knee joint. The activity of the VAS muscle in Fig. 10a has the same tendency as our proposed load-estimate index. In Fig. 10b, the estimated load exceeds the threshold ( $\mu^{\max} = 40(\%)$ ), and our robot assists with force for the standing motion. Therefore, the load of the subject decreases during the knees' 40–75% movement. These results show that our proposed load estimate scheme is effective.

Figure 11 shows the ratio,  $\rho$ , which shows the correct answer rate of the estimated physical activity from (14):

$$\rho = \frac{t_{match}}{t_s}, \quad (14)$$

where  $t_s$  is the time required to complete the standing operation and  $t_{match}$  is the time at which the estimated physical activity exceeds the threshold  $\mu^{\max}$  and the measured muscle activity exceeds this threshold as well.

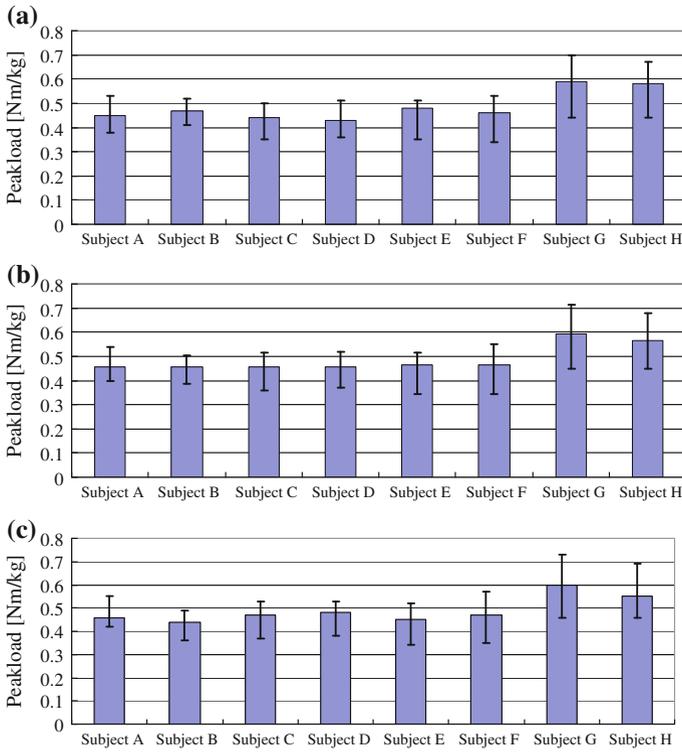


**Fig. 12** Workload of the knee joint. **a** Case 2A (with proposed estimate, rehabilitation mode). **b** Case 2B (with proposed estimate, load-reducing mode). **c** Case 3 (with previous scheme)

In Case 2A (Fig. 11a), our system uses the rehabilitation mode and the proposed activity ratio  $\mu^{\max} = 40(\%)$  as an index of high physical activity; in Case 2B (Fig. 11b), our system uses the load-reducing mode with assistance ratio  $r = 0.3$  as the index; in Case 3 (Fig. 11c), our system uses joint traction  $\tau_{prev}^{\max} = 0.5(\text{Nm/kg})$  as the index.

These results show that our proposed physical activity estimate scheme (Cases 2A and 2B) is more accurate than the previous index using joint traction (Case 3). Two subjects (Subjects G and H) are hemiplegics and the estimate results for both cases are inaccurate because their standing motions were different from the motion recommended by nursing specialists [16]; therefore, different muscles may be used when they stand up. Future work will discuss the muscle model for hemiplegics.

Using the estimated physical activity of the subject, our robot assists with force control only when necessary under the rehabilitation mode in Case 2A and the load-reducing mode in Case 2B. As a result, Fig. 12 shows the maximum traction output,  $\tau_{knee}^{req}$  (peak load), which the subject is required to output to stand completely and Fig. 13 shows the required output power for one standing motion of a knee joint.



**Fig. 13** Peak load of the knee joint. **a** Case 2A (with proposed estimate, rehabilitation mode). **b** Case 2B (with proposed estimate, load-reducing mode). **c** Case 3 (with previous scheme)

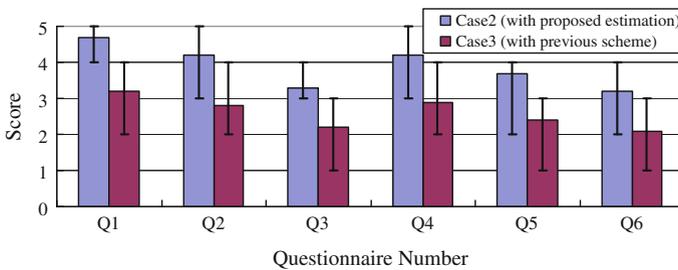
From Fig. 12a, c, we see that the workload in Case 2A is larger than that in Case 3, which means that the subject uses more physical strength in the rehabilitation mode with our proposed load-estimate (Case 2A). Furthermore, from Fig. 12b, c, we see that the workload in Case 2B is smaller than that in Case 3, which means that our system assists more efficiently in the load-reducing mode under our proposed load-estimate scheme (Case 2B).

On the other hand, from Fig. 13a–c, we see that the peak load is almost the same and does not exceed the capability of the subject,  $\tau_{prev}^{max} = 0.5(Nm/kg)$ , which means that our robot assists with enough force when necessary. These results show that our proposed load-estimate method allows the robot to assist with standing in such a way that the subject’s remaining physical strength is used as much as possible.

Finally, to investigate whether subjects find our assistance scheme oppressive, we administer a questionnaire survey to all subjects, as shown in Fig. 14. This Japanese questionnaire form is proposed by [20]. Figure 15 shows the questionnaire results. From Fig. 15, we find that, using our proposed idea, the feeling of oppression arising from a standing assistance system seems to be reduced. This means that our assistance

①	不安な	← 1	— 2	— 3	— 4	— 5	→	安心な
②	不快な	← 1	— 2	— 3	— 4	— 5	→	快適な
③	親しみを感じない	← 1	— 2	— 3	— 4	— 5	→	親しみを感じる
④	頼もしくない	← 1	— 2	— 3	— 4	— 5	→	頼もしい
⑤	恐怖を感じる	← 1	— 2	— 3	— 4	— 5	→	恐怖を感じない
⑥	違和感を感じる	← 1	— 2	— 3	— 4	— 5	→	違和感がない

**Fig. 14** Questionnaire form. [不安な] means uneasy and [安心な] means safe. [不快な] means uncomfortable and [快適な] means comfortable. [親しみを感じない] means the subject does not feel familiarity and [親しみを感じる] means he feels it. [頼もしくない] means the subject feels the robot is not reliable and [頼もしい] means he feels it is reliable. [恐怖を感じる] means the subject feels fear and [恐怖を感じない] means he does not feel it. [違和感を感じる] means the subject feels incongruity and [違和感を感じない] means he does not feel it



**Fig. 15** Questionnaire results of the 8 subjects (A–H)

fits the condition of using a subject’s physical strength, and that subject does not feel fear of falling or other feelings of oppression. Thus, the proposed system succeeds in leading the standing motion by the subject.

## 5 Conclusions

This paper proposes both a physical activity estimate scheme that considers muscle arrangements and a novel assistance system that uses results of such estimates to take advantage of a patient’s remaining physical strength in such a way as to prevent their muscular strength from declining over time. By using our proposed scheme, our system can reduce a patient’s load when the patient’s posture is such that it is difficult to use any of the patient’s own physical strength.

In our system, the subject is required to set parameters, such as the cross-sectional area of each muscle. Previous researchers have proposed a derivation method of these values using easy gymnastics [11]. We plan to develop an automatic individual parameter derivation scheme in future work.

**Acknowledgments** This research is supported in part by Grant-in-Aid for Scientific Research C (25350693) from the Japan Society for the Promotion of Science (JSPS).

## References

1. Alexander, N.B., Schultz, A.B., Warwick, D.N.: Rising from a chair: effects of age and functional ability on performance biomechanics. *J. Geom.: Med. Sci.* **46**(3), M91–98 (1991)
2. Hughes, M.A., Schenkman, M.L.: Chair rise strategy in the functionally impaired elderly. *J. Rehabil. Res. Dev.* **33**(4), 409–412 (1996)
3. Cabinet Office, Government of Japan, KOUREISHA HAKUSHO (The whitepaper on the aged society), p. 25 (2011). ISBN: 4904681010 (in Japanese)
4. Nagai, K., Nakanishi, I., Hanabusa, H.: Assistance of self-transfer of patients using a power-assisting device. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4008–4015 (2003)
5. Funakubo, A., Tanishiro, H., Fukui, Y.: Power assist system for transfer aid. *J. Soc. Instrum. Control Eng.* **40**(5), 391–395 (2001)
6. Hirvensalo, M., Rantanen, T., Heikkinen, E.: Mobility difficulties and physical activity as predictors of mortality and loss of independence in the community-living older population. *J. Am. Geriatr. Soc.* **48**, 493–498 (2000)
7. Chugo, D., Morita, Y., Sakaida, Y., Yokota, S., Kobayashi, H., Hashimoto, H., Takase, K.: Standing assistance control using a physical strength of a patient with load estimation. In: *Proceedings of 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 234–239 (2012)
8. Nuzik, S., Lamb, R., Vansant, A., Hirt, S.: Sit-to-stand movement pattern. A kinematic study. *Phys. Ther.* **66**(11), 1708–1713 (1986)
9. Hatsukari, T., Kuroko, S., Miyake, N., Kawazoe, R., Higuchi, J., Hirata, Y., Kosuge, K.: Self-help standing-up method based on quasi-static motion. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 342–347 (2009)
10. Nishida, I., Maeda, M., Kawano, T., Shirase, K.: Estimation method of muscle forces of lower limb considering the role of antagonistic muscles and biarticular muscles-estimation of muscle forces of lower limb during vertical jumping. *J. Jpn. Ergonomics Soc.* **47**(6), 244–251 (2011)
11. Okada, H., Ae, M., Fujii, N., Morioka, Y.: Body segment inertia properties of Japanese elderly. *Biomechanisms* **13**, 125–139 (1996)
12. Oshima, T., Fujikawa, T., Kumamoto, M.: Functional evaluation of effective muscle strength based on a muscle coordinate system consisted of bi-articular and mono-articular muscles-contraction forces and output forces of human limbs. *J. Precis. Eng.* **65**(12), 1772–1777 (1999)
13. Spector, S.A., Gardiner, P.F., Zernicke, R.F., Roy, R.R., Edgerton, V.R.: Muscle architecture and force-velocity characteristics of cat soleus and medial gastrocnemius: implications for neural control. *J. Neuro-physiol.* **44**, 951–960 (1980)
14. Hoy, M.G., Zajac, F.E., Gordon, M.E.: A musculoskeletal model of the human lower extremity: The effect of muscle, tendon, and moment arm on the moment-angle relationship of musculotendon actuators at the hip, knee, and ankle. *J. Biomech.* **23**(2), 157–169 (1990)
15. Maki, E., Holliday, P.J., Topper, A.K.: Fear of falling and postural performance in the elderly. *J. Gerontol.* **46**(4), 123–131 (1991)
16. Kamiya, K.: Development and evaluation of life support technology in nursing. In: *Proceedings of 7th RACE Symposium Research into Intelligent Artifacts for the Generalization of Engineering*, pp. 116–121 (2005)
17. Chugo, D., Matsuoka, W., Songmin, J., Takase, K.: Rehabilitation walker with standing-assistance device. *J. Robot. Mechatronics* **19**(6), 604–611 (2007)
18. Takeda, K., Kanemitsu, Y., Futouy, Y.: Understanding the problem of the elderly through a simulation experience-difference in the effect between before and after clinical practice. *Kawasaki Medical Welfare J.* **11**(1), 64–73 (2001)

19. Omori, K., Yamazaki, Y., Yokoyama, H., Aoki, U., Kasahara, M., Hiraki, K.: The relationship between strength in the lower extremity and the ability to stand up from a chair in elderly inpatients. *Sogo Rehabil.* **30**(2), 167–171 (2001)
20. Matsui, Y., Kanoh, M., Kato, S., Itoh, H.: Generating interactive facial expressions of kansei robots using simple recurrent network. *J. Robot. Soc. Jpn.* **28**(3), 360–368 (2010) (in Japanese)

# Multimodal Image Registration and Visual Servoing

M. Ourak, B. Tamadazte, N. Andreff and E. Marchand

**Abstract** This paper deals with multimodal imaging in the surgical robotics context. On the first hand, it addresses numerical registration of a preoperative image obtained by fluorescence with an intraoperative image grabbed by a conventional white-light endoscope. This registration involves displacement and rotation in the image plane as well as a scale factor. On the second hand, a method is developed to visually servo the endoscope to the preoperative imaging location. Both methods are original and dually based on the use of mutual information between a pair of fluorescence and white-light images and of a modified Nelder-Mead simplex algorithm. Numerical registration is validated on real images whereas visual servoing is validated experimentally in two set-ups: a planar microrobotic platform and a 6DOF parallel robot.

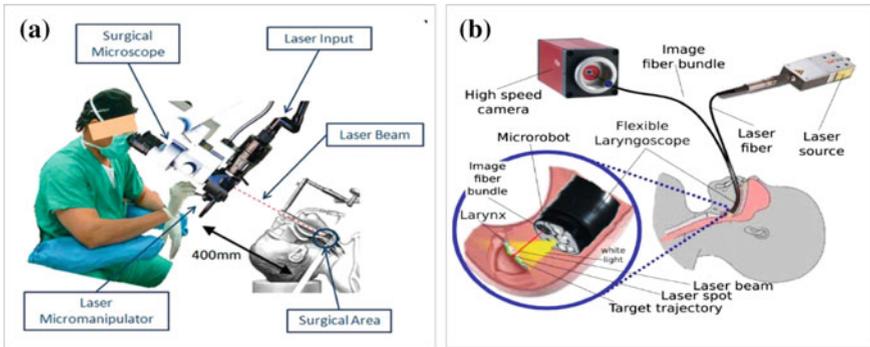
## 1 Introduction

This work is grounded into robot assisted laser phonosurgery (RALP). The current gold standard procedure for the vocal folds surgery is certainly suspension micro-laryngoscopy (Fig. 1a) which requires direct visualization of the larynx and the trachea as proposed in [9]. This system is widely deployed in hospitals but it features many drawbacks related to patient and staff safety and comfort. Therefore, alternative endoscopic approaches are under investigation: the extended use of the HARP (Highly Articulated Robotic Probe) highly flexible robot, designed for conventional surgery [6] or the use of an endoscopic laser micro-manipulator [17] (Fig. 1b). In all aforementioned cases, cancer diagnosis can be performed thanks to fluorescence

---

M. Ourak (✉) · B. Tamadazte · N. Andreff  
FEMTO-ST Institute, AS2M Department, Université Bourgogne  
Franche-Comté/CNRS/ENSMM, 24 Rue Savary, 25000 Besançon, France  
e-mail: mouloud.ourak@femto-st.fr

E. Marchand  
Université de Rennes 1, IRISA, Rennes, France



**Fig. 1** Global view of the microphonosurgery system: **a** the current laser microphonosurgery system and **b** the targeted final system

imaging [16], (a few) days before the surgical intervention. The latter is usually performed under white-light conditions because fluorescence may require longer exposure time than real time can allow. Therefore, during a surgical intervention the fluorescence diagnosis image must be registered to the real-time white light images grabbed by the endoscopic system in order to define the incision path of the laser ablation or resection. Registration can be done either numerically or by physically servoing the endoscope to the place where the preoperative fluorescence image was grabbed.

In this paper, our aim is to control a robot based on direct visual servoing, i.e. using image information coming from white light and fluorescence sensors. Several visual servoing approaches based on the use of features (line, Region of interest (ROI)) [2] or the image global information (gradient [11], photometry [3] or mutual information [5]) can be used. Nevertheless, the use of mutual information (MI) in visual servoing problems has proved to be especially effective in the case of multimodal and less contrasted images [4]. In fact, these control techniques assume that the kinematic model of the robot and the camera intrinsic parameters are at least partially known, but would fail if the system parameters were fully unknown. In practice, the initial position cannot be very distant from the desired position to ensure convergence. To enlarge the stability domain, [12] proposed to use the Simplex method [13] instead of the usual gradient-like methods (which require at least a rough calibration of the camera and a computation of the camera/robot transformation). However, the work in [12] relies on the extraction from the image of geometrical visual features.

Furthermore, in the surgical robotics context, it is preferable to free ourselves from any calibration procedure (camera, robot or robot/camera system) for several reasons:

1. Calibration procedures are often difficult to perform, especially by non-specialist operators i.e., clinicians.

2. Surgeons entering in the operating room are perfectly sterilized to avoid any risk of contamination, and then it is highly recommended to limit the manipulation of the different devices inside the operating room.

For these reasons, we opted for uncalibrated and model-free multimodal registration and visual servoing schemes using mutual information as a global visual feature and a Simplex as optimization approach. Thereby, it is not necessary to compute the interaction matrix (Jacobian image); the kinematic model of the robot may be totally unknown, without any constraint in the initial position of the robot with respect to its desired position. A preliminary version of this work was presented in [14] in the case of planar positioning and is extended in this paper to positioning in the 3D space.

This paper is structured as follows: Sect. 2 explains the medical application of the proposed approach. Section 3 gives the basic background on mutual information. Section 4 presents a modified Simplex method. Section 5 describes multimodal registration and multimodal visual servoing. Finally, Sect. 6 deals with the validation results.

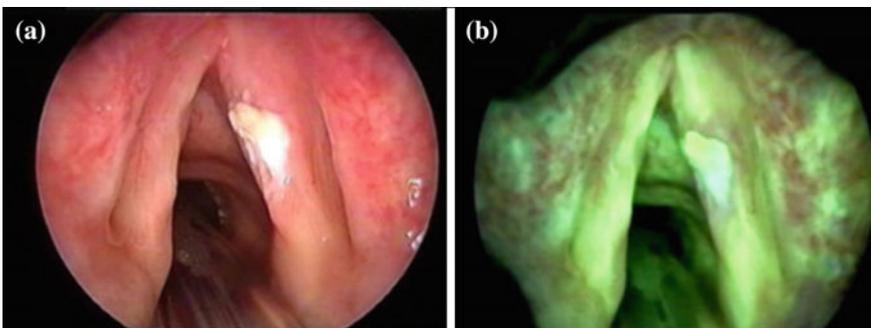
## 2 Medical Application

The vocal folds are situated at the center and across the larynx and form a V-shaped structure. They are used to create the phonation by modulating the air flow being expelled from the lungs through quasi-periodic vibrations. They can be affected by benign lesions, such as cysts or nodules (for instance, when they are highly stressed, e.g. when singing) or, in the worst case, cancer tumors (especially for smokers). These lesions change the configuration of the folds and thereby the patient's voice. Nowadays, medical tools can be used to suppress this trouble and recover the original voice in particular for cyst and nodules. Appeared in 1960, *phonosurgery*—the surgery of the vocal folds—can be divided into laryngoplastic, laryngeal injection, renovation of the larynx and phonomicrosurgery. Specifically, laser phonomicrosurgery consists of a straight rigid laryngoscope, a stereoscopic microscope, a laser source, and a controlled 2DOF device to orient the laser beam [8], as shown in Fig. 1a. Nevertheless, the current system requires extreme skill from the clinician. Specifically, high dexterity is required because both the laser source is located out of the patient, 400 mm away from the vocal folds. This distance increases the risk of inaccuracy when the laser cutting process is running. Moreover, the uncomfortable position of the patient's neck in a straight position all along the operation can be traumatic. The drawbacks of the conventional procedure are taken into account in the new set-up developed within the European project  $\mu$ RALP, which consists on embedding all the elements (i.e., cameras, laser and guided mirror) inside an endoscope Fig. 1b. More precisely, the endoscope is composed of white light, high speed camera imaging the laser evolution with 3D feedback to the clinician. Additionally, a low framerate, high sensitivity fluorescence imaging system is to be used preoperatively to detect cancerous lesions.

The global approach is based on the use of 2 degrees of freedom (DOF) to guide the laser along the trajectory drawn by the surgeon on a preoperative fluorescence image. However, since the preoperative image is not necessarily taken by the same instrument on the same location, this approach requires the preoperative fluorescence image (where the surgeon decides the trajectory) and the white light image (where the control of the robot is developed) to be registered (Fig. 2). This can be done in two ways: registration or servoing. Registration deals with the estimation of the transformation between both images, which can then be used to morph the fluorescence image onto the real-time endoscopic image flow (for instance, as an augmented reality). Visual servoing deals with bringing the endoscope back to the place where the fluorescence image was grabbed and stabilizing it in that configuration, which amounts to a physical registration and should turn useful in many other applications, such as surgery in the stomach to compensate for physiological motions.

### 3 Mutual Information and Registration

In the literature, multimodal image registration has been widely discussed. Zitova et al. [19] classified registration techniques for medical applications into two main categories: area-based and features-based methods. In these cases, the registration process follows mainly four steps: feature detection, feature matching, transformation estimation, and image resampling. As previously stated, our approach is based on mutual information rather than geometrical visual features. Therefore, the most critical steps (feature detection and matching) of a conventional registration method are removed. Instead, from the joint and marginal entropy of two images, it is possible to compute their similarities. This means that the higher the mutual information is, the better the images are aligned [4].



**Fig. 2** Vocal folds endoscopic images: **a** *white light* endoscopic image, **b** *fluorescence* endoscopic image [18]

### 3.1 Mutual Information in the Image

Mutual information is based on the measure of information, commonly called entropy in 1D signal. By extension, the entropy expression in an image  $\mathbf{I}$  is given by

$$\mathbf{H}(\mathbf{I}) = - \sum_{i=0}^{N_I} p_{\mathbf{I}}(i) \log_2(p_{\mathbf{I}}(i)) \quad (1)$$

where  $\mathbf{H}(\mathbf{I})$  represents the marginal entropy, also called Shannon entropy of an image  $\mathbf{I}$ ;  $i \in [0, N_I]$  (with  $N_I = 255$ ) defines a possible gray value of an image pixel; and  $p_{\mathbf{I}}$  is the probability distribution function, also called marginal probability of  $i$ . This can be estimated using the normalized histogram of  $\mathbf{I}$ .

Moreover, the entropy between two images  $\mathbf{I}_1$  and  $\mathbf{I}_2$  is known as joint entropy  $\mathbf{H}(\mathbf{I}_1, \mathbf{I}_2)$ . It is defined as the joint variability of both images

$$\mathbf{H}(\mathbf{I}_1, \mathbf{I}_2) = - \sum_{i=0}^{N_{I_1}} \sum_{j=0}^{N_{I_2}} p_{\mathbf{I}_1\mathbf{I}_2}(i, j) \log_2(p_{\mathbf{I}_1\mathbf{I}_2}(i, j)) \quad (2)$$

where  $i$  and  $j$  are the pixel intensities of the two images  $\mathbf{I}_1$  and  $\mathbf{I}_2$  respectively; and  $p_{\mathbf{I}_1\mathbf{I}_2}(i, j)$  is the joint probability for each pixel value. The joint probability is accessible by computing the  $(N_{I_1} + 1) \times (N_{I_2} + 1) \times (N_{bin} + 1)$  joint histogram which is built with two axes defining the bin-size representation of the image gray levels and an axis defining the number of occurrences between  $\mathbf{I}_1$  and  $\mathbf{I}_2$ .

From (1) and (2), the mutual information contained in  $\mathbf{I}_1$  and  $\mathbf{I}_2$  is defined as

$$\mathbf{MI}(\mathbf{I}_1, \mathbf{I}_2) = \mathbf{H}(\mathbf{I}_1) + \mathbf{H}(\mathbf{I}_2) - \mathbf{H}(\mathbf{I}_1, \mathbf{I}_2) \quad (3)$$

and can be expressed using the marginal probability  $p_{\mathbf{I}}$  and joint probability  $p_{\mathbf{I}_1\mathbf{I}_2}(i, j)$ , by replacing (1) and (2) in (3) with some mathematical manipulations

$$\mathbf{MI}(\mathbf{I}_1, \mathbf{I}_2) = \sum_{i,j} p_{\mathbf{I}_1\mathbf{I}_2}(i, j) \log \left( \frac{p_{\mathbf{I}_1\mathbf{I}_2}(i, j)}{p_{\mathbf{I}_1}(i)p_{\mathbf{I}_2}(j)} \right) \quad (4)$$

This cost-function has to be maximized if  $\mathbf{I}_1$  and  $\mathbf{I}_2$  are requested to “look like each other”.

In practice, the cost-function computed using (4) is not very smooth. This creates local maxima, hence complicating the convergence optimization process [4]. To reduce the joint histogram space as well as the irregularities in the mutual information, and thereby local maxima (at least for the less significant ones), Dawson et al. [7] proposed to use the *in-Parzen* windowing formulation when computing the mutual information:

$$\mathbf{I}_{b1}(k) = \mathbf{I}_1(k) \frac{N_c}{I_{max}} \quad \text{and} \quad \mathbf{I}_{b2}(k) = \mathbf{I}_2(k) \frac{N_c}{I_{max}} \quad (5)$$

where  $t_{max} = r_{max} = 255$  and  $N_c$  are the new bin-size of the joint histogram and  $\mathbf{I}_{b1}$ ,  $\mathbf{I}_{b2}$  are the new gray level value of  $\mathbf{I}_1$  and  $\mathbf{I}_2$ , respectively.

In addition to re-sampling of the joint histogram, it is advisable to introduce a filtering method based on *B-splines* interpolation in order to further smooth the mutual information cost-function. As far as multimodal images are concern, the abrupt change in the cost-function creating local maxima are *flattened* in order to reduce again these irregularities. In practice, we opted for a third-order interpolation  $\psi$ , which presents a good balance between smoothing quality and time computation. Thereby, both marginal and joint probabilities become

$$p_{\mathbf{I}_{b1}\mathbf{I}_{b2}}(i, j) = \frac{1}{N_k} \sum_k \psi(i - \mathbf{I}_{b1}(k)) \psi(j - \mathbf{I}_{b2}(k)) \quad (6)$$

$$p_{\mathbf{I}_{b1}}(i) = \frac{1}{N_k} \sum_k \psi(i - \mathbf{I}_{b1}(k, x)) \quad (7)$$

$$p_{\mathbf{I}_{b2}}(j) = \frac{1}{N_k} \sum_k \psi(j - \mathbf{I}_{b2}(k)) \quad (8)$$

with  $N_k$  is the number of pixels in the new images  $\mathbf{I}_{b1}$  and  $\mathbf{I}_{b2}$  and  $\psi$  is the used B-spline function.

## 4 Simplex-Based Registration

This section deals with the method for solving the mutual information maximization problem. However, before describing the chosen optimization approach among the many existing ones [10] to solve this problem, it is necessary to know the exact shape of the cost-function in the case of bimodal images (fluorescence vs. white light) of the vocal cords.

In practice, rather than maximizing mutual information, we minimize the cost-function

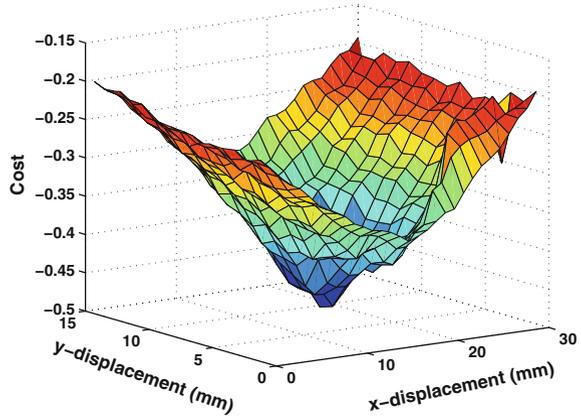
$$\mathbf{f}(\mathbf{r}) = -\mathbf{MI}[\mathbf{I}_{b1}(\mathbf{r}), \mathbf{I}_{b2}] \quad (9)$$

In the general case, because the mutual information depends on a Euclidean displacement (i.e. in  $SE(3)$ ) between both image viewpoints, the problem to solve is

$$\hat{\mathbf{r}} = \arg \min_{\mathbf{r} \in SE(3)} \mathbf{f}(\mathbf{r}) \quad (10)$$

where  $\mathbf{r}$  is the camera pose with respect to the world reference frame, attached to the fluorescence image.

**Fig. 3** MI cost-function in nominal conditions (representation of -MI)

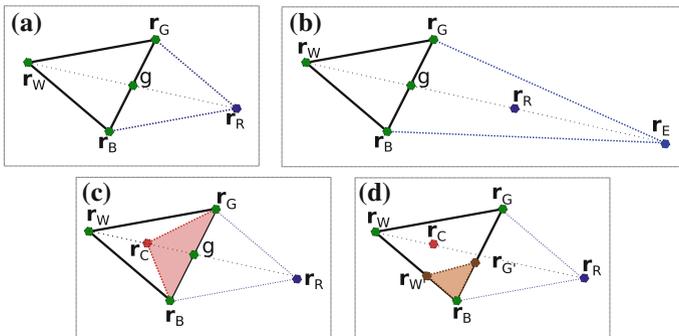


### 4.1 Cost-Function Shape

Figure 3 shows the computed cost-function in nominal conditions (i.e., the high definition images shown in Fig. 8). It has a global convex shape but still has many irregularities. Consequently, derivative based methods such as gradient descent could not necessarily guarantee convergence. Thereby, an unconstrained optimization technique was chosen to overcome this problem, i.e., a modified Simplex algorithm.

### 4.2 Modified Simplex Algorithm

The Nelder-Mead Simplex algorithm [13] roughly works as follows. A Simplex shape  $S$  defined by vertices  $\mathbf{r}_1$  to  $\mathbf{r}_{k+1}$  with  $k = \dim(6)$  is iteratively updated until convergence using four operators: reflection, contraction, expansion, and shrinkage (see Fig. 4), defined on a linear space.



**Fig. 4** Example of the Simplex steps: **a** reflection, **b** expansion, **c** contraction, and **d** shrinkage

In order to apply this algorithm in the non linear Euclidean space, we represent any rigid displacement  $\mathbf{r} \in SE(3)$  as

$$\mathbf{r} = \begin{pmatrix} \mathbf{t} \\ \mathbf{u}\theta \end{pmatrix} \quad \text{such that} \quad [\mathbf{r}] \stackrel{\text{def}}{=} \begin{pmatrix} [\mathbf{u}]_{\wedge} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 0 \end{pmatrix} \stackrel{\text{def}}{=} \text{logm} \mathbf{T} \quad (11)$$

where  $\text{logm}$  is the matrix logarithm and  $\mathbf{T}$  is the  $4 \times 4$  homogeneous matrix representation of  $\mathbf{r}$ .

Thus, the usual four steps of the Simplex  $S$  can be used:

$$\text{reflection} : \mathbf{r}_R = (1 - \alpha)g + \alpha\mathbf{r}_W \quad (12)$$

where  $\mathbf{r}_R$  is the reflection vertex,  $\alpha$  is the reflection coefficient and  $g$  is the centroid between  $\mathbf{r}_G$  and  $\mathbf{r}_B$ .

$$\text{expansion} : \mathbf{r}_E = (1 - \gamma)g + \gamma\mathbf{r}_R \quad (13)$$

where  $\mathbf{r}_E$  is the expansion vertex and  $\gamma$  is the expansion coefficient, and

$$\text{contraction} : \mathbf{r}_C = (1 - \beta)g + \beta\mathbf{r}_W \quad (14)$$

where  $\mathbf{r}_C$  is the contraction vertex, and  $\beta$  is the contraction coefficient.

$$\begin{aligned} \text{shrinkage} : \mathbf{r}'_G &= (\mathbf{r}_G + \mathbf{r}_B)/2 \\ \mathbf{r}'_W &= (\mathbf{r}_W + \mathbf{r}_B)/2 \end{aligned} \quad (15)$$

where the vertices are updated as:  $\mathbf{r}_G = \mathbf{r}'_G$  and  $\mathbf{r}_W = \mathbf{r}'_W$ .

Finally, the algorithm ends when  $\text{val}(S) \leq \varepsilon$  where  $\varepsilon$  is a predefined eligible small distance,  $\text{val}(S)$  is defined as

$$\text{val}(S) = \max \left( \text{dist}(\mathbf{r}_W, \mathbf{r}_B), \text{dist}(\mathbf{r}_W, \mathbf{r}_G), \text{dist}(\mathbf{r}_G, \mathbf{r}_B) \right) \quad (16)$$

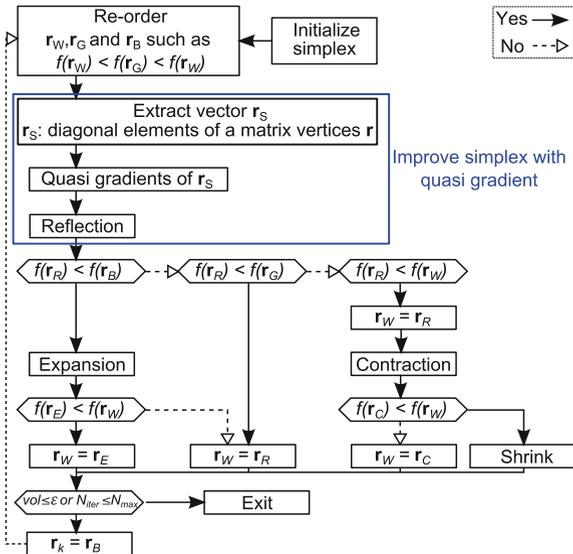
and  $\text{dist}$  is the distance between two vertices. By convention, the vertices are ordered as

$$f(\mathbf{r}_1) \leq f(\mathbf{r}_2) \leq \dots \leq f(\mathbf{r}_{k+1}) \quad (17)$$

where  $\mathbf{r}_1$  is the best vertex and  $\mathbf{r}_{k+1}$  is the worst vertex.

The minimization of the cost-function using the Simplex algorithm is shown in Fig. 5. In our case, the Simplex was modified, by introducing the quasi-gradient convergence instead of reflection stage method [15], in order to improve the convergence direction of  $f$  (without getting trapped in local minima) when the controller approaches the desired position. This combination of an unconstrained and non-linear method with a quasi-gradient technique allows a higher rate, faster and smooth convergence speed. This returns to combine the advantages of a Simplex and gradient-based optimization methods.

**Fig. 5** Modified simplex algorithm



Therefore, (12) is replaced with

$$\mathbf{r}_R = \mathbf{r}_B - \alpha \mathbf{Q} \tag{18}$$

where  $\mathbf{Q}$  is the quasi-gradient vector based on the diagonal elements of the vertices matrix [15].

## 5 Registration Versus Visual Servoing

### 5.1 Image Transformation

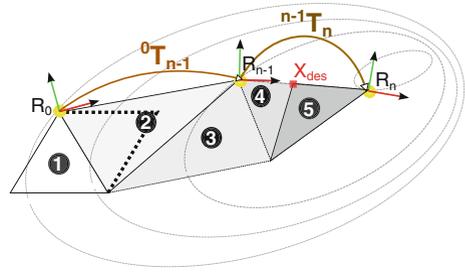
First, the considered registration is defined as a rigid transformation between two images. Let us assume the transformation  $\hat{\mathbf{r}} \in SE(3) = \mathcal{R}(3) \times SO(3)$  between the white light image  $\mathbf{I}_{b1}$  and the fluorescence image  $\mathbf{I}_{b2}$ . Thereby, this transformation can be estimated by minimizing the value of  $\mathbf{MI}(\mathbf{I}_{b1}, \mathbf{I}_{b2})$ :

$$\hat{\mathbf{r}} = arg \min -\mathbf{MI}[\mathbf{I}_{b1}(\mathbf{r}), \mathbf{I}_{b2}] \mid \mathbf{r} \in SE(3) \tag{19}$$

where  $\mathbf{r}$  is a possible rigid transformation.

The process allowing to carry out this registration is operating as follows: acquisition of both white light image  $\mathbf{I}_{b1}$  and fluorescence image  $\mathbf{I}_{b2}$  then computing  $\mathbf{MI}(\mathbf{I}_{b1}, \mathbf{I}_{b2})$ . The obtained transformation  $\hat{\mathbf{r}}$  from the first optimization is then applied

**Fig. 6** Possible evolution of the simplex



to synthesize a new image  $I_{b1}(\mathbf{r})$  from the image  $I_{b1}$ . These steps are repeated until the predefined stop criterion is reached (Fig. 6).

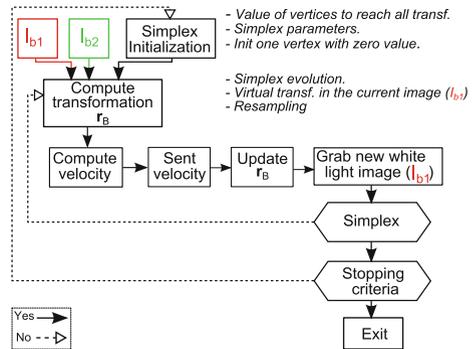
### 5.2 Visual Servoing

Let us assume that we have the cost-function shown in Fig. 3, then our objective is to find the global minimum

$$\hat{\mathbf{r}} = \underset{\mathbf{r} \in SE(3)}{\text{arg min}} -\mathbf{MI} [I_{b1}(\mathbf{r}), I_{b2}] \quad (20)$$

A first way to move the robot so that the current (smoothed) image  $I_{b1}$  superimpose onto the desired fluorescence (smoothed) image  $I_{b2}$  is to use the look-than-move approach: let the Simplex method converge, then apply  $\hat{\mathbf{r}}^{-1}$  to the robot and start again (Fig. 7). However, this requires a very fine tuning of the Simplex algorithm. The chosen approach allows interlacing the Simplex loop and the vision-based control loop. At each iteration  $n$ , the Simplex provides  $\mathbf{r}_B^n$ , the best vertex so far, which is associated to the best transformation  ${}^0\mathbf{T}_n = e^{[\mathbf{r}_B^n]}$ , with  $[\mathbf{r}_B^n] = \begin{pmatrix} [\mathbf{u}_n \theta_n]^\wedge & \mathbf{t}_n \\ 0 & 0 \end{pmatrix}$ , from

**Fig. 7** MI-based visual servoing scheme



the initial to the current pose thanks to the exponential mapping. Thus, applying directly the Simplex would require displacing the robot by

$${}^{n-1}\mathbf{T}_n = ({}^0\mathbf{T}_{n-1})^{-1} {}^0\mathbf{T}_n \quad (21)$$

where  ${}^0\mathbf{T}_{n-1} = e \begin{pmatrix} [\mathbf{u}_{n-1}\theta_{n-1}]_{\wedge} & \mathbf{t}_{n-1} \\ 0 & 0 \end{pmatrix}$

This displacement will not be applied to the complete transformation  ${}^{n-1}\mathbf{T}_n$  found, because that may have the robot to take too large motion. Instead, we extract the screw  $(\Delta\mathbf{t}, \mathbf{u}\theta)^\top$  associated to  ${}^{n-1}\mathbf{T}_n$  and convert it to a damped velocity over the sample period  $T_s$  which is  $\mathbf{v} = (\lambda \cdot \Delta\mathbf{t})/T_s$  and  $\omega = (\lambda \cdot \mathbf{u}\Delta\theta)/T_s$ .

Applying this velocity to the robot requires to update the Simplex vertex  $\mathbf{r}_B^n$  according to the current (estimated) transformation (Fig. 6):

$$(\mathbf{r}_B^n)^{update} \Leftrightarrow {}^0\mathbf{T}_n^{update} = ({}^0\mathbf{T}_{n-1})^{-1} e \begin{pmatrix} [\omega]_{\wedge} & \mathbf{v} \\ \mathbf{0} & 0 \end{pmatrix} T_s \quad (22)$$

## 6 Real-World Validation

### 6.1 Planar Positioning

#### Numerical Registration

The proposed numerical registration method is validated using two vocal folds images: real fluorescence and white light. These images taken from [1] were acquired in two different points of view with known pose as shown in Fig. 8. It can be

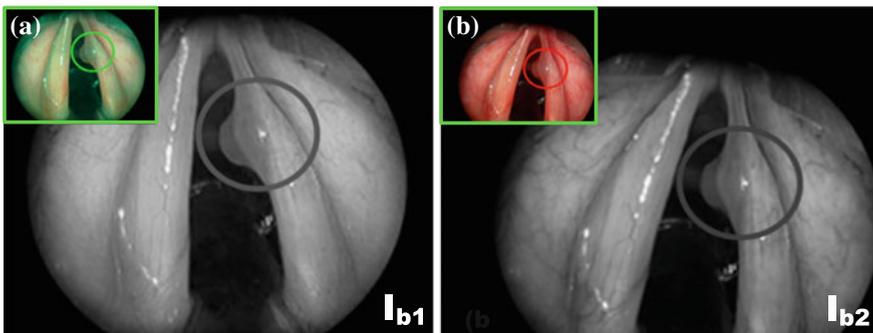
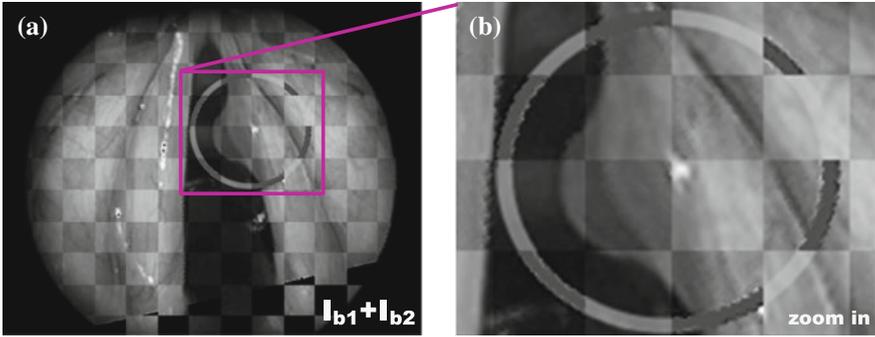


Fig. 8 a Fluorescence image  $I_{b2}$  and b white light image  $I_{b1}$



**Fig. 9** Numerical registration results: **a** shows  $\mathbf{I}_{b1}$  integrated in  $\mathbf{I}_{b2}$ , and **b** a zoom in the region of interest

highlighted that  $\hat{\mathbf{r}}$  between  $\mathbf{I}_{b1}$  and  $\mathbf{I}_{b2}$  includes four parameters ( $x$ ,  $y$ ,  $\theta$  and  $zoom$ ). To be more realistic in our validation tests, we added a circular trajectory (i.e., virtual incision mark done by a surgeon), to be tracked by the surgical laser spot, in the fluorescence image delimiting the tumor (Fig. 8). Then by analyzing Fig. 9a, can be underlined the continuity of the combination ( $\mathbf{I}_{b1} + \mathbf{I}_{b2}$ ), which relates to the high accuracy of the registration method. This accuracy is clearly visible on the zoom in the incision mark (Fig. 9b). For this example, the numerical values are summarized in Table 1.

### Visual Servoing

For ethical reasons, we have not yet performed trials in a clinical set-up. Therefore, we validated the method on two benchmarks. The first one is a 3 DOF ( $x$ ,  $y$ ,  $\theta$ ) microrobotic cell (Fig. 10).

Firstly, the MI-based visual servoing is validated on monomodal images in aim to verify the validity of our controller. Figure 11a represents an example of white light images registration in visual servoing mode. More precisely, Fig. 11a(a, b) represent the initial and desired images, respectively. In the same way, Fig. 11a(c, d) show the initial and final error  $\mathbf{I}_{b1} - \mathbf{I}_{b2}$ . It can be noticed that the final position of the positioning platform matches perfectly with the desired position indicating good accuracy of our method.

**Table 1** Numerical values of  $\hat{\mathbf{r}}$ ,  $\hat{z}$  ( $1pix = 0.088\text{ mm}$ )

DOF	Real pose	Obtained pose	Errors
$x$ (mm)	-8.000	-7.767	0.233
$y$ (mm)	-12.000	-12.059	0.059
$\theta$ (deg)	12.000	12.500	0.500
$z$	1.09	1.089	0.010

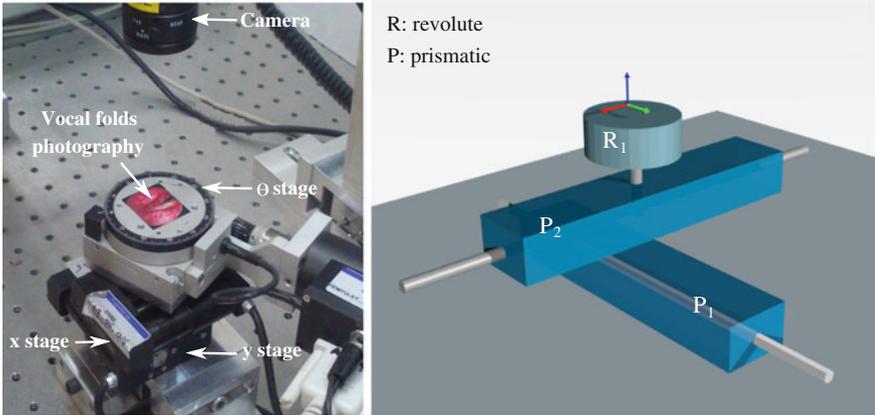


Fig. 10 Global view on the 3DOF experimental platform

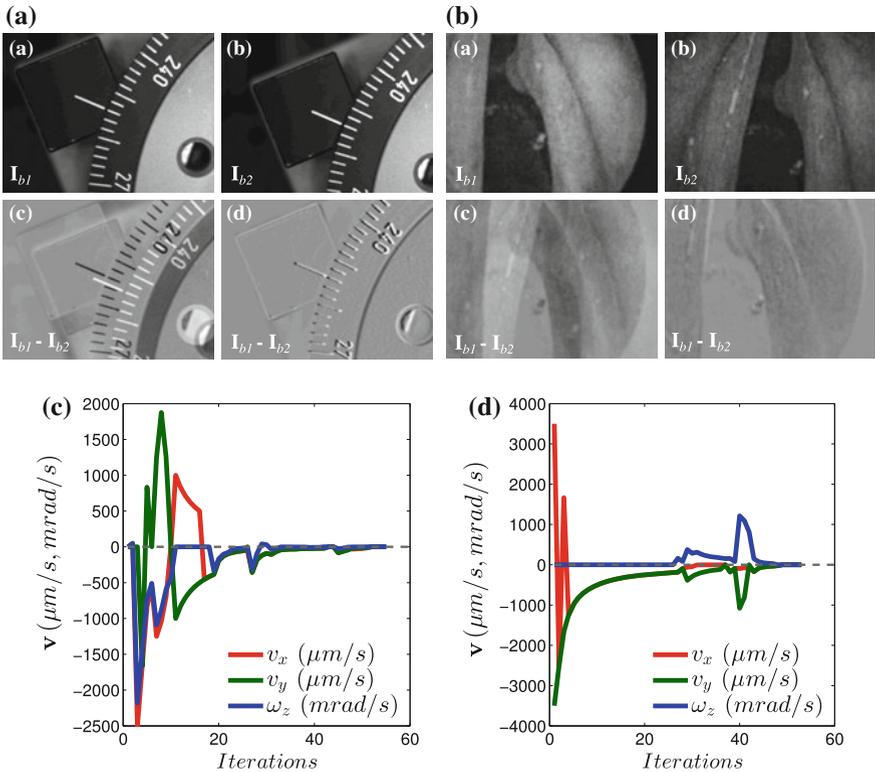


Fig. 11 Image snapshots acquired during the  $SE(2)$  positioning: **a** white light versus white light images, **b** white light versus fluorescence images. Velocities  $v_x$ ,  $v_y$  and  $\omega_z$  (in  $\mu\text{m/s}$ , mrad/s) versus iterations in the case of: **c** white light versus white light image, **d** fluorescence versus white light image

Figure 11c shows the evolution of the velocities  $v_x$ ,  $v_y$  and  $\omega_z$  in the different DOF versus number of iterations. It can be underlined that the developed controller converges with accuracy in fifty iterations (*each iteration takes about 0.5s*). Also, the speed varies in the iteration 40 because the Simplex after initialization found a new best minimum.

Secondly, vocal folds multimodal images are also used to test the proposed controller. In this scenario, the desired image is in fluorescence mode (prerecorded image) and the current images are in white light mode as it would be in the surgical context. Figure 11b(a, b) show the initial image  $\mathbf{I}_{b1}$  and the desired image  $\mathbf{I}_{b2}$ , respectively. Figure 11b illustrate the error ( $\mathbf{I}_{b1} - \mathbf{I}_{b2}$ ) during the visual servoing process. As shown in this figure, the controller converges also to the desired position with a good accuracy. Note that the image ( $\mathbf{I}_{b1} - \mathbf{I}_{b2}$ ) is not completely gray (if two pixels are exactly the same, it is assigned the gray value of 128 for a better visualization of ( $\mathbf{I}_{b1} - \mathbf{I}_{b2}$ ), this is due to the fact that both images are acquired from two different modalities, then the difference will never be zero (respectively 128 in our case).

In the same manner, Fig. 11d shows the evolution of the velocities  $v_x$ ,  $v_y$  and  $\omega_z$  with respect number of iterations. It can be also underlined that the controller converges with the accuracy to the desired position despite the large difference between  $\mathbf{I}_{b1}$  and  $\mathbf{I}_{b2}$ .

Additional validation tests were performed to assess the repeatability and behavior (convergence and robustness) of the controller. Therefore, for each test, the experimental conditions (lighting conditions, initial position and image quality) were deliberately altered. Table 2 gives the results of a sample of these experiments.

**Table 2** Repeatability test for visual servoing ( $x$ ,  $y$ ,  $error$  in mm,  $\theta$  in  $^\circ$  and  $t$  in seconds)

No.	DOF	Des. pos.	Ini. pos.	Error	$t$
1	$x$	5.37	2.47	-0.33	25.2
	$y$	2.94	0.66	0.37	
	$\theta$	-2.61	-8.43	2.51	
2	$x$	4.02	-0.66	0.37	36.5
	$y$	-5.57	-5.05	1.45	
	$\theta$	2.47	-5.05	2.41	
3	$x$	6.05	3.14	0.16	49.2
	$y$	1.47	0.21	0.88	
	$\theta$	-14.59	-24.19	0.64	
4	$x$	4.09	2.1	0.17	36.3
	$y$	2.12	0.44	0.4	
	$\theta$	14.56	6.63	1.15	
5	$x$	3	0.31	0.55	57.3
	$y$	2.5	0.19	0.53	
	$\theta$	-4.81	14.53	0.83	

## 6.2 3D Positioning

### Numerical Registration

This numerical registration was tested in the same condition as in the planar numerical registration experiment. However, in this case the transformation between  $I_{b1}$  and  $I_{b2}$  is  $\hat{\mathbf{r}} \in SE(3)$ . As in the previous experiment, we use the fluorescence image (Fig. 12a) versus white light (Fig. 12a) image, with circular trajectory of the laser spot draw by the surgeon in both images. The initial Cartesian error between the desired image  $I_{b1}$  and the current image  $I_{b2}$ , was  $\mathbf{r} = (30, 30, 40 \text{ mm}, 4^\circ, 10^\circ, 5^\circ)$ .

Again in this experiment we can see overlapping between the reference and the transformed image in the combined image Fig. 13c. The resulting image is the sum between a region of current image (Fig. 13a) and the transformed one with the returned registration values (Fig. 13b) to show the continuity of the vocal fold shape. Besides, the real final error is  $\delta \mathbf{r} = (0.22, 1.29, 9.5 \text{ mm}, 0.29^\circ, 0.86^\circ, 1.02^\circ)$ , with a computation time of 6.564 s.

### Visual Servoing

The previous experiment on the visual servoing was extended to the 6 DOF robot platform with an eye-to-hand configuration as shown in the Fig. 14 (left). The test

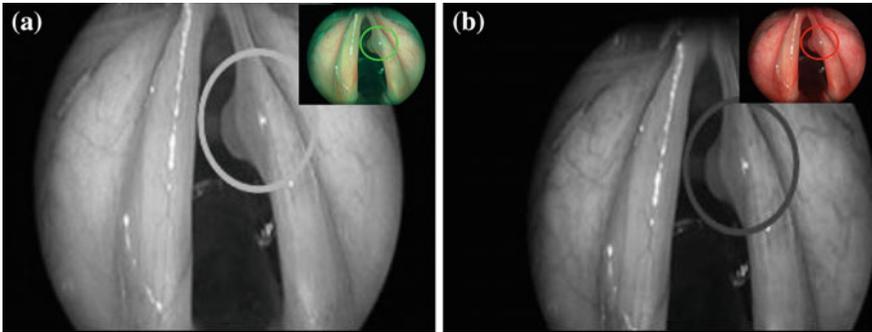
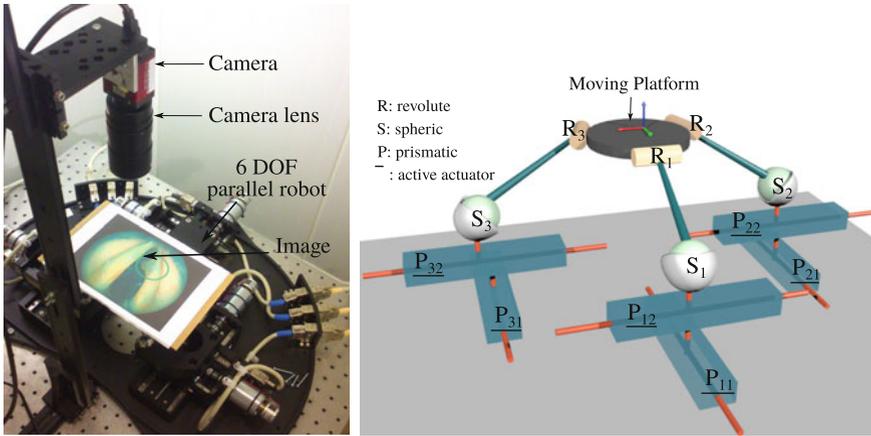


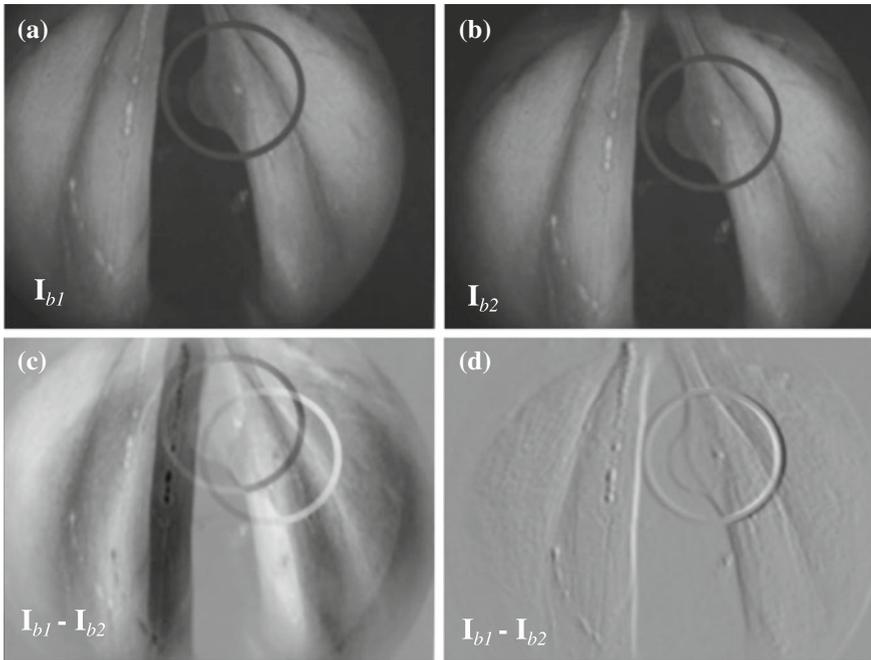
Fig. 12 a Fluorescence image  $I_{b2}$  and b white light image  $I_{b1}$



Fig. 13 Numerical registration results: a shows a sample region of  $I_{b1}$ , b shows a sample region of  $I_{b2}$  after applying the numerical registration transformation, and c the combination of the images (a) + (b)



**Fig. 14** Global view on the 6 DOF experimental platform

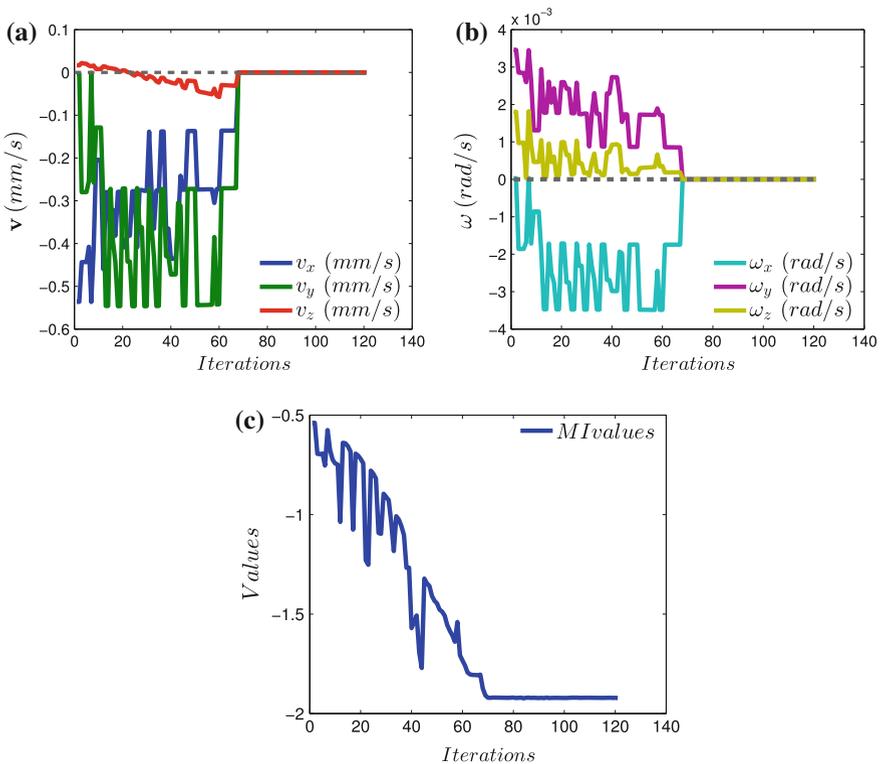


**Fig. 15** Image sequence captured during the positioning task. **a** Desired image  $I_{b1}$ , **b** current image  $I_{b2}$ , **c** initial difference  $I_{b1} - I_{b2}$  and **d** final difference  $I_{b1} - I_{b2}$  showing that the controller reaches the desired position

consists in the validation of our controller without any information of the setup as an interaction matrix or calibration parameters.

The approach consists of 3D positioning of the robot based on desired image, Fig. 15a (planer image (i.e., photography of vocal fold)) from current image Fig. 15b chosen arbitrary at the workspace of the robot. To do so, the robot is placed at an initial position  $\mathbf{r} = (-6, 6, 75 \text{ mm}, -1^\circ, -1^\circ)$  and must reach the desired position  $\mathbf{r}^* = (6, -6, 74 \text{ mm}, -4^\circ, 2^\circ, 1^\circ)$ . While, the Fig. 15c presents the initial image difference ( $\mathbf{I}_{b1} - \mathbf{I}_{b2}$ ) and Fig. 15d the final image difference when the controller reaches the desired position. The positioning errors in each DOF are computed using the robot encoders. The final error obtained is  $\delta \mathbf{r} = (1.22, 0.352, 0.320 \text{ mm}, 1.230^\circ, 1.123^\circ, 0.623^\circ)$ . By analyzing this numerical value, it can be underlined the convergence of the proposed method.

In Fig. 16a, b illustrate the velocities  $\mathbf{v}$  evolution sends to the robot during the positioning task relative to the number of iterations (each iteration takes 0.5 s). Furthermore, the mutual information values evolution decay is presented in Fig. 16c with respect to the number of iterations.



**Fig. 16** a Translation velocities  $\mathbf{v}$  (in mm/s), b rotation velocities  $\omega$  (in rad/s), c mutual information values evolution

## 7 Conclusion

In this paper, a novel metric visual servoing-based on mutual information has been presented. Unlike the traditional methods, the developed approach was based only on the use of a modified Simplex optimization. It has been shown that the designed controller works even in the presence of many local minima in the mutual information cost-function. Beside this, the controller has shown good behavior in terms of repeatability and convergence. Also, we have validated the controller in  $SE(3)$  using a 6 DOF robot.

Future work will be devoted to optimize the computation time to reach the video rate and improve the velocity control trajectories.

**Acknowledgments** This work was supported by  $\mu$ RALP, the EC FP7 ICT Collaborative Project no. 288663 (<http://www.microralp.eu>), by French ANR NEMRO no ANR-14-CE17-0013-001, and by LABEX ACTION, the French ANR Labex no. ANR-11-LABX-0001-01 (<http://www.labex-action.fr>).

## References

1. Arens, C., Dreyer, T., Glanz, H., Malzahn, K.: Indirect autofluorescence laryngoscopy in the diagnosis of laryngeal cancer and its precursor lesions. *Eur. Arch. Oto-Rhino-Laryngol. Head Neck* **261**(2), 71–76 (2004)
2. Chaumette, F., Hutchinson, S.: Visual servo control, Part 1: basic approaches. *IEEE Robot. Autom. Mag.* **13**(1), 82–90 (2006)
3. Collewet, C., Marchand, E.: Photometric visual servoing. *IEEE Trans. Robot.* **27**(4), 828–834 (2011)
4. Dame, A., Marchand, E.: Entropy-based visual servoing. In: *IEEE International Conference on Robotics and Automation*, pp. 707–713 (2009)
5. Dame, A., Marchand, E.: Mutual information-based visual servoing. *IEEE Trans. Robot.* **27**(5), 958–969 (2011)
6. Degani, A., Choset, H., Wolf, A., Zenati, M.A.: Highly articulated robotic probe for minimally invasive surgery. In: *IEEE International Conference on Robotics and Automation*, pp. 4167–4172 (2006)
7. Dowson, N., Bowden, R.: A unifying framework for mutual information methods for use in non-linear optimisation. *Lect. Notes Comput. Sci.* **3951**, 365–378 (2006)
8. Eckel, H., Berendes, S., Damm, M., Klusmann, J.: Suspension laryngoscopy for endotracheal stenting. *Laryngoscope* **113**, 11–15 (2003)
9. Jackel, M., Martin, A., Steine, W.: Twenty-five years experience with laser surgery for head and neck tumors. *Eur. Arch. Oto-Rhino-Laryngol.* **264**, 577–585 (2013)
10. Kelley, C.: *Iterative Methods for Optimization*. *Frontiers in Applied Mathematics*, vol. 18 (1999)
11. Marchand, E., Collewet, C.: Using image gradient as a visual feature for visual servoing. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5687–5692 (2010)
12. Miura, K., Hashimoto, K., Gangloff, J., de Mathelin, M.: Visual servoing without Jacobian using modified simplex optimization. In: *IEEE International Conference on Robotics and Automation*, pp. 3504–3509 (2005)
13. Nelder, A., Mead, R.: A simplex method for function minimization. *Comput. J.* **7**, 308–313 (1965)

14. Ourak, M., Tamadazte, B., Andreff, N., Marchand, E.: Visual servoing-based registration of multimodal images. In: International Conference on Informatics in Control, Automation and Robotics, pp. 1–8 (2015)
15. Pham, N., Wilamowski, B.: Improved Nelder Mead's simplex method and applications. *J. Comput.* **3**(3), 55–63 (2011)
16. Sevick-Muraca, E.: Fluorescence-enhanced optical imaging and tomography for cancer diagnostics. In: IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2004, vol. 2, pp. 1482–1485 (2004)
17. Tamadazte, B., Andreff, N.: Weakly calibrated stereoscopic visual servoing for laser steering: application to phonomicrosurgery. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 743–748 (2014)
18. Upile, T., Jerjes, W., Sterenborg, H.J., El-Naggar, A.K., Sandison, A., Witjes, M.J., Biel, M.A., Bigio, I., Wong, B.J., Gillenwater, A., et al.: Head and neck oncology. *Head Neck* **1**(1), 16 (2009)
19. Zitová, B., Flusser, J.: Image registration methods: a survey. *Image Vis. Comput.* **21**(11), 977–1000 (2003)

# Trajectory Tracking Control of an Excavator Arm Using Guaranteed Cost Control

Alexander Gurko, Oleg Sergiyenko, Juan Ivan Nieto Hipólito,  
Igor Kirichenko, Vera Tyrsa and Wilmar Hernandez

**Abstract** Improving excavators control systems allows increasing efficiency and productivity of these machines, reduces loads on an operator and improves his safety. One of the stumbling blocks in the way of improving an excavator control system is uncertainty of high dynamic loads during the bucket and soil interaction. Therefore, this paper deals with the problem of robust trajectory tracking control of an excavator bucket during digging. The optimal inverse kinematic solution is used to determine the joint angles, given in the base coordinate system. To control an excavator arm movement, the computed torque control with the guaranteed cost control is designed. The mathematical tool of R-functions for state estimation of an excavator arm is used. Simulation results and functional ability analysis for the proposed control system are given. Some intermediate outcomes concerning the sensing part of the robotic excavator prototype, that is being developed, are presented.

**Keywords** Robotic excavator · Digging control · Guaranteed control · Multiple identification

---

A. Gurko (✉) · I. Kirichenko  
Kharkov National Automobile and Highway University, Kharkov, Ukraine  
e-mail: gurko@khadi.kharkov.ua; agrk@mail.ru

I. Kirichenko  
e-mail: igor.kyrychenko@mail.ru

O. Sergiyenko · J.I.N. Hipólito  
Autonomous University of Baja California, Mexicali, Baja California, Mexico  
e-mail: srgnk@uabc.edu.mx

J.I.N. Hipólito  
e-mail: jnieto@uabc.edu.mx

V. Tyrsa  
Kharkov Petro Vasylenko National Technical University of Agriculture,  
Kharkov, Ukraine  
e-mail: vera-tyrsa@mail.ru

W. Hernandez  
Universidad Técnica Particular de Loja, Loja, Ecuador  
e-mail: whernandez@utpl.edu.ec

## 1 Introduction

Increasing quality and productivity of construction and road-building works, as well as their cost saving, require improving control systems of corresponding earthmoving machines, and one of the most widespread of them is a hydraulic excavator. Development of excavator control systems is also stimulated by the need to reduce the load on the operator and improve his safety.

However, the problem of an excavator control is generally made complicated by a range of factors that include a lot of nonlinearities and uncertainties such as backlashes between machine parts, variability of the inertial and gravitational forces with joint motions as well as variation of a fluid viscosity in hydraulic actuators, oil leaks, etc. The problem is aggravated by external uncertainties such as high dynamic loads during a bucket and soil interaction. Such nonlinear systems present the designer with a difficult challenge, which researchers meet using a wide range of approaches [1–12].

On the basis of aforementioned it may be said that, among several tasks, in order an excavator to be automated, the trajectory tracking control of excavator's tool during digging is one of the most fundamental and essential tasks.

Much research has been done on the design of the digging process control system. For example, some works [1–3] describe PD and PID controllers application to control a robotic excavator arm movement. Besides, in one of the papers [3] a proportional-integral-plus (PIP) controller and a nonlinear PIP controller based on a state-depended parameter model structure were offered.

In one of the works [4] a disturbance observer was offered in addition to PI-controller to control a mini excavator arm. Along with the computed torque control, the adaptive and robust controls of the excavator arm were designed in [5].

In [6] a fuzzy plus PI controller with fuzzy rules based on the soft-switch method was developed. In [7] an adaptive fuzzy sliding mode control to realize the trajectory tracking control of an automatic excavator was designed. Two controllers based on fuzzy logic, including the fuzzy PID controller and fuzzy self-tuning with neural network, were developed in [8] to control the electrohydraulic mini excavator. In [9] the Time-Varying Sliding Mode Controller with fuzzy algorithm was applied to the tracking control system of the hydraulic excavator. Time-delay controllers were offered for motion control of a hydraulic excavator arm in [10, 11]. A time-delay control with switching action using an integral sliding surface for the control of a heavy-duty robotic excavator was offered in [12].

All these works have made a valuable contribution in solving the problem of robotic excavator creating, but nevertheless, the development and improvement of digging control systems is still a very important and urgent task, as evidenced by the large number of new publications on this subject.

The focus of this paper is on the guaranteed cost control for the trajectory tracking control of the excavator arm during digging operation. The control guarantees robustness against uncertainties of modelling and unexpected disturbances due to, for instance, the bucket and soil interaction.

## 2 Excavator Modelling

### 2.1 Modelling of an Excavator Arm

A normal excavator working mechanism is a 4 DOF hydraulic manipulator mounted on a mobile base. The dynamic model of an excavator arm can be obtained using the Lagrange equation and can be expressed concisely in a matrix form as the well-known equations for a rigid-link manipulator [13]:

$$D(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + B(\dot{\theta}) = \tau - \tau_L, \tag{1}$$

where  $\theta, \dot{\theta}, \ddot{\theta}$  are the  $4 \times 1$  vectors of the measured joint position, velocity and acceleration angles as shown in Fig. 1;  $D(\theta)$  is the  $4 \times 4$  symmetric, positive-definite inertia matrix;  $C(\theta, \dot{\theta})$  is the  $4 \times 4$  Coriolis and centripetal matrix;  $G(\theta)$  is the  $4 \times 1$  vector of gravity terms;  $B(\dot{\theta})$  is the  $4 \times 1$  vector of frictions;  $\tau$  is the  $4 \times 1$  vector specifying the torques acting on the joint shafts; and  $\tau_L$  is the  $4 \times 1$  vector representing the interactive torques between the links and environment during the digging operation. As research suggests [14–17] the main contribution to the torque  $\tau_L$  creation is made by the Digging Resistance Force  $F_r$ . The force  $F_r$  will be defined in the next section.

For convenience, the dynamic equation (1) can be rewritten as follows:

$$D(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) + \tau_L = \tau, \tag{2}$$

where  $N(\theta, \dot{\theta}) = C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + B(\dot{\theta})$ .

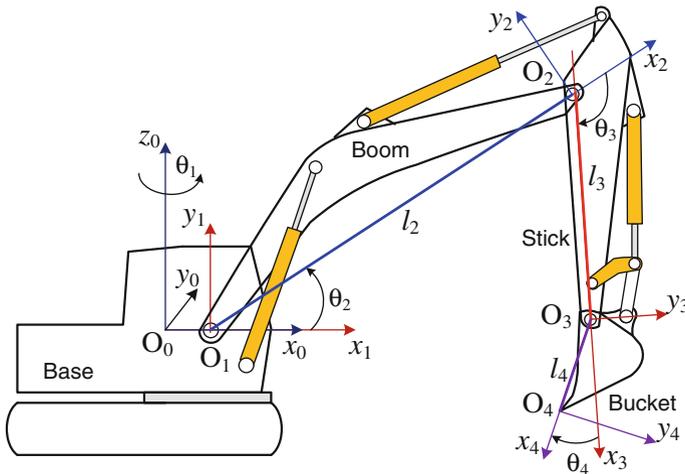


Fig. 1 Coordinate frames of an excavator

As a rule, a skilled operator combines parts of an excavator work cycle, e.g. lifting the bucket loaded with soil and swinging the base towards a vehicle. Here, for simplicity we will suppose that during the digging operation the joint variable  $\theta_1$  is not changed, it is therefore assumed that  $\dot{\theta}_1 = \ddot{\theta}_1 = 0$ .

### 2.2 Digging Resistance Force

Digging by an excavator is performed due to the bucket movement in two directions. The main movement, named lifting, cuts a slice of soil. The second movement (penetration) is perpendicular to the main movement and regulates the thickness of the cut slice of the soil.

During digging soil by an excavator the resistance force  $F_r$  acts at the cutting edge of the bucket teeth (Fig. 2).  $F_r$  is a resultant reaction force of the tangential  $F_t$  and the normal  $F_n$  forces. There are a lot of equations that model these forces [14, 16, 18], but the simplest ones and at the same time accurate enough are Dombrovskiy equations.

According to Dombrovskiy [14, 18], the tangential force can simplistically be determined as

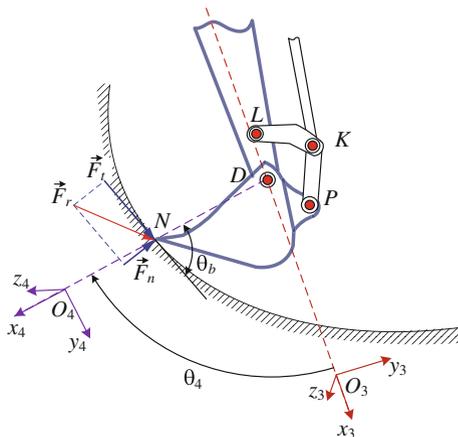
$$F_t = k_c b h, \tag{3}$$

where  $k_c$  is the specific cutting force in  $\text{N/m}^2$  that takes into account soil resistance to cutting as well as all other forces (frictional resistance of the bucket to the soil, resistance to the movement of the prism of soil etc.);  $h$  and  $b$  are the thickness and width of the cut slice of soil.

The normal component  $F_n$  is calculated as:

$$F_n = \psi F_t, \tag{4}$$

**Fig. 2** Bucket and soil intersection



where  $\psi = 0.1 - 0.45$  is a dimensionless factor depending on the digging angle, digging conditions and the cutting edge where  $\psi = 0.1 - 0.45$ . Higher values of  $\psi$  correspond to blunting of the bucket teeth edge.

It should be noted that the factors  $k_c$ ,  $\psi$  and the soil density  $\rho_s$  can vary within wide ranges.

Thus, the torques of resistance forces for each link of an excavator arm can be calculated as:

$$\tau_L = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\tau_{L2} \\ \Delta\tau_{L3} \\ \Delta\tau_{L4} \end{bmatrix}, \quad (5)$$

where  $\Delta\tau_{L4} = l_4 (F_t \sin \theta_b - F_n \cos \theta_b)$ ;  $\Delta\tau_{L3} = -l_3 (F_t \sin(\theta_4 - \theta_b) + F_n \cos(\theta_4 - \theta_b))$ ;  $\Delta\tau_{L2} = l_2 (F_t \sin(\theta_{34} - \theta_b) + F_n \cos(\theta_{34} - \theta_b))$ ;  $\theta_b$  is the angle between the axes  $x_4$  and the direction of the force  $F_t$  (Fig. 2);  $\theta_{34} = \theta_3 + \theta_4$  (Fig. 1);  $l_j, j = \overline{2, 4}$  are the lengths of the excavator arm links.

It is obvious that using more accurate models of a bucket and soil interaction, for example, given in [16], it is still possible to improve the performance of the proposed control system.

### 2.3 Controller Model

In classical case of manipulator control, the computed-torque control (CTC) and computed-torque-like controls are widely used. The equation for the CTC is given by Spong [13]

$$u = D(\theta)a + N(\theta, \dot{\theta}) + \tau_L, \quad (6)$$

where  $u$  is the control vector;  $a = \ddot{\theta}^d + K_v \dot{e} + K_p e$ ;  $K_p$  and  $K_v$  are symmetric positive-definite matrices;  $e = \theta^d - \theta$  is the position error vector;  $\dot{e} = \dot{\theta}^d - \dot{\theta}$  is the velocity error vector; and superscript “ $d$ ” means “desired”.

As far as the values of the parameters in (2) are not known exactly due to the uncertainties in the system, we have to rewrite the control (6) as

$$u = \hat{D}(\theta)a + \hat{N}(\theta, \dot{\theta}) + \hat{\tau}_L, \quad (7)$$

where the notation  $\hat{(\cdot)}$  represents the estimates of the terms in the dynamic model.

Having substituted (7) in (2), we can obtain  $\ddot{\theta} = a - \eta$ , where  $\eta$  is the uncertainty. Hence,  $\ddot{e} = \ddot{\theta}^d - a + \eta$ . We can set the outer loop control as  $\ddot{\theta} = a + \delta a$ , where  $\delta a$  is to be chosen to guarantee robustness to the uncertainty effects  $\eta$ . By taking  $x = [e^T \ \dot{e}^T]^T$  as the system state, the following first-order differential matrix equation is obtained:

$$\dot{x} = Ax + B(\eta - \delta a), \quad (8)$$

where  $A$  and  $B$  are the block matrices of the dimensions  $(6 \times 6)$  and  $(6 \times 3)$  respectively:

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

Thus, the issue of the control of an excavator arm movement is reduced to finding an additional control input  $\delta a$  to overcome the influence of the uncertainty  $\eta$  in the nonlinear time-varying system (7) and to guarantee ultimate boundedness of the state trajectory  $x$  in (8).

### 3 Controller Design

#### 3.1 Trajectories planning

Before development of control system as a subject to improve excavator dynamics, it is necessary to solve the problem of its kinematic control. In [19] an optimal solution of the trajectories planning task for robotic excavator that provides bucket teeth movement along the desired path was considered. As optimality criterion the minimizing quadratic function (9) of joint angles associated with the respective weights was accepted:

$$J_0 = \sum_{j=2}^4 \gamma_j \cdot (\theta_j^I - \theta_j^0)^2 \xrightarrow{\theta \in \Omega_{\theta}} \min, \quad (9)$$

where  $\theta_j^0$  and  $\theta_j^I$  are the initial and the final values of the angles  $\theta_j$ ,  $j = \overline{2, 4}$ , respectively (Fig. 1);  $\gamma_j$  are the weighting factors, that prioritize the angles changing  $\theta_j$ ;  $\Omega_{\theta}$  is the given subset.

To solve the problem (9) it is necessary to solve the matrix equation (10):

$$H_i \Theta_i = F_i, \quad (10)$$

where  $\Theta_i = [\Delta \theta_2^i \quad \Delta \theta_3^i \quad \Delta \theta_4^i]^T$ ;  $\Delta \theta_j^i$  are increments of the joint angles of an excavator arm at each step  $i$  in time domain;  $F_i = [-\Delta x_b^i \quad \Delta z_b^i]^T$ ;  $\Delta x_b^i$  and  $\Delta z_b^i$  are increments of a bucket teeth coordinates in a Cartesian frame at each  $i$ -th step in time domain;

$$H_i = \begin{bmatrix} \sum_{j=2}^4 l_j \sin \alpha_j^{i-1} & \sum_{j=3}^4 l_j \sin \alpha_j^{i-1} & l_4 \sin \alpha_4^{i-1} \\ \sum_{j=2}^4 l_j \cos \alpha_j^{i-1} & \sum_{j=3}^4 l_j \cos \alpha_j^{i-1} & l_4 \cos \alpha_4^{i-1} \end{bmatrix}, \quad \alpha_j = \sum_{k=2}^j \theta_k, \quad j = \overline{2, 4}.$$

To solve such underdetermined systems as (10) either the Moore-Penrose pseudoinverse or the Tikhonov regularization methods are used. Here, taking into account the specifics of the issue, we will use the Tikhonov regularization method. Using this method we can write the original Eq. (10) in the following form

$$(H_i^T H_i + \lambda \Gamma) \Theta_i = H_i^T F_i, \quad (11)$$

where  $\lambda$  is an arbitrary small positive parameter that provides stability of the matrix  $(H_i^T H_i + \lambda \Gamma)^{-1}$  computation;  $\Gamma$  is the square  $3 \times 3$  matrix.

In classical problems the matrix  $\Gamma$  has equal diagonal elements. Taking into account the specifics of the vector  $\Theta_i$ , we will use the diagonal matrix  $\Gamma$  where non-zero elements are determined as:

$$(\Gamma)_{j-1, j-1} = \gamma_j, j = \overline{2, 4}. \quad (12)$$

If the values of  $\gamma_j$  are known, solution of (12) is trivial. We offer to determine the weighting coefficients  $\gamma_j$  in the following way. The value of  $\gamma_2$  is selected wittingly large to minimize the boom motion. Values  $\gamma_3$  and  $\gamma_4$  depend on the method of digging:

- when digging with the bucket  $\gamma_3 \gg \gamma_4$ ;
- when digging with the stick  $\gamma_4 \gg \gamma_3$ ;
- when excavator digs simultaneously with the stick and with the bucket, the  $\gamma_3$  and  $\gamma_4$  ratio is chosen to equate the maximum angular acceleration of  $\varepsilon_3$  and  $\varepsilon_4$ . Accelerations  $\varepsilon_j$  are calculated by the well-known formula:

$$\varepsilon_j^i = \frac{\theta_j^{i-1} - 2\theta_j^i + \theta_j^{i+1}}{\Delta t^2}. \quad (13)$$

### 3.2 Robust Control

For an additional control  $\delta a$  determining, we propose the optimal guaranteed cost control approach. According to this approach, it is assumed that uncertainties in the system are known with accuracy to a certain guaranteed bounded set. During the control system operation the new sets representing the estimates of the system state are built. The advantage of this approach is in providing an upper bound on a given performance index and thus, the system performance degradation incurred by the uncertainties is guaranteed to be less than this bound [20].

Let's derive the digital version of the Eq. (8) for the digital control system implementation [21]:

$$x_{k+1} = A_d x_k + B_d \{\eta_k - \delta a_k\} \quad (k = 0, 1, \dots, n-1), \quad (14)$$

where  $A_d$  and  $B_d$  are the digital versions of matrices  $A$  and  $B$  in (8); the uncertainty  $\eta_k$  is bounded by the known set  $\Omega_k^\eta$ ;  $k$ —moments of quantization.

Control is formed on the basis of joint angles  $\theta$  measurements, which are represented in the form of the vector  $y_k$ :

$$y_k = C_d(x_k + v_k), \quad (k = 1, 2, \dots, n - 1), \quad (15)$$

where  $C_d$  is the output vector;  $v_k$  is the vector of measurement noises bounded by the known set  $\Omega_k^v$ .

In (14) and (15)  $A_d$ ,  $B_d$  are assumed to be controllable, and  $A_d$ ,  $C_d$  are assumed to be observable.

We assume the minimizing of the following cost function as the aim of the control:

$$J_k(x_k, \delta a_k) = V_k(x_{k+1}) + \overset{k}{\omega}(x_k, \delta a_k), \quad (16)$$

where  $V_k$  is Lyapunov function that allows estimating the quality of the further excavator arm motion in the absence of perturbations;  $\omega_k$  is the given function, which defines the control costs and assigns limitations on their value.

Lyapunov function can be defined as a quadratic form

$$V_k(x_k) = x_k^T P x_k, \quad (17)$$

where  $P$  is the  $6 \times 6$  symmetric, positive-definite matrix, i.e.  $P = P^T > 0$ .

For the well-posed problem (16) formulation, information about the uncertainty  $\eta_k$  has to be redefined. As far as the  $\eta_k$  can take on any value inside the set  $\Omega_k^\eta$ , we have to consider the values maximizing the cost function (16).

Moreover, the fact that  $\eta_k$  and  $v_k$  belong to the proper sets  $\Omega_k^\eta$  and  $\Omega_k^v$  enables to suppose that as a result of measurement (15) of the excavator arm joint angles  $\theta$ , information about the current state is obtained in the form of the set  $x_k \in \Omega_k^r$ . For the additional control  $\delta a$  determining the point estimation of  $x_k \in \Omega_k^r$  is required. For this purpose we will consider the point maximizing the cost function (16). So, the objective of the additional control  $\delta a_k$  is to solve the following task:

$$\min_{\delta a_k \in \Omega_k^u} \max_{\eta_k \in \Omega_k^\eta} \max_{v_k \in \Omega_k^v} \max_{x_k \in \Omega_k^r} J_k(x_k, \delta a_k). \quad (18)$$

It's obvious that the task (18) solution guarantees the proper excavator control system performance that depends on  $J_k$  at any allowed  $\eta_k$  and  $v_k$ .

The description of the sets of the possible states of the excavator arm will be carried out according to the following algorithm [20].

1. Let there be an estimate of the excavator arm state as  $x_k \in \Omega_k^r$  at an arbitrary moment of quantization  $k$ . The transformation (19) should be realized to find the set of states  $\Omega_{k,k+1}^f$

$$\Omega_{k,k+1}^f = A_d \Omega_k^r, \quad (19)$$

where  $\Omega_{k,k+1}^f$  is a prediction of possible system states  $x_{k+1}^f \in \Omega_{k,k+1}^f$  at the  $[k + 1]$ th moment to which it must transit moving freely from the state  $x_k \in \Omega_k^r$ .

2. A new set  $\Omega_{k,k+1}^w$  of possible system states is developed by transformation (blurring) of the set of states  $\Omega_{k,k+1}^f$ :

$$\Omega_{k,k+1}^w = \Omega_{k,k+1}^f \bigcup B_d \partial \Omega_k^\eta, \quad (20)$$

where  $\partial \Omega_k^\eta$  is the aggregate of boundary elements of the set  $\Omega_k^\eta$ . Thus, the set  $\Omega_{k,k+1}^w$  is a prediction of the excavator arm state at the  $[k + 1]$ th moment with allowance for the influence exerted by uncertainties  $\eta_k$  on the values of parameters of vector  $x_{k+1}^f$ .

3. The value  $x_{k+1}^u \in \Omega_{k,k+1}^w$  of the system state is found. The  $x_{k+1}^u$  is used for an additional control  $\delta a_k$  determined to solve the task (16).
4. Moving set  $\Omega_{k,k+1}^w$  by the additional control  $\delta a_k$  is provided and a new set  $\Omega_{k+1}^u$  is constructed. The set  $\Omega_{k+1}^u$  is an estimation of the system state to which it must transit at the  $[k + 1]$ th moment under  $\delta a_k$  and  $\eta_k$  action.
5. The new measurement of joint angles  $\theta_k$  is carried out to find a posteriori estimate  $x_{k+1} \in \Omega_{k+1}^r$  of the system state at the  $[k + 1]$ th moment:

$$\Omega_{k+1}^r = \Omega_{k+1}^u \bigcap \Omega_{k+1}^v. \quad (21)$$

Further, the mentioned procedure is repeated iteratively.

## 4 Determination of a Set of Possible States

Until recently linear matrix inequalities have been used to construct sets of control system possible states. In [22] we offered to use  $R$ -functions for this purpose.

The  $R$ -function  $\varphi(x_k)$  of the set  $\Omega_k$  has the following properties:

$$\begin{cases} \varphi(x_k) > 0, & \text{when } x_k \in \Omega_k, \\ \varphi(x_k) = 0, & \text{when } x_k \in \partial \Omega_k, \\ \varphi(x_k) < 0, & \text{when } x_k \notin \Omega_k \bigcup \partial \Omega_k, \end{cases}$$

where  $\partial \Omega_k$  is the aggregate of boundary elements of set  $\Omega_k$ .

Let's denote  $R$ -functions of sets  $\Omega_k^r$ ,  $\Omega_{k,k+1}^f$ ,  $\Omega_k^\eta$ ,  $\Omega_k^v$ ,  $\Omega_k^u$  and  $\Omega_{k,k+1}^w$  as  $\varphi^r(x_k)$ ,  $\varphi^f(x_k)$ ,  $\varphi^\eta(x_k)$ ,  $\varphi^v(x_k)$ ,  $\varphi^u(x_k)$  and  $\varphi^w(x_k)$ . For instance, set  $\Omega_k^r$  is constructed using the following  $R$ -function:

$$\varphi^r(x_k) = \varphi^u(x_k) \underset{R}{\wedge} \varphi^v(x_k), \quad (22)$$

where  $\wedge_R$  is the  $R$ - operation of conjunction:

$$\varphi^u \wedge_R \varphi^v = \varphi^u + \varphi^v - \sqrt{(\varphi^u)^2 + (\varphi^v)^2}. \tag{23}$$

Using  $R$ -functions significantly simplifies the estimation of sets of an excavator arm states. For instance, the simple MATLAB code:

```
f(1)=P(1)+P(3)-sqrt(P(1)^2+P(3)^2);
f(2)=f(1)+P(4)-sqrt(f(1)^2+P(4)^2);
f(3)=f(2)+P(5)-sqrt(f(2)^2+P(5)^2);
f(4)=f(3)+P(6)-sqrt(f(3)^2+P(6)^2);
f(5)=f(4)+P(2)-sqrt(f(4)^2+P(2)^2);
```

allows building the hexagon set. In the code  $P(k)$  is a straight-line equation, that contains the  $k$ -th side of hexagon. Using Linear Matrix Inequalities for this purpose gives a more complex code.

## 5 Simulations

Simulation study of the excavator arm motion with the numerical values given in Table 1 [1] was performed in MATLAB.

A desired bucket trajectory is presented in Fig. 3.

The desired joint angles  $\theta_j, j = \overline{2, 4}$  were calculated by the Eq.(11) and are shown in Fig. 4.

At simulating only the joint angles  $\theta_j, j = \overline{2, 4}$  have been measured. It was assumed that the measurement noise is in the foregoing range  $|v_j| \leq 0.5^\circ$  and is subject to the uniform distribution law. The resistance forces, experienced when the bucket penetrates into the soil, are calculated by (3)–(4).

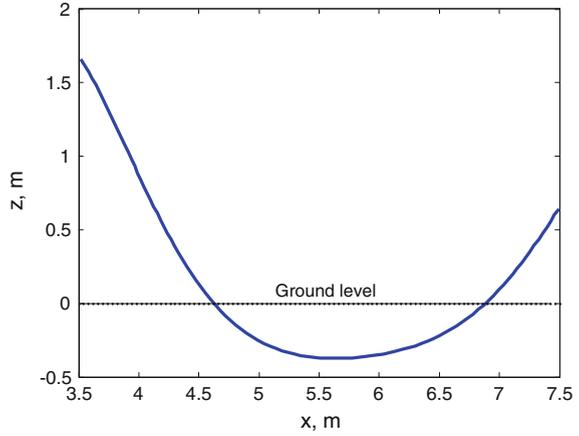
Loam as the type of soil has been considered; the loam density varied arbitrarily in the range  $1600 \leq \rho_s \leq 1900 \text{ kg/m}^3$ . The exact value of the force  $k_c$  in (3) was considered to be unknown except for the fact that it belongs to the set  $117600 \leq k_c \leq 245000 \text{ N/m}^2$ . The value of the factor  $\psi$  in (4) was assumed to be 0.25. Change of the bucket mass has also been taken into account. The true load torques  $\tau_L$  acting at the links are shown in Fig. 5.

As the aim of the control the task (16)–(18) solution has been assumed, where  $V_k = x_{k+1}^T P x_{k+1}; \omega_k = \delta a_k^T R \delta a_k; R = \text{diag}\{0.7, 0.5, 0.2\}$  and

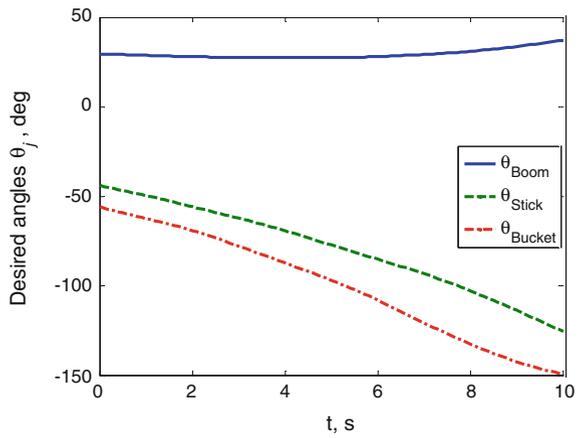
**Table 1** Excavator parameters

Link	Mass (kg)	Inertia (kg m <sup>2</sup> )	Length (m)
Boom	1566	14250.6	5.16
Stick	735	727.7	2.59
Bucket	432	224.6	1.33

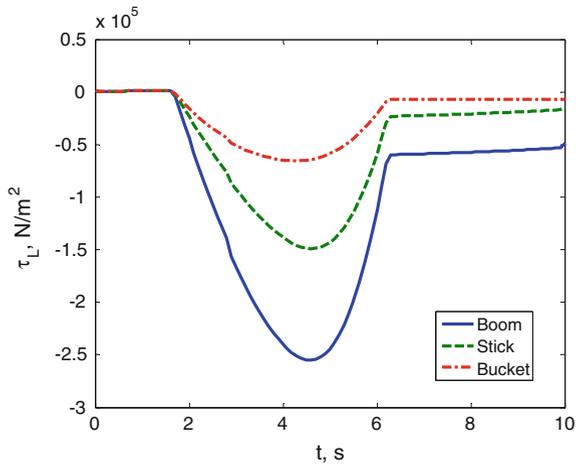
**Fig. 3** Desired bucket trajectory



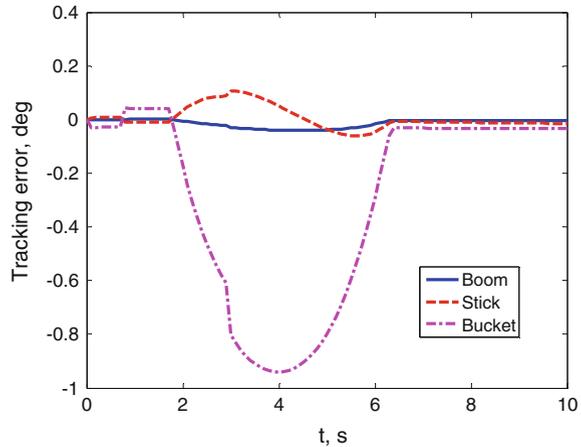
**Fig. 4** Desired angles



**Fig. 5** Load torques  $\tau_L$  acting at the links



**Fig. 6** Joint angles tracking errors versus time



$$P = \begin{bmatrix} 3.2 & 0 & 0 & 1.12 & 0 & 0 \\ 0 & 3.2 & 0 & 0 & 1.12 & 0 \\ 0 & 0 & 3.2 & 0 & 0 & 1.12 \\ 1.12 & 0 & 0 & 1.86 & 0 & 0 \\ 0 & 1.12 & 0 & 0 & 1.86 & 0 \\ 0 & 0 & 1.12 & 0 & 0 & 1.86 \end{bmatrix}.$$

Sampling time was  $T_s = 0.1$  s. For the sets of the system possible states  $R$ -functions have been used.

The simulation results are presented in Figs. 6, 7 and 8. As depicted in Fig. 6, the joint angles tracking errors are less than 0.1, 0.2, and 1 degree for the boom, the stick and the bucket, respectively.

In Fig. 7 the predicted sets of possible states  $\Omega^r$  vs. the true system states  $X_t$  at  $t = 4$  s are shown. It corresponds to the maximum value of the bucket tracking error. To determine the sets  $\Omega^r$  expressions (19)–(21) have been used.

The digging error is less than 1.2 cm or 3.5% (Fig. 8).

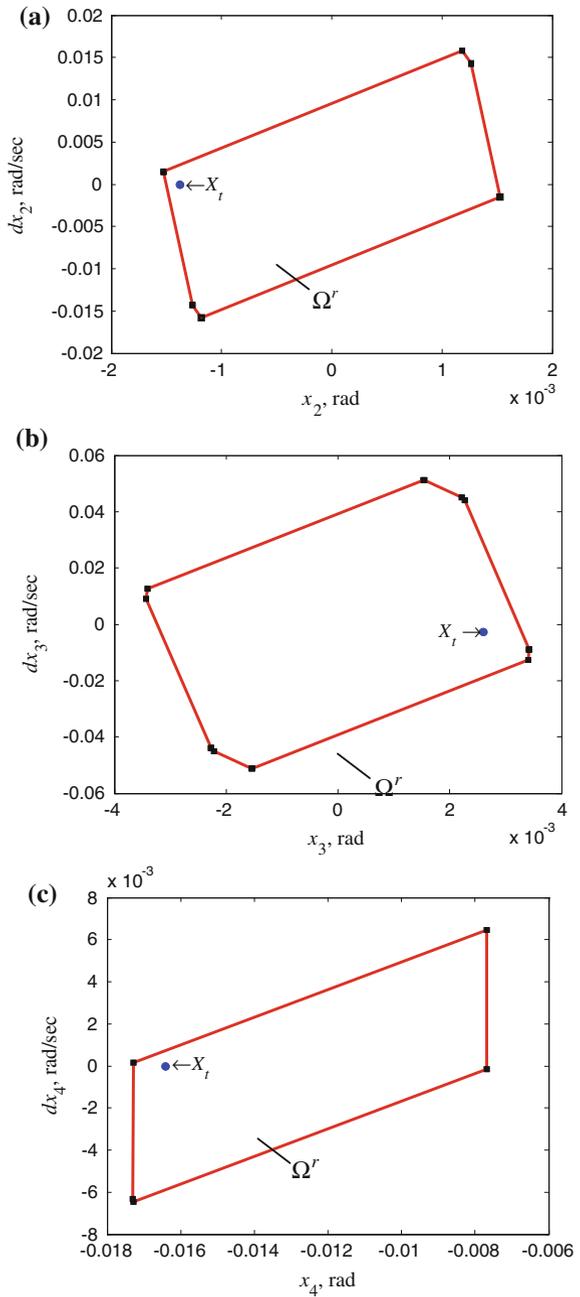
The simulation results illustrate that the proposed controller provides high quality of digging under uncertainties.

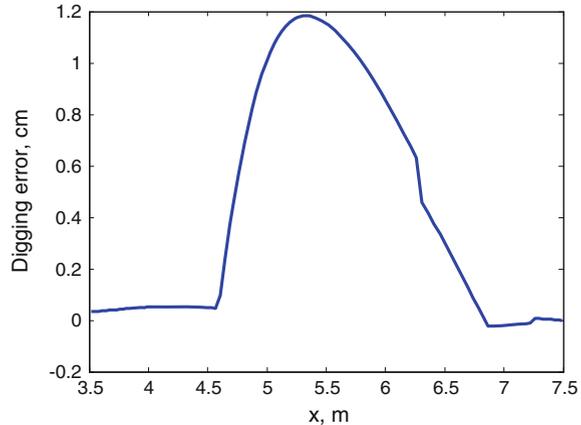
## 6 Sensing

### 6.1 Joint Angle Sensor

To test the aforementioned control system the robotic excavator prototype is being created. To realize the control it should take the current information about joint angles from corresponding sensors as well as excavator position from the Laser scanning

**Fig. 7** Predicted sets  $\Omega^r$  and the true system states  $X_t$ :  
 a—for the boom; b—for the stick; c—for the bucket



**Fig. 8** Digging error

Technical Vision System (TVS). The positioning system is to form a same depth plane in a wide area at work sites with uneven ground and when the excavator body height changes when the machine body moves [23]. In this section we present some results concerning the sensing part of the robotic excavator prototype.

The most common methods to sense joint angles are to use position sensors inside hydraulic cylinders and accelerometers. The downside of these sensors is that position sensing can be noisy [24]. For noise elimination various kinds of software (digital filters) and circuitry solutions are implemented. For example, in [25] a method for measuring the rotation angles based on the processing of signals from the pair of accelerometers located on adjacent links is proposed. However, the accelerometer noise remains a serious problem.

An alternative is to use potentiometers for our robotic excavator prototype. These sensors produce a signal that is proportional to joint displacement. But as the practice shows, the static characteristic of these sensors can be significantly different from the linear ones. The maximum scatter linearly depends on the function of the angle, resistance and the tracks size of potentiometers [26]. Therefore, for successful operation of the control system, sensors have to be calibrated.

Calibration is establishing relationship between the sensor output and input, and its data is used for potentiometers characteristics linearization in smart sensors. Calibration is performed as follows (Fig. 9). The position of the stepper motor (SM) shaft changing within the range from  $0^\circ$  to  $310. . . 350^\circ$  is transmitted to the potentiometer shaft (R). Microstepping modes of polyphase stepper motors allow obtaining the accuracy of angular positioning within the range of  $0.25\text{--}0.5^\circ$ .

Information about changing the SM shaft angle is fed to the microcontroller unit (MCU), but data from the potentiometer—to the ADC. Obtained from the ADC code values  $N$  (Table 2) are fed to a microcontroller, which compares values  $N$  with the SM reference values, and are transmitted to the PC for graphical (Fig. 10) or tabular representation.

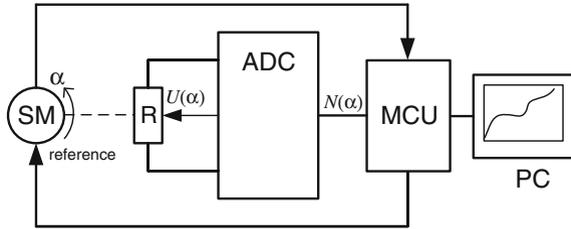


Fig. 9 The diagram of potentiometers calibration

Table 2 Experimental data

Angle (deg.)	$N$	Angle (deg.)	$N$	Angle (deg.)	$N$
0	0	90	162	180	746
10	0	100	228	190	807
20	0	110	289	200	870
30	0	120	356	210	924
40	18	130	423	220	951
50	33	140	488	230	970
60	52	150	554	240	989
70	73	160	619	250	1004
80	103	170	683	260	1016

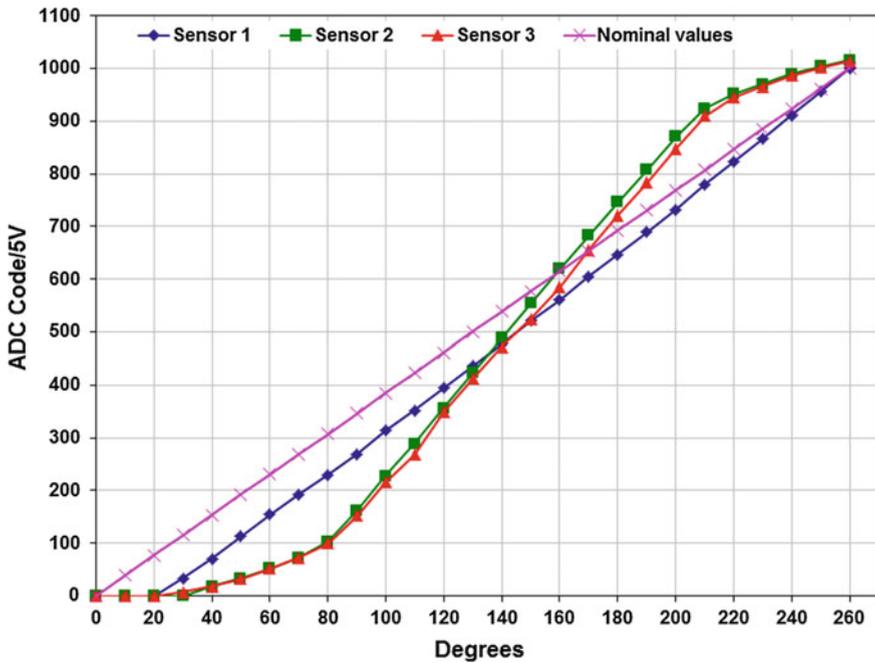


Fig. 10 Linearity graphs of three single-type potentiometers

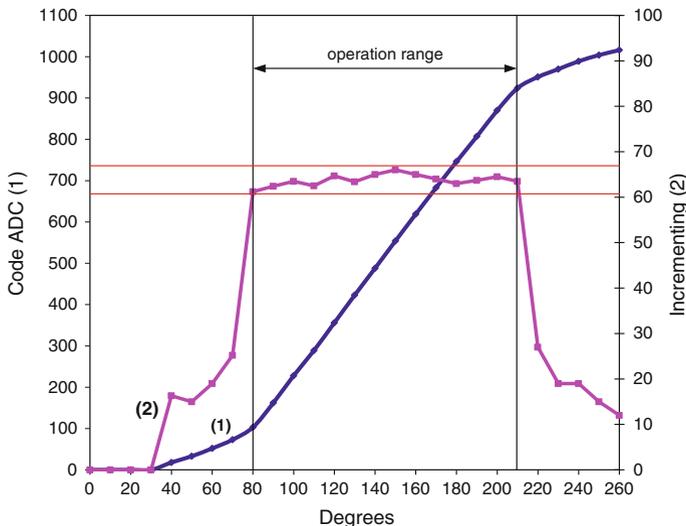


Fig. 11 Operation range of the potentiometer

Each item of the single-type sensors has individual characteristics which are significantly different from the nominal (Fig. 10). This problem is solved by the characteristics linearization using microcontrollers.

To plot the sensor characteristic with sufficiently small step of tabular representation (Table 2) a linear interpolation can be used to find the value  $y(x)$  according to a well-known relationship:

$$\frac{y - f(x_0)}{f(x_1) - f(x_0)} = \frac{x - x_0}{x_1 - x_0}, \tag{24}$$

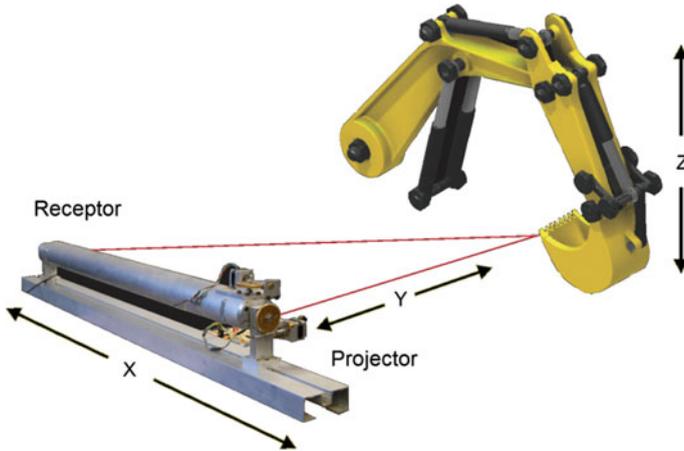
where  $x_0 \leq x \leq x_1$ .

Step data tables can be fixed (Table 2) or variable. The characteristic (Fig. 11, curve 1) being plotted from experimental data taking into account the amplitude quantization step in the ADC, has a marked non-linear character. Curve 2 in Fig. 11 shows the ADC code increments which correspond to specific values of angles.

Analyzing Fig. 11 the characteristics, close to linear parts of the potentiometer curves, can be detected. It is obvious that it is preferred to use the sensor in this operating range of angles. The inappreciable nonlinearities of the characteristics in this range can be eliminated relatively simple by using expression (25):

$$y = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0). \tag{25}$$

Modern microcontrollers with embedded or external ADC can perform the mentioned conversion in real time.



**Fig. 12** Excavator positioning by the laser scanning TVS

## 6.2 Laser Scanning Technical Vision System

In [19] for the purposes of excavator control we introduced the TVS (Fig. 12) that uses the dynamic triangulation principle, has the object tracking offset accuracy less than 1% for the central sector of field-of-view (FOV), this accuracy being valid only at the distance of 3–5 m.

The TVS is fixed on the ground, in a lateral view of the excavator bucket working zone (as shown on Fig. 12). Initial imperfections in TVS's coordinate system positioning can be detected by accelerometer platform with efficient noise filter. These none zero initial values of  $RPY$ -angles can be measured and used as correction factors for automatic coordinate system self-calibration.

The TVS based on original principle (see Fig. 13) of dynamic triangulation [27] is able to detect with metrological accuracy the 3D coordinates of up to 1000 points per second inside predetermined FOV (frame).

Dynamic triangulation consists of detection of laser instantly highlighted point coordinates based on two detected angles  $B_{ij}$  and  $C_{ij}$  (here  $ij$  means the number of horizontal and vertical scanning steps consequently) and fixed distance  $a$  between a projector and a receptor. Such triangle's lifetime is about  $0.039 \times 0.5/2000 \approx 0.0000975$  s (where 0.039 s is minimal time of semi-sphere scanning at 7–13 rev/s motor speed; 0.5 cm is averaged laser spot size on experimental striking distance, and 2000 cm is the selected frame width). In such triangle (Fig. 13) if 3 parameters are known, it makes possible to calculate all others. Angle  $B_{ij}$  is calculated as simple ratio of two counters codes: number of clock pulses between two home pulses and in interval 'home pulse—spot pulse'.

TVS measurement accuracy can also be improved (Fig. 14) by neuronal network methods implementation [28] in fast postprocessing.

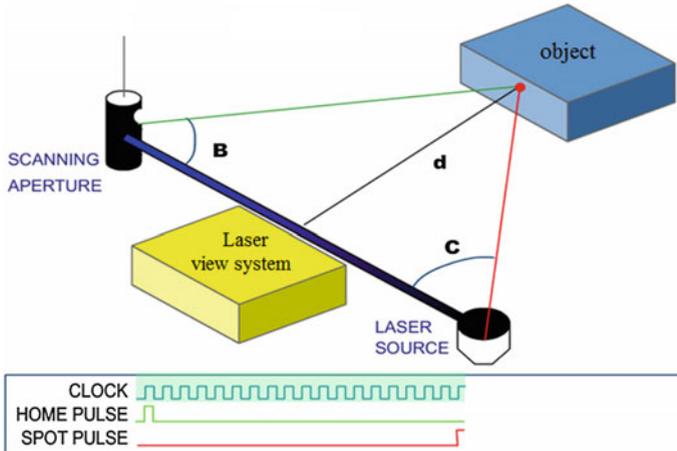
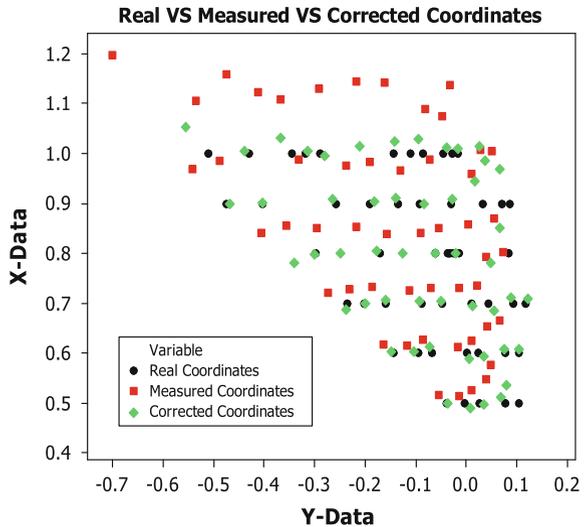


Fig. 13 Dynamic triangulation Principle of laser scanning TVS

Fig. 14 Comparison graph of object’s real, measured and corrected coordinates with the TVS (scale in m)



## 7 Conclusions

The motion control system of an excavator has been developed for the purpose of improving digging work efficiency. For this the optimal solution of the trajectories planning, the dynamics model of an excavator arm, and also the model of the bucket-soil interaction were considered, and it was assumed that the parameters of the latter are not precisely known and can vary according to the uniform law within the specified range. The same suggestion was made for the joint angles sensors. To compensate these uncertainties the control based on guaranteed cost control was

designed. The practical value of the proposed controller is in providing an upper bound on a given performance index at any uncertainties from the given bounded set, as well as in requiring a relatively low computational capability compared to other reviewed methods. The controller can have other applications, for instance, in automotive industry [29].

Some simulation work was conducted to demonstrate that the proposed system good tracking performance is gained in execution of excavation tasks with soil contact considerations.

In addition, some results concerning the preparation of the measurement part of the robotic excavator prototype, which is being developed, is described.

The presented work can be extended in a number of ways.

We will take into account the dynamics of the hydraulic actuators with their nonlinearities, leakages and uncertainties. Parametric uncertainties of the excavator arm dynamic model will also be considered.

Since the uncertainties do not always tend to maximize the cost function, the implementation of an additional circuit of adaptation, which adjusts the bounds of sets of uncertain parameters, is desirable, e.g. one that relies on measurements of digging forces. In future it is planned to conduct the experimental study as well.

## References

1. Koivo, A.J., Thoma, M., Kocaoglan, E., Andrade-Cetto, J.: Modeling and control of excavator dynamics during digging operation. *J. Aerosp. Eng.* **9**(1), 10–18 (1996)
2. Yingjie, G., Yanchao, J., Zhang, Q.: Motion planning based coordinated control for hydraulic excavators. *Chin. J. Mech. Eng.* **22**(1), 97–101 (2009)
3. Gu, J., Taylor, J., Seward, D.: Proportional-integral-plus control of an intelligent excavator. *Comput.-Aided Civil Infrastruct. Eng.* **19**(1), 16–27 (2004)
4. Yokota, S., Sasao, M., Ichiryu, K.: Trajectory Control of the boom and arm system of hydraulic excavators. *Trans. Jpn. Soc. Mech. Eng. Part C* **62**(593), 161–167 (1996) (in Japanese)
5. Yu, H., Liu, Y., Hasan, M.S.: Review of modelling and remote control for excavators. *Int. J. Adv. Mechatron. Syst.* **2**(1), 68–80 (2010)
6. Bo, L., Jun, Y., Gang, G., Yonghua, Z., Wenxing, L.: High performance control of hydraulic excavator based on fuzzy-pi soft-switch controller. In: *IEEE International Conference on Computer Science and Automation Engineering*. IEEE Press, Shanghai (2011), vol. 2, pp. 676–679
7. Zhang, J., Liu, K., Lin, J., Ma, X., Yu, W.: 4-DOF Adaptive fuzzy sliding mode control of excavator. *Jixie Gongcheng Xuebao (Chin. J. Mech. Eng.)* **46**(21), 87–92 (2010)
8. Le Hanh, D., Ahn, K.K., Kha, N.B., Jo, W.K.: Trajectory control of electro-hydraulic excavator using fuzzy self tuning algorithm with neural network. *J. Mech. Sci. Technol.* **23**(1), 149–160 (2009)
9. Choi, J.: Tracking control of hydraulic excavator using time varying sliding mode controller with fuzzy system. *Adv. Sci. Lett.* **15**(1), 78–82 (2012)
10. Chang, P.H., Lee, S.J.: A straight-line motion tracking control of hydraulic excavator system. *Mechatronics* **12**(1), 119–138 (2002)
11. Vidolov, B.: A discrete-time design of time delay control focus on control experimentations of coordinated motion control of hydraulic excavator. In: *10th International IFAC Symposium on Robot Control*, p. 6 (2012)
12. Lee, S.U., Chang, P.H.: Control of a heavy-duty robotic excavator using time delay control with integral sliding surface. *Control Eng. Pract.* **10**(7), 697–711 (2002)

13. Spong, M.W., Hutchinson, S., Vidyasagar, M.: *Robot Modeling and Control*. Wiley, New York (2006)
14. Volkov, D.P.: *Dynamics and Strength of One-Bucket Excavators*. Mashgiz, Moscow (1965). (in Russian)
15. Maeda, G.J., Rye, D.C.: Learning disturbances in autonomous excavation. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE Press, Vilamoura, Algarve, Portugal (2012), pp. 2599–2605
16. Luengo, O., Singh, S., Cannon, H.: Modeling and identification of soil-tool interaction in automated excavation. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE Press, Victoria (1998), pp. 1900–1906
17. Towarek, Z.: Dynamics of a single-bucket excavator on a deformable soil foundation during the digging of ground. *Int. J. Mech. Sci.* **45**(6–7), 1053–1076 (2003)
18. Alekseeva, T.V., et al.: *Machines for Earthmoving Work: Theory and Calculations*. Translation of *Mashiny dlia Zemlianykh Rabot: Teoriia i Raschet*. Amerind Publishing Company, New Delhi (1985)
19. Sergiyenko, O. Yu., et al.: Optimal kinematic control of a robotic excavator with laser TVS feedback. In: *39th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013)*. IEEE Press, Vienna (2013), pp. 4239–4244
20. Gurko, A.G., et al.: Robust control of excavation mobile robot with dynamic triangulation vision. In: *9th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2012)*, vol. 2. SciTePress, Rome, Italy (2012), pp. 481–484
21. Gurko, A.G., et al.: Guaranteed control of a robotic excavator during digging process. In: *12th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2015)*, vol. 2. SciTePress, Colmar (2015), pp. 52–59
22. Gurko, A.G., Kolodyazhny, V.M.: Estimating the state vector of a dynamic system under uncertainty. *Cybern. Syst. Anal.* **49**(6), 821–828 (2013)
23. Haga, M., Hiroshi, W., Fujishima, K.: Digging control system for hydraulic excavator. *Mechatronics* **11**(6), 665–676 (2001)
24. Sanjiv, S.: The state of the art in automation of earthmoving. *J. Aerosp. Eng.* **10**(4), 1–29 (1997)
25. Ghassemi, F., Tafazoli, S., Lawrence, P.D., Hashtrudi-Zaad, K.: An accelerometer-based joint angle sensor for heavy-duty manipulators. In: *IEEE International Conference on Robotics and Automation (ICRA'02)*, vol. 2, pp. 1771–1776 (2002)
26. Wiegleb, G.: *Sensortechnik: Übersicht, Applikation, Anwendungen*. Franzis, München (1986)
27. Sergiyenko, O.Yu., Hernandez, W., Tyrsa, V.V., Hernández-Balbuena, D.: Precise optical scanning for practical multiapplications. In: *34th Annual Conference of IEEE Industrial Electronics (IECON'08)*. IEEE Press, Florida (2008), pp. 1656–1661
28. Rodriguez-Quiñones, J.C., Sergiyenko, O.Yu., Gonzalez-Navarro, F.F., Basaca-Preciado, L., Tyrsa, V.V.: Surface recognition improvement in 3D medical laser scanner using Levenberg–Marquardt method. *Signal Process. (Elsevier)* **93**(2), 378–386 (2013)
29. Mercorelli, P.: A two-stage sliding-mode high-gain observer to reduce uncertainties and disturbances effects for sensorless control in automotive applications. *IEEE Trans. Ind. Electron.* **62**(9), 5929–5940 (2015)

# Computation of Curvature Skeleton to Measure Deformations in Surfaces

Carlos M. Mateo, Pablo Gil and Fernando Torres

**Abstract** This work presents a method to analyse 3D flat objects and to measure variations of its surface. The method represents of the objects using processing techniques based on point cloud. The proposed method is focused on the deformation detection of elastic objects which are formed by flat faces. These deformations are usually caused when two bodies, a solid and another elastic object, come in contact and there are contact pressures among their faces. Our method describes an algorithm to estimate the deformation shape. It is done by calculating its skeleton. Particularly, the algorithm calculates the curvature values of the surface points on the object using an analysis of eigenvectors and eigenvalues. Afterwards, the points of the similar curvature are grouped in level curvatures. Finally, a set of the minimum path among level curvatures are calculated to obtain the skeleton. The paper shows a set of experiments which simulate the deformations caused by a robot hand in manipulation tasks of flat objects.

**Keywords** 3D shape · Surface · Curvature · Surface normal · Geometric modelling · Deformations visual perception

---

C.M. Mateo (✉)  
Institute for Computer Research, University of Alicante,  
San Vicente del Raspeig, Alicante, Spain  
e-mail: cm.mateo@ua.es

P. Gil · F. Torres  
Physics, Systems Engineering and Signal Theory Department,  
University of Alicante, San Vicente del Raspeig, Alicante, Spain  
e-mail: pablo.gil@ua.es

F. Torres  
e-mail: fernando.torres@ua.es

## 1 Introduction

Generally, researchers about robotic manipulation have been focused to recognize rigid objects such as solids [1, 2]. But, in recent years, the manipulation process has changed to recognize articulated objects [3], deformable objects [4] such as soft objects [5] semi-solids such as organic material [6] or tissue [7]. Therefore, three basic types of rigid objects can be considered solids, elastics and deformable objects.

The object rigidity can be mathematically measured with three different interpretations: stiffness which is dependent on the force and the size of area where it is applied, hardness which defines the forces required to penetrate the material and toughness which is the amount of energy that a material can tolerate before it can be fractured.

Using tactile and/or force control is usually to manage the grasping processes [8]. Thus, a robot hand is able to do task manipulation of objects applying forces that is not usually enough large to break the surface structure or to drill it. In a rule, the forces never exceed a value which can cause a rupture or penetration. Notwithstanding, this is a complex task in which just the tactile control does not allow us to avoid deformations. More data are required to control the manipulation process if the deformations wish be controlled and measured [9]. Sometimes, the tactile and force information is poor, inconsistent or ambiguous to detect and analyse deformations in an object which is being manipulated [10]. In contrast to [11], the goal of this paper is to model and identify deformation in semi-solids with elastic properties from contactless sensors. Thus, visual sensors can assist to other sensors [1, 12, 13]. Firstly, they built a geometrically modeled object and secondly, they identify the object deformation for comparison between points of surface with and without deformation.

Consequently, the elastic object can be construed just as a stiff object because there is not surface penetration or rupture of its structure. The elastic modulus has often been used to measure the stiffness properties in the study of materials when they are known. The elastic modulus measures the applied force per unit area to deform an object surface. However, the elastic modulus cannot be used when the objects were made with an unknown material. In this paper, the stiffness can be measured by the curvature in surface points from the object geometry. The curvature features are computed from mesh points which models the object surface.

The paper is structured as follows: the concept of curvature to measure curves by means differential geometry is shown in Sect. 2. In Sect. 3, we present the method based on surface variation to model surfaces, detect and measure deformation in those. Experimental results of the deformations caused in virtual robot manipulation tasks are shown in Sect. 4. Finally, Sect. 5, contains the conclusions.

## 2 Mathematical Approach to Compute Curvature of Surfaces

In this paper, the differential geometry of curves is used as a tool to propose a method which allows us to analyse 3D surfaces. Generally, in computer vision, the 3D object surface consists of an unstructured point cloud represented as  $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3\}$ . If a plane object in the Euclidean space is deformed by pressing on its outside surface, the geometric properties change and some smooth curves could appear. Here, the curves are understood as a variation of surface, and they can be computed by measuring of the orientation change of normal vector to the surface.

Generally, each point  $\mathbf{p}_i$  satisfies a set of axioms in relation with its neighbourhood environment. Thus, the computation of the geometric properties of the curves depends on the  $k$ -nearest neighbour points to each  $\mathbf{p}_{ij}$ . Therefore,  $\mathbf{P}$  can be sampled as a set of patch  $\mathbf{N}_j$  and each one is a subset of points  $\mathbf{p}_{ij}$ . Both, the size (radius) and number of points of a patch (dense) influence the accuracy to compute the geometric properties of the curves. Few points cause inaccuracy. In contrast, many points distort the values hindering the detection of orientation changes in the surface. In this last case, the detection is smoothed.

An analysis of the eigenvalues computed from covariance matrix of points in a neighbourhood environment according to (1) can be used to estimate the local geometric properties of a patch of the surface [14]. To define the covariance matrix is applied PCA (Principle Component Analysis) as follows:

$$\mathbf{C}_P = \mathbf{P}\mathbf{P}^T = \begin{bmatrix} \mathbf{p}_{i1} - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_{ik} - \bar{\mathbf{p}} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{i1} - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_{ik} - \bar{\mathbf{p}} \end{bmatrix}^T \quad (1)$$

where each  $\mathbf{p}_{ij}$  is a point of the neighbourhood environment  $\mathbf{N}_j$  and  $\bar{\mathbf{p}}$  is the centroid of the patch. And  $k$  defines the number of points  $\mathbf{N}_j$ .

Equation (1) is solved by Turk and Pentland method [15] that allow us the computation of eigenvalues and eigenvectors of  $\mathbf{C}_P$  with low computational cost by building a matrix  $\mathbf{A} = \mathbf{P}\mathbf{P}^T$  and by applying singular value decomposition (SVD) to  $\mathbf{A}$  as follows:

$$\mathbf{A} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{v}'_j \rightarrow \mathbf{P}^T \mathbf{P} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{v}'_j \quad (2)$$

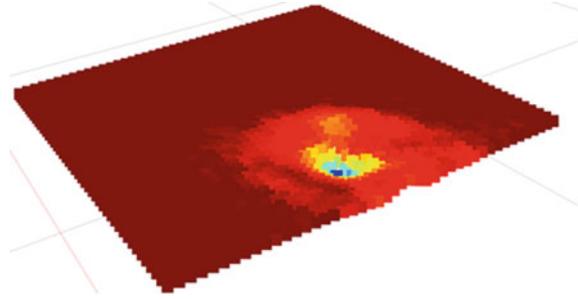
where  $\mathbf{v}'_j$  are the eigenvectors and  $\lambda'_j$  are the singular values of the matrix  $\mathbf{A}$ . Multiplying by  $\mathbf{P}$  is obtained:

$$\mathbf{P}\mathbf{P}^T \mathbf{P} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{P}\mathbf{v}'_j \rightarrow \mathbf{C}_P \mathbf{P} \cdot \mathbf{v}'_j = \lambda'_j \cdot \mathbf{P}\mathbf{v}'_j \quad (3)$$

Then, the eigenvalues and associated eigenvectors of  $\mathbf{C}_P$  can be obtained as:

$$\lambda_j = \lambda'_j \text{ and } \mathbf{v}_j = \frac{\mathbf{P} \cdot \mathbf{v}'_j}{\sqrt{\lambda'_j}} \quad (4)$$

**Fig. 1** Level curves set  $\mathbf{S}_P$  computed from a surface  $\mathbf{P}$  with a deformation



The eigenvalues sum provides information about the surface variation between each point of the patch  $\mathbf{N}_j$  and its centroid. In addition, the smallest eigenvalue provides a measure of the variation along the normal vector to the surface. Consequently, the eigenvalues can help to classify the concavity of the surface at each point, and the set of all defines the curvature parameter. Thus, the local maximum curvature [16] of  $\mathbf{p}_{ij}$  within the patch  $\mathbf{N}_j$  can be computed by:

$$c_{ij} = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (5)$$

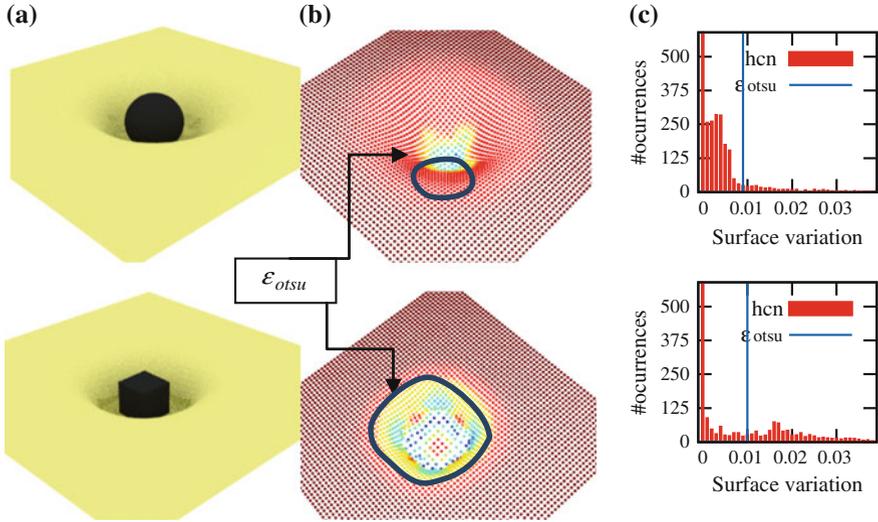
where  $\lambda_0 \leq \lambda_1 \leq \lambda_2$  are eigenvalues of  $\mathbf{C}_P$ . The associated eigenvector  $\mathbf{v}_0$  is the normal vector to the tangent plane of the patch surface  $\mathbf{P}$  that define  $\mathbf{v}_1$  and  $\mathbf{v}_2$ . The set of level curves is defined as a function  $\mathbf{S}_P : \mathfrak{R}^3 \rightarrow \mathfrak{R}$  as follows:

$$\mathbf{S}_P(\Phi) = \{(x, y, z) \in \mathfrak{R}^3 : \Phi(x, y, z) = l\} \quad (6)$$

where  $l$  is a constant value and it represents a level curve in the surface. Then, every level curve is computed like a cluster with the same colour that represents the points of  $\mathbf{P}$  with a similar value of curvature (Fig. 1).

### 3 Our Method to Find Surface Variations

In this work, the surfaces are represented as a point clouds. It is worth nothing that the data of  $\mathbf{P}$  can be obtained from a structured or unstructured way.  $\mathbf{P}$  is structured whether it can be stored in a matrix  $\mathbf{P}_{X \times Y}$  where  $X$  and  $Y$  are the number of rows and columns, respectively. This occurs whether  $\mathbf{P}$  is acquired from a sensor like a RGBD or ToF, then each point corresponds with a position of the sensor. In contrast,  $\mathbf{P}$  is unstructured whether it can only be stored as a vector matrix  $\mathbf{P}_{X \times 1}$ . In this last case,  $\mathbf{P}$  has no direct relationship with the data of range image.  $\mathbf{P}$  is usually unstructured when it is built from a virtual CAD model (Fig. 2a). Consequently, in order to find the neighbor of each point of  $\mathbf{P}$  is typically used either kd-tree or octree data structure.



**Fig. 2** **a** CAD Model of a deformation caused in a planar surface by contacting of two different geometric objects: *Sphere* and *Cube*. **b** Level curves set computed from the deformation and represented with a different colour  $l$ . **c** Histogram

The proposed algorithm has two phases: Initialization and Extract-Curvatures. The first step is only computed when  $\mathbf{P}$  is unstructured because it needs to be structured to get sorted points. The closest points in the Euclidean space must be stored as neighbors within  $\mathbf{P}$ . Thereby, it is possible to search points with the same geometrical properties and located in the same neighbourhood environment. This is essential to calculate the level curves from the curvature parameter and to detect surface variations given by the transversal paths to the level curves.

The proposed method detects deformations of any surface represented as  $\mathbf{P}$ . It is based on finding the critical points of  $\mathbf{P}$  in which there are surface variations. In this work, the critical points are points belonging to different level curves  $\Phi$  of the surface but also, they lie in the transversal path that goes across the level curves by fitting of singular points. The singular points are defined as the points with maximum curvature values, and whose curvature value is estimated like a curvature threshold computed from a Histogram of curvature.

### 3.1 Histogram of Curvatures

Once the curvature parameter, according to (5), is calculated for all points within  $\mathbf{P}$  (Fig. 2b), a curvature histogram is built (Fig. 2c). It represents the distribution of the surface variation and it can be computed as follows:

$$H_P = \frac{\text{Number}(c_{ij})}{\text{size}(\mathbf{P})} \quad (7)$$

where  $c_{ij}$  is the curvature value, and  $\text{size}(\mathbf{P})$  represents the density or number of points used to sample the surface.  $H_P$  changes depending on the size of the neighbourhood environment  $\mathbf{N}_j$  used to compute each  $c_{ij}$ . The user must choose the radio according to the accuracy for the detection of abrupt changes in the surface (Fig. 5).

The histogram  $H_P$  is useful to find the singular points, that is to say both the points with  $\max(c_{ij})$  and the boundary points which define the border where there is no curvature variation in the surface. The boundary points split the curvature region and the non-curvature region. They are computed from  $H_P$  using the technique presented in [17]. The histogram allows us to find the curvature threshold  $\varepsilon_{otsu}$  by minimizing the standard deviation of the Gaussian distributions that represent the two zones (Fig. 2c).

```

Algorithm Finding curvatures
Begin
  // Initialization:
   $O \leftarrow P$ 
  For each leaf of  $O$  with points do:
    For  $i = \text{leaf}_x - 1 : \text{leaf}_x + 1$  do:
05    For  $j = \text{leaf}_y - 1 : \text{leaf}_y + 1$  do:
      For  $k = \text{leaf}_z - 1 : \text{leaf}_z + 1$  do:
         $N \xleftarrow{\text{add}} p_{ijk}$ 
      End For
    End For
10    End For
     $Q \xleftarrow{\text{add}} \{p, N\}$ 
  End For
  // Extract-Curvatures:
   $R \xleftarrow{\text{curvature}(p) \geq \varepsilon_{otsu}} Q$ 
15 While  $R \neq \{\}$  do:
     $C \xleftarrow{\text{add}} \text{head}(R)$  // Candidate list of point for def.  $D'$ 
     $R \xleftarrow{\text{remove}} \text{head}(R)$ 
    While  $C \neq \{\}$  do:
20     $D' \xleftarrow{\text{add}} \text{head}(D')$ 
    If  $\text{curvature}(\text{head}(D')) \leq \varepsilon_{otsu}$  then:
       $C \xleftarrow{\text{addAdjacent Points}} \text{head}(C)$ 
    End If
     $C \xleftarrow{\text{remove}} \text{head}(C)$ 
    End while
25     $D \xleftarrow{\text{add}} D'$ 
    While  $D' \neq \{\}$  do:
      If  $\text{curvature}(\text{head}(D')) \equiv \varepsilon_{otsu}$  then:
        FindMinPath(head(tail(D)), head(D'))
      End If
30     $D' \xleftarrow{\text{remove}} \text{head}(D')$ 

```

```

    End while
End while

```

### 3.2 Path of Critical Points to Measure the Surface Variations

We assume that the surface is always differentiable, then the surface gradient at a point is either zero or perpendicular to the level curve which represents the surface variation at that point. We inspect the critical points of the surface function along its level curves. Our critical points are the curve points which follow the direction of gradient.

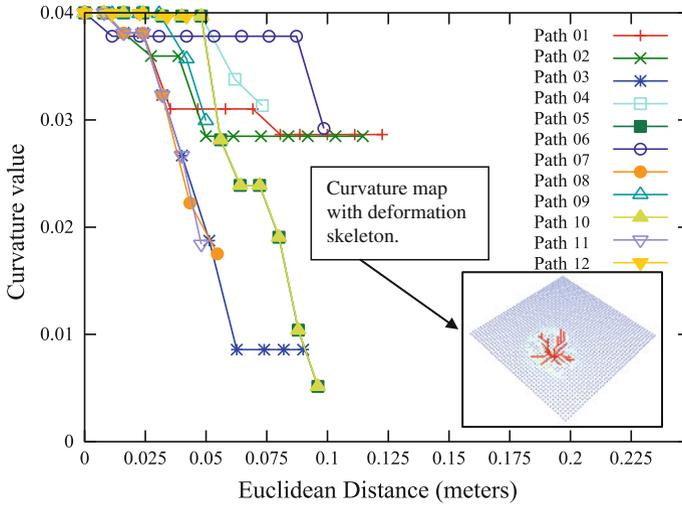
The proposed algorithm is used to find the critical and boundary points and the transversal path which define the deformation on the surface. The algorithm uses an octree  $O$  like [18] in order to contain all points  $p$  of  $\mathbf{P}$  as a sorted structured. Later, we create a priority queue  $Q$  of pairs  $\{p, N\}$ , where  $p$  is each point of  $O$  and  $N$  is its adjacency list.  $N$  represents the 26-connected neighbour points.  $Q$  puts on the head, the points  $p$  with the greater curvature values. Later,  $Q$  is filtered to only obtain the values of  $Q$  which are greater than  $\varepsilon_{otsu}$  then they are stored in  $R$ . Subsequently,  $R$  is crossed according to algorithm from line 18 until 24 in order to get a list with the deformations from the clusters  $D$ . Finally, we use Dijkstra's algorithm  $\text{FindMinPath}(v_1, v_2)$  to find the minimum paths. Let  $v_1$  be the point with maximum curvature (the head into each  $D'$ ) and let  $v_2$  be the target point (points into the boundaries).

## 4 Experiments

Several experiments and tests have been done. We have modelled the movements of a real robot hand considering both the kinematics and the virtual model (shape and structure) of their fingers and palm without physical constraints and without considering singularities, by using free software Blender. In particular, the model of robot hand corresponds to a Shadow Hand Robot available in our research laboratory.

### 4.1 Experiments to Compute Curvatures Skeleton

The deformations are described by curvature skeletons. A curvature skeleton is a set of minimum transversal paths. These path goes across the level curvatures, thus they start in the same critical point (maximum curvature) and end over the boundary points. An example of the skeleton is represented in the curvature map of the Fig.3. The chart of that figure shows the minimum transversal paths which define the behaviour of the deformation. Thereby, if all paths have similar distances then



**Fig. 3** Characterization of the transversal path variations computed from critical points

the deformations are homogeneous (in this case, the values are all close to 0.1 m and then the deformation shape could be understand like a circle). In opposition, if the paths represent different distances then the deformation is heterogeneous (in this case, other irregular deformation shapes can be read). Furthermore, the chart shows the deformation as a set of branches which represents several minimum transversal paths. The number of branches and the value of slope of them determine how is the deformation: steep, smooth, flat, etc. In Fig. 3, there are two groups of path (group 1: path 1, 2, 4, 6, 8, 11; group 2: path 3, 5, 7, 9, 10, 12) and each one tends to curvature values around to 0.03 (group 1) and 0.005 (group 2).

## 4.2 Experiments Applied to Grasping Tasks

Our experiments simulate several deformations which are caused by contact between fingers of the virtual robot hand and a planar object. The deformations depend on the orientation of the fingers, the trajectory used in the contact process, the shape of the finger and the pressure produced by it when the contact occurs (Fig. 4).

Each experiment consists of a movement that shows an image sequences. Each sequence is done from 60 images for testing. The algorithm computes the level curves set from both the curvature parameter and the histogram of curvatures. They allow us to obtain the evolution of the curvatures during a time sequence through the formation process of the deformation (Figs. 4 and 6). Afterwards, we estimate the path of critical points to measure the surface variation and the topographic profile of the deformation (Figs. 5 and 6).



The algorithm is implemented with the C++ language and using the open source libraries PCL, Boost and Eigen. It runs over a computer with a Core i7-4770k processor, equipped with 8GB of system memory and an nVidia GeForce 760GTX.

Figure 5 shows the dependence of the curvature measures when the radius of the neighborhood is chosen in a proper way or not. If the radius is not enough large then the smallest gaps on the surface are not detected by the algorithm. Also, Fig. 5 shows the evolution of the boundary points and how they grow quickly in the first iterations. This fact indicates that the contact between robot hand and elastic surface is occurring in that moment. Later, the growth is smoother and more progressive. The curve slope indicates the deformation level generated during the grasping over time. Figure 6 shows us how the minimum transversal paths retrieve us information to determine the deformation in an instant time. In the test 2, there are two dominant paths 2 and 3 because they are far from the rest of paths (those other are close and have similar length). The paths 2 and 3 determine the shape of the deformation caused on the surface. But also, the paths 0, 1, 4, 5 and 6 show a sharp slope which represents an abrupt deformation. This is much variation among level curves located near on the surface.

## 5 Conclusions

This paper has described a novel approach to analyse the deformation of flat surfaces. The proposed method builds curvature variation maps to measure the local unevenness in the surface by comparison among the points that lie on the same surface. Also, the method uses a novel algorithm to characterize and to describe the curvature variation caused by deformation. To do it, our algorithm identifies the level curves and later, it finds the critical points which define transversal paths among those level curves. Thus, we generate topographic profiles in the local maximum gradient directions. The gap, between the curvature value and the Euclidean distance of the critical points, determines whether or not deformation and the degree of slope that defines if the deformation is smooth or abrupt.

The results reveal that this surface analysis provides an empirical investigation on the effect of the deformations about the surface of elastic objects when several grasping tasks with robot hand are performed. And most importantly, an algorithm to measure these deformations has been implemented and tested successfully. It is programed in C++, and it can be integrated into robotic platforms.

**Acknowledgments** The research leading to these results has received funding from the Spanish Government and European FEDER funds (DPI2012-32390 and DPI2015-68087R) and the Valencia Regional Government (PROMETEO/2013/085).

## References

1. Saxena, A., Driemeyer, J., Ng, A.Y.: Robotic grasping of novel objects using vision. *Int. J. Robot. Res.* **27**(2), 157–173 (2008). doi:[10.1177/0278364907087172](https://doi.org/10.1177/0278364907087172)
2. Mateo, C.M., Gil, P., Torres, F.: Visual perception for the 3D recognition of geometric pieces in robotic manipulation. *Int. J. Adv. Manuf. Technol. (IJAMT)* (Springer, London) (2015). doi:[10.1007/s00170-015-7708-8](https://doi.org/10.1007/s00170-015-7708-8)
3. Katz, D.: Manipulating articulated objects with interactive perception. In: *Proceedings of the IEEE Robotics and Automation (ICRA)*, pp. 272–277 (2008). doi:[10.1109/ROBOT.2008.4543220](https://doi.org/10.1109/ROBOT.2008.4543220)
4. Hirai, S., Tsuboi, T., Wada, T.: Robust grasping manipulation of deformable objects. In: *Proceedings of the IEEE Symposium on Assembly and Task Planning*, pp. 411–416 (2001)
5. Berenson, D.: Manipulation of deformable objects without modeling and simulating deformation. In: *Proceedings of the Intelligent Robots and Systems (IROS)*, pp. 4525–4532 (2013). doi:[10.1109/IROS.2013.6697007](https://doi.org/10.1109/IROS.2013.6697007)
6. Gemici, M.C., Saxena, A.: Learning haptic representation for manipulating deformable food objects. In: *Intelligent Robots and Systems (IROS)*, pp. 638–645 (2014). doi:[10.1109/IROS.2014.6942626](https://doi.org/10.1109/IROS.2014.6942626)
7. Ramisa, A., Alenya, G., Moreno-Noguer, F., Torras, C.: FINDDD: a fast 3D descriptor to characterize textiles for robot manipulation. In: *Proceedings of the IEEE/RSJ Intelligent Robots and Systems (IROS)*, pp. 824–830 (2013)
8. Khalil, F.F., Payeur, P.: Dexterous robotic manipulation of deformable objects with multi-sensory feedback—a review. In: *Robot Manipulators Trends and Development*, Chap. 28 (2010). doi:[10.5772/9183](https://doi.org/10.5772/9183)
9. Khalil, F.F., Curtis, P., Payeur, P.: Visual monitoring of surface deformations on objects manipulated with a robotic hand. In: *Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE)*, pp. 1–6 (2010). doi:[10.1109/ROSE.2010.5675327](https://doi.org/10.1109/ROSE.2010.5675327)
10. Pomares, J., Gil, P., García, G.J., Sebastián, J.M., Torres, F.: Improving detection of surface discontinuities in visual-force control Systems. *Image Vis. Comput.* **26**(10), 1435–1447 (2008)
11. Boonvisut, P., Cenk-Cavusoglu, M.: Identification and active exploration of deformable object boundary constraints through robotic manipulation. *Int. J. Robot. Res.* **33**(11), 1445–1461 (2014). doi:[10.1177/0278364914536939](https://doi.org/10.1177/0278364914536939)
12. Smith, P.W.: Vision based manipulation of non-rigid objects. In: *Proceedings of the IEEE Robotics and Automation (ICRA)*, pp. 3191–3196 (1996). doi:[10.1109/ROBOT.1996.509198](https://doi.org/10.1109/ROBOT.1996.509198)
13. Mkhitarian, A., Burschka, D.: Vision based haptic multisensory for manipulation of soft, fragile objects. In: *Proceedings of the IEEE Sensors*, pp. 1–4 (2012)
14. Shaffer, E., Garland, M.: Efficient adaptive simplification of massive meshes. In: *Proceedings of the IEEE Conference on Visualization*, pp. 127–134 (2001). doi:[10.1109/VISUAL.2001.964503](https://doi.org/10.1109/VISUAL.2001.964503)
15. Turk, M., Pentland, A.P.: Face recognition using eigenfaces. In: *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 586–591 (1991). doi:[10.1109/CVPR.1991.139758](https://doi.org/10.1109/CVPR.1991.139758)
16. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: *Proceedings of the IEEE Conference on Visualization*, pp. 163–170 (2002)
17. Otsu N.: A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979). doi:[10.1109/TSMC.1979.4310076](https://doi.org/10.1109/TSMC.1979.4310076)
18. Papon, J., Abramov, A., Scholer, M., Woergoetter, F.: Voxel cloud connectivity segmentation-supervoxels form pointclouds. In: *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 2027–2034 (2013). doi:[10.1109/CVPR.2013.264](https://doi.org/10.1109/CVPR.2013.264)

# On Structure and Distribution of Software for Mobile Manipulators

Andreas Schierl, Andreas Angerer, Alwin Hoffmann,  
Michael Vistein and Wolfgang Reif

**Abstract** Complex robot applications or the cooperation of multiple mobile robots are use cases of increasing popularity where software distribution becomes important. When developing mobile robot systems and applications, software structure and distribution has to be considered on various levels, with effects on the organization and exchange of data. With respect to structure and distribution, this work proposes to distinguish between real-time level, system level and application level. Ways of structuring the software, as well as advantages and disadvantages of distribution on each level are analyzed. Moreover, examples are given how this structure and distribution can be realized in the robotics frameworks OROCOS, ROS and the Robotics API. The results are demonstrated using a case study of two cooperating KUKA youBots handing over a work-piece while in motion, which is shown both in simulation and in a real world setup.

**Keywords** Mobile robots · Cooperative manipulators · Software distribution · Robot architectures

---

A. Schierl (✉) · A. Angerer · A. Hoffmann · M. Vistein · W. Reif  
Institute for Software and Systems Engineering, University of Augsburg,  
Augsburg, Germany  
e-mail: schierl@isse.de

A. Angerer  
e-mail: angerer@isse.de

A. Hoffmann  
e-mail: hoffmann@isse.de

M. Vistein  
e-mail: vistein@isse.de

W. Reif  
e-mail: reif@isse.de

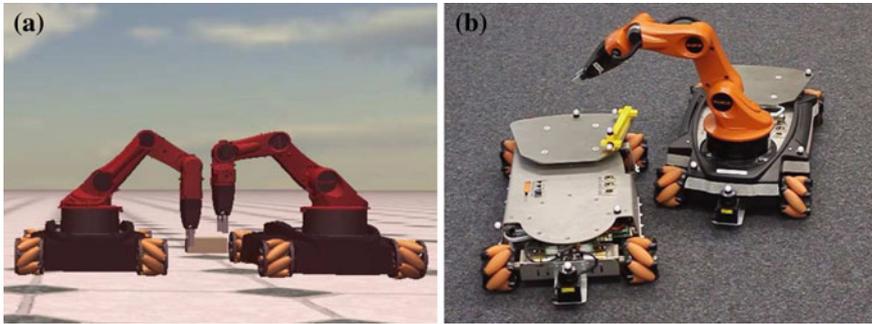
## 1 Introduction

With service robotics getting more and more important, robot demand has extended from factory automation towards mobile robot systems. Thus, the topic of mobile robotics has become important, and a lot of research has been performed. However, in some cases a single mobile robot is not sufficient to execute a task or deal with all problems [1]. For example, Knepper et al. [2] introduced the *IkeaBot* which is a coordinated furniture assembly system with multiple KUKA youBots. Based on that work, a flexible assembly system with cooperative robots was suggested in [3]. When multiple robots work together, cooperative mobile manipulation becomes important and poses new challenges—especially to the software structure.

Already in the 1990s, Dudek et al. [4] and Cao et al. [5] described a classification for cooperative mobile robotics. Dudek et al. [4] defined a taxonomy for multi-agent mobile robotics, where such a system can be e.g. differentiated by the number of agents, the communication range, topology and bandwidth as well as the reconfigurability and the composition of homogeneous or heterogeneous agents. Similarly, Cao et al. [5] defined research axes such as the differentiation, the communication structure or different types of modeling other agents. Moreover, they made a distinction between centralized and decentralized control in the software for mobile robots. Later, Farinelli et al. [6] added a classification based on coordination, where they compared cooperation, knowledge, coordination and organization of agents in multi-agent systems. This classification defines aware multi-robot systems where each robot knows that it is cooperating with other robots.

Concentrating on aware systems, this work analyzes which software structure and distribution can be used. However, there is not only *one* software structure and distribution that is possible in cooperative mobile robotics. From our point of view, there are different levels and aspects in the software architecture of a multi-robot system that can be distributed, ranging from the application level to low-level real-time device control. Typical early software architectures such as 3T [7] include reactive behavior as well as planning and execution, but are limited to single robot systems. The decision to distribute the software solution on one of the mentioned levels may affect the other levels and the complexity of the solution. Every possible solution, i.e. the chosen distributed software architecture, has its advantages but also its shortcomings that must be considered. For example, in current robotics systems component-based software frameworks which facilitate transparent distribution (e.g. ROS) are commonly used (cf. [8]). However, although communication between distributed components is easy using these frameworks, the decision about distribution as well as the assignment of responsibilities to certain components affects the overall capabilities of the solution (e.g. support for hard real-time).

In this work, we introduce a taxonomy for distributed software architectures in cooperative mobile robotics. However, the concepts are not specific to mobile robots, but also apply to other cooperating robots such as teams of industrial robots in automation systems. Hence, we are interested in finding a generalized representation and description of distributed robotics systems that can be used to classify and compare



**Fig. 1** Case study of two cooperating youBots: **a** Simulated scenario. **b** Real world scenario

software architectures of distributed robots. Additionally, we give advantages and disadvantages of applying distribution on different levels. It is important to be able to compare distributed robotics systems as the chosen software architecture often influences or sometimes even determines the complexity of the solution.

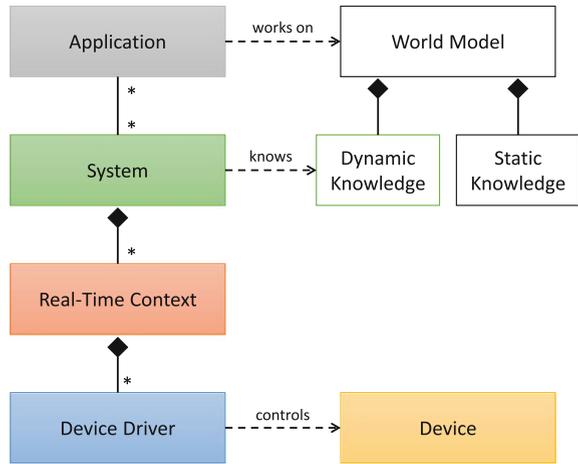
For experimental results, a case study is used where two KUKA youBots [9] physically interact with each other to transfer a work piece from one robot to the other. This scenario is inspected in different levels of difficulty. In simulation (cf. Fig. 1a), both robots can be coordinated in real-time, exact position information is available and all control inputs and trajectories are exactly followed. Initially, workpiece transfer happens while the first robot is standing still, and then while both robots are moving. As some of the assumptions made for simulation are not valid for real robots, a second scenario with real youBots is analyzed (cf. Fig. 1b). There, a youBot platform (left, without arm) is initially carrying a workpiece, which is then picked up by the second youBot (right) while both youBots are moving. The youBots and the workpiece are tracked externally using a Vicon optical tracking system, so precise position information is available.

In Sect. 2, the different levels for structuring and distribution of software for (mobile) robots is introduced. Subsequently, the identified levels (i.e. the real-time, system, and application level) are discussed in Sects. 3–5. Implications on the world model of robotics software are addressed in Sect. 6. To show the general validity of the suggested taxonomy, the possibilities of structuring and distribution on each level are explained using three different robotic frameworks in Sect. 7. Experimental results with different possible solutions of the case study are presented in Sect. 8. Finally, Sect. 9 concludes this work and gives an outlook.

## 2 Different Levels of Structuring Robotics Software

When designing a software architecture for a distributed robot scenario, we propose to group the software components into different layers as illustrated in Fig. 2. Each of the hardware devices present in the robot solution is represented and controlled

**Fig. 2** Software structure for distributed robots



by a *device driver* which is defined as the component that communicates with the hardware device through the vendor-specific interface. Additionally, the device driver is responsible for exchanging data with the surrounding software components. It has to derive control inputs and forward them to the device, as well as receive feedback from the device and make it available to other software components.

Each device driver can belong to a *real-time context* where data transfer and coordination between components occur with given timing guarantees. Depending on the implementation, the real-time context can contain only one device or span over multiple devices. Within a real-time context, reactions to events or the processing of sensor data can be guaranteed to happen before a given time limit. This allows to handle safety-critical situations that require timing guarantees (e.g. to stop the robot if an obstacle occurs), or to execute precise behaviors (such as actions that happen at a given point on a trajectory).

Above the real-time level, one or multiple real-time contexts belong to a *system* where all knowledge is shared between the components. Hence, all components within one system are allowed to access each other’s data, as well as to communicate with and send commands to each other. This allows components to directly include other components’ data into planning or computation, however no real-time guarantees are given unless the communication is handled within one real-time context.

To perform a desired task, systems can be controlled from *applications* that coordinate the work flow. Within an application, data is read and commands are sent to the controlled systems, so that the corresponding devices execute the task. However, if data from one system is required for an action in another system, it is the responsibility of the application or the deployment to facilitate the data transfer, as there is no concept of implicit shared data between systems. The overall behavior of cooperating robots results from the interplay of all applications that coordinate the robots.

Each application performs its work based on its *world model*, i.e. the knowledge about the controlled devices and systems, as well as about the environment including other (cooperating) devices. This includes geometric information such as positions and orientations of the relevant objects, as well as also physical data (such as mass and friction), shape data (such as 3D models for visualization or collision checks), semantics and ontologies. Information from the world model can be stored and used in applications, systems or real-time contexts, and can also be shared between different applications and systems. Structurally, the world model data can be differentiated into dynamic and static knowledge, with static knowledge (e.g. maps, shapes and ontologies) being valid everywhere, while dynamic knowledge (such as positions and sensor data) may be known in only one system or be different in different systems.

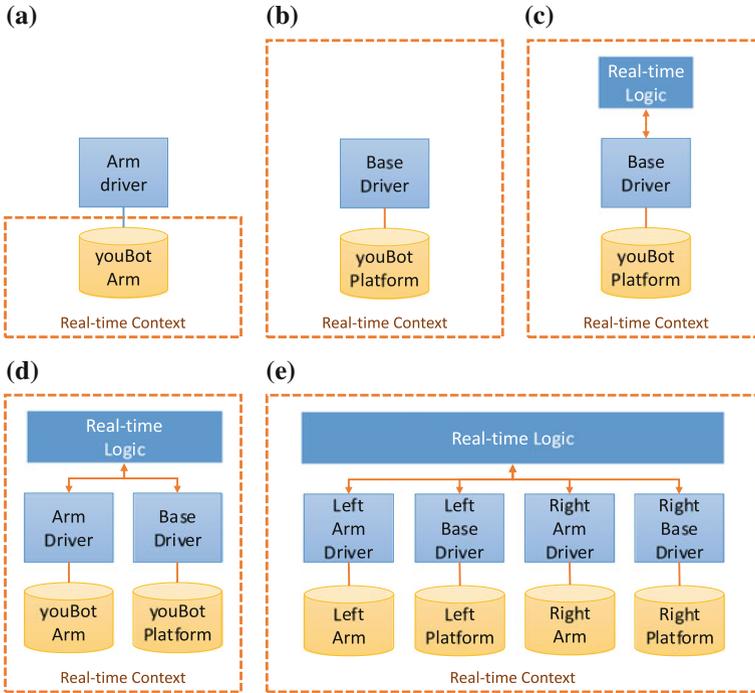
Depending on the requirements and technical limitations of the robot solution, the size and distribution of real-time contexts, systems and applications and thus the structure of the software can vary. The following sections discuss different design decisions concerning this structure based on the examples of the case study and using the three popular frameworks OROCOS, ROS and the Robotics API. OROCOS as a component framework mainly targets control systems with real-time guarantees [10]. The main focus of ROS is to be a component framework with transparent distribution, which over time has collected a large amount of algorithms as reusable components [11]. The Robotics API focuses on high-level robot programming using modern programming languages (such as Java) while still providing real-time guarantees [12].

### 3 Real-Time Level

First, the existing hardware devices and device drivers have to be grouped into one or more real-time contexts. Within a real-time context, reactions to events or the processing of sensor data can be guaranteed to happen before a given time limit. Having hard real-time guarantees allows to control precise behaviors or to handle safety-critical situations that need strict timing. In the mobile manipulator example of the case study, the available device drivers have to be grouped into real-time contexts, especially focusing on the two youBot platforms and one or two arms.

#### 3.1 *Software Structure on the Real-Time Level*

Generally speaking, there are five different choices to structure these devices (and their device drivers) into real-time contexts. In the first case (i.e. the real-time context in Fig. 3a), the device driver software is written without real-time in mind. Here, the real-time context only spans the (possibly real-time capable) firmware or controller present in the device itself. For the youBot arm, this could mean that only the position control mode of the arm motor controllers is used. Thus, it is sufficient



**Fig. 3** Variants of real-time contexts: **a** No real-time. **b** One device. **c** One device incl. logic. **d** Two devices. **e** All used devices, together with logic

to give one joint configuration that the robot is expected to move to. While easy to implement, no synchronization between the joints or support for precise Cartesian space motions is possible. Moreover, no guarantees can be given regarding the interpolation quality of user-defined trajectories or the timing of reactions to events (unless supported directly by the device).

In the next case (cf. Fig. 3b), the device driver and the communication with the device is implemented in a real-time capable fashion. This requires to use a real-time operating system (RTOS) and more care when implementing the device driver, but allows to execute precise custom trajectories and handle sensor events with timing guarantees. Besides the device driver, additional real-time logic (cf. Fig. 3c) can be present that implements control, trajectory tracking or coordination of the device. For example, a real-time capable driver running at 250Hz can be implemented for the youBot Platform on a RTOS such as VxWorks or Xenomai. It provides the motor controllers with smooth control set points for velocity or torque control, which allows precise user-defined trajectories or custom feed-forward or feedback control laws. However, as a drawback only information that is provided by the device itself can be included in the control law. For example, only the wheel position can be controlled exactly, but the position of the entire robot in Cartesian space is inaccurate (due to

wheel slip and other factors limiting odometry precision), and the platform motion cannot be synchronized with the arm motion.

Increasing the real-time context, multiple devices can be combined up to all devices that are physically connected to the controlling computer (cf. Fig. 3d). Both the youBot arm and the platform—connected to the onboard computer via EtherCAT—can be controlled from a real-time capable software on a RTOS. In this way, coordinated motions between platform and arm are possible by combining the five joints of the arm and the three degrees of freedom provided by the omni-directional platform. This allows, for example, to execute Cartesian space motions of the end-effector relative to a point in Cartesian space known to the youBot (such as the position where the youBot started assuming that odometry exactly provides the current position relative to this origin based on wheel rotations). Additionally, one device can react to events that occur at other devices or are detected by other sensors. However, in order to be able to specify these reactions, either this part of the real-time logic has to be changed for a specific application, or a flexible specification language is required in the real-time logic [13]. To enable easy cooperation between multiple robots, the devices of all robots could be combined into one real-time context (cf. Fig. 3e). However, if the corresponding devices are connected to different PCs, real-time distribution becomes important to establish this kind of real-time context.

### 3.2 Distribution of Real-Time Contexts

In the simplest cases, all devices structured into one real-time context are connected to the same computer (cf. Fig. 4a), which requires no distribution. Regarding our case study where both youBot platforms are not connected to one computer using a single (wired) EtherCAT bus, the real-time context has to be distributed in order to achieve

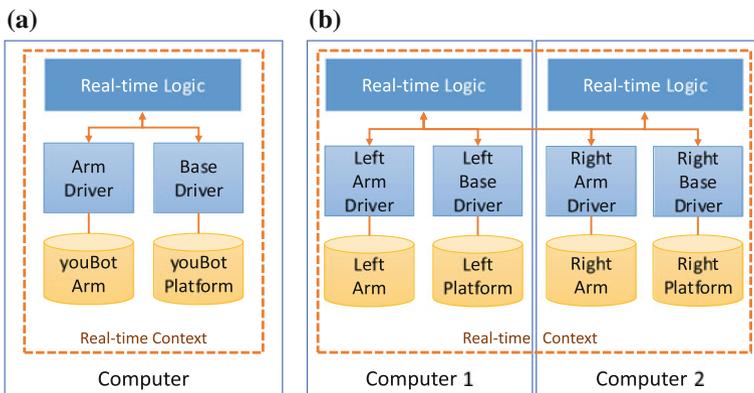


Fig. 4 Examples for distributing real-time contexts: a No distribution. b Distribution over two computers

this structure (cf. Fig. 4b). However, to distribute a real-time context, special real-time capable communication is required. For stationary robots such as manipulators or automation systems, as well as for complex mobile robots where different devices are connected to the different on-board computers (such as the PR2 or DLR's Justin), this is possible through Ethernet or a field bus like EtherCAT. In the automation domain, standard equipment such as PLCs are used, while in robot research software frameworks such as aRDx [14] or OROCOS [10] are preferred. But between mobile robots, using a wired connection usually is no option, and standards for general purpose real-time capable wireless connections are not yet common, so providing a single real-time context is not yet widely usable. In summary, while distributing a real-time context over multiple computers can improve the scalability of the solution (w.r.t. processing power or device connectivity), the need for deterministic communication implies special requirements (such as field bus hardware or dedicated networks) that make the solution more complex or expensive.

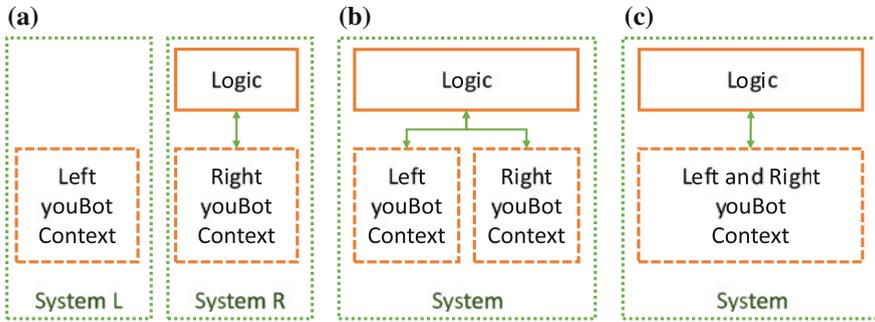
## 4 System Level

Proceeding to the system level, one or more real-time contexts can be grouped into one system. Within a system, all components are allowed to access each other's data, and to communicate with and send commands to each other. This allows components to directly include other components' data into planning or computation, though no real-time guarantees are given (unless handled within one real-time context).

### 4.1 *Software Structure on the System Level*

Working with the two youBots, three different structures of systems are possible: As shown in Fig. 5a, each youBot can work in its own system optionally together with further logic used in the system. Then, no implicit data transfer occurs between both youBots. The second option is to group both real-time contexts for the two youBots into one system (cf. Fig. 5b), in addition with further computation logic. Then, the system introduces communication between both real-time contexts that however does not provide real-time guarantees. Finally, one real-time context spanning both youBots can be used in a system (cf. Fig. 5c), which allows real-time cooperation of the youBots. However, this may require a distribution of the real-time context.

Using a big system spanning all robots (cf. Fig. 5b, c) has the advantage of simplifying application programming or deployment: All the data that any component might need is made available everywhere in the system. Hence, no manual data transfer is required. This especially covers the world model. Within one system, a consistent world model is possible, because the best knowledge about the world is available to every component. Moreover, every change to the common world model is available to every component immediately.



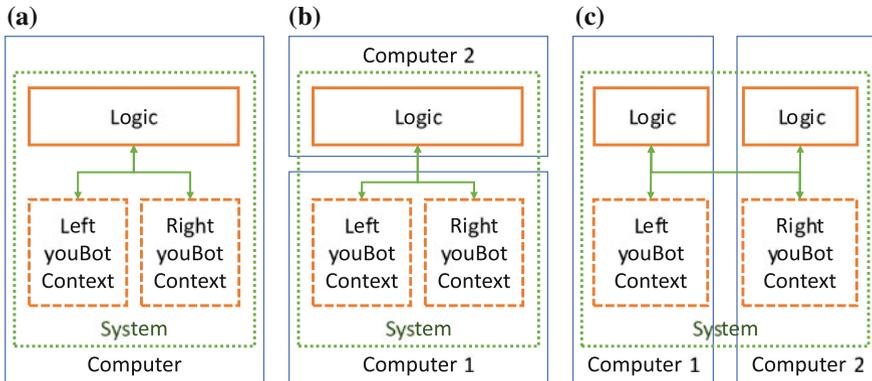
**Fig. 5** Variants of system structure: **a** Two separate systems. **b** Two real-time contexts. **c** One real-time context

However, there can be various reasons to use multiple systems: The sheer amount of data present in a big (multi) robot system can be a technical reason. Scalability can be limited by the management overhead induced by the data transfer between a great amount of components, and the addressing or mapping of data to components can become problematic. Furthermore, network bandwidth or reliability can be a limiting factor. In particular, this can be a problem when multiple robots are used that cooperate in varying teams. While for constant teams the corresponding robots could be joined into one system, varying teams quickly increase the required system size as all robots that might work together in a team at any time have to be within the same system.

But also more political reasons can opt for the separation into multiple systems, if cooperating robots belong to different people or parties. In this situation, not everyone might want to provide access to all of the robot’s data, or allow everyone else to control the robot. Then, matters of trust or access control become important that do not fit into the share-everything theme of a system. However, these reasons do not occur between the different devices within one robot (such as the arm and the platform of one youBot), so grouping them into two different systems does not really make sense and has been omitted in the figure.

## 4.2 Distribution of Systems

Looking at the distribution aspect of a system, some constraints are given by the real-time contexts: As the real-time contexts have to communicate with the corresponding hardware devices, the assignment to a certain computer is usually given. Depending on the devices and their connectivity, this can lead to a solution without distribution (cf. Fig. 6a), if all devices are connected to the same computer. In this case however, further logic components can be moved to a different computer as shown in Fig. 6b, based on performance deliberations. When the devices are connected to different



**Fig. 6** Examples for distributing systems: **a** No distribution. **b** Separate logic. **c** Distributed devices

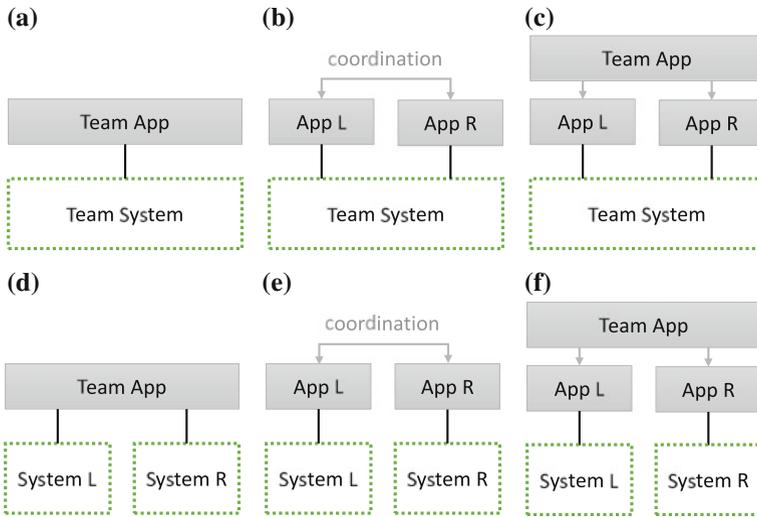
computers and are not handled within a distributed real-time context, distribution of the system is required. Here, each real-time context's assignment is given, while the further logic components can be assigned to the different computers based on further requirements (cf. Fig. 6c). For communication between the different components and real-time contexts, no timing guarantees are required. When data is to be exchanged between components on the same computer, communication can be performed locally, while otherwise network connections can be used to transfer the corresponding data and make necessary information available. Therefore, standard communication methods including wireless ones such as WiFi are applicable, however reliability or bandwidth can be limiting factors.

## 5 Application Level

To perform a requested task, one or multiple systems have to execute actions that are controlled and coordinated by one or multiple applications. There are various ways to specify applications: Mainly sequential workflows can well be expressed as a programming language control flow. For reactive behavior, model-based approaches such as statecharts (e.g. [12, 15, 16]) or Petri nets (e.g. [17]) can be more appropriate. Solutions offered by different robotics frameworks are discussed in Sect. 7.

### 5.1 Software Structure on the Application Level

Figure 7 gives the different possibilities to structure the application(s) for controlling two youBots. One way is to control all robots from one application as shown in Fig. 7a. This defines all the interaction present in the solution in one place and thus makes



**Fig. 7** Variants of application structure: **a** One app. **b** Two apps. **c** Three apps. **d** Two systems, like (a). **e** Two systems, like (b). **f** Two system, like (c)

it easier to understand. However, if varying teams are needed in a certain scenario, the corresponding application has to coordinate all robots at the same time. This can become confusing if the concurrent execution of multiple subtasks is encoded in one control flow or sequential state machine. Thus, separating concerns into subtasks, one for each team, should be considered within the application.

Another way is to use multiple applications, e.g. one for each controlled robot (cf. Fig. 7b). In this way, team behavior can be implemented by only locally and independently describing the behavior of each robot. However, the applications have to coordinate, either through explicit communication or through observation of the environment or other robots. As a third way, a further application coordinating both applications is a possible solution (cf. Fig. 7c), which however leads to a coordination application similar to the one in Fig. 7a. Using separate applications can also be required for political reasons, as described in Sect. 4.

In a multi-application cooperation scheme, however, the resulting behavior is not easily understandable by looking at one place, but only by examining the interaction of all different applications involved. In multi-agent robot systems (e.g. [18, 19]) every agent is controlled by one application. The overall behavior is given by the agents' interaction. As shown by Hoffmann et al. [20], this can lead to self-organizing properties making the overall solution more robust. It is even possible that the application for each robot only implements low-level behaviors, and the resulting behavior completely emerges from the interaction [21].

Another structuring approach looks at the relation between applications and systems. It is possible to control (different devices of) one system from multiple applications (cf. Fig. 7b, c), one system from one application (cf. Fig. 7a, e) or multiple

systems from one application (cf. Fig. 7d). The software framework should allow an application that was intended for use with multiple systems (e.g. Fig. 7d) to also work when all corresponding devices exist in one system, while the distribution aspect on the system and real-time level is handled by deployment (e.g. through configuration). This may however not be possible the other way round, if the application relies on having one system (or even real-time context).

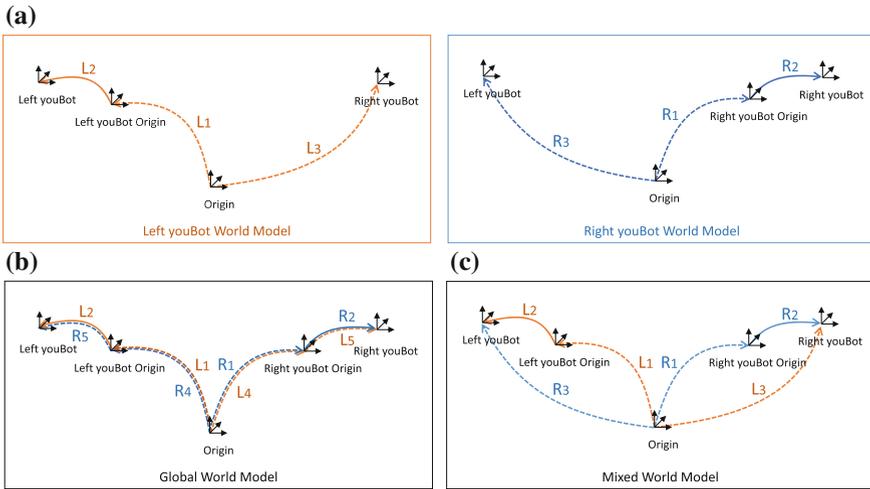
## 5.2 Distribution of Applications

When looking at the application level, different possible variants of distribution can occur. An application can run on another computer than the controlled system(s). In this case, network communication is required to transfer data between the system(s) and the application. However, this kind of distribution is equivalent to distribution on the system level, when seeing the application as a component in the system. When multiple applications are used, they can be executed on different computers. Then, coordination between the applications has to be implemented in a way supporting remote communication. Finally, a single application can be distributed onto multiple computers. Therefore, programming language concepts such as remote procedure calls or service-oriented architectures can be used, which can however be used in a standard fashion and do not require any special, robotics-related treatment.

## 6 World Model

If a robot is expected to cooperate with another robot, it needs the relevant data about that other robot. In the easiest case, both robots belong to the same system, thus the required information is already accessible. Then we refer to both robots as *controlled robots* within this system. Once robots from multiple systems are to be coordinated from one application, the world model becomes more complex. The information about robots in other systems can originate either from communication or observation. If the other system provides (maybe read-only) access to the information, then we term the other robot a *remote robot*. However, if the information is only available through observation, we define the other robot as an *observed robot*. Thus, the world model has to keep (a descending amount of) information about controlled, remote and observed robots.

When working with two youBots, each of which belongs to its own system, some information (e.g. the robot positions) is available in different systems with varying precision, and has to be organized. One way is to keep one *world model per system*. Figure 8a shows an example of two resulting world models. The figure concentrates on geometric information, with coordinate systems (frames) indicating positions of objects in Cartesian space, while arrows represent positions or transformations between these frames. Solid arrows indicate controllable positions, while dashed



**Fig. 8** Different options to represent the world model of multiple systems: **a** Separate world models. **b** Global world model. **c** Mixed world model

arrows indicate positions retrieved through measurement or communication. Orange arrows (i.e. L1–L3) belong to the system of the left youBot, while blue arrows (i.e. R1–R3) belong to the right youBot system. The frames could be augmented by additional information such as shape data, which is omitted here for clarity. In this case, the system of the left youBot knows information about the left youBot’s start position together with the distance traveled, as well as current position information about the right youBot. On the right youBot system, the opposite situation occurs.

Using separate models has the advantage that multiple systems can just use different instances of the same world model, which allows to re-use models designed for single systems to a large extent. However, for cooperation a lot of information has to be duplicated, such as robot models of robots that occur as controlled robot in one and as a remote robot in another system. Additionally, the different world models have to be kept consistent. For example, a workpiece that is grasped and moved in one system also has to appear as grasped and moved in the other system.

The second way is to keep one *global world model* with all the objects and relations, and to provide access to the different transformations or sensor values for each system (cf. Fig. 8b). This has the advantage that the world model is always consistent (as far as topology and static information is concerned, however different systems can still disagree about object transformations), and structural changes performed in one system are automatically present in other systems. However, this scheme lacks flexibility when dealing with observed robots: While a mobile robot can keep track of its movement since the start through odometry measurements, an observer has no way to achieve this information from outside. Thus, the frame graph contains two transformations for controlled and remote robots (cf. R1 and R2 in Fig. 8a), while for observed robots only one transformation is available (cf. R3 in Fig. 8a).

To solve this problem, we propose a *mixed world model* scheme (cf. Fig. 8c). In a mixed world model, the static data is shared between all systems, while dynamic data can be different for each system. For example, information about physical objects (such as youBot geometry) as well as static connections (such as the position of the youBot arm relative to the youBot platform) are shared. Dynamic connections (such as the position of the youBot relative to the origin, or the fact that the youBot is positioned relative to the world origin or youBot origin) can be different for each system. Still, in both systems it should be possible to compute the transformation of the left youBot to the right youBot, using the data and topology present in each system (and to use it for planning and execution). This combines the advantages of a shared world model with the flexibility to include limited observations, while still allowing the application to address one youBot in a uniform way.

## 7 Support in Software Frameworks

When looking at the different software frameworks, a different focus becomes obvious. ROS itself puts no emphasis on real-time guarantees, so typically a real-time context only spans one device (cf. Fig. 3a–c), where a single device is encapsulated into a ROS node, providing an interface to execute the required local commands. Sometimes multiple devices (such as a youBot arm and platform) are combined into one node, however this leads to higher coupling. On the system level, all nodes that run using the same ROS master are seen as a system. In this situation, all these nodes can subscribe to any data published by other nodes, and post messages or actions, thus providing the transparent data exchange required for a system.

To implement applications in ROS, Python scripts can be used for sequential workflows, while SMACH state machines introduced by Bohren and Cousins [22] allow reactive behavior. However, communication with multiple ROS masters from one application is not natively supported. To share data between different systems (i.e. ROS masters), concepts like *multimaster* or *foreign\_relay* [23] can be used to forward topics between multiple masters. The forwarded topics have to be configured during deployment. Additionally, working with multiple masters is one of the new use cases motivating ROS 2.<sup>1</sup> Within one ROS system, the world model is managed through data periodically published by different nodes. It includes transformation data as provided by the *tf service*, as well as robot models and data (e.g. sensor values) from other nodes.

In OROCOS usually most devices are combined into one real-time context (cf. Fig. 3b–e), because the framework targets real-time capable component systems with device drivers implemented in C++ on a RTOS. When using multiple robots, wired connections are used to ensure one real-time context, sometimes even for mobile systems [24]. Looking at the system level, OROCOS as a framework does not contain features for non-real-time communication, however often ROS is used

---

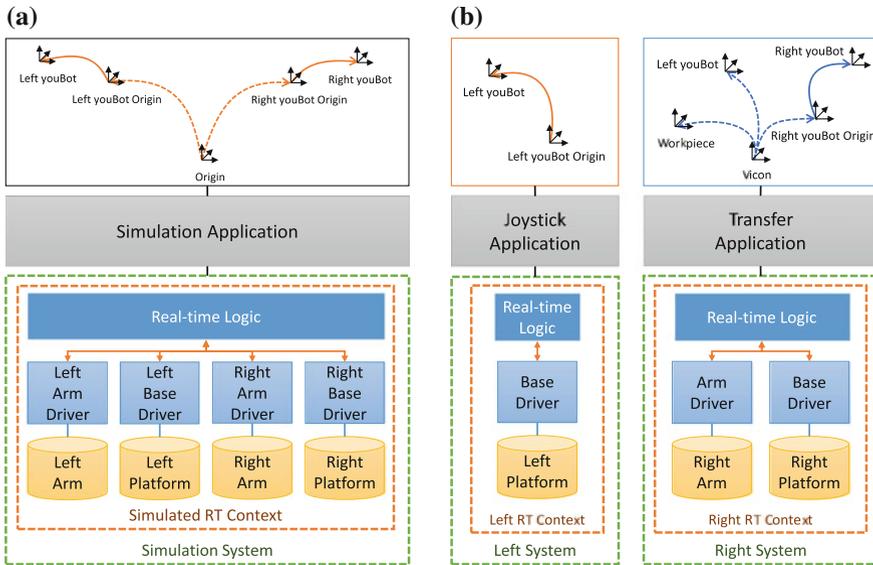
<sup>1</sup>[http://design.ros2.org/articles/why\\_ros2.html](http://design.ros2.org/articles/why_ros2.html).

to combine multiple real-time contexts into one common system [25] that allows for non-real-time communication and data sharing. Consequently, OROCOS does not provide direct support for access to multiple systems. Control flow can be expressed in LUA scripts, while complex coordination is possible using *rFSM statecharts* as suggested by Klotzbücher and Bruyninckx [15]. World models are usually implemented as components in an application dependent manner within the real-time context, which can include geometry, semantics and history [26].

Using the Robotics API, the underlying *Robot Control Core* [27] is implemented in C++ for Xenomai and includes real-time capable drivers for devices connected to the corresponding computer. Additionally, the *Realtime Primitives Interface* [27] allows for the flexible definition of real-time logic to execute user-defined tasks. In this way, all devices physically connected to the computer can form a real-time context (cf. Fig. 3c–e). The system term here refers to the concept of a *RoboticsRuntime*, which represents one real-time context and makes the data available to applications in a non-real-time way as well as to other devices in the same context for real-time reactions. Control flow can be expressed directly in Java applications, as well as through statecharts, e.g. by using Java libraries for SCXML. It is easily possible to use multiple systems in one application (as in the Factory 2020 case study [12]) and to share limited amount of data between different systems or applications—either through common data sources (such as a Vicon system connected to both youBots through WiFi) or through explicit direct transfer. The frame graph [28] contains semantic information such as the relation between frames (e.g. if they are static, temporary or have a transformation that can change during execution). With further information about physical objects (e.g. their physical properties or shape), it serves as a world model in an application and can be used for planning as well as task execution.

## 8 Experimental Results

The simulation experiments described in the case study have been performed using the Robotics API as a framework and its corresponding simulation and visualization engine. A single application as shown in Fig. 9a was used to control one system (cf. Fig. 5c) containing a single real-time context (cf. Fig. 3e). In this real-time context, both youBot arms and platforms were simulated, as well as the youBot grippers. The application was programmed in an object-oriented fashion, referring to the robots and work pieces as software objects. Moreover, the application used a single world model and expressed all interaction in the control flow of a Java method. Its initial version, where the first robot was not moving during transfer, was easily extended into the second version where both platforms were moving. This extension mainly consisted of adding commands to move the first platform, and to make the second platform synchronize its motion to the position and movement of the first platform. This showed the simplicity of programming and synchronizing robots in the



**Fig. 9** Software structure for the example application: **a** Simulation application. **b** Real application

idealized simulation case where every device belongs to the same real-time context and can be synchronized, and where exact positioning is available.

Transferring the results from simulation to reality, various changes had to be done. The tests were conducted on two KUKA youBots using a Vicon optical tracking system for external localization. A straightforward approach would be to combine both youBots into one real-time context. Then, the same distribution scheme as in simulation could be re-used, as well as large parts of the implementation. However, lacking real-time communication over wireless networks (cf. Sect. 3.2), this was not easily possible. On the other hand, while for stationary manipulators adding a common real-time context greatly simplifies and improves the precision of physical cooperation, in the mobile case the gains are less clear. This is because precise cooperation does not only require exact timing synchronization, but also exact spatial synchronization. For stationary manipulators, this can be achieved by appropriate calibration procedures. For mobile systems, this is in general more problematic due to sensor inaccuracies. External positioning systems—e.g. the Vicon system used here—can mitigate this problem. However, wireless real-time communication becomes a problem again when it comes to transmitting the position information to the youBots.

Thus, we decided to choose an alternative distribution approach as shown in Fig. 9b. On each internal computer an instance of the *Robot Control Core* was running, which formed the real-time context (cf. Fig. 3d) as well as the system for the corresponding youBot. On the system level, this structure corresponds to Fig. 5a. Vicon tracking data for both youBots and the workpiece was streamed to both youBot systems through a WiFi connection from an external PC running the Vicon software.

Looking at the application level, each youBot was controlled from a separate application (cf. Fig. 7e). The motion of the platform carrying the workpiece was controlled through teleoperation. The other youBot was controlled by a Java method similar to the one in the simulation case.

However, both applications used separate world models (cf. Fig. 8a). The workpiece and the other youBot platform were not modeled as Java objects, but only the Vicon position data was used to synchronize the motion and find the grasp position. The youBot arm used joint impedance control to mitigate position inaccuracies. Still, the experiment succeeded and the work piece could be transferred. Instead of separate world models, a single application with a mixed world model could have been used, which would be closer to the (single) world model in the simulation case. This way, changes to the world model topology (e.g. the information that the object has been grasped) would have automatically been transferred to the second youBot's system and static position data would be known to both youBots.

## 9 Conclusion and Outlook

In this paper, we introduced different levels for structuring the software for distributed robot applications: real-time, system, application level. Moreover, implications on the world model have been discussed. The structure on the different levels can be used and combined independently, motivated by technical as well as political constraints. The different options for structuring and distribution have been explained based on a case study of cooperating mobile manipulators and various robot frameworks, and evaluated both in simulation and in real world setup with two KUKA youBots. In the example applications, different ways to distribute the software on different levels have been introduced, and the advantages and drawbacks for the given scenario have been shown.

Overall, it became clear that there is not a single optimal way of structuring and distributing the software. The levels presented in this work will hopefully be a starting point that can help developers in designing and discussing their software architecture. Based on non-functional requirements to the developed solution (e.g. reactivity, synchronization quality, data privacy, trust), the choice of the appropriate distribution scheme and framework(s) for implementation should become easier. As next steps, we plan to implement the mentioned other ways of distribution and to evaluate the gains for the given scenario. This especially includes the use of a mixed world model, as well as ways to share a world model between multiple applications or to synchronize relevant structural changes.

## References

1. Parker, L.E.: Heterogeneous multi-robot cooperation. PhD thesis, Massachusetts Institute of Technology Cambridge, MA, USA (1994)
2. Knepper, R., Layton, T., Romanishin, J., Rus, D.: Ikeabot: an autonomous multi-robot coordinated furniture assembly system. In: Proceedings of 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 855–862. Karlsruhe, Germany, IEEE (2013)
3. Dogar, M., Knepper, R.A., Spielberg, A., Choi, C., Christensen, H.I., Rus, D.: Towards coordinated precision assembly with robot teams. In: Proceedings of 14th International Symposium of Experimental Robotics, Marrakech, Morocco, IFRR (2014)
4. Dudek, G., Jenkin, M.R.M., Milios, E., Wilkes, D.: A taxonomy for multi-agent robotics. *Auton. Robots* **3**, 375–397 (1996)
5. Cao, Y.U., Fukunaga, A., Kahng, A.: Cooperative mobile robotics: antecedents and directions. *Auton. Robots* **4**, 7–27 (1997)
6. Farinelli, A., Iocchi, L., Nardi, D.: Multirobot systems: a classification focused on coordination. *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* **34**, 2015–2028 (2004)
7. Bonasso, R.P., Kortenkamp, D., Miller, D.P., Slack, M.: Experiences with an architecture for intelligent, reactive agents. *J. Exp. Theor. Artif. Intell.* **9**, 237–256 (1995)
8. Brugali, D., Shakhimardanov, A.: Component-based robotic engineering (Part II). *IEEE Robot. Autom. Mag.* **20**, 100–112 (2010)
9. Bischoff, R., Huggenberger, U., Prassler, E.: KUKA youBot—a mobile manipulator for research and education. In: Proceedings of 2011 IEEE International Conference on Robotics and Automation, pp. 1–4. Shanghai, China, IEEE (2011)
10. Bruyninckx, H.: Open robot control software: the OROCOS project. In: Proceedings of 2001 IEEE International Conference on Robotics and Automation, pp. 2523–2528. Seoul, Korea, IEEE (2001)
11. Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source Robot Operating System. In: Workshop on Open Source Software, IEEE International Conference on Robotics and Automation, Kobe, Japan, IEEE (2009)
12. Angerer, A., Hoffmann, A., Schierl, A., Vistein, M., Reif, W.: Robotics API: object-oriented software development for industrial robots. *J. Softw. Eng. Robot.* **4**, 1–22 (2013)
13. Schierl, A., Hoffmann, A., Angerer, A., Vistein, M., Reif, W.: Towards realtime robot reactions: Patterns for modular device driver interfaces. In: Workshop on Software Development and Integration in Robotics. IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, IEEE (2013)
14. Hammer, T., Bauml, B.: The highly performant and realtime deterministic communication layer of the aRDx software framework. In: Proceedings of the 16th International Conference on Advanced Robotics, pp. 1–8. Montevideo, Uruguay, IEEE (2013)
15. Klotzbücher, M., Bruyninckx, H.: Coordinating robotic tasks and systems with rFSM statecharts. *J. Softw. Eng. Robot.* **3**, 28–56 (2012)
16. Stampfer, D., Schlegel, C.: Dynamic state charts: composition and coordination of complex robot behavior and reuse of action plots. *Intell. Serv. Robot.* **7**, 53–65 (2014)
17. Costelha, H., Lima, P.: Modelling, analysis and execution of multi-robot tasks using petri nets. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 3, pp. 1187–1190. Estoril, Portugal (2008)
18. Barbulescu, L., Rubinstein, Z.B., Smith, S.F., Zimmerman, T.L.: Distributed coordination of mobile agent teams: the advantage of planning ahead. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 1331–1338. Toronto, Canada (2010)
19. Chaimowicz, L., Kumar, V., Campos, M.: A paradigm for dynamic coordination of multiple robots. *Auton. Robots* **17**, 7–21 (2004)
20. Hoffmann, A., Nafz, F., Seebach, H., Schierl, A., Reif, W.: Developing self-organizing robotic cells using organic computing principles. In: Meng, Y., Jin, Y. (eds.) *Bio-Inspired Self-Organizing Robotic Systems. Studies in Computational Intelligence*, vol. 355, pp. 253–274. Springer, Heidelberg (2011)

21. Mataric, M.J.: Designing emergent behaviors: From local interactions to collective intelligence. In: Meyer, J.A., Roitblat, H.L., Wilson, S.W. (eds.) *From Animals to Animats 2: Proceedings of the 2nd International Conference on Simulation of Adaptive Behavior*, pp. 432–441. MIT Press (1993)
22. Bohren, J., Cousins, S.: The SMACH high-level executive. *IEEE Robot. Autom. Mag.* **17**, 18–20 (2010)
23. Schneider, T.: Distributed networks using ROS—cross-network middleware communication using ipv6. Master’s thesis, Technische Universität München (2012)
24. Klotzbücher, M., Biggs, G., Bruyninckx, H.: Pure coordination using the coordinator-configurator pattern. *CoRR abs/1303.0066* (2013)
25. Smits, R., Bruyninckx, H.: Composition of complex robot applications via data flow integration. In: *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, pp. 5576–5580. Shanghai, China, IEEE (2011)
26. Blumenthal, S., Bruyninckx, H., Nowak, W., Prassler, E.: A scene graph based shared 3D world model for robotic applications. In: *Proceedings of 2013 IEEE International Conference on Robotics and Automation*, pp. 453–460. Karlsruhe, Germany, IEEE (2013)
27. Vistein, M., Angerer, A., Hoffmann, A., Schierl, A., Reif, W.: Interfacing industrial robots using realtime primitives. In: *Proceedings of 2010 IEEE International Conference on Automation and Logistics*, pp. 468–473. Hong Kong, China, IEEE (2010)
28. Hoffmann, A., Angerer, A., Schierl, A., Vistein, M., Reif, W.: Service-oriented robotics manufacturing by reasoning about the scene graph of a robotics cell. In: *Proceedings of the 41st International Symposium on Robotics*, pp. 1–8. Munich, Germany, VDE (2014)

# Combination of HMM and DTW for 3D Dynamic Gesture Recognition Using Depth Only

Hajar Hiyadi, Fakhreddine Ababsa, Christophe Montagne, El Houssine Bouyakhf and Fakhita Regragui

**Abstract** Gesture recognition is one of the important tasks for human Robot Interaction (HRI). This paper describes a novel system intended to recognize 3D dynamic gestures based on depth information provided by Kinect sensor. The proposed system utilizes tracking for the upper body part and combines the hidden Markov models (HMM) and dynamic time warping (DTW) to avoid gestures misclassification. By using the skeleton algorithm provided by the Kinect SDK, body is tracked and joints information are extracted. Each gesture is characterized by one of the angles which remains active when executing it. The variations of the angles throughout the gesture are used as inputs of Hidden Markov Models (HMM) in order to recognize the dynamic gestures. By feeding the output of (HMM) back to (DTW), we achieved good classification performances without any misallocation. Besides that, using depth information only makes our method robust against environmental conditions such as illumination changes and scene complexity.

**Keywords** 3D gesture recognition · Gesture tracking · Depth image · Hidden Markov models · Dynamic time warping

## 1 Introduction

### 1.1 Motivation

The goal of Human Robot Interaction (HRI) research is to increase the performance of human robot interaction in order to make it similar to human-human interaction, allowing robots to assist people in natural human environments. As for communication between humans, gestural communication is also widely used in human robot

---

H. Hiyadi (✉) · F. Ababsa · C. Montagne  
Laboratoire IBISC, Université d'Évry Val d'Essonne, Évry, France  
e-mail: hajar.Hiyadi@ufrst.univ-evry.fr; hiyadi.h89@gmail.com

H. Hiyadi · E.H. Bouyakhf · F. Regragui  
Laboratoire LIMIARF, Université Mohammed V Agdal, Rabat, Morocco

© Springer International Publishing Switzerland 2016

J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*, Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_13

interaction. Several approaches have been developed over the last few years. Some approaches are based on data markers or gloves and use mechanical or optical sensors attached to these devices that transform reflexion of the members into electrical signals to determine the posture. These methods are based on various information such as the angles and the joints of the hand which contain data position and orientation. However, these approaches require that the user wear a glove or a boring device with a load of cables connected to the computer, which slows the natural human robot interaction. In the other side, computer vision is a non intrusive technology which allows gesture recognition, without any interference between the human and the robot. The vision-based sensors include 2D and 3D sensors. However, gesture recognition based on 2D images had some limitations. Firstly, the images can not be in a consistent level lighting. Second, the background elements can make the recognition task more difficult. With the emergence of Kinect [1], depth capturing in real time becomes very easy and allows us to obtain not only the location information, but also the orientation one. In this paper we aim to use only the depth information to build a 3D gesture recognition system for human robot interaction.

## 1.2 Related Work

A gesture recognition system includes several steps: detection of one or more members of the human body, tracking, gesture extraction and finally classification. Hand tracking can be done based on skin color. This can be accomplished by using color classification into a color space. In [2], skin color is used to extract the hand and then track the center of the corresponding region. The extracted surface into each chrominance space has an elliptical shape. Thus, taking into account this fact, the authors proposed a skin color model called elliptical contour. This work was extended in [3] to detect and localize the head and hands. In addition, the segmentation process is also an important step in tracking. It consists of removing non-relevant objects leaving behind only the regions of interest. Segmentation methods based on clustering are widely used in hand detection and especially K-means and expectation maximization. In [4] the authors combine the advantages of both approaches and propose a new robust technique named KEM (K-means Expectation Maximization). Other detection methods based on 2D/3D template matching were also developed [5–7]. However, skin color based approaches are greatly affected by illumination changes and background scene complexity. Therefore, recent studies tend to integrate new information such as depth. Indeed, depth information given by depth sensors can improve the performance of gesture recognition systems. There are several studies that combine color and depth information, either in tracking or segmentation [8–11]. Other works combine depth information, color and speech [12]. In [10], the authors use a silhouette shape based technique to segment the human body, then they combine 3D coordinates and motion to track the human in the scene. Filtering approaches are also used in tracking such as the Unscented Kalman Filter [13], the Extended Kalman Filter [14] and the Particle Filter [15]. Other methods are based

on points of interest which have more constraints on the intensity function and are more reliable than the contour based approaches [16]. They are robust to occlusions present in a large majority of images.

The most challenging problem in dynamic gesture recognition is the spatial-temporal variability, when the same gesture could be different in velocity, shape and duration. These characteristics make recognition of dynamic hand gestures very difficult compared to static gestures [17]. As in speech, hand writing and character recognition [18, 19], HMM were successfully used in gesture recognition [20–22]. Actually, HMM can model spatial-temporal time series and preserve the spatial-temporal identity of gesture. The authors in [23] developed a dynamic gesture recognition system based on the roll, yaw and pitch orientations of the left arm joints. Other mathematical models such as Input-Output Hidden Markov Model (IOHMM) [24], Hidden Conditional Random Fields (HCRF) [25] and Dynamic Time Warping [26] are also used to model and recognize sequences of gestures.

In this paper, we propose a robust classification realized by combining HMM and DTW methods for 3D dynamic gesture recognition. The basic framework of the technique is shown in Fig. 1. The Skeleton algorithm given by the Kinect SDK is used for body tracking. Only depth information is recorded. The 3D joints information are extracted and used to calculate new and more relevant features which are the angles between joints. Discrete HMM with Left-Right Banded topology are used to model and classify gestures. Finally, the output of HMM is given as input for DTW algorithm in order to measure the distance between the gesture sequence and a reference sequence. The final decision is given by comparing the distance calculated

**Fig. 1** Flowchart of the proposed 3D dynamic gesture recognition technique



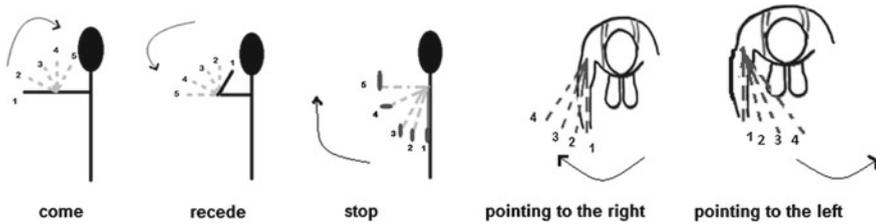


Fig. 2 Five distinct gesture kind

by DTW to a fixed threshold. The evaluation experiments show the effectiveness of the proposed technique. The performance of our technique is further demonstrated with the validation step which yielded good recognition even without training phase.

The rest of the paper is organized as follows: Sect. 2 describes our 3D dynamic gesture approach and the features we used. Section 3 gives some experimental results. Finally, Sect. 4 ends the paper with a conclusion and future work.

## 2 Proposed Approach

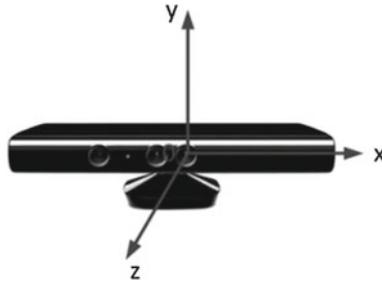
In the context of human robot interaction, the aim of our work is to recognize five 3D dynamic gestures based on depth information. We are interested in deictic gestures. The five gestures we want to recognize are: {*come*, *recede*, *stop*, *pointing to the right* and *pointing to the left*}. Figure 2 shows the execution of each gesture to be recognized. Our gesture recognition approach consists of two main parts: (1) Human tracking and data extraction, and (2) gesture classification.

### 2.1 Human Tracking and Data Extraction

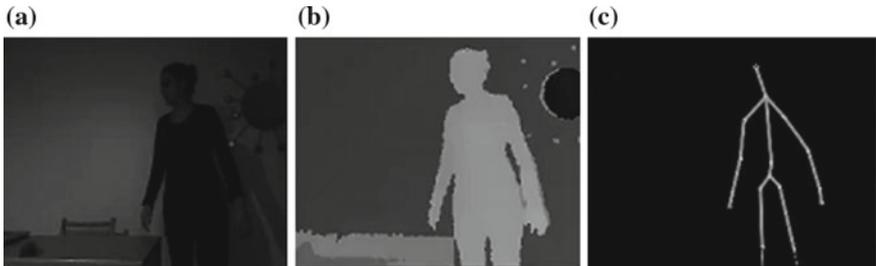
In order to proceed to the gesture recognition, we need first to achieve a robust tracking for Human body and arms. Most recent tracking methods use color information. However, color is not a stable cue, and is generally influenced by several factors such as brightness changing and occlusions. Hence, color-based tracking approaches fail often and don't success to provide 3D human postures at several times. In our work we choose to use a depth sensor (Kinect) in order to extract 3d reliable data. Figure 3 shows the reference coordinate frames associated to the acquisition system.

The coordinates  $x$ ,  $y$  and  $z$  denote, respectively, the  $x$  and  $y$  positions and the depth value. Human tracking is performed using the Skeletal Tracking method given by the kinect SDK.<sup>1</sup> This method projects a skeleton on the human body image so each joint

<sup>1</sup><http://msdn.microsoft.com/en-us/library/jj131025.aspx>.



**Fig. 3** Kinect system coordinate



**Fig. 4** a RGB image, b depth image, c skeleton tracking

of the body is related to a joint of the projected skeleton. In this manner, it creates a collection of 20 joints to each detected person. Figure 4 shows the information used in our approach: depth image (b) and skeleton tracking (c).

The idea is to estimate in real time the variations of the active angles while executing the gestures. The considered angles are:  $\alpha$  elbow,  $\beta$  shoulder and  $\gamma$  armpit angle, as shown in Fig. 5. Each angle is then computed from the 3D coordinates of the three joints that are commonly accounted to it:

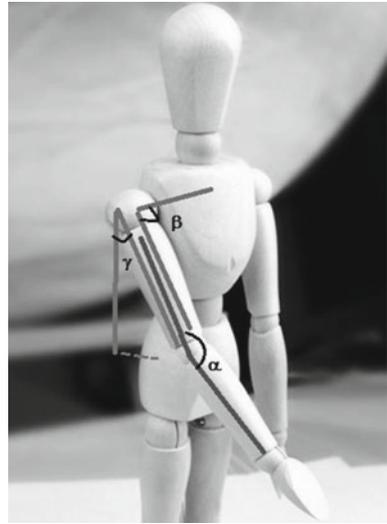
- $\alpha$  elbow angle is computed from the 3D coordinates of elbow, wrist and shoulder joints.
- $\beta$  shoulder angle is computed from the 3D coordinates of shoulder, elbow and shoulder center joints.
- $\gamma$  armpit angle is computed from the 3D coordinates of shoulder, elbow and hip joints.

When performing a gesture we record the values given by each of these three angles and we store the results in vectors as follow:

$$V_{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_T] \tag{1}$$

$$V_{\beta} = [\beta_1, \beta_2, \dots, \beta_T] \tag{2}$$

**Fig. 5**  $\alpha$ ,  $\beta$  and  $\gamma$  angles



$$V_\gamma = [\gamma_1, \gamma_2, \dots, \gamma_T] \tag{3}$$

where  $T$  is the length of the gesture sequence, it is variable from a gesture to another and from a person to another. The input vector of our 3D dynamic gesture recognition system will be then written as:

$$V_\alpha = [\alpha_1, \alpha_2, \dots, \alpha_T, \beta_1, \beta_2, \dots, \beta_T, \gamma_1, \gamma_2, \dots, \gamma_T] \tag{4}$$

The gesture description based on angles variation allows distinguishing between different human gestures. Thus, for every canonical gesture, there is one main angle which changes throughout the gesture and the remaining two angles vary slightly. We consider the five gestures defined previously. The angle which is varying for *come* and *recede* is the angle  $\alpha$ . Likewise, the angle  $\gamma$  for *stop* gesture, and angle  $\beta$  for both pointing gestures. The main angle's variations in each gesture are showing in the Table 1.

**Table 1** The main angle's variations in each gesture

	$\alpha$	$\beta$	$\gamma$
Come	180°–30°	–	–
Recede	30°–180°	–	–
Pointing to right	–	90°–150°	–
Pointing to left	–	90°–40°	–
Stop	–	–	30°–80°

In this work, we propose to use the sequences of angles variations as an input of our gesture recognition system as explained in the next section.

## 2.2 Gesture Classification Method

Our recognition method is based on a combination of Hidden Markov Models (HMM) and Dynamic Time Warping (DTW) method. HMM are widely used in temporal pattern, speech, and handwriting recognition, they generally yield good results. The problem in the dynamic gestures is their spatial and temporal variability which make their recognition very difficult, compared to the static gestures. In fact, the same gesture can vary in speed, shape, length. However, HMM have the ability to maintain the identity of spatio-temporal gesture even if its speed and/or duration change. Since we work with time series data, we use Dynamic Time Warping algorithm to measure similarity between two sequences that may vary in time and speed. DTW warps the sequences and gives a distance like quantity between them.

In the first stage, we classify the gesture using HMM [27]. Based on the best probability of belonging to one of the five classes, the gesture kind is recognized. In the second stage, we measure the similarity between the variations of the main angle sequence that characterizes the gesture class which HMM gave as output and another variations sequence of the same angle taken as a reference using DTW. Next, the distance is compared to a precalculated threshold. If the distance is less than the threshold we keep the result provided by HMM method else the gesture will be considered as an unknown gesture then rejected. Therefore a bad performed gesture will be rejected instead of being misclassified. Figure 6 shows the steps of our recognition system. First, HMM method will classify a given gesture ( $G_{test}$ ) into one of the five classes. Then, HMM method will give the result which is the type of gesture (for example: *Come*). As mentioned before, the angle which characterizes the gesture *Come* is elbow angle designed by  $\alpha$ . Thus, we take the first part of the gesture sequence ( $G_{test}$ ) which corresponds to  $\alpha$  angle variations, and we take a reference sequence of  $\alpha$  angle variations in *Come* gesture from the database. Next, we calculate the distance between these two sequences using DTW method. The resulting distance is compared to a threshold that was fixed for the gesture *Come*.

**Hidden Markov Models.** An HMM can be expressed as  $\lambda = (A, B, \pi)$  and described by:

- (a) A set of  $N$  states  $S = \{s_1, s_2, \dots, s_n\}$ .
- (b) An initial probability distribution for each state  $\Pi = \{\pi_j\}$ ,  $j = \{1, 2, \dots, N\}$ , with  $\pi_j = \text{Prob}(S_j \text{ at } t = 1)$ .
- (c) A  $N$ -by- $N$  transition matrix  $A = \{a_{ij}\}$ , where  $a_{ij}$  is the transition probability of  $s_i$  to  $s_j$ ;  $1 \leq i, j \leq N$  and the sum of the entries in each row of the matrix  $A$  must be equal to 1 because it corresponds to the sum of the probabilities of making a transition from a given state to each of the other states.

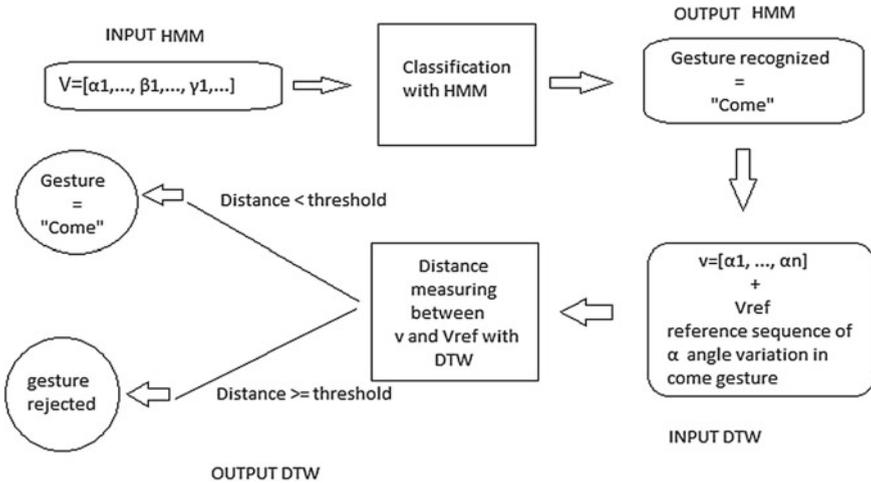
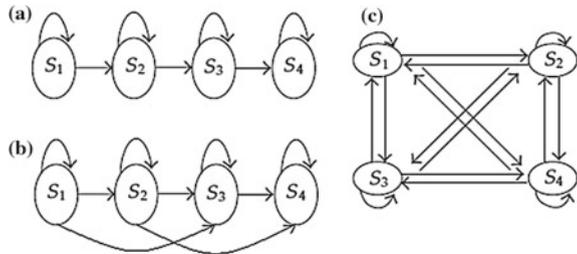


Fig. 6 Our recognition system combining HMM and DTW

- (d) A set of observations  $O = \{o_1, o_2, \dots, o_t\}$ ,  $t = \{1, 2, \dots, T\}$  where  $T$  is the length of the longest gesture path.
- (e) A set of  $k$  discrete symbols  $V = \{v_1, v_2, \dots, v_k\}$ .
- (f) The  $N$ -by- $M$  observation matrix  $B = \{b_{im}\}$ , where  $b_{im}$  is the probability of generating the symbol  $v_k$  from state  $s_j$  and the sum of the entries in each row of the matrix  $B$  must be 1 for the same previous reason.

There are three main problems for HMM: evaluation, decoding, and training, which are solved by using Forward algorithm, Viterbi algorithm, and Baum-Welch algorithm, respectively [28]. Also, HMM has three topologies: Fully Connected (Ergodic model) where each state can be reached from any other state, Left-Right (LR) model where each state can go back to itself or to the following states and Left-Right Banded (LRB) model in which each state can go back to itself or the following state only (Fig. 7). We choose left-right banded model Fig. 7a as the HMM topology, because the left-right banded model is good for modeling-order-constrained time-

Fig. 7 HMM topologies. **a** Left-right banded topology, **b** left-right topology, **c** ergodic topology



series whose properties sequentially change over time. We realized five HMM, one HMM for each gesture type.

**Initializing Parameters for LRB Model.** We created five HMM, one for each gesture. First of all, every parameter of each HMM should be initialized. We start with the number of states. In our case this number is not the same for all the five HMM, it depends on the complexity and duration of the gesture. We use 12 states as maximum number and 8 as minimum one in which the HMM initial vector parameters  $\Pi$  will be designed by;

$$\Pi = (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0) \tag{5}$$

To ensure that the HMM begins from the first state, the first element of the vector must be 1. The second parameter to be defined is the Matrix  $A$  which can be written as:

$$A = \begin{matrix} & a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & a_{ii} & 1-a_{ii} & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & a_{ii} & 1-a_{ii} & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & a_{ii} & 1-a_{ii} & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & a_{ii} & 1-a_{ii} & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & a_{ii} & 1-a_{ii} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{ii} \end{matrix} \tag{6}$$

where  $a_{ii}$  is initialized by a random value. The Matrix  $B$  is determined by:

$$B = \{b_{im}\} \tag{7}$$

where  $b_{im}$  is initialized by a random value.

**Training and Evaluation.** Our database is composed of 100 videos for each kind gesture (50 for training and 50 for testing). In the training phase the Baum-Welch algorithm [28] is used to do a full training for the initialized HMM parameters  $\lambda = (\Pi, A, B)$ . Our system is trained on 50 sequences of discrete vector for each kind of gesture by using LRB topology with the number of states ranging from 3 to 12. After the training process, we obtain new HMM parameters  $(\Pi', A', B')$  for each type of gesture. According to the forward algorithm with Viterbi path, the other 50 video sequences for each type of gesture are tested using the new parameters. The forward algorithm computes the probability of the discrete vector sequences for all the five HMM models with different states. Thereby, the gesture path is recognized corresponding to the maximal likelihood of 5 gesture HMM models over the best path that is determined by Viterbi algorithm. The following steps demonstrate how the Viterbi algorithm works on LRB topology [29]:

- Initialization:
  - for  $1 \leq i \leq N$ ,
  - $\delta_1(i) = \Pi_i \cdot b_i(o_1)$
  - $\phi_1(i) = 0$
- Recursion:
  - for  $2 \leq t \leq T, 1 \leq j \leq N$ ,

$$\delta_t(i) = \max[\delta_{t-1}(i) \cdot a_{ij}] \cdot b_j(o_t)$$

$$\phi_t(i) = \operatorname{argmax}[\delta_{t-1}(i) \cdot a_{ij}]$$

- Termination:

$$p^* = \max[\delta_T(i)]$$

$$q_T^* = \operatorname{argmax}[\delta_T(i)]$$

- Reconstruction:

$$\text{for } T - 1 \leq t \leq 1$$

$$q_t^* = \phi_{t+1}(q_{t+1}^*)$$

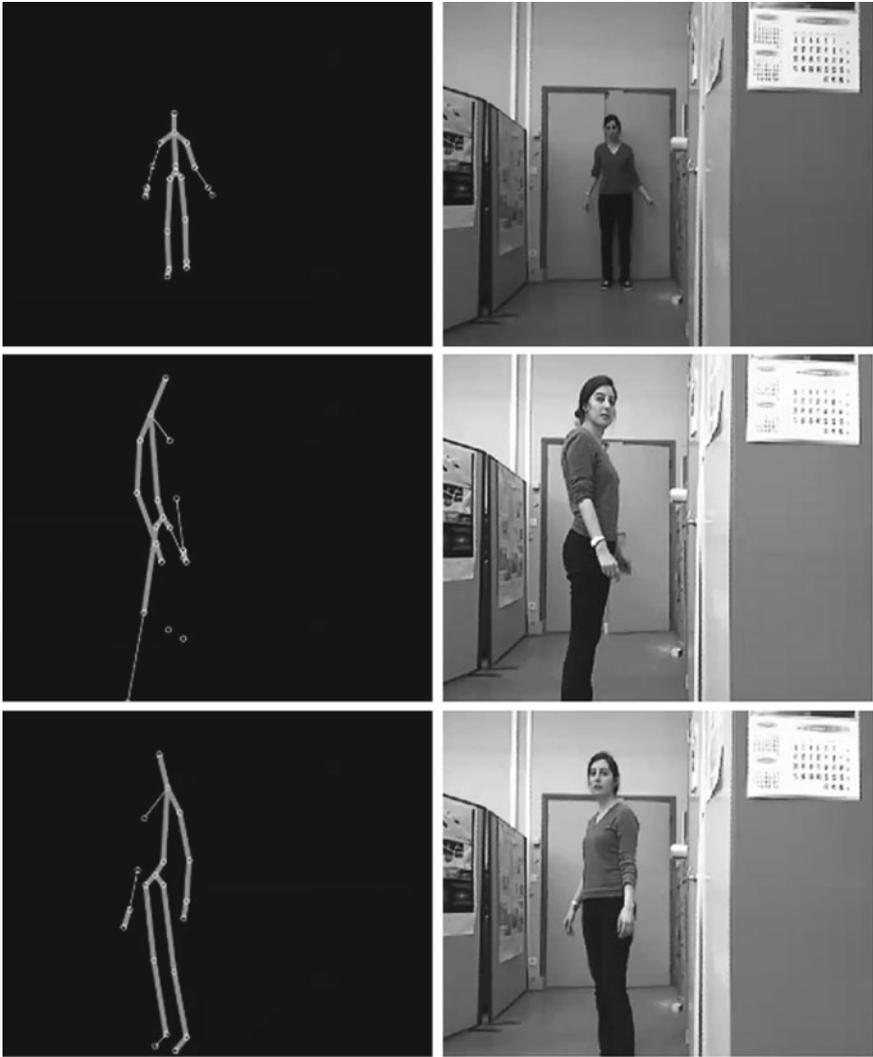
The resulting trajectory (optimal states sequence) is  $q_1^*, q_2^*, \dots, q_T^*$  where  $a_{ij}$  is the transition probability from state  $s_i$  to state  $s_j$ ,  $b_j(o_t)$  is the probability of emitting  $o$  at time  $t$  in state  $s_j$ ,  $\delta_t(j)$  represents the maximum value of  $s_j$  at time  $t$ ,  $\phi_t(j)$  is the index of  $s_j$  at time  $t$  and  $p^*$  is the state optimized likelihood function.

**Calculating the Threshold for DTW Distance.** We have calculated empirically five threshold values one for each class of gesture. First we consider for each gesture its own reference sequence. For *Come* class, the reference sequence contains the variations of  $\alpha$  angle throughout a *Come* gesture. For *Recede* class, the reference sequence contains the variations of  $\alpha$  angle throughout a *Recede* gesture. For *Pointing to the right* class, the reference sequence contains the variations of  $\beta$  angle throughout a *Pointing to the right* gesture. For *Pointing to the left* class, the reference sequence contains the variations of  $\beta$  angle throughout a *Pointing to the left* gesture. And for *Stop* class, the reference sequence contains the variations of  $\gamma$  angle throughout a *Stop* gesture. The threshold of a gesture class corresponds to the maximum distance between its appropriate reference sequence and 50 sequences of test. The distance is given by DTW algorithm and the sequences of test are extracted from the training database.

## 3 Experimental Results

### 3.1 Experimental Protocol

Before the experiment, the experimental protocol was given to the subjects which describes the beginning and the end of the five gestures. The gesture duration is not fixed. The person can do a gesture whether slowly or speedy. We used the Kinect sensor that must remain stable. The distance between the kinect and the person should be between 80 cm and 3 m in order to detect the person properly. Figure 8 shows some cases when the Kinect can not totally detect the body. The environment is more or less crowded with no obstacles between the subject and the Kinect. While performing a gesture, the person should be standing and remains in front of the kinect.

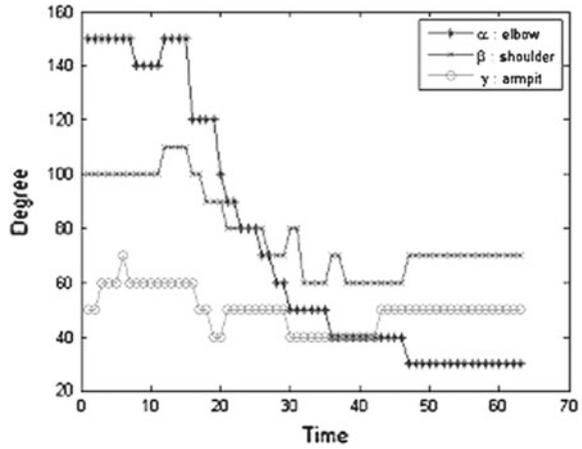


**Fig. 8** The cases of detection failure by the Kinect. The first image: the distance is greater than 3 m. The second and third images: the person is not in front of the Kinect

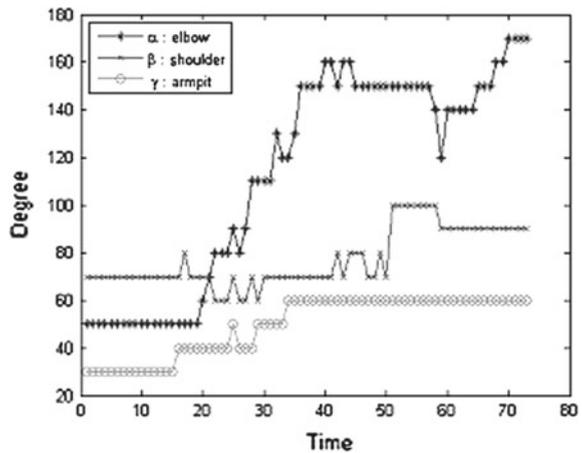
### 3.2 Recognition Results

Angles variations are plotted in Figs. 9, 10, 11, 12 and 13. As it is shown, each gesture is characterized by the most changing angle comparing to the two others. We choose the state number of HMM for each gesture according to the experiment results and find that the recognition rate is maximum when the state number is 11 states for the

**Fig. 9** Angles variations for *come* gesture



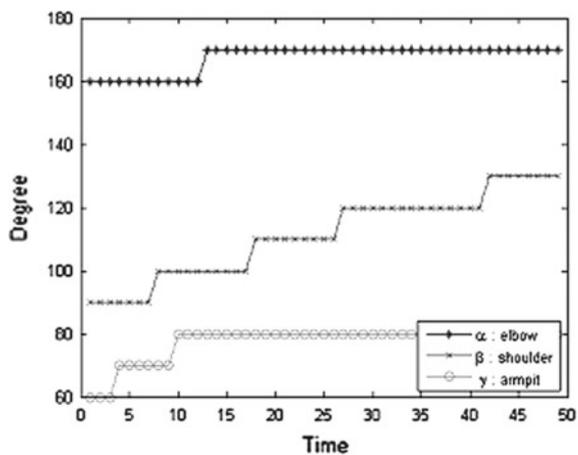
**Fig. 10** Angles variations for *recede* gesture



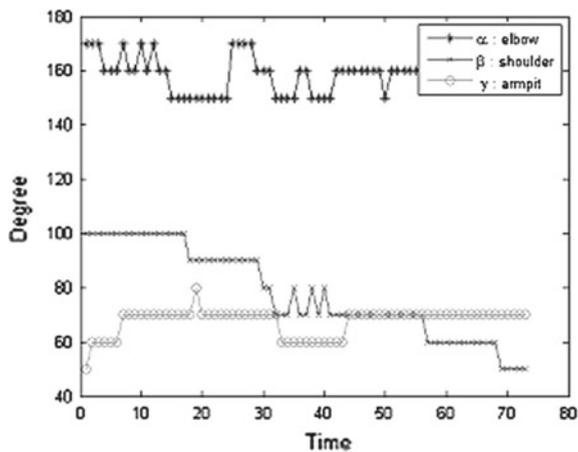
gestures *come*, *recede* and *pointing to the right*, 12 for the gesture *pointing to the left*, and 8 for the last gesture *stop* as shown in Fig. 14. Therefore, we use this setting in the following experiments. A given gesture sequence is recognized in 0.1508 s. The recognition results are listed in Table 2. We can see that the proposed method can greatly improve the recognition process, especially for opposed gestures like *come* and *recede*, *pointing to the right* and *pointing to the left*. We can also see that there is no confusing between some gestures such as *come* and *recede*. In this case, it is due to the fact that the angle  $\alpha$  changes during these two gestures decreases in *come* and increases in *recede*.

The same reasoning can be given in the case of the tow opposed gestures, *pointing to the right* and *pointing to the left*. As a matter of fact, even if the same angle varies in two different gestures, our method can distinguish them.

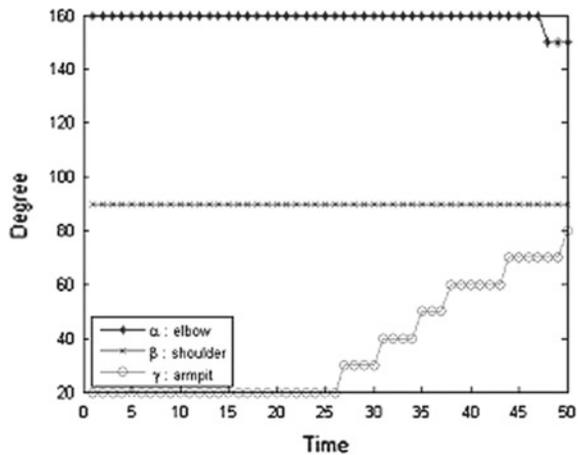
**Fig. 11** Angles variations for *pointing to the right* gesture



**Fig. 12** Angles variations for *pointing to the left* gesture



**Fig. 13** Angles variations for *stop* gesture



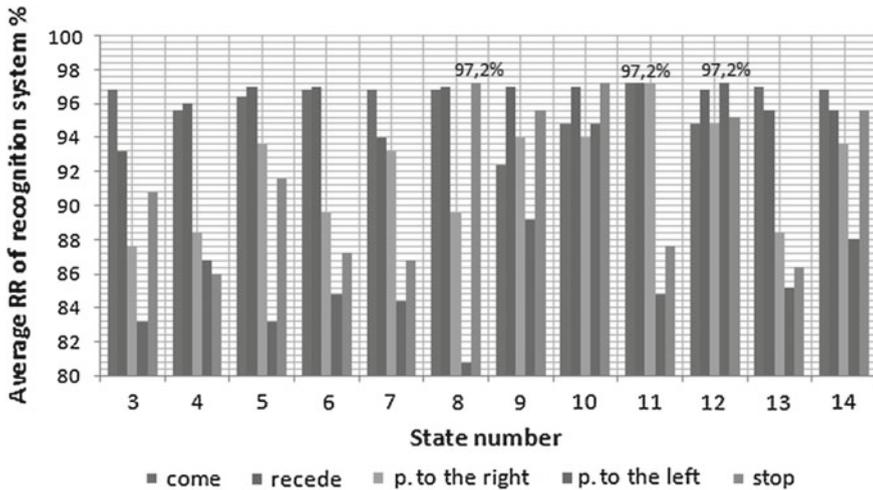


Fig. 14 Recognition accuracy when changing the number of state of HMM from 3 to 14 states

Table 2 Confusing matrix and recognition accuracy

	Come	Recede	P. to the right	P. to the left	Stop	Unknown gesture	Accuracy (%)
Come	50	0	0	0	0	0	100
Recede	0	50	0	0	0	0	100
P. to the right	0	0	49	0	0	1	98
P. to the left	0	0	0	48	0	2	96
Stop	0	0	0	0	46	4	92

Average accuracy 97.2%

Table 3 presents a comparison of our approach with that of the authors in [30]. They use raw, roll and pitch orientations of *elbow* and *shoulder* joints of the left arm. Their database contains five gestures trained by one person and tested by two. The gesture duration is fixed beforehand. In offline mode, the accuracy of recognizing gestures executed by persons who did training was found to be 85 % with their method and 97.2% with our method. And without training, the recognition accuracy attained 73% with their method and 82% with our method. The gestures we have defined for the human robot interaction are natural. They are almost the same that we use daily and between people. Whereas, most methods in the state of the art are based on constrained gestures that use signs which are not natural. The proposed gesture recognition approach is based only on depth information that is what makes it very robust against the environment complexity and illumination variation.

**Table 3** Comparison between the performance of our approach and Ye and Ha [30]’s approach

Methods	Ye and Ha [30]	Our approach
Gesture nature	Dynamic	Dynamic
Used Info.	Raw, roll and pitch orientations	Angles between joints
Gestures number	5	5
Joints number	2	5
Used data	Segmented	Brute
Classification	HMM	HMM
Database	75	500
People for test	2	21
Gesture duration	Fixed	Variable
Accuracy	73 %	97.2 %

## 4 Conclusion and Future Work

In this paper we have presented an efficient method for 3D natural and dynamic gesture recognition intended for human robot interaction. The proposed gesture recognition system is able to recognize five deictic gestures described by depth information only. The upper body part is tracked using Kinect camera and angles are computing from the 3d coordinates of five different joints. Our five gesture are represented by a sequence which combines the variations of three angles. This sequence is the input of our classification system that combines HMM and DTW method. First, HMM affects the given gesture to one of five classes corresponding to the maximum probability. Based on this result, the DTW measures the similarity between the variations sequence of the main angle that characterizes the gesture class which HMM method gave as output and its reference sequence. The output distance is compared to the threshold corresponding to the same class; if the distance is less than the threshold then we keep the HMM result else we reject the gesture. Experimental results presented in this paper, confirm the effectiveness and the efficiency of the proposed approach. In one hand, the recognition rate can reach up to 100 % for some kind of gestures. The combination of HMM and DTW avoid misclassification and reject bad performed gestures. In the second hand, the system can recognize gestures even if the distance or the location of people change knowing that some conditions should be respected as given in the experimental protocol. Finally, The environment and the brightness do not affect the data collection and analyzing because we rely on depth only.

Nevertheless, and despite the vast amount of relevant research efforts, the problem of efficient and robust vision based recognition of natural gestures in unprepared environments still remains open and challenging, and is expected to remain of central importance in human-robot interaction in the forthcoming years. In this context we intend to continue our research efforts towards enhancing the current system. At first, the training and test data-sets will be expanded to include richer gesture types in order

to recognize different gestures in the same sequence. Then, we intend to introduce other information such as speech to improve the proposed recognition system by detecting the beginning and the end of the gesture.

## References

1. Zhang, Z.: Microsoft kinect sensor its effect. *Multi Media* (2012)
2. Rautaray, S., Agrawal, A.: A real time hand tracking system for interactive applications. *Int. J. Comput. Appl.* (2011)
3. Dan, X., Chen, Y.-L., Wu, X., Xu, Y.: Integrated approach of skincolor detection depth information for hand and face localization. In: *IEEE International Conference on Robotics Biomimetics-ROBIO* (2011)
4. Ghobadi, S.E., Leopprich, O.E., Hartmann, K., Loffeld, O.: Hand segmentation using 2d/3d images. In: *Proceeding of image Vision Computationg* (2007)
5. Barczak, A., Dadgostar, F.: Real-time hand tracking using a set of cooperative classifiers based on haar-like features. *Res. Lett. Inf. Math. Sci.* (2005)
6. Xu, J., Wu, Y., Katsaggelos, A.: Part-based initialization for h tracking. In: *The 17th IEEE International Conference on Image Processing (ICIP)* (2010)
7. Chen, Q., Georganas, N., Petriu, E.: H gesture recognition using haar-like features a stochastic context-free grammar. *IEEE Trans. Instrum. Meas.* (2008)
8. Bleiweiss, A., Werman, M.I.: Fusion time-of-flight depth and color for real-time segmentation and tracking. In: *DAGM Symposium for Pattern Recognition* (2009)
9. Xu, D., Wu, X., Chen, Y., Xu, Y.: Online dynamic gesture recognition for human robot interaction. *IEEE J. Intell. Robot. Syst.* (2014)
10. Lu, X., Chen, C.-C., Aggarwal, J.K.: Human detection using depth information by kinect. In: *Computer Society Conference on Computer Vision Pattern Recognition-CVPR* (2011)
11. Qin, S., Zhu, X., Yang, Y., Jiang, Y.: Real-time hand gesture recognition from depth images using convex shape decomposition method. *J. Signal Process. Syst.* (2014)
12. Matuszek, C., Bo, L., Zettlemoyer, L., Fox, D.: Learning from unscripted deictic gesture language for human-robot interactions. *I. J. Robotic* (2014)
13. Boesen, A., Larsen, L., Hauberg, S., Pedersen, K.S.: Unscented Kalman filtering for articulated human tracking. In: *17th Scinavian Conference SCIA* (2011)
14. Ababsa, F.: Robust extended Kalman filtering for camera pose tracking using 2D to 3D lines correspondences. In: *International Conference on Advanced Intelligent Mechatronics* (2009)
15. Ababsa, F., Mallem, M.: Robust line tracking using a particle filter for camera pose estimation. In: *Proceedings of the ACM Symposium on Virtual Reality Software Technology* (2006)
16. Koller, D., Thrun, S., PlagemannVarun, C., Ganapathi, V.: Real time identification localization of body parts from depth images. In: *IEEE International Conference on Robotics Automation (ICRA)* (2010)
17. Wang, X., Xia, M., Cai, H., Gao, Y., Cattani, C.: Hidden Markov models based dynamic hand gesture recognition. *Math. Probl. Eng.* (2012)
18. Saon, G., Chien, J.T.: Bayesian sensing hidden Markov models. *IEEE Trans. Audio Speech Lang. Process.* (2012)
19. Li, M., Cattani, C., Chen, S.Y.: Viewing sea level by a one-dimensional random function with long memory. *Math. Probl. Eng.* (2011)
20. Elmezain, M., Al-Hamadi, A., Michaelis, B.: Real-time capable system for hand gesture recognition using hidden Markov models in stereo color image sequences. *J. WSCG* (2008)
21. Binh, N.D., Ejima, T.: Real-time hand gesture recognition using pseudo 3-d hidden Markov model. In: *Proceedings of the 5th IEEE International Conference on Cognitive Informatics (ICCI'06)* (2002)

22. Eickeler, S., Kosmala, A., Rigoll, G.: Hidden Markov model based continuous online gesture recognition. In: Proceedings of 14th International Conference on Pattern Recognition (1998)
23. Gu, Y., Do, H., Ou, Y., Sheng, W.: Human gesture recognition through a kinect sensor. In: International Conference on Robotics Biomimetics (2012)
24. Bengio, Y., Frasconi, P.: Input-output HMMs for sequence processing. *IEEE Trans. Neural Netw.* (1996)
25. Wang, S., Quattoni, A., Morency, L., Demirdjian, D., Darrell, T.: Hidden conditional rom fields for gesture recognition. In: IEEE Computer Society Conference on Computer Vision Pattern Recognition (CVPR) (2006)
26. Corradini, A.: Dynamic time warping for off-line recognition of a small gesture vocabulary. In: ICCV Workshop on Recognition Analysis Tracking of Faces Gestures in Real-Time Systems (2001)
27. Hiyadi, H., Ababsa, F., Montagne, Ch., Bouyakhf, E.H., Regragui, F.: A depth-based approach for 3D dynamic gesture recognition. *ICINCO* 103-110 (2015)
28. Lawrence, R.: A tutorial on hidden Markov models selected applications in speech recognition. *Proc. IEEE* (1989)
29. Elmezain, M., Al-Hamadi, A., Appenrodt, J., Michaelis, B.: A hidden Markov model-based isolated meaningful hand gesture recognition. *J. WSCG* (2009)
30. Ye, G., Ha, D., Yongsheng, O., Weihua, S.: Human gesture recognition through a kinect sensor. *Robotics Biomimetics (ROBIO)* (2012)

# Model-Free (Human) Tracking Based on Ground Truth with Time Delay: A 3D Camera Based Approach for Minimizing Tracking Latency and Increasing Tracking Quality

Philip Nicolai, Jörg Raczkowski and Heinz Wörn

**Abstract** Model-free tracking allows tracking of objects without prior knowledge of their characteristics. However, many algorithms require a manual initialization to select the target object(s) and perform only a coarse tracking. This article presents a new hybrid approach that allows combining a new fast, model-free tracking algorithm using 3D cameras with an arbitrary separate, slower tracking method that provides a time-delayed ground truth. In particular, we focus on human tracking human and employ Time-of-Flight cameras for the model-free tracking, based on ground truth provided by (multiple) Kinect cameras. The article describes the setup of the system, the model-free tracking algorithm and presents evaluation results for two different scenarios. Results show a high precision and recall, even with large time-delays of the ground truth of up to 10 s.

**Keywords** 3D camera · Model-free tracking · Probability propagation

## 1 Introduction

The interaction of humans and robots in a shared workspace is an ongoing field of research. Applications cover a wide field from domains where robotic technologies have been traditionally employed, e.g. industrial scenarios, to relatively new fields such as surgical applications. In all domains, the safety of the human interacting with the robotic system is paramount. For appropriate safety considerations as well as for many applications in human-robot interaction, humans in the environment

---

P. Nicolai (✉) · J. Raczkowski · H. Wörn  
Institute for Anthropomatics and Robotics (IAR), Intelligent Process Control  
and Robotics Lab (IPR), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany  
e-mail: philip.nicolai@kit.edu

J. Raczkowski  
e-mail: joerg.raczkowski@kit.edu

H. Wörn  
e-mail: heinz.woern@kit.edu

have to be perceived, e.g. detected and located in 3D space. Both the latency of the perception system and the frame rate heavily influence the possible application scenarios, especially for safety critical applications.

This paper, which is an extension of [1], presents an approach for combining a model-free tracking algorithm using a fast 3D camera, i.e. with low latency and/or high frame rate, with a ground truth provided by a secondary, slower 3D camera. The faster camera pre-calculates the full point cloud of the tracked target in real time. To achieve this, the time-delayed ground truth of the slower camera is propagated forward in the data stream of the faster camera using 2D optical flow and then refined to segment the full point cloud from the scene. Segmentation is performed by calculating connected regions, rejecting outliers based on a simple tracking model and applying background subtraction. This results in a highly accurate tracking estimation in time with the faster camera, based on the time-delayed ground truth of the secondary system. In addition, the model-free tracking algorithm will continue tracking even if there is no ground truth available for a period of time, e.g. because the tracking algorithm that provides the ground truth has lost the target(s).

While the approach is implemented and evaluated on the scenario of human tracking, i.e. using human tracking by a 3D camera as ground truth, the algorithm is not tailored to this application (either implicitly or explicitly). On the contrary, as a model-free tracking algorithm it is designed with the goal to be adaptable to other applications, i.e. different combinations of tracking tasks and modalities. Examples might be 3D tracking of objects with specific temperatures by using a thermal imaging camera as delayed ground truth or even 3D tracking of objects based on positional audio information provided by a microphone array.

The developed algorithm is applied to three different scenarios for tracking a human body as a point cloud in 3D space: (a) low-latency tracking based on ground truth with a latency of one to several seconds, (b) high frame rate tracking based on ground truth with a lower frame rate, (c) continuous tracking based on non-continuous ground truth.

Optical flow and depth information have been used in various works for segmenting and tracking humans and objects. Examples are [2], where depth and optical flow were used to estimate the 3D position and motion of a person; using optical flow to track persons between multiple cameras to avoid occlusions [3] or applying optical flow and depth cues to vehicle-based moving object segmentation [4]. The combination of 2D and 3D Kinect data has been researched e.g. in [5] with the purpose of mobile mapping in indoor environments.

However, the combination of different 3D cameras with 2D/3D propagation of tracking probabilities has not been investigated before.

## 2 System Setup

The Robot Operating System has been used as a communication framework [6]. It is based on sending time-stamped messages on named topics and provides transport mechanisms for both 2D and 3D image data.

Processing of data acquired by the different 3D cameras has been performed using OpenCV [7] for 2D images and Point Cloud Library [8] for 3D data.

In the following, we give a short description of the scenarios and camera systems to which the algorithm has been applied.

### 2.1 Latency Minimization Scenario

The first scenario is based on the sensing system of OP: Sense, a research platform for surgical robotics [9]. Four RGB-D Microsoft Kinect cameras (first generation), featuring a resolution of  $640 \times 480$  pixel for both depth and color image at 30 frames/s (fps), supervise a narrow scene from different points of view. Human tracking and fusion is performed based on the OpenNI tracking libraries [10]. Due to the distributed setup of the system, the Kinect system features a latency of about 950 ms.

A secondary camera system consists of six Time of Flight (ToF) pmd[vision] S3 cameras. With a resolution of  $64 \times 48$  pixels, they provide depth sensing (e.g. point clouds and depth image) as well as an amplitude image that contains the signal strength of the measurement. The left side of Fig. 1 shows one sensor node with both Kinect and ToF camera.



**Fig. 1** *Left* Sensor node with Kinect (*top*), ToF camera (*bottom center*); the optical tracking (*bottom right*) was not used for this work. *Right* Argos P100 3D mounted on top of Kinect II

## 2.2 *Frame Rate Optimization Scenario*

In this scenario, a RGB-D Microsoft Kinect II has been used for human tracking. The camera provides a color stream with  $1920 \times 1080$  pixels and a depth data stream with  $512 \times 424$  pixels, both at 30 fps. Human tracking was performed using the Microsoft Kinect SDK 2.0 on a Windows system and streaming to ROS has been realized using a custom bridge based on the win\_ros stack.

A Bluetechnix Argos 3D P100 ToF camera with a resolution of  $160 \times 120$  pixels provides depth data and an amplitude image, both at a rate of up to 160 fps. The right side of Fig. 1 shows the demonstration setup.

## 2.3 *Tracking Reconstruction Scenario*

This scenario is based on the same setup as scenario 2.1, i.e. four Kinect cameras provide the ground truth and six pmd[vision] S3 cameras are used for running the pre-calculation.

# 3 **Methods**

For easier reading and consistency with the scenarios and evaluation, we designate the source of the ground truth in the following as “Kinect camera” and the secondary camera as “ToF camera”. However, the presented algorithm is naturally applicable to a wide range of different cameras.

Similarly, the tracking application, which will be referred to throughout the article, is the tracking of humans (based on ground truth provided by the Kinect camera). As the presented approach is a model-free algorithm which is deliberately based on processing an external ground truth (opposed to implementing custom detection and/or tracking algorithms), applications to arbitrary different tracking scenarios are possible. In general, the only requirement is that an external ground truth is available and that correspondences can be established between ground truth and data acquired by the secondary camera.

## 3.1 *Processing Pipelines*

The proposed algorithm consists of two different processing pipelines which are executed in parallel as depicted in Fig. 2. The first one processes all data acquired by the ToF camera (data which doesn't contain any tracking information) and propagates tracking information based on the delayed ground truth. Thereby, a tracking estimate

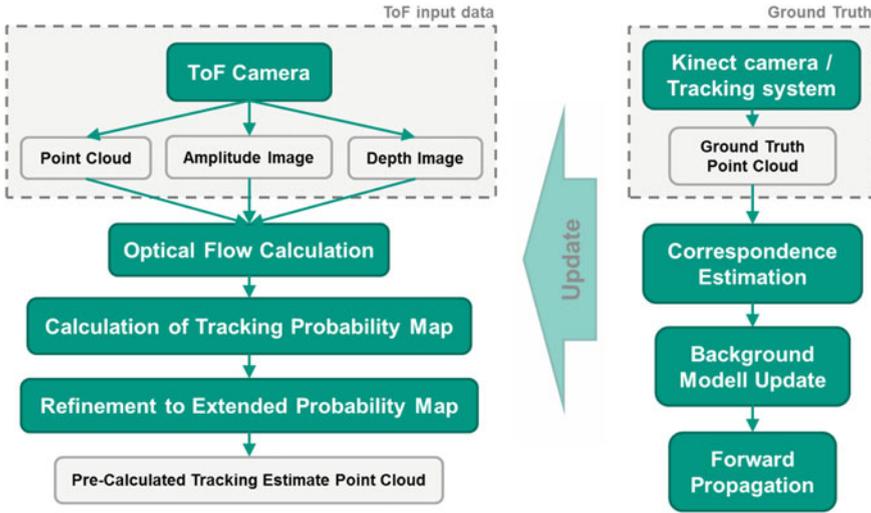
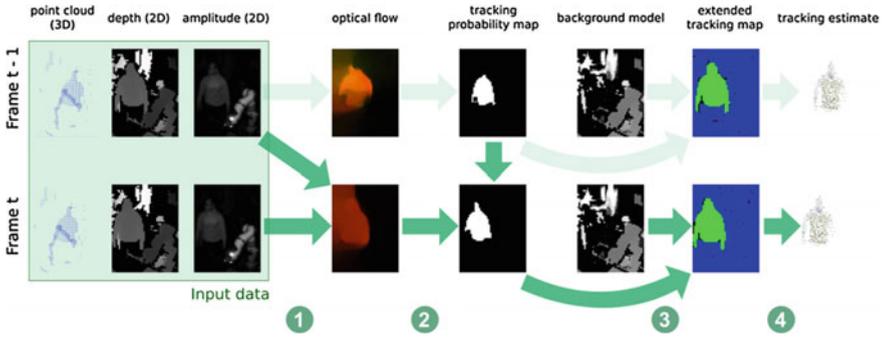


Fig. 2 Overview of the high-level structure of the proposed algorithm. Two pipelines (left ToF pre-calculation, right ground truth processing) are executed in parallel such that the pre-calculation pipeline based on ToF data is continuously updated with the latest available ground truth

is provided in each time step. The second one processes the user tracking information from the Kinect camera (ground truth) and updates the ToF tracking state as well as the background model.

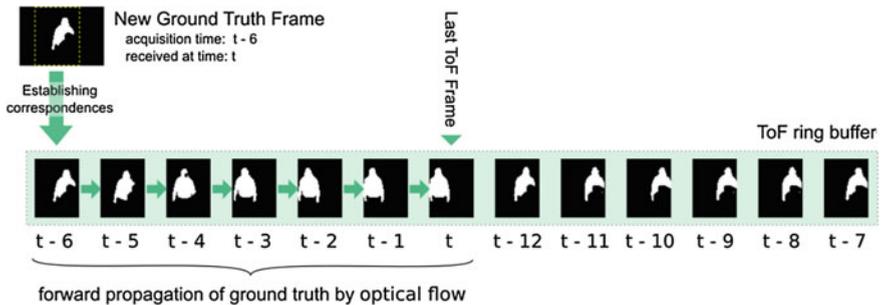
**ToF Processing.** In the following, we use the term “ToF frame” to refer to all ToF data associated to a single time step: source data such as the 3D point cloud, the amplitude image, the depth image and the time stamp of the data acquisition as well as processed data such as a flow field, a tracking probability map and geometric information about tracked targets. To enable applying the results of filtering in the 2D image domain to the 3D space of the point cloud, the pixel-to-point correspondences have to be preserved. For this reason, only operations are employed on the ToF point clouds that keep them organized, i.e. that don’t alter the original points in the cloud.

Figure 3 visualizes the data processing of incoming ToF frames: Upon receiving a new ToF frame, the point cloud is transformed into a shared coordinate system and 2D optical flow from the previous frame is calculated based on the respective amplitude images (see Sect. 3.3). The ToF frame is then stored in a ring buffer. A tracking probability map is calculated that provides a first estimation of the current position of the tracked target(s), based on the optical flow and the tracking probability map stored in the previous ToF frame. Last, a refinement and rejection step is performed based on the tracking probability map, the background model and the spatial information encoded in the depth map (see Sect. 3.5). This yields the extended tracking map for the current time step which is then applied to the point cloud to calculate the human body point cloud tracking estimate.



**Fig. 3** Processing pipeline for newly acquired ToF frame at time  $t$ . 1 A flow field is calculated based on amplitude images of frame  $t-1$  and frame  $t$ . 2 The flow field is applied to the tracking probability map of frame  $t-1$ , resulting in a tracking probability map for frame  $t$ . 3 The tracking probability map is processed based on the tracking information of frame  $t-1$  and a global background model to provide an extended tracking map. 4 Applying the extended tracking map to the point cloud results in the final tracking estimate

**Ground Truth Processing.** Upon reception, the point cloud corresponding to the tracked human(s) is transformed to the shared coordinate system. Based on the acquisition time of the received point cloud, the closest matching ToF frame is located in the ring buffer (see Fig. 4). By determining correspondences between the ground truth and the point cloud stored in the ToF frame, a tracking probability map with full certainties is established and the ToF frame is marked as a key frame. The background model is updated using this tracking probability map and the corresponding depth



**Fig. 4** Processing pipeline of new ground truth data acquired at time  $t-6$  and is received at time  $t$ . First, the corresponding ToF frame in the ring buffer is identified using the associated time stamps. Next, correspondences are estimated to calculate a tracking probability map for the ToF frame at time  $t-6$ . Last, the tracking probability map is propagated forwards using the flow fields associated with each ToF frame

map (see Sect. 3.2). These calculations are performed for each incoming ground truth frame and are therefore independent of the actual delay of the ground truth.

Using the respective flow fields, the tracking probability map is propagated forward throughout the ring buffer until the most recent ToF frame (see Sect. 3.4). Here, the number of forward-propagations is directly proportional to the length of the delay. Thereby, the tracking probability map of the next arriving ToF frame will be calculated based on the updated information from this frame.

## 3.2 Background Modelling

In the presented approach, almost all information is stored and processed on a frame-by-frame basis, e.g. optical flow between two frames and the tracking probability map are directly assigned to a specific ToF frame. There are two exceptions which are modelled as global components: The number of tracked humans and a background model of the scene.

Our approach to modelling the background of the scene is based on the works of [11] that extended the common Gaussian mixture models for pixel-wise background subtraction by an automatic calculation of the correct number of Gaussian distributions per pixel. We have modified the OpenCV implementation of this algorithm in two ways in order to take advantage of the data flow in our approach. First, we introduce a masking capability that enables restricting an update of the background model to specific areas of the image. Second, we split the update step of the original algorithm into two different parts: A background maintenance that only updates the model (without performing background subtraction on the input) and a foreground detection stage that allows performing background subtraction on an image and calculating a foreground mask without updating the background model.

Based on these modifications, the background model is being used as follows:

When a new ground truth frame arrives and correspondences to the according ToF frame have been calculated, the background model is updated using the depth image of this ToF frame. The tracking probability map is used to mask the tracked humans, thereby ensuring that they are not incorporated into the background model. This prevents the common problem that non-moving entities will be included in the background after a certain number of update-steps [12].

When a new ToF frame is processed, an extended tracking map is calculated that contains the location of all pixels belonging to a tracked human. However, this map is prone to inclusion of false positives, e.g. pixels that belong to the background. For correction, a foreground mask is retrieved by querying the background model with the depth image of the ToF frame. By masking the extended tracking map with the foreground mask, we remove potential false positives.

### 3.3 *Optical Flow Estimation*

As described in Sect. 3.1, optical flow applied to 2D images is used to propagate the tracking probability map between the ToF frames.

When using RGB images, the sensitivity of optical flow for moving targets such as humans or objects is highly dependent on the kind of motion performed. When applying optical for the purpose of tracking, rotations prove more difficult to detect than translations: During rotation of a tracked target, previously visible parts of the object vanish from the image while new parts appear. For these new elements, no corresponding parts exist in the previous image. Performing optical flow calculations on the amplitude images acquired by ToF cameras partially overcomes this problem: The reflectivity of a tracked target, especially in the case of human tracking, is usually less affected by rotations than its appearance in color space.

For the actual calculation of optical flow between two amplitude images, we use the TV-L1 algorithm proposed by [13]. The flow field is calculated upon receiving a new ToF frame and stored within the frame. As the flow field based propagation of the tracking probability map is only used as a first approximation which is refined in subsequent steps, our parameterization of the optical flow algorithm is targeted on a higher computation speed rather than an optimal accuracy. Therefore, we set the number of warps to 2 with 3 levels.

### 3.4 *Tracking Probability Propagation*

In ToF frames, information about the location of tracked humans has to be stored and propagated. We represent this information as a 2D probability map where the value of each pixel denotes the probability of this pixel belonging to a tracked human.

When a ground truth frame is received and the ToF frame with the closest matching timestamp was located in the ring buffer, point-to-point correspondences between both frames have to be established. These correspondences are calculated by creating a k-d-tree of the downsampled ground truth cloud, iterating over all points in the ToF point cloud and determining whether the distance to the ground truth cloud is smaller than a pre-defined threshold. For all points where this check is successful, the according pixel in the zero-initialized probability map is set to one.

Propagation of the tracking probability map from ToF frame  $F_t$  to subsequent frame  $F_{t+1}$  is performed using the flow fields associated with each ToF frame: Using the flow field, each pixel  $p_{i,t}$  with a positive probability value is projected onto the tracking probability map of frame  $F_{t+1}$ . To map its new coordinates  $(x_{i,t+1}, y_{i,t+1})$  to whole-numbered coordinates, the probability value associated with  $p_{i,t}$  is distributed onto the four adjacent pixels  $p_{j1,t+1} \dots p_{j4,t+1}$  based on their L2 distance to the new position, provided that these pixels are inside the region of the image.

In addition to populating the tracking probability map, the current total number of tracked targets is determined based on the ground truth frame and stored as part of the global tracking state.

### 3.5 Tracking Estimation

At the arrival of each new ToF frame, a tracking probability map is calculated that provides a first estimation which points in the point cloud correspond to the tracked human. However, this estimation has to be refined due to potential errors introduced by the flow field based propagation of the tracking probability. In our experience, especially human extremities such as arms are prone to misdetection during optical flow propagation with low-resolution ToF cameras (false negatives). Also, tracking probabilities might be erroneously associated to non-tracked objects in the surrounding environment (false positives).

For this reason, the tracking estimation step is split into two stages: tracking refinement stage and outlier rejection stage.

**Tracking Refinement Stage.** The tracking refinement stage is primarily targeted at correcting false negative detections, e.g. non-detected extremities. The tracking probability map is first binarized by comparison against a pre-defined threshold and then segmented into connected probable tracking regions  $r_i$ . For each region, the center of mass  $m_i$  is calculated. Using  $m_i$  as a seed, a floodfill operation is performed on the associated depth image in order to connect previously undetected pixels with local continuity in 3D space. The result is a refined tracking estimate  $r'_i$  for each connected region.

**Outlier Rejection Stage.** While false negative detections have been resolved in the previous stage, there is still a possibility for false positive detections to be present due to erroneous propagation of the tracking probability map onto untracked pixels. To reject these outliers, the current number of probable tracking regions is first checked against the number of tracked targets (see Sect. 3.4). If there are more regions than tracked targets, we perform a similarity comparison between each tracked region  $r_{j,t-1'}$  of the last frame and all current probable tracked regions  $r_{i,t'}$  in order to detect the correct correspondences. The similarity comparison is based on both 2D similarity metrics (e.g. 2D center location and area of a region) and 3D similarity metrics (e.g. Euclidean distance between the center points in 3D space). For each region  $r_{j,t-1'}$  of the previous frame, the best matching region  $r_{i,t'}$  is determined and its features are stored as detected tracked regions in the current ToF frame. In order to avoid merging of multiple regions  $r_{j,t-1'}$  onto a single region  $r_{i,t'}$ , regions  $r_{i,t'}$  are exempt from further similarity comparisons once they have been successfully matched.

As a last step, for each detected tracked region all corresponding points in the ToF frame cloud are selected. This results in the full body point cloud of the respective tracked human being available for further processing.

## 4 Results

The developed algorithm has been evaluated in the three scenarios presented in Sects. 2.1–2.3. For both the latency minimization and the frame rate optimization scenario, evaluation was performed by comparing the extended tracking map, which is calculated immediately on the arrival of each new ToF frame, against the corresponding ground truth, which becomes available with a certain delay. This also means that only frames for which a corresponding ground truth was received are taken into account. For the third scenario, which is the reconstruction of lost tracking (gaps in the ground truth), the evaluation method and results are described in Sect. 4.3.

All tests were performed under Linux Ubuntu 12.04 using an AMD Phenom II 1090T processor with six cores at 3.2 GHz and 12 GB of RAM. All cameras have been registered against an optical tracking system.

Table 1 lists the metrics employed for accuracy evaluation.

### 4.1 Latency Minimization

For the latency minimization scenario, evaluation was performed on three recorded data sets. Set A has a duration of 53.5 s, contains 317 ToF frames and 265 ground truth frames. The cameras are located with a distance of 31.2 cm between each other and share the same field of view. The desired latency for evaluation was artificially introduced by playing back the Kinect data with a delay between 1 and 10 s. The average processing time per ToF frame was 39 ms, independently of the induced delay.

In set A, the tracked person comes into the field of view two times. To allow for a detailed examination, evaluation has been performed on two different subsets of the measurements: A1 takes into account all frames of each measurement, A2 includes

**Table 1** Metrics for accuracy evaluation

Metric	Definition
True positives $tp$	Pixels correctly classified as part of the tracked human
True negatives $tn$	Pixels correctly classified as not part of the tracked human
False positives $fp$	Pixels incorrectly classified as part of the tracked human
False negatives $fn$	Pixels incorrectly classified as not part of the tracked human
Precision	$\frac{tp}{(tp+fp)}$
Recall	$\frac{tp}{(tp+fn)}$
ToF frame processing time	Time required for processing a single ToF frame (ms)
Ground truth processing time	Time required for forward propagation of the ground truth of a single Kinect frame (ms)
Tracking loss	Percentage of frames with complete loss of tracking

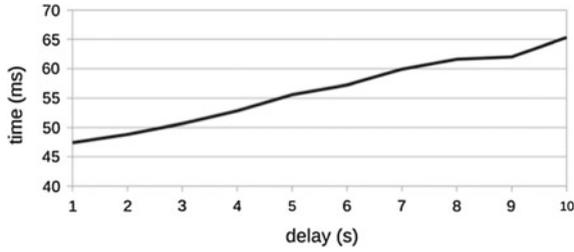


Fig. 5 Ground truth processing time (shown for subset A1 only)

only the frames in which recall and precision were positive, i.e. tracking was actually performed. As a consequence, subset A1 is directly influenced by the delay of the ground truth: On entry of a person into the field of view, there is no ground truth available until the delayed ground truth is received. A higher delay therefore directly results in more frames in which no forward propagation happens and no tracking is performed which in turn lead to a higher rate of false negative classifications and thereby a lower recall.

In all following figures, obtained results are shown over the respective delay; the continuous line corresponds to subset A1 whereas the dotted line corresponds to subset A2. All reported results are averaged over all frames of each measurement.

Figure 5 shows the ground truth processing time. Figure 6 shows the numbers of false negative and false positive classifications. Figure 7 shows the resulting precision of the tracking estimate and the achieved recall of the tracking estimate.

Set B was recorded with the aim of evaluating the proposed algorithm in terms of robustness against data acquired from different points of view.

It contains data of six ToF cameras that are ceiling-mounted in four corners as well as on the sides of a rectangle of about 2 m × 2 m (see Fig. 8). A Kinect camera mounted in one of the corners is used as ground truth. Set B has a duration of 85 s, contains approximately 230 ToF frames per camera and 294 ground truth frames. Again, the results are split into two subsets B1 and B2 where B2 only contains frames where a detection was performed. Further information about the spatial relation between each ToF camera and the Kinect camera as well as the achieved results (recall and precision) for both subsets B1 and B2 are shown in Table 2.

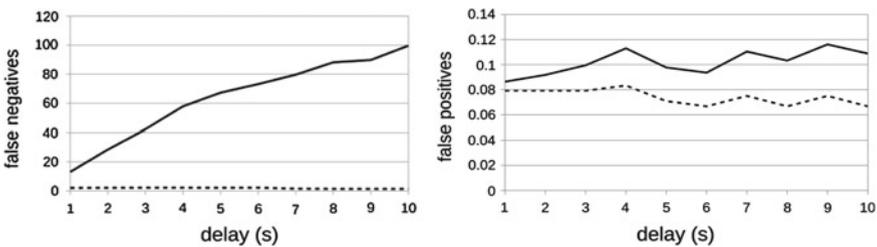


Fig. 6 Number of false negative classifications (left) and false positive classifications (right)

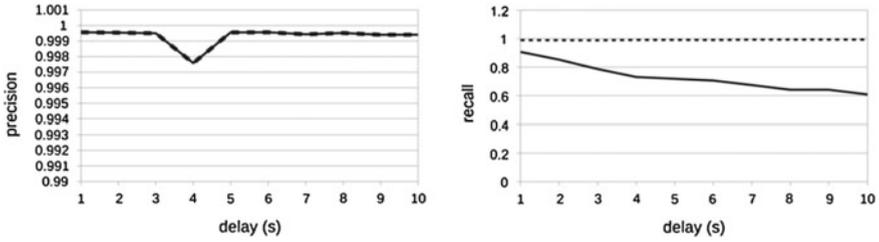


Fig. 7 Precision of the tracking estimate (left) and recall of the tracking estimate (right)

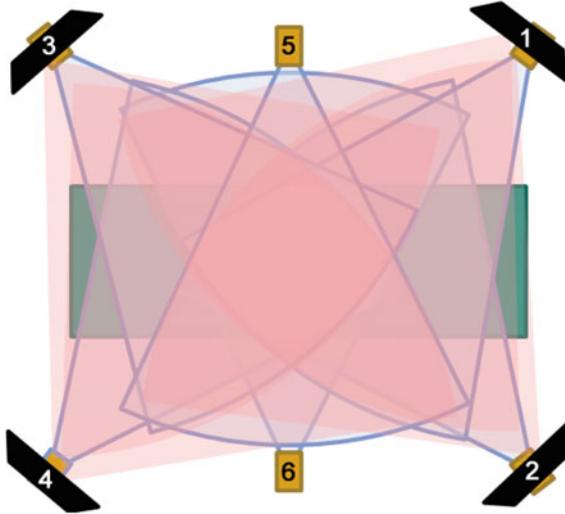
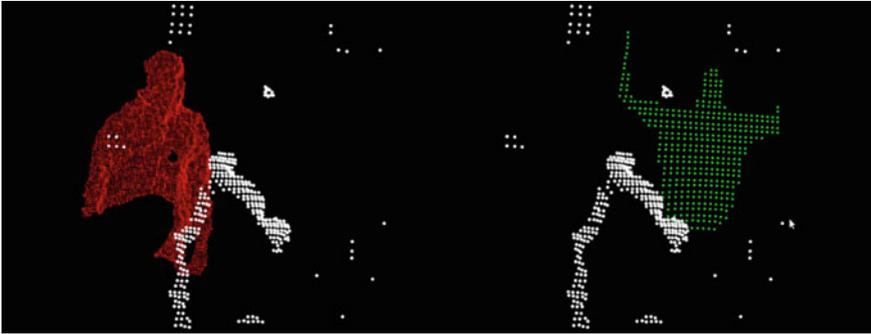


Fig. 8 Schematic top-down view of spatial camera configuration: Six ToF cameras (small rectangles) and four Kinect cameras (trapezes) are mounted at a rectangular rig to supervise the surroundings of an operating table (large rectangle)

Table 2 Spatial configuration and accuracy evaluation for six ToF cameras with different points of view compared to the Kinect camera and latency of 1 s

	1	2	3	4	5	6
Angle compared to Kinect (est.)	0°	90°	90°	180°	45°	135°
Distance to Kinect (cm)	31	163	192	251	92	189
Recall B1	0.71	0.71	0.80	0.66	0.80	0.64
Precision B1	0.99	0.96	0.97	0.88	0.97	0.92
Recall B2	0.90	0.90	0.91	0.96	0.91	0.96
Precision B2	0.99	0.90	0.97	0.88	0.97	0.96



**Fig. 9** Delayed ground truth (*left scene*) and pre-calculated tracking estimate (*right scene*) in latency minimization scenario

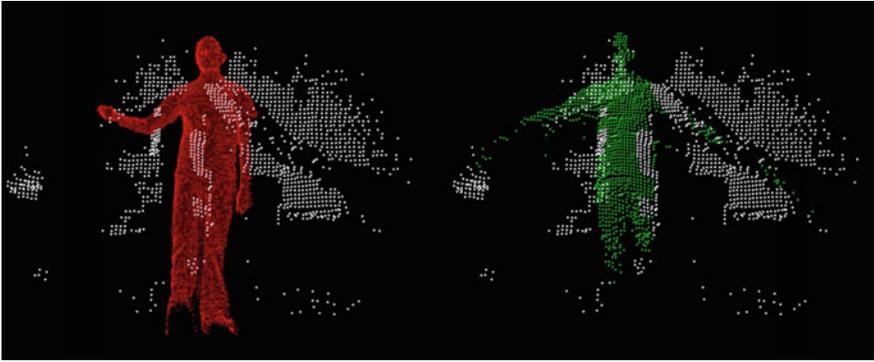
Figure 9 shows a side-by-side exemplary view of the point cloud of a single ToF camera with the delayed ground truth (left) and the pre-calculated tracking estimate for this scenario (right). All 3D points that are pre-calculated as corresponding to the tracked human have been omitted from the scene for better visibility.

## 4.2 Frame Rate Optimization

Contrary to the camera system used in the scenario above, which has already been well-tested and optimized, e.g. with regards to crosstalk of the different cameras illuminating the scene with infrared light, the combination of the Kinect II with the Argos 3D P100 is employed as a proof of concept for the purpose of evaluating the presented algorithm. Currently, the maximum frame rate of 160fps for the Argos camera can only be achieved with a low integration time that drastically decreases the sensing range of the camera. As a compromise, we operated the camera at 80fps which yielded an acceptable sensing range for object with a medium to high reflectivity (i.e. people wearing white clothes). In addition, we observed infrequent crosstalk. Figure 10 shows the pre-calculation with the Argos 3D P100.

Evaluation was performed using four different data sets of lengths between 30 and 68 s. Each data set contains at least 2.300 frames acquired by the ToF camera and 600 frames taken by the Kinect II. Again, the measurements were split as before into subsets D1 and D2.

As the processing of each ToF frame took more than 230 ms on average, which resulted in dropped frames, we slowed back the playback of the recorded data by a factor of 10. In proportion, this corresponds to a processing time of about 20 ms, and can serve as an indication for the potential accuracy of the algorithm. Table 3 lists the resulting accuracy metrics.



**Fig. 10** Delayed ground truth (*left scene*) and pre-calculated tracking estimate (*right scene*) in frame rate optimization scenario

**Table 3** Accuracy evaluation for high frame rate ToF at normal and reduced speed

	$1 \times \text{speed}$	$0.1 \times \text{speed}$
Average recall D1	0.73	0.87
Average precision D1	0.87	0.90
Average recall D2	0.96	0.94
Average precision D2	0.95	0.91

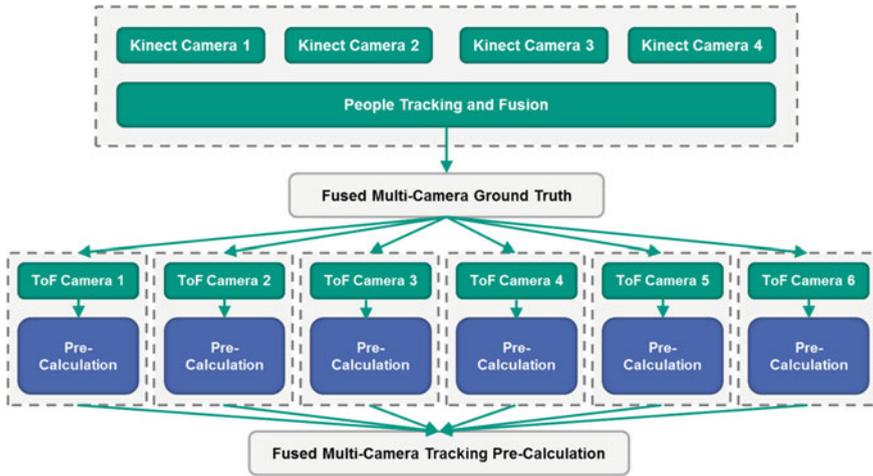
### 4.3 Tracking Reconstruction

Set E with a length of 151 s represents the full configuration of the camera system. It contains 926 ToF frames for each of the six ToF cameras and 453 (460) point clouds representing the ground truth. One person was performing different tasks on both sides of the OR table.

In this scenario, no effective time delay was introduced for the fused ground truth. For each ToF frame, the correctness of the corresponding ground truth was manually annotated using the classification shown in Table 4.

**Table 4** Classifications for manual annotation of ground truth quality

Correct	Ground truth corresponds to the tracked human
Loss	No ground truth despite the human being visible in the scene
Noise	“Tracking” of objects not corresponding to the human in addition the correct tracking of the human
Holes	Parts of the human were not detected by ground truth



**Fig. 11** Data flow for tracking reconstruction scenario: Four Kinect cameras supervise the scene and independently perform human tracking. Their fused output is used as ground truth for six parallel instances of the model-free pre-calculation algorithm, each based on scene data acquired by one ToF camera. The results are then fused into a point cloud containing the final pre-calculation

In difference to set B, the full human tracking Kinect system has been used as ground truth. Figure 11 shows an overview of the data flow for set E.

As can be seen from the spatial setup of the cameras as depicted in Fig. 8, for ToF cameras 1, 2, 3 and 4 there is a Kinect camera located close by with a similar angle of view. Each camera only supervises a part of the scene: Even-numbered cameras and odd-numbered cameras acquire different sides of the OR table in the OP:Sense setup. For this evaluation, the results of camera ToF 1, ToF 2 and ToF 3 are used as these represent the three different cases (ToF 1 and ToF 2 look at different sides of the OR table; ToF 3 does not have a spatially close Kinect camera).

In the evaluation data set, there is a frequent loss of tracking by the Kinect cameras. In some instances, this has been intentionally caused by the tracked person standing still which is a common reason of failure for tracking algorithms that rely on detection motion in their first stage.

Contrary to the previous Sects. 4.1 and 4.2, where the accuracy of the pre-calculation against a known ground truth was evaluated, evaluation here takes into account all frames, including those where no ground truth was available. This is necessary for evaluating the capability to fill gaps in the ground truth, e.g. continue tracking the target even if the ground truth incorrectly fails.

**Table 5** Accuracy evaluation for tracking reconstruction scenario based on all frames where tracking was performed

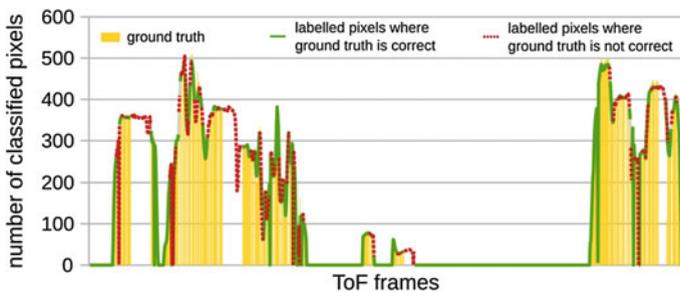
	ToF 1	ToF 2	ToF 3
Average % of false positives	4.13	1.44	8.02
Average % of false negatives	10.31	10.85	11.29
Average recall	0.90	0.89	0.89
Average precision	0.98	0.99	0.97

**Table 6** Accuracy evaluation for tracking reconstruction scenario based on frames with correct ground truth only

	ToF 1	ToF 2	ToF 3
Average % of false positives	0.59	1.36	2.17
Average % of false negatives	12.81	11.17	7.9
Average recall	0.87	0.89	0.92
Average precision	0.99	0.99	0.98

To determine the accuracy, recall and precision were calculated. Calculations only took into account the frames where a valid ground truth was available and tracking was performed. Tables 5 and 6 show the achieved results when taking into account either all frames where ground truth was not classified as “loss” or all frames where ground truth was classified as “correct”.

Figures 12, 13 and 14 show graphs of the tracking results obtained in evaluation as bar/line diagrams with the number of tracked pixels on x-axis and the frame number on y-axis. The bars represent the ground truth as obtained by the Kinect camera; for better visibility, bars have been plotted without interleaving intervals. The line represents the result of the pre-calculation algorithm. For a better understanding of the graphs, the line is plotted solidly where ground truth was labelled as tracking the human; where ground truth incorrectly failed to track (loss in tracking), the line representing the result of the pre-calculation algorithm is plotted as points.



**Fig. 12** Tracking reconstruction results for camera 1

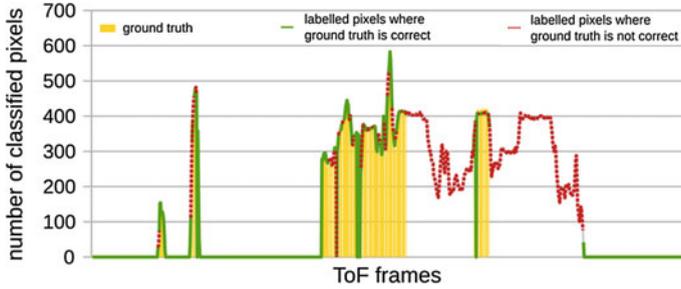


Fig. 13 Tracking reconstruction results for camera 2

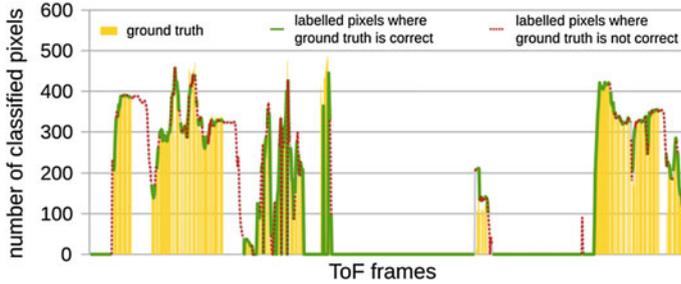


Fig. 14 Tracking reconstruction results for camera 3

## 5 Discussion

For the latency minimization scenario, Fig. 5 shows that the ground truth processing time starts at 47 ms at a delay of 1 s and increases with longer delays. This corresponds to a first processing step of about 45 ms, in which transformation of the ground truth cloud and correspondence calculation are performed, followed by the forward propagation of the ground truth which takes about 1.7 ms/s of delay and is therefore also applicable to longer delays.

The total latency of the pre-calculated tracking can be calculated as the sum of the latency of the ToF cameras in the six-camera setup of about 240 ms and the ToF frame processing time of 39 ms. The resulting total latency of less than 300 ms is independent of the induced delay, so the observed speedup of the tracking is between  $3\times$  and  $33\times$  for a respective delay of 1–10 s.

As expected, the number of false negative classifications as depicted in Fig. 6 is approximately proportional to the induced delay for the subset A1 (see Sect. 4.1). For subset A2, from which frames without a ground truth were excluded, the number of false negative classifications was negligible and clearly independent of the delay.

These results lead to a high precision (see Fig. 7), e.g. close to nil points are erroneously classified as belonging to the tracked human. For subset A1, recall is again proportional to the delay as with a higher delay, there is no ground truth for a

large number of frames. If only frames for which a ground truth was available during the measurement are taken into account (subset A2), recall is close to 1 which means that almost all points that belong to the tracked human have been classified as such (see Fig. 9).

Measurements with six ToF cameras show that the proposed algorithm shows good results also on different camera configurations, i.e. when the ToF camera and the Kinect camera are not mounted with a similar point of view, as can be seen from Table 2. Subset B1 shows worse results on recall than subset B2, due to the fact that with different fields of view, the tracked human is often not visible in both cameras at once, so no correspondences can be established. For the six-camera scenario specifically, we expect to solve this by utilizing the fused output of four spatially distributed Kinect cameras as ground truth.

The rather long processing time when using the Argos 3D P100 camera is consistent with the timings measured for the pmd[vision] S3 cameras: The Argos3D P100 nominally provides about six times more points per frame, for which correspondences have to be determined, which leads to an increase in processing time from 46 ms to about 230 ms. However, this calculation is currently performed on CPU in a single thread so we are expecting to achieve a large speedup by parallelizing on CPU and/or GPU. Further optimizations of the frame rate and image quality are expected by using a different high-speed ToF camera, the upcoming Argos 3D P320, which features 12 instead of 2 LEDs for illumination and thereby increases the effective sensing range.

For the tracking reconstruction scenario, results show a good fit between the ground truth that was classified as “correct” and the according result of our model-free pre-calculation tracking algorithm. It is also clearly visible that the tracking continued when ground truth was lost.

When taking into account the tracking results depicted in Figs. 12, 13 and 14, one can further see that the tracking results of the ToF cameras complement each other. E.g. when the tracked human is not visible for cameras ToF 1 and ToF 3, ToF 2 performs the pre-calculation. This is the desired behavior based on the spatial camera setup of OP:Sense (as shown in Fig. 8).

For the accuracy, there is again a high recall and precision as given in Tables 5 and 6. While both metrics yield slightly higher results when only taking into account “correct” ground truth, the difference is negligible. However, it can be seen that the position of the ToF camera does not affect pre-calculation result.

## 6 Conclusions

We have proposed a new approach for pre-calculating the body point cloud of a human using a model-free algorithm which is based on time-delayed ground truth. It features two distinct processing pipelines: One pipeline processes the ground truth, that corresponds to a past measurement frame, and propagates it forward to the current frame. The other pipeline handles the incoming data from the faster 3D camera system

and calculates a tracking estimate based on 2D optical flow in combination with a customized background model and various refinement steps.

The algorithm has been implemented and evaluation has been performed on three different scenarios. Results for the latency minimization scenario show that the presented approach consistently achieves very good results for the evaluated data sets. The distinction between two different data sets for each evaluation shows that apart from the initial delay until a tracking is established, the magnitude of the latency doesn't affect the high tracking quality of the algorithm. While still good, the accuracy of the second scenario is lower than that of the first scenario and the current processing time prohibits its intended usage. For this reason, optimization of the algorithm in terms of computational costs and the optimization of our test bed for the second scenario will be addressed as detailed above. The results obtained for the third scenario show that using the proposed algorithm with multiple cameras, both for providing the ground truth and for pre-calculating the tracking, gives an accurate and robust tracking estimation. It is also capable of covering gaps in the ground truth, i.e. continues tracking even if tracking was lost as ground truth.

As next steps, we aim to apply the developed algorithm to a number of use cases in human-robot-interaction, e.g. camera-based switching of robot control modes. We also intend to apply the algorithm to other kinds of tracking scenarios using different input modalities.

## References

1. Nicolai, P., Raczkowsky, J., Wörn, H.: Continuous Pre-calculation of human tracking with time-delayed ground-truth—a hybrid approach to minimizing tracking latency by combination of different 3D cameras. In: Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, pp. 121–130 (2015). ISBN 978-989-758-123-6
2. Okada, R., Shirai, Y., Miura, J.: Tracking a person with 3-D motion by integrating optical flow and depth. In: 4th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 336–341 (2000)
3. Tsutsui, H., Miura, J., Shirai, Y.: Optical flow-based person tracking by multiple cameras. In: International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 91–96 (2001)
4. Klappstein, J., et al.: Moving Object Segmentation Using Optical Flow and Depth Information. Lecture Notes in Computer Science: Advances in Image and Video Technology, vol. 5414, pp. 611–623 (2009)
5. Józków, G., et al.: Combined matching of 2D and 3D Kinect<sup>TM</sup> data to support indoor mapping and navigation. In: Proceedings of Annual Conference of American Society for Photogrammetry and Remote Sensing (2014)
6. Quigley, M. et al.: ROS: an open source Robot Operating System. In: ICRA '09, International Conference on Robotics and Automation Workshop on Open Source Software (2009)
7. Bradski, G.R., Pisarevsky, V.: Intel's computer vision library: applications in calibration, stereo, segmentation, tracking, gesture, face and object recognition. In: CVPR '00, IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 796–797 (2000)
8. Rusu, R.B., Cousins, S.: 3D is here: point cloud library (PCL). In: ICRA '11, International Conference on Robotics and Automation, pp. 1–4 (2011)

9. Moennich, H. et al.: A supervision system for the intuitive usage of a telemanipulated surgical robotic setup. In: ROBIO '11, IEEE International conference on Robotics and Biomimetics, pp. 449–454 (2011)
10. Beyl, T. et al.: Multi kinect people detection for intuitive and safe human robot cooperation in the operating room. In: ICAR '13, International Conference on Advanced Robotics, pp. 1–6 (2013)
11. Živkovic, Z., Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.* **27**, 773–780 (2005)
12. Sobral, A., Vacavant, A.: A comprehensive review of background subtraction algorithms evaluated with synthetic and real video. In: *Computer Vision and Understanding*, vol. 122, pp. 4–21. Elsevier (2014)
13. Sánchez, J., Meinhardt-Llopis, E., Facciolo, G.: TV-L1 Optical Flow Estimation. *Image Process. Online* **3**, 137–150 (2013)

# Highly Parallelizable Algorithm for Keypoint Detection in 3-D Point Clouds

Jens Garstka and Gabriele Peters

**Abstract** In computer vision a reliable recognition and classification of objects is an essential milestone on the way to autonomous scene understanding. In particular, keypoint detection is an essential prerequisite towards its successful implementation. The aim of keypoint algorithms is the identification of such areas within 2-D or 3-D representations of objects which have a particularly high saliency and which are as unambiguous as possible. While keypoints are widely used in the 2-D domain, their 3-D counterparts are more rare in practice. One of the reasons often consists in their long computation time. We present a highly parallelizable algorithm for 3-D keypoint detection which can be implemented on modern GPUs for fast execution. In addition to its speed, the algorithm is characterized by a high robustness against rotations and translations of the objects and a moderate robustness against noise. We evaluate our approach in a direct comparison with state-of-the-art keypoint detection algorithms in terms of repeatability and computation time.

**Keywords** 3-D keypoint detection · 3-D object recognition · 3-D computer vision

## 1 Introduction

3-D object recognition and classification is a fundamental part of computer vision research. Many of the existing recognition systems use local feature based methods as they are more robust to occlusion and clutter. In those systems keypoint detection should be the first major step to get distinctive local areas for discriminative local feature descriptions. But a recent survey on feature based 3-D object recognition

---

J. Garstka (✉) · G. Peters  
Faculty of Mathematics and Computer Science, Human-Computer Interaction,  
FernUniversität in Hagen, University of Hagen, 58084 Hagen, Germany  
e-mail: jens.garstka@fernuni-hagen.de  
URL: <http://www.fernuni-hagen.de/mci>

G. Peters  
e-mail: [gabriele.peters@fernuni-hagen.de](mailto:gabriele.peters@fernuni-hagen.de)

systems by Guo et al. [7] shows, that many major systems for local feature based 3-D object recognition use sparse sampling or mesh decimation methods to create a set of points on which the feature descriptors will be computed. In terms of repeatability and informativeness these methods do not result in qualified 3-D keypoints.

There are a variety of reasons why existing 3-D keypoint detection algorithms are not used: Some of them are sensitive to noise and some are time consuming. Furthermore, there is only a handful of methods, that work with unstructured 3-D point clouds without a time consuming approximation of local surface patches or normal vectors [5, 12–14, 20, 21].

This paper addresses the problems described above. The proposed method is a fast and robust algorithm for automatic identification of 3-D keypoints in unstructured 3-D point clouds. We create a filled and watertight voxel representation (in terms of a dense voxel connectivity) of a point cloud. This voxel representation is convolved with a spherical convolution kernel. The sphere works as an integral operator on all voxels containing points of the point cloud. The convolution gives the proportion of voxels of the sphere which are inside the point cloud. The proportion values are used to identify regions of interest, and from these robust keypoints are extracted. All parts of the algorithm are highly parallelized and thus will be computed very quickly. The size of the convolution kernel can be adopted to the size of the area which is used by the local feature descriptor. Furthermore, we can easily simulate a lower resolution point cloud by increasing the voxel size. Therefore, we can create keypoints for multiple resolutions. Finally we will show, that our approach provides robust keypoints, even if we add noise to the point cloud.

## 2 Related Work

There are many 3-D keypoint detection algorithms that work on meshes or use surface reconstruction methods. A brief overview is given in a recent survey paper by Guo et al. [7]. But there are only a few of them that work directly on unstructured 3-D point cloud data. They have been compared multiple times, e.g., by Salti et al. [15], Dutagaci et al. [3], and Filipe and Alexandre [4] and therefore, we will give just a short overview of algorithms, which are designed to work with point clouds only.

Pauly et al. [14] use a principal component analysis to compute a covariance matrix  $C$  for the local neighborhood of each point  $\mathbf{p}$ . With the eigenvalues  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  they introduce the surface variation  $\sigma_n(\mathbf{p}) = \lambda_1/(\lambda_1 + \lambda_2 + \lambda_3)$ , for a neighborhood of size  $n$ , i.e., the  $n$  nearest neighbors to  $\mathbf{p}$ . Within a smoothed map of surface variations Pauly et al. do a local maxima search to find the keypoints. A major drawback of this method is, that the surface variation is sensitive to noise (Guo et al. [7]).

Matei et al. [12] use a similar approach as Pauly et al., but they use only the smallest eigenvalue  $\lambda_3$  of the covariance matrix  $C$  for a local neighborhood of a point  $\mathbf{p}$  to determine the surface variation. But in contrast to Pauly et al., the method from Matei et al. provides only a fixed-scale keypoint detection.

The algorithm presented by Flint et al. [5] is a 3-D extension of 2-D algorithms like SIFT [11] and SURF [2] called THRIFT. They divide the spatial space by a uniform voxel grid and calculate a normalized quantity  $D$  for each voxel. To construct a density scale-space Flint et al. convolve  $D$  with a series of 3-D Gaussian kernels  $g(\sigma)$ . This gives rise to a scale-space  $S(\mathbf{p}, \sigma) = (D \otimes g(\sigma))(\mathbf{p})$  for each 3-D point  $\mathbf{p}$ . Finally, they compute the determinant of Hessian matrix at each point of the scale space. Within the resulting  $3 \times 3 \times 3 \times 3$  matrix, a non maximal suppression reduces the entries to local maxima, which become interest points.

Unnikrishnan and Hebert [20] introduce a 3-D keypoint detection algorithm based on an integral operator for point clouds, which captures surface variations. The surface variations are determined by an exponential damped displacement of the points along the normal vectors weighted by the mean curvature. The difference between the original points and the displaced points are the surface variations which will be used to extract the 3-D keypoints, i.e., if a displacement is an extremum within the geodesic neighborhood the corresponding 3-D point is used as keypoint.

Zhong [21] propose another surface variation method. In their work they use the ratio of two successive eigenvalues to discard keypoint candidates. This is done, because two of the eigenvalues can become equal and thus ambiguous, when the corresponding local part of the point cloud is symmetric. Apart from this, they use the smallest eigenvalue to extract 3-D keypoints, as proposed by Matei et al.

Finally, also Mian et al. [13] propose a surface variation method. For each point  $\mathbf{p}$  they rotate the local point cloud neighborhood in order to align its normal vector  $n_{\mathbf{p}}$  to the  $z$ -axis. To calculate the surface variation they apply a principal component analysis to the oriented point cloud and use the ratio between the first two principal axes of the local surface as measure to extract the 3-D keypoints.

### 3 Our Algorithm

The basic concept of our algorithms is adopted from a keypoint detection algorithm for 3-D surfaces introduced by Gelfand et al. [6]. To be able to use an integral volume to calculate the inner part of a sphere without structural information of the point cloud, we designed a volumetric convolution of a watertight voxel representation of the point cloud and a spherical convolution kernel. This convolution calculates the ratio between inner and outer voxels for all voxels that contain at least one point of the 3-D point cloud. The convolution values of the point cloud get filled into a histogram. Keypoint candidates are 3-D points with rare values, i.e., points corresponding to histogram bins with a low level of filling. We cluster these candidates, find the nearest neighbor of the centroid for each cluster, and use these points as 3-D keypoints.

Thus, our method for getting stable keypoints in an unstructured 3-D point cloud primarily consists of the following steps:

1. Estimate the point cloud resolution to get an appropriate size for the voxel grid.
2. Transfer the point cloud to a watertight voxel representation and fill all voxels inside of this watertight voxel model with values of 1.
3. Calculate a convolution with a voxel representation of a spherical convolution kernel.
4. For each 3-D point fill the convolution results of its corresponding voxel into a histogram.
5. Cluster 3-D points with rare values, i.e., 3-D points of less filled histogram bins, and use the centroid of each cluster to get the nearest 3-D point as stable keypoint.

The details of these steps are provided in the sections below.

### ***3.1 Point Cloud Resolution***

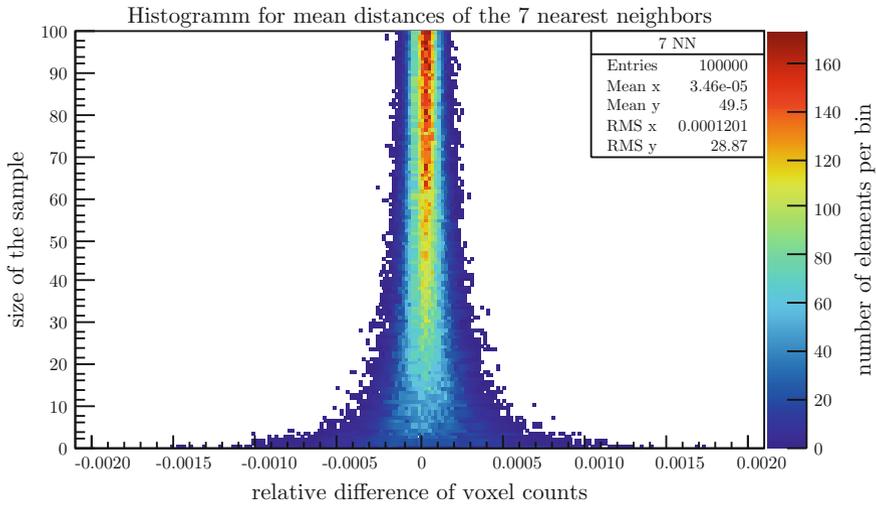
A common way to calculate a point cloud resolution is to calculate the mean distance between each point and its nearest neighbor. Since we use the point cloud resolution to get an appropriate size for the voxel grid (which is to be made watertight in the following steps), we are looking for a voxel size which leads to a voxel representation of the point cloud, where only a few voxels corresponding to the surface of the object remain empty, while as much as possible of the structural information is preserved.

To get appropriate approximations of point cloud resolutions, we carried out an experiment on datasets obtained from ‘The Stanford 3-D Scanning Repository’ [19]. We examined the mean Euclidean distances between  $n$  nearest neighbors of  $m$  randomly selected 3-D points, with  $n \in [2, 10]$  and  $m \in [1, 100]$ . The relative difference between the number of voxels based on the 3-D points to the number of voxels based on the triangle mesh were filled into a separate histogram for each value of  $n$ . The histogram of the ‘Stanford Bunny’ shown in Fig. 1 is illustrative for all results.

With  $n = 7$  the absolute mean of the relative difference is at a minimum. Thus, the experiments using different 3-D objects show that an approximated point cloud resolution with the use of 7 nearest neighbors and with a sample size of 50 randomly selected points is a good choice to get a densely filled initial voxel grid within a small computation time.

### ***3.2 Fast Creation of a Watertight 3-D Model***

Initially we create a voxel grid of cubic voxels with an edge length of the point cloud resolution as described above. Each voxel containing a 3-D point is initialized with a value of 1.0. This creates an approximated voxel representation of the surface. The voxels representing the point cloud are defined as watertight, if the voxels result in a densely connected structure without gaps.



**Fig. 1** The quality of the voxel grid based on the point cloud resolution calculated by the mean of  $n = 7$  nearest neighbors. The x-axis shows the relative difference between the number of voxels based on the 3-D points to the number of voxels based on the triangle mesh of ‘Stanford Bunny’. The y-axis shows the number of randomly selected points which have been used to calculate the values

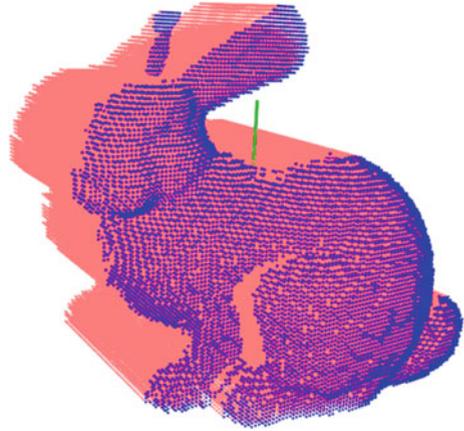
To be able to fill inner values of objects, we assume that it is known whether the point cloud represents a closed model or a depth scan, which has its depth direction along the z-axis.

In case of a depth scan, which is very common in robotics, we take the maximum depth value  $z_{max}$  and the radius of the convolution kernel  $r_{conv}$ , and set the value of all voxels along the z-axis beginning with the first voxel with a value of 1 (a surface voxel) and ending with depth of at least  $z_{max} + r_{conv}$  to a value of 1, too. An illustration of this can be seen in Fig. 2.

In case we assume a closed model we have to fill all inner voxels. Due to variations in density of the point cloud and the desire to keep the voxel size as small as possible, it often occurs that the voxel surface is not watertight. For first tests we implemented two different approaches to close holes in the voxel grid. The first implementation was based on the method from Adamson and Alexa [1]. Their approach is in fact intended to enable a ray tracing of a point cloud. They use spheres around all points of the point cloud to dilate the points to a closed surface. Shooting rays through this surface can be used to close holes. The second implementation was based on a method by Hornung and Kobbelt [8]. Their method creates a watertight model with a combination of adaptive voxel scales and local triangulations. Both methods create watertight models without normal estimation. But the major drawback of both methods consists in their long computation times.

Since the method we propose should be fast, these concepts were discarded for our approach. Instead we use a straightforward solution, which appears to be sufficient

**Fig. 2** This is a voxel representation of a view dependent patch (depth scan) from the ‘Bunny’. *Blue dots* represent voxels corresponding to 3-D points. *Red dots* represent the filled voxels below the surface



for good but fast results. The filling of the voxel grid is described exemplary for the one direction. The calculation of the other directions is performed analogously.

Let  $u$ ,  $v$  and  $w$  the indexes of the 3-D voxel grid in each dimension. For each pair  $(u, v)$  we iterate along all voxels in  $w$ -direction. Beginning with the first occurrence of a surface voxel (with a value of 1) we mark all inner voxels by adding  $-1$  to the subsequent voxels until we reach the next surface voxel. These steps are repeated until we reach the  $w$  boundary of the voxel grid. If we added a value of  $-1$  to the last voxel, we must have passed a surface through a hole. In this case we need to reset all values for  $(u, v)$  back to the previous values.

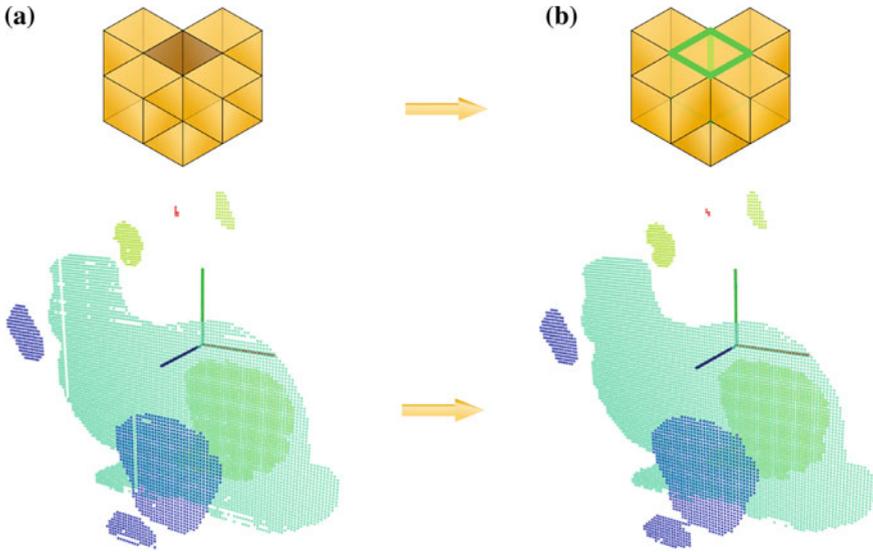
After we did this for each dimension, all voxels with a value  $\leq -2$ , i.e., all voxel which have been marked as inner voxels by passes for at least two dimensions, will be treated as inner voxels and their value will be set to a value of 1. All other voxels will get a value of 0.

We already mentioned, that, if we pass the surface through a hole, we set back all voxel values to previous values. This might result in tubes with a width of one voxel (see Fig. 3a). Because of that, we fill these tubes iteratively in a post-processing step, with the following rule.

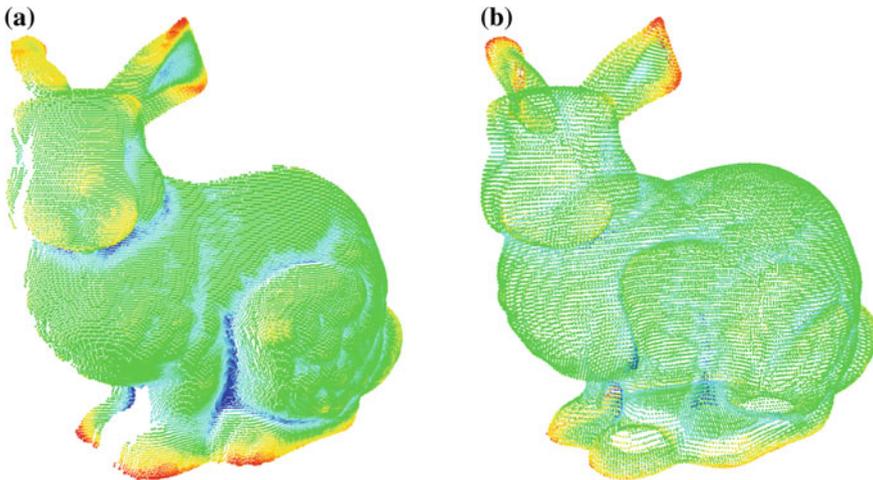
If a voxel at position  $(u, v, w)$  has a value of 0 and at least each of the 26 neighbor voxels except of one of the six direct neighbors  $(u \pm 1$  or  $v \pm 1$  or  $w \pm 1)$  has a value of 0, the voxel at  $(u, v, w)$  gets a value of 1, too. The result is shown in Fig. 3b.

### 3.3 Convolution

The convolution is done with a voxelized sphere of radius  $r_{conv}$ . For a fast GPU based implementation we use the NVIDIA FFT-implementation cuFFT. Figure 4a, b visualize the results showing those voxels which contain 3-D points from the initial point cloud. The convolution values are in  $[0, 1]$ . While values near 0 are depicted



**Fig. 3** The *lower part* of this figure shows sliced versions of the filled voxel grid of the ‘Bunny’. All inner voxels get filled after the surface is passed. This is done for each direction  $u, v$  and  $w$ . If the surface gets missed through a hole, it might lead to tubes with a width of one voxel as shown in (a). If we post-process this voxel grid, we can identify and fill those tubes. This is done as shown in the *upper part* of this figure. The *upper part* of (a) shows a configuration, where a non-filled voxel (tinted in *brown*) is bounded by at least 5 filled direct neighbor voxels (tinted in *beige*). In such a case the voxel framed in green is filled as illustrated in the *upper part* of (b). This leads to a nearly complete filled voxel grid as shown in the *lower part* of (b)



**Fig. 4** The two figures show colored convolution results of the ‘Bunny’ in full resolution using a convolution kernel of radius  $r_{conv} = 10pcr$  ( $pcr =$  point cloud resolution). **a** Shows a depth scan ( $pcr = 0.00167$ ) with convolution values between 0.23 (*red*) and 0.79 (*blue*) while **b** shows the closed point cloud ( $pcr = 0.00150$ ) with values between 0.08 (*red*) and 0.83 (*blue*)

red, values near 1 are depicted blue to purple, and values of about 0.5 are depicted green.

### 3.4 Histograms

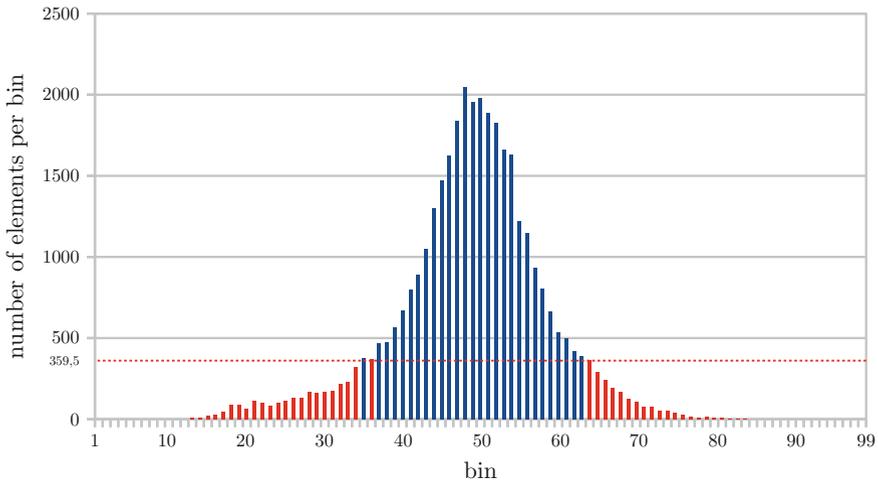
Following the computation of the convolution, we have to identify all convolution values which are interesting, i.e., which are less frequent. To find those regions of values which are less frequent, we fill the convolution values into a histogram. To get an appropriate amount of bins we use Scott’s rule [16] to get a bin width  $b$ :

$$b = \frac{3.49\sigma}{\sqrt[3]{N}}, \tag{1}$$

where  $\sigma$  is the standard deviation of  $N$  values. In case of the Stanford bunny the value of  $b$  is:

$$b = \frac{3.49 \cdot 0.096}{\sqrt[3]{35947}} \approx 0.01015 \tag{2}$$

The corresponding histogram is shown in Fig. 5.



**Fig. 5** This histogram illustrates the distribution of convolution values for the Stanford Bunny. The smallest convolution value is 0.12, while the largest convolution value is 0.85. With  $\sigma = 0.096$  the interval  $[0, 1]$  is divided into 99 bins with a bin size of 0.0101. The *red dotted line* indicates the 1% limit. All 3-D points with a convolution value of bins colored in *red* will be used as keypoint candidates

In case of a depth scan we need to take into account that the convolution values at outer margins do not show the correct values. Thus, we ignore all values of points within an outer margin  $r_{conv}$ .

### 3.5 Clustering

From the histogram we select all bins filled with values of at most 1 % of all points (see Fig. 5). It turned out that this is a good choice for an upper limit since higher values lead to large clusters. On the other hand, a significant smaller limit leads to fragmented and unstable clusters.

All points corresponding to the values in the selected bins will be used as keypoint candidates for clustering. Figure 6a shows all keypoint candidates for the ‘Bunny’. We cluster these points using the Euclidean distance with a range limit of  $3pcr$ . This enables us to handle small primarily longish clusters, e.g., the region above the bunny’s hind legs, as a single connected cluster. Figure 6b illustrates the different clusters with separate colors.

For each cluster we calculate the centroid. Each centroid is used to find its nearest neighbor among the 3-D points of the corresponding cluster. This nearest neighbor is used as 3-D keypoint. Figure 6c, d show the results for the ‘Bunny’.

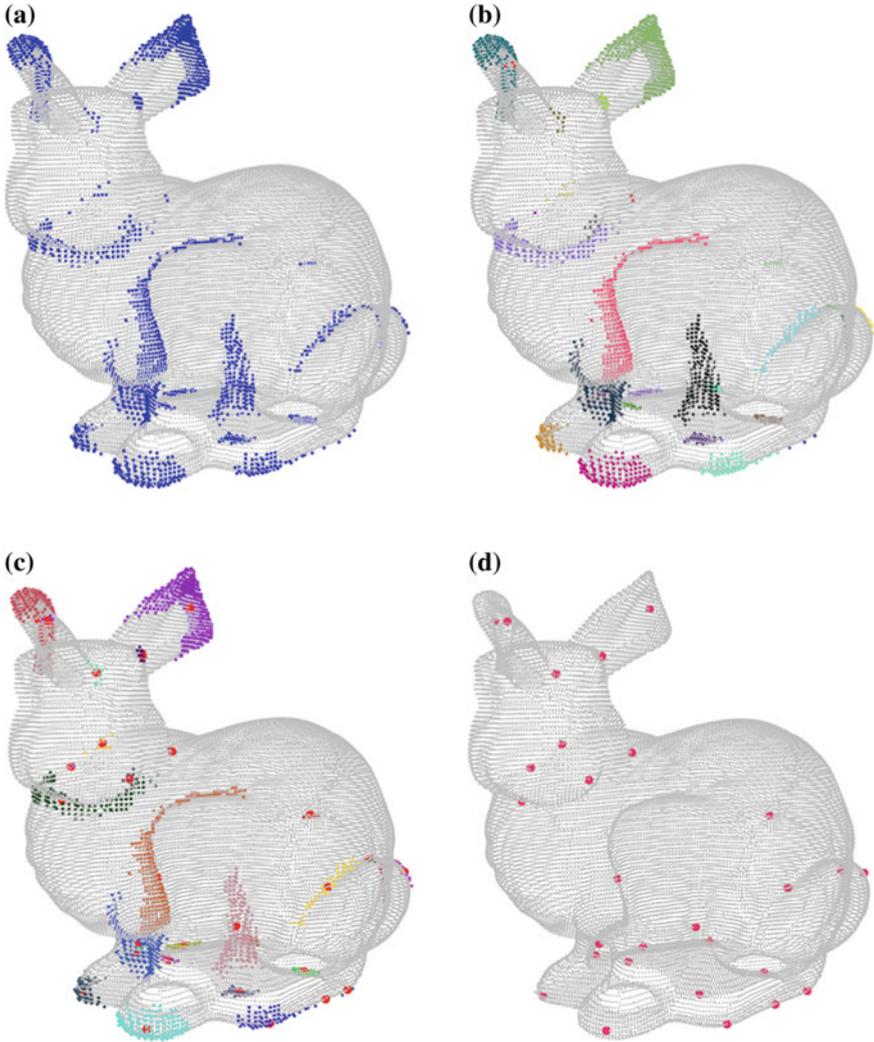
Additional examples of further objects from the Stanford 3-D Scanning Repository are given in the Appendix.

## 4 Results

We evaluated our results with respect to the two main quality features repeatability and computation time. To be comparable to other approaches we used the same method of comparing different keypoint detection algorithms and the same dataset as described by Filipe and Alexandre [4], i.e., we used the large-scale hierarchical multi-view RGB-D object dataset from Lai et al. [9]. The dataset contains over 200000 point clouds from 300 distinct objects. The point clouds have been collected using a turntable and an RGB-D camera. More details can be found in another article by Lai et al. [10].

### 4.1 Repeatability Under Rotation

Filipe and Alexandre [4] use two different measures to compare the repeatability: the relative and absolute repeatability. The relative repeatability is the proportion of keypoints determined from the rotated point cloud, that fall into a neighborhood of



**Fig. 6** These figures illustrate the process of obtaining clusters and keypoints from the set of possible candidates. **a** Shows all 3-D keypoint candidates (*blue*) for the ‘Bunny’. **b** Shows all clusters found for an Euclidean distance with a range limit of  $3\text{ pc}$ . **c** Shows a combination of colored clusters and the corresponding 3-D keypoints. Finally, **d** shows the isolated keypoints

a keypoint of the reference point cloud. The absolute repeatability is the absolute number of keypoints determined in the same manner as the relative repeatability.

To compute the relative and absolute repeatability, we randomly selected five object classes (‘cap’, ‘greens’, ‘kleenex’, ‘scissor’, and ‘stapler’) and picked 10



**Fig. 7** The five object classes obtained from ‘The Stanford 3-D Scanning Repository’ [19] are *cap*, *greens*, *kleenex*, *scissor*, and *stapler*

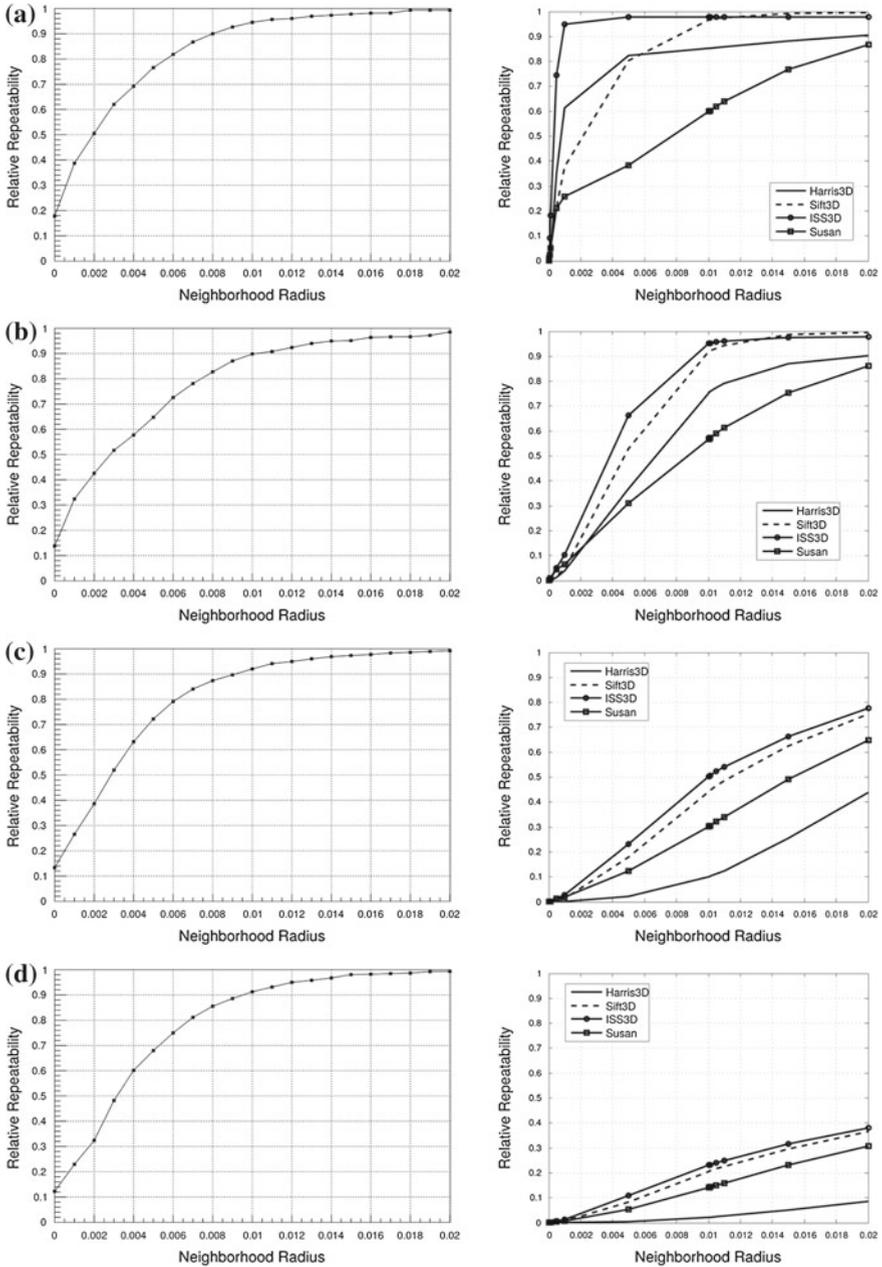
point clouds within each object class randomly as well. A color image of one pose and one point cloud from each of the used object classes is shown in Fig. 7.

Additionally, these 50 base point clouds were rotated around random axes at angles of  $5^\circ$ ,  $15^\circ$ ,  $25^\circ$ , and  $35^\circ$ . Afterwards, we applied our algorithm on each of these 250 point clouds. For neighborhood sizes  $n$  from 0.00 to 0.02 in steps of 0.001 the keypoints of the rotated point clouds were compared with the keypoints of the base point clouds. If a keypoint of a rotated point cloud fell within a neighborhood  $n$  of a keypoint of the base point cloud, the keypoint was counted. Finally, the absolute repeatability was determined based on these counts for each  $n$  as an average of the 50 corresponding point clouds for each angle. The relative repeatability rates are the relations between the absolute repeatabilities of the rotated point clouds and the number of keypoints of the base point clouds.

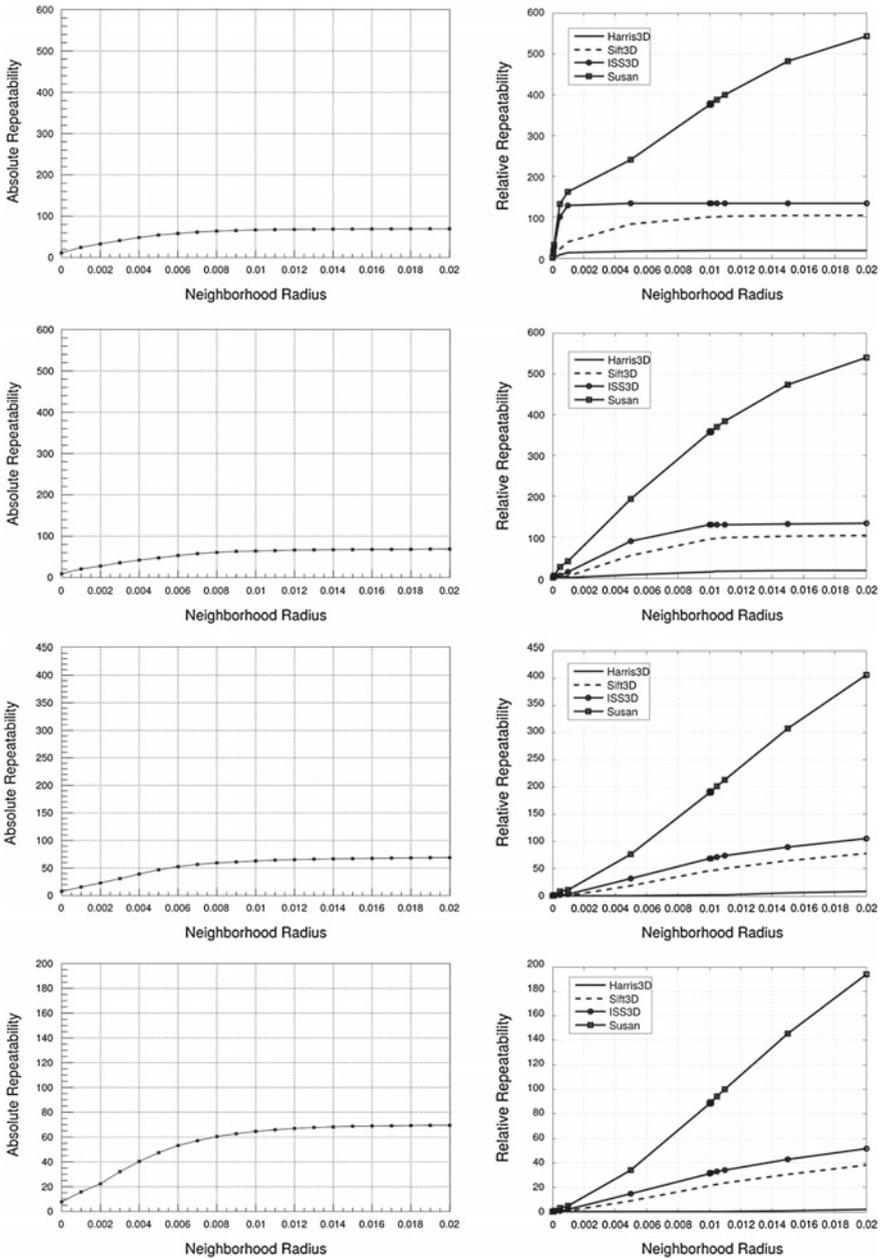
Figure 8 opposes relative repeatability rates of our approach (left hand side) to the corresponding results of four state-of-the-art keypoint detection algorithms (right hand side). Figure 9 does the same for absolute repeatability rates. The approaches evaluated by Filipe and Alexandre [4] are Harris3D [17], SIFT3D [5], ISS3D [21], and SUSAN [18] which they extended to 3-D.

In more detail, the graphs on the left of both Figs. 8 and 9 show the average relative, resp. absolute, repeatability of keypoints computed with our algorithm for 5 randomly selected objects over 10 iterations, i.e., with 10 different rotation axes. The graphs on the right of both figures are taken from the evaluation done by Filipe and Alexandre [4].

It is striking, that the relative repeatability rates of our approach are considerably higher for large rotation angles than those of all other state-of-the-art approaches that have been compared by Filipe and Alexandre. Only for an rotation angle of  $5^\circ$  one of the other algorithms (ISS3D) is able to outperform our approach significantly in the range of small neighborhood radii. On the other hand, the results of our algorithm in terms of absolute repeatability of keypoints are in general less convincing, although for all of the considered rotation angles it outperforms one of the other approaches (Harris3D). For a rotation angle of  $35^\circ$  our algorithm outperforms even three of the other approaches considered (Harris3D, SIFT3D, and ISS3D).



**Fig. 8** Relative repeatability. *Left* our approach. *Right* four state-of-the-art approaches, diagrams taken from the evaluation done by Filipe and Alexandre [4]. **a** Relative repeatability of keypoints at a rotation angle of 5°, **b** Relative repeatability of keypoints at a rotation angle of 15°. **c** Relative repeatability of keypoints at a rotation angle of 25°. **d** Relative repeatability of keypoints at a rotation angle of 35°



**Fig. 9** Absolute repeatability. *Left* our approach. *Right* four state-of-the-art approaches, diagrams taken from the evaluation done by Filipe and Alexandre [4]. *Note* the diagrams taken from Filipe and Alexandre have wrong axis labels. However, they contain the absolute repeatability of keypoints. **a** Absolute repeatability of keypoints at a rotation angle of 5°. **b** Absolute repeatability of keypoints at a rotation angle of 15°. **c** Absolute repeatability of keypoints at a rotation angle of 25°. **d** Absolute repeatability of keypoints at a rotation angle of 35°

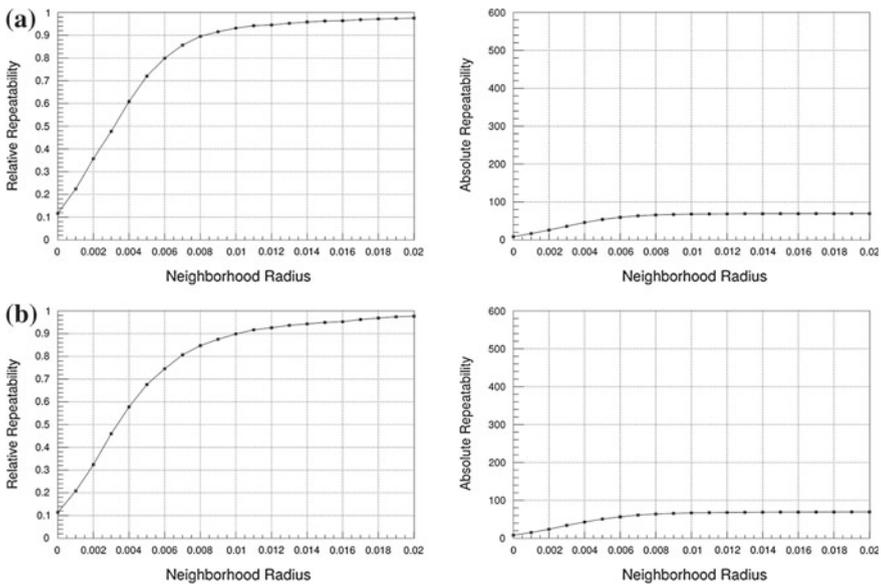
### 4.2 Repeatability Under Noise

Furthermore, we have repeated the described simulations for point clouds with additional random noise at a level of 0.5 times of the point cloud resolution. The graphs of Fig. 10 display the average repeatability rates (relative and absolute) for point clouds with added noise. The differences between these curves and their corresponding curves from the simulations without noise are negligible, which shows that our approach is able to cope with a fair amount of noise, as well.

### 4.3 Computation Time

To calculate average computation times we computed the 3-D keypoints 10 times for each point cloud. The system configuration we used for all experiments is given in Table 1.

The computation time of our algorithm correlates with the number of voxels, i.e., with the dimensions of the voxel grids, which must be powers of two. Measured average computation times in dependence of voxel grid dimensions are given in Table 2.



**Fig. 10** Our approach with additional noise. *Left* relative repeatability. *Right* absolute repeatability. **a** Repeatability for a point cloud with 0.5 *pcr* additional random noise at a rotation angle of 5°. **b** Repeatability for a point cloud with 0.5 *pcr* additional random noise at a rotation angle of 15°

**Table 1** System configuration for experiments

System configuration	
Processor	Intel Xeon E5630 @2.53 GHz
Memory	12 GB
GPU	NVIDIA GeForce GTX 670
GPU memory	2 GB GDDR5
OS	Debian GNU/Linux
NVIDIA driver	340.32

**Table 2** Average computation times

Voxel grid	Time (ms)
$512 \times 256 \times 128$	$\approx 1500$
$256 \times 256 \times 128$	$\approx 700$
$256 \times 256 \times 64$	$\approx 300$
$256 \times 128 \times 128$	$\approx 350$
$256 \times 128 \times 64$	$\approx 80$
$128 \times 128 \times 128$	$\approx 200$
$128 \times 128 \times 64$	$\approx 110$
$128 \times 64 \times 64$	$\approx 60$
$64 \times 64 \times 64$	$\approx 35$

The computation time for most of the point clouds was below 1 s. For many of the point clouds the computation time fell within a range below 300 ms. The average computation time for all 250 point clouds which were used to compare the repeatability rates was 457 ms. This is considerably fast, especially in comparison to the average computation times of Harris3D (1010 ms) and ISS3D (1197 ms), which have been determined on the same system.

## 5 Conclusion

In this paper we have presented a fast and robust approach for extracting keypoints from an unstructured 3-D point cloud. The algorithm is highly parallelizable and can be implemented on modern GPUs.

We have analyzed the performance of our approach in comparison to four state-of-the-art 3-D keypoint detection algorithms by comparing their results on a number of 3-D objects from a large-scale hierarchical multi-view RGB-D object dataset.

Our approach has been proven to outperform other 3-D keypoint detection algorithms in terms of relative repeatability of keypoints. Results in terms of absolute

repeatability rates are less significant. An important advantage of our approach is its speed. We are able to compute the 3-D keypoints within a time of 300 ms for most of the tested objects.

Furthermore, the results show a stable behavior of the keypoint detection algorithm even on point clouds with added noise. Thus, our algorithm might be a fast and more robust alternative for systems that use sparse sampling or mesh decimation methods to create a set of 3-D keypoints. Additional examples can be found in the appendix.

## Appendix—Additional Examples

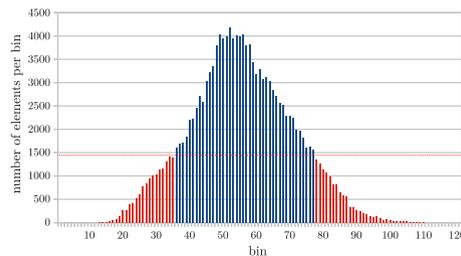
### Happy Buddha

This object is obtained from ‘The Stanford 3-D Scanning Repository’ [19] and is characterized by the following properties:

- Points: 144647
- $pcr = 0.00071$
- Voxel grid:  $135 \times 299 \times 135$
- $r_{conv} = 10 \cdot pcr$
- $\sigma = 0.124884$
- Bins: 121
- Keypoints: 210

The histogram below illustrates the distribution of convolution values for the ‘Happy Buddha’. To save space the labels are not included in the histogram. They correspond to those shown in Fig. 5, i.e., the abscissa shows the bin number, while the ordinate shows the number of elements per bin.

The 3-D point cloud of the ‘Happy Buddha’ shown right is a combination of two types of figures which have already been used to illustrate the results of the ‘Stanford Bunny’. The color gradient used to tint the point of the point cloud illustrates the convolution values from the smallest value (red) to the largest value (blue). This was already used in Fig. 4. The purple markers illustrate the final keypoints. This was already used in Fig. 6d.



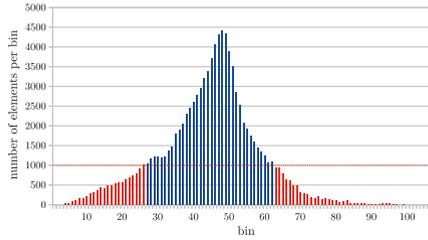


### Dragon

This object is obtained from ‘The Stanford 3-D Scanning Repository’ [19] and is characterized by the following properties:

- Points: 100250
- $pcr = 0.00097$
- Voxel grid:  $236 \times 174 \times 120$
- $r_{conv} = 10 \cdot pcr$
- $\sigma = 0.124507$
- Bins: 107
- Keypoints: 92

The histogram below illustrates the distribution of convolution values for the ‘Dragon’. To save space the labels are not included in the histogram. They correspond to those shown in Fig. 5, i.e., the abscissa shows the bin number, while the ordinate shows the number of elements per bin.



The 3-D point cloud of the ‘Dragon’ shown above is a combination of two types of figures which have already been used to illustrate the results of the ‘Stanford Bunny’. The color gradient used to tint the point of the point cloud illustrates the convolution values from the smallest value (red) to the largest value (blue). This was already used in Fig. 4. The purple markers illustrate the final keypoints. This was already used in Fig. 6d.

## References

1. Adamson, A., Alexa, M.: Ray tracing point set surfaces. In: Shape Modeling International, pp. 272–279. IEEE (2003)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: speeded up robust features. In: Computer Vision—ECCV 2006, pp. 404–417. Springer (2006)
3. Dutagaci, H., Cheung, C.P., Godil, A.: Evaluation of 3d interest point detection techniques via human-generated ground truth. *Vis. Comput.* **28**(9), 901–917 (2012)
4. Filipe, S., Alexandre, L.A.: A comparative evaluation of 3d keypoint detectors. In: 9th Conference on Telecommunications. Conftel 2013, Castelo Branco, Portugal, pp. 145–148 (2013)
5. Flint, A., Dick, A., Hengel, A.V.D.: Thrift: local 3d structure recognition. In: 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, pp. 182–188. IEEE (2007)
6. Gelfand, N., Mitra, N.J., Guibas, L.J., Pottmann, H.: Robust global registration. In: Symposium on Geometry Processing, vol. 2, p. 5 (2005)
7. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J.: 3d object recognition in cluttered scenes with local surface features: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **99**(PrePrints), 1 (2014)
8. Hornung, A., Kobbelt, L.: Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, pp. 41–50. SGP'06, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2006)
9. 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 (ICRA), pp. 1817–1824. IEEE (2011)
10. Lai, K., Bo, L., Ren, X., Fox, D.: A scalable tree-based approach for joint object and pose recognition. In: AAAI (2011)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**, 91–110 (2004)
12. Matei, B., Shan, Y., Sawhney, H.S., Tan, Y., Kumar, R., Huber, D., Hebert, M.: Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(7), 1111–1126 (2006)
13. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *Int. J. Comput. Vis.* **89**(2–3), 348–361 (2010)
14. Pauly, M., Keiser, R., Gross, M.: Multi-scale feature extraction on point-sampled surfaces. In: Computer Graphics Forum, vol. 22, pp. 281–289. Wiley Online Library (2003)
15. Salti, S., Tombari, F., Stefano, L.D.: A performance evaluation of 3d keypoint detectors. In: International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), pp. 236–243. IEEE (2011)
16. Scott, D.W.: On optimal and data-based histograms. *Biometrika* **66**(3), 605–610 (1979)
17. Sipiran, I., Bustos, B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *Vis. Comput.* **27**(11), 963–976 (2011)
18. Smith, S.M., Brady, J.M.: Susana new approach to low level image processing. *Int. J. Comput. Vis.* **23**(1), 45–78 (1997)
19. The stanford 3d scanning repository (2014). <http://graphics.stanford.edu/data/3Dscanrep>
20. Unnikrishnan, R., Hebert, M.: Multi-scale interest regions from unorganized point clouds. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08, pp. 1–8. IEEE (2008)
21. Zhong, Y.: Intrinsic shape signatures: a shape descriptor for 3d object recognition. In: IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), pp. 689–696. IEEE (2009)

# 3D Modelling of Biped Robot Locomotion with Walking Primitives Approach in Simulink Environment

Ramil Khusainov, Ilya Shimchik, Ilya Afanasyev and Evgeni Magid

**Abstract** In the past decades bipedal robots related research gained significant attention as the technology progresses towards acceptable humanoid robot assistants. Serious challenges of human-like biped robot locomotion include such issues as obtaining a human gait multi-functionality, energy efficiency and flexibility. In this paper we present Russian biped robot AR-601M and its locomotion modelling in Simulink environment using walking primitives approach. We consider two robot models: with 6 and 12 Degrees of Freedom (DoFs) per legs, using the same walking strategies. While the 6-DoF model is constrained to move only in sagittal plan, the 12-DoF model supports 3D motion and precisely reflects the hardware of AR-601M robot legs. The locomotion algorithm utilizes position control and involves inverse kinematics computations for the joints. The resulting simulation of robot locomotion is dynamically stable for both models at a small step length and short step time with relatively long damping pauses between the steps.

**Keywords** Biped robot · Robot locomotion · Walking primitives · Dynamic stability · Modelling · Simulink · AR-601M robot

---

R. Khusainov · I. Shimchik · I. Afanasyev · E. Magid (✉)  
Intelligent Robotic Systems Laboratory (LIRS), Innopolis University,  
Universitetskaya street 1, Innopolis, Russian Federation  
e-mail: e.magid@innopolis.ru  
URL: <http://university.innopolis.ru/en/research/robolab/>

R. Khusainov  
e-mail: r.khusainov@innopolis.ru

I. Shimchik  
e-mail: i.shimchik@innopolis.ru

I. Afanasyev  
e-mail: i.afanasyev@innopolis.ru

## 1 Introduction

At present, human-like biped robot locomotion becomes scientific and technical challenge, targeting to provide similar to human gait features of multi-functionality, energy efficiency and flexibility. Over the last few decades, a number of approaches for stable robot walking has been developed [1].

Most of modern bipeds use analytical or controller-based methods to ensure locomotion stability. The analytical approaches are based on physics with consideration of multi-body dynamics, pre-calculated joint trajectories, zero-moment point (ZMP), and trajectory optimization methods. Alternatively, controller-based approaches are able to provide stable biped locomotion by using feedback strategies, continually adapting robot forward dynamics to the stability criteria. However, beside their considerable achievements both analytical and controller-based methods demonstrate serious disadvantages like strong model dependency, high energetic and computational costs, and susceptibility to different disturbances. Therefore, locomotion of biped robots is still far from achieving a human level of anthropomorphism, energy efficiency, stability and robustness.

To reduce cost and time of software development with a real robot while providing system debugging on early stages of algorithm formation, it is crucial to build a relevant simulation model which would take into account robot physical properties and its interaction with the environment. Among other positive effects of simulation prior to experimentation, such simulations help to develop and test balance-based biped locomotion algorithms before experiencing potentially destructive failures with real robots—and these simulations could provide extremely valuable clues on dynamical and static stability behavior and potential issues of robot gaits.

Our long-term research goal targets to develop dynamically stable walking for Russian biped robot AR-601M. This paper presents one particular approach of generating walking primitives of biped robot locomotion in Simulink. We consider two robot models which differ with their legs modelling: a simple initial model with 6-DoF legs (3 DoFs per each leg) and a realistic model with 12-DoF legs (6 DoFs per each leg), while we use the same walking strategies for both models. In such strategy a robot makes a step and then joins another leg with a pause between the steps in order to damp the robot body inertia. The 6 DoFs simulation is the earlier model which was previously presented in [2]; the robot model was constrained to sagittal plane motions only and had four input angles for hip and knee joints and two input of torques which actively stabilize the robot at the end of each step. In this paper we introduce the current stage of our study and, in addition to the simple model, present the extended 12 DoFs simulation model of the biped robot in Simulink/SimMechanics. This advanced 3D model precisely reflects the kinematic structure and the hardware of AR-601M robot legs and implies no artificial restrictions on the motion directions and planes. To calculate joint angles with inverse kinematics in Simulink the model requires three input parameters: step length, step time, and sine feet trajectories. The further goal of our research is to apply 3D model of AR-601M biped robot to the robust and high graphical quality robot simulator

Gazebo<sup>1</sup> and to perform evaluative testing of our locomotion algorithms under ROS framework<sup>2</sup> [3].

The rest of the paper is organized as follows. Section 2 introduces Russian humanoid robot AR-601M. Section 3 considers the theoretical background of a biped robot locomotion methods. Section 4 describes the simple (6 DoFs per legs) and the extended (12 DoFs per legs) biped robot models which were designed in Simulink environment. The simulation results and their discussion appear in Sect. 5. Finally, Sect. 6 is used to familiarize the reader with our future research plans and to conclude.

## 2 System Setup

The biped robot AR-601M (Fig. 1) is being developed by a Russian company Android Technics.<sup>3</sup> It is a human-like biped robot with the height of 144 cm and weight of 65 kg, having 57 DoFs (41 active DoFs, including 12 DoFs in robot legs) [2]. Nowadays, the robot comes with a software that allows it to walk at a low-speed gait with position control when its center of mass (CoM) trajectory is constrained to stay inside the support polygon of a supporting leg footprint during each step.

## 3 Biped Robot Locomotion Modelling

This section familiarizes the reader with popular models of biped robot locomotion and methods for robot motion control.

**Inverted pendulum models** are widely used approaches for building biped robot models. When the robot is approximated with inverted pendulum, the biped robot locomotion is considered as motion of an inverted pendulum. During a step of the biped robot, its swinging leg is similar to an inverted pendulum with a massless rod, which connects the foot and robot's CoM [4]. The linear inverted pendulum model is employed widely as a useful model which simplifies trajectory generation task [5]. There are various modifications of the inverted pendulum: virtual height inverted pendulum model, multiple masses inverted pendulum model, gravity compensated inverted pendulum model, etc. [6]. Among these models, we emphasize a **spring loaded inverted pendulum** model, which describes the spring-like leg behavior of

---

<sup>1</sup>Gazebo robot simulator, <http://www.gazebosim.org>.

<sup>2</sup>Willow Garage Robot Operating System (ROS)—Robotics middleware for robot software development, providing operating system-like functionality, [ros.org](http://ros.org).

<sup>3</sup>Android Technics (in Russian “Androidnaya Tehnika”) is a Russian company, which develops AR-600 robot series (including AR-601M and AR-603E models), <http://en.npo-at.com/products/ar-600>.

**Fig. 1** Android Technics AR-601M robot



human and animal walking or running [7]. The model is also used to approximate bipedal robot locomotion with a human gait [8].

Our research is aimed to achieve energy efficient locomotion with an acceptable walking speed while maintaining the dynamically-stable robot motion. Below we briefly describe methods, which are applied for biped robot locomotion control by various research teams.

**Passive walking** control is based on the body's momentum. Passive walking on a slope is known as the most efficient and natural walking style because there is no extra power without potential energy, and the motion occurs inherently according to its mechanical property [9]. It creates movements in a similar way as a human does, but due to low stability for disturbances, this approach is not broadly used [10].

**Zero Moment Point (ZMP)** defines a special point on the surface where the sum of horizontal inertial and gravitational forces equals to zero [11]. Vukobratovic in his work [12] proposed to use ZMP as a criteria for dynamic stability for biped locomotion. It says that to keep dynamic balance robot's ZMP should lie within the boundaries of a stability area. This approach plays a role of a criterion in the stability analysis of biped robot locomotion and could be considered as a dynamic analogue of CoM or center of gravity (CoG) criterion for static stability analysis [13].

ZMP approach causes poor energy efficiency of the locomotion, because it defines trajectory of robot bodies without considering the load of each particular joint.

**Walking primitives** approach generates trajectories for the joints. Moreover, each walking primitive is calculated off-line, whereas during the locomotion a planning algorithm can generate motions as a sequence of predefined walking primitives [14] in order to move a robot from the current posture and location to the goal location. Each primitive implies the following restrictions: at the beginning and at the end of each primitive the accelerations and velocities of all joints are equal to zero, and, therefore, ZMP always stays inside a support polygon [15]. In order to combine two walking primitives together, the initial state of the second primitive should be the end state of the previous one. Combining together a set of walking primitives is used to obtain a stable locomotion of the robot.

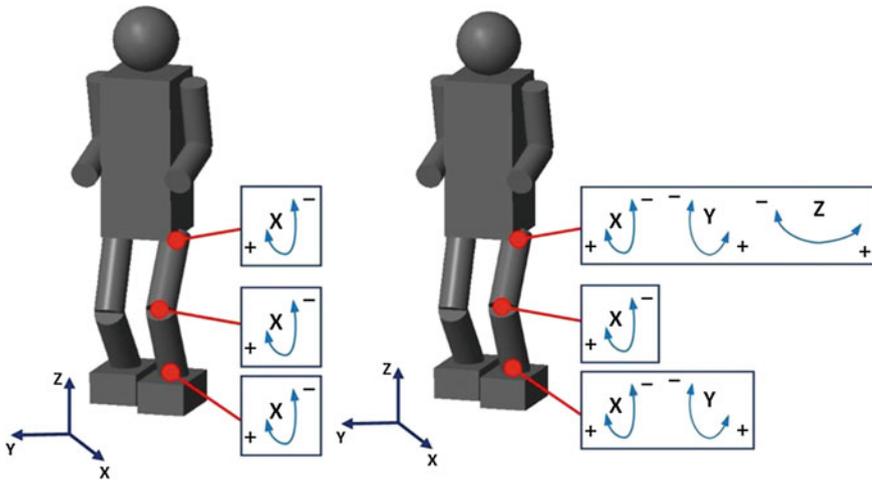
**Artificial neural network** approach uses a neuroadaptive control algorithm to learn the biped robot locomotion, based on the dynamic balance optimization criterion, such as ZMP. This strategy does not use analytical approach, and there is no need to describe all details of robot motion. This approach is used by Boston Dynamics for locomotion control of Atlas robot [16]. Major drawback of neural networks is in its practical use for real-time control applications—as far as a system becomes more complicated it has the exponential growth of the number of parameters and the computational cost becomes too high for real-time usage.

**Central Pattern Generator (CPG)** approach is borrowed from animal locomotion. CPG is capable of producing the rhythms which are associated with each gait. CPG is commonly modeled as a network of identical differential equation systems [17]. Instead of prescribing a trajectory and playing it back on the robot, these approaches focus on the limit cycle behavior of legged locomotion. It is shown that limit-cycle motion control could have better energy efficiency than the trajectory-based control [18]. CPG is a network of neuron oscillators coupled by mutual inhibition, which can be used to control the robot's limb motion, and displays characteristics close to natural ones [19, 20].

**Full-body posture goal** strategy for path planning performs a search through all achievable robot states with a goal to find a collision-free path to the goal posture while keeping dynamic balance. This approach is divided into two phases: the first phase computes a statically-stable, collision-free path, and the second phase smoothens and transforms this path into a dynamically stable trajectory for the entire body [21].

## 4 3D Modelling of Biped Robot in Simulink

Robotics simulators significantly accelerate robotic research, promote advanced designs, help to test new control algorithm and solve tasks for complex problems. We model and simulate robot mechanical systems in Matlab/Simulink environment using



**Fig. 2** 6 DoFs (*left*) and 12 DoFs (*right*) robot models in Matlab/Simulink environment

SimMechanics tool<sup>4</sup> as the mechanical developer module. SimMechanics simulates 3D translational and rotational motion by predicting the system's future from the current state through differential equations of Newtonian dynamics. SimMechanics allows quite simple 3D modelling of a biped robot as a multibody system using blocks which represent bodies, joints, constraints, and force elements; it supports such features as mass distribution and inertia assignments, external forces, trajectory generation etc. Moreover, SimMechanics automatically generates 3D animation which helps to visualize system dynamics.

In this paper we present two simplified models of AR-601M robot: with 6 DoFs and 12 DoFs per legs (Fig. 2, 3DoF and 6DoF per each leg respectively). The model concentrates only on leg's kinematics and dynamics, anchoring all other joints for upper body of the human-like robot model. We use crude approximation of the robot body based on simplified geometrical shapes and mass distribution. However, all main parameters of the models that influence walking dynamics of the robot (such as foot area, leg length, CoM position) precisely correspond to AR-601M kinematical structure and hardware properties.

The following steps were performed in SimMechanics environment:

- Specification of robot's rigid parts geometry and their inertial properties (bodies and links).
- Connecting the robot bodies and links with rotational joints.
- Setting up the ground reaction forces.
- Assigning the reference signals at each joint position.
- Running Simulink solver.
- Performing the simulation visualization.

<sup>4</sup>Multibody Simulation—SimMechanics, [www.mathworks.com/products/simmechanics/](http://www.mathworks.com/products/simmechanics/).

Our current robot model consists of 12 rigid bodies (see, Fig. 2) which are connected to each other with revolute joints for knees, ankles, hips and anchored joints for a neck, elbows and shoulders. We use the geometric primitives for robot bodies simulation such as (1) parallelepipeds for a torso and feet, (2) a sphere as a head, (3) cylinders for all other bodies. Figure 3 shows top level abstraction block diagram of the whole biped robot in SimMechanics, and Table 1 describes all the parameters in details. The next two subsections present the walking primitives approach for a simple 6-DoF legs robot model (3 DoFs per leg) which moves only in sagittal plane and for a 12-DoF legs model (6 DoFs per leg) which can locomote both in sagittal and frontal planes.

### 4.1 The 6 DoFs Robot Model

The simple 6 DoFs robot model (3 DoFs per leg) allows the motion in sagittal plane only. It has single revolutionary joints with one rotational DoF in ankles, knees and hips (Fig. 4). The torso has 3 DoFs in sagittal plane with regard to the global (ground-fixed) coordinate system: 2 translational DoFs in vertical and horizontal directions and 1 rotational DoF.

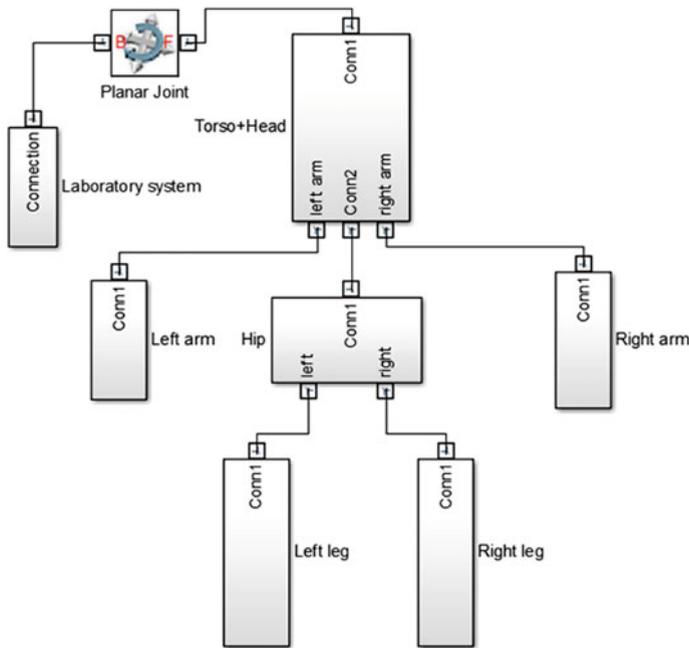


Fig. 3 SimMechanics block diagram of a biped robot: top level of abstraction

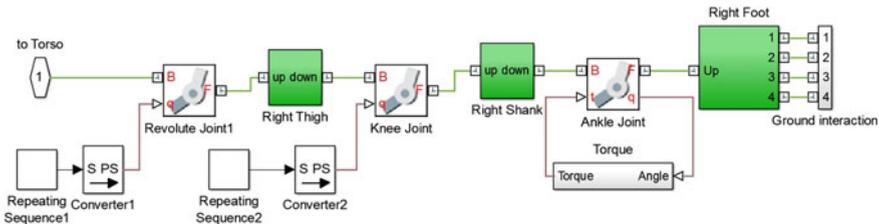
In [2] we considered the 6 DoFs robot locomotion based on walking primitives, i.e. on the gait parts with identical time periods, at the beginning and at the end of which the robot returned to the same positions. The leg motion pattern is the same in each primitive and consists of four phases: (1) raising the left foot; (2) performing the inverted pendulum motion until the left foot touches the ground; (3) moving the right foot forward to make the knee and hip joint’s angles equal to zero; (4) damping forward motion by applying torque in the ankles. In the next step, the same phases are repeated for the right leg. Thus, we realized the robot locomotion by setting four angles for hip and knee joints as inputs and two torques on the ankle joints, damping the robot inertia at the end of each step. The input angles of knee and thigh joints were found empirically and are shown in Figs.5 and 6 respectively.

### 4.2 The 12 DoFs Robot Model

As a step forward from the presented above simple model towards a complete 41 active DoFs AR-601M robot model, we created a newer model with 12 DoFs by

**Table 1** The robot bodies parameters

Torso parameters		Foot parameters	
Width (mm)	248	Width (mm)	160
Length (mm)	242	Length (mm)	254
Height (mm)	544	Height (mm)	106
Mass (kg)	17	Mass (kg)	3
Shank and thigh parameters		Upper/lower arm parameters	
Radius (mm)	50	Radius (mm)	45
Height (mm)	280	Length (mm)	300
Mass (kg)	7	Mass (kg)	3
Head parameters			
Radius (mm)	100		
Mass (kg)	2		



**Fig. 4** SimMechanics block diagram for a leg in the 6 DoFs robot model

adding one rotational DoF in ankle and two rotational DoFs in hip joints (Fig. 2, on the right). This advanced 12 DoFs legs model with 6 DoFs per leg precisely reflects the kinematic structure and the hardware of AR-601M robot legs and implies no artificial restrictions on the motion directions and planes. The top abstraction level SimMechanics block diagram of the robot remains the same as shown in Fig. 3, whereas the leg diagram implies significant changes which are depicted in Fig. 7.

The robot starts the motion from the initial position when the feet are located next to each other. Similar to the simple 6FoFs model we divide each walking primitive into four phases: (1) forward movement of the left foot (from the initial position); (2) waiting until oscillations are damped (the opposite to the 6 DoFs model where no pause before the next motion is required); (3) joining the right leg to the left leg with its forward movement (this way the robot returns into the state which is identical to its initial state, but has its global position incremented for one step); (4) waiting until oscillations are damped.

For the 12 DoFs robot model we cannot build walking primitive pattern based on empirical values of joint angles because of motion complexity. Therefore, at first we define trajectory in Cartesian space and then find joint angle functions, solving the inverse kinematics task. To solve inverse kinematics of 12 joint angles, it is necessary to define 12 Cartesian coordinates: 6 coordinates for a torso and 6 coordinates for a moving foot—3 rotational and 3 translational coordinates for each. We assume that rotational angles of the torso and a swinging foot are always zero, i.e. the foot is parallel to the ground and the torso is always kept vertical, and there are no torso motion and foot swinging in frontal plane which allows to keep the torso at the constant height. This way, foot swinging and the torso trajectories in sagittal plane could be calculated from trigonometric functions. We set the reference frame center in the middle of supporting foot and summarize Cartesian coordinates functions for a moving foot and the torso as follows:

$$x_{foot} = x_0; \quad y_{foot} = \frac{1}{2}a(1 - \cos(\frac{\pi t}{T})); \quad z_{foot} = \frac{1}{2}h(1 - \cos(\frac{2\pi t}{T})); \quad (1)$$

$$x_{torso} = 0.5x_0; \quad y_{torso} = \frac{1}{4}a(1 - \cos(\frac{\pi t}{T})); \quad z_{torso} = z_0; \quad (2)$$

where  $x_0$  is the distance between feet centers,  $a$  is the step length,  $h$  is the maximum height of the moving foot,  $T$  is the step period,  $z_0$  is the height of torso movement.

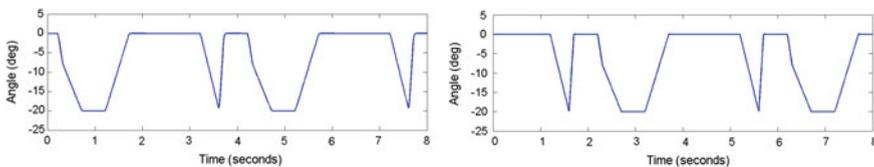
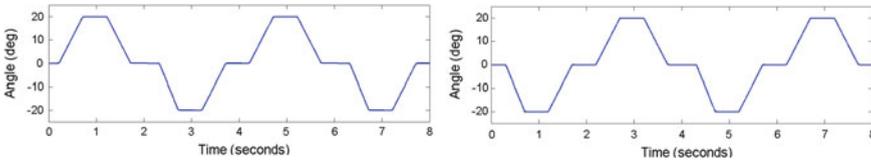


Fig. 5 The input angle for the left knee (on the left) and the right knee (on the right)



**Fig. 6** The input angle for the left hip (on the *left*) and the right hip (on the *right*)

These trajectories are applied in the first phase of the walking primitive pattern, where the left foot moves forward. For the third phase, where the right foot returns to the initial position, we use similar equations with reference frame center in the middle of left foot. Notice that to solve inverse kinematics we anchor supporting foot to the ground. It means that if we apply calculated angle functions to move the robot, its trajectory will be different due to supporting foot lifting. In order to damp oscillations, occurring after each step, we fix angle values at the end of the first and third phases for certain amount of time  $\Delta t$ .

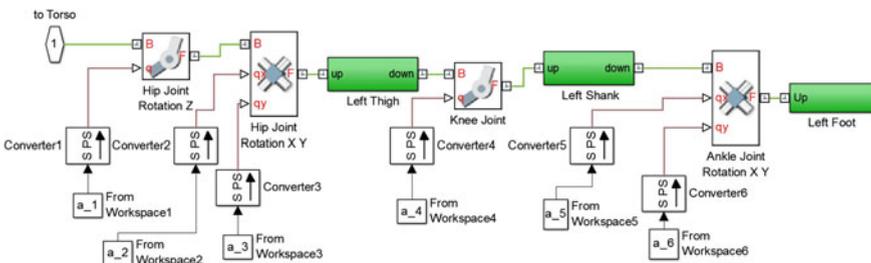
### 4.3 Contact Force Modelling in SimMechanics

The foot-ground interaction is a very important issue in biped robot simulation and should be carefully modeled. Generally, the contact force, which takes place when robot’s feet touch the ground, can be studied in terms of normal and tangential (friction) forces.

**Normal Force.** Two compliant contact models for normal force were considered in our work. The first one is the Linear Spring-Damper model. Then the normal force is given as:

$$F_n = k_n z + b_n \dot{z} \tag{3}$$

where  $z$  is the coordinate of the leg tip ( $y = 0$  corresponds to ground surface),  $k_n$  and  $b_n$  are elastic constant and damping coefficient respectively. The normal force is



**Fig. 7** SimMechanics block diagram for a 6DoF leg within the 12 DoFs robot model

applied only when  $z \leq 0$ , i.e. when the leg touches the ground. Being computationally simple, this approach has some drawbacks: (1) the contact force is not continuous at the beginning of impact due to the damping term character; (2) there is a sticking effect, which holds feet at the lifting moment. The second issue can be solved by limiting  $F_n$  to positive values only, while the first issue can not be solved using linear model. Furthermore, the equivalent coefficient of restitution, which shows the relation of body velocity before and after collision, in linear model depends on body mass and does not depend on body velocity, which is contrary to empirical results presented in [22]. The optimal solution to these problems, proposed in [23], is to replace the Linear Spring-Damper parallel combination with a nonlinear one, changing the contact normal force equation as follows:

$$F_n = k_n z^n + b_n z^n \dot{z} \quad (4)$$

where the power  $n$  is close to one and depends on the surface geometry of the impact. The advantage of this model is that while the computational time is similar, the contact force increases continuously with the impact and coefficient of restitution depends on impact velocity. In our work we used nonlinear model with the power  $n$  equal to one.

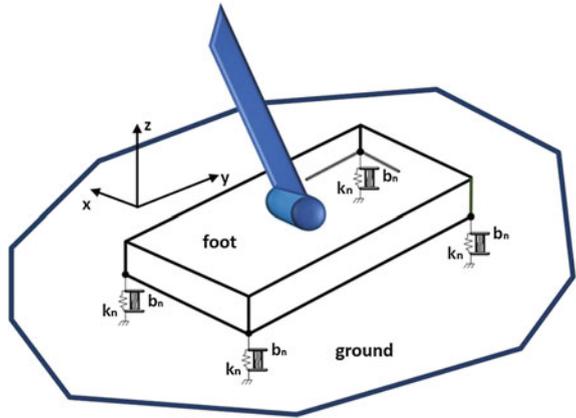
**Friction Force.** In our work tangential forces (friction) were modeled similar to viscous forces and given as:

$$F_t = b_t z \dot{x} \quad (5)$$

where  $b_t$  is tangential damping coefficient. Firstly, notice that force is nonzero only when a motion in tangential direction exists. It means that we do not have sticking, where body is fully at rest. However this motion is negligible, if we take large values for coefficient  $b_t$ . Secondly, tangential force is proportional to penetration depth, which again increases the force with the continuous impact. Equation 5 works perfect at leg landing, whereas at leg raising we cannot provide strict vertical movement and have a horizontal component of velocity, which results in large tangential force. To avoid this effect, the absolute value of tangential force should be limited to  $\mu F_n$ , where  $\mu$  is a friction coefficient. At leg landing  $F_n$  value is large, so the tangential force will be large enough to prevent the robot from sliding. At leg raising  $F_n$  is small enough to avoid sticking. Similarly to normal force, tangential force is applied only when  $z \leq 0$ . The ground coefficients  $k_n$ ,  $b_n$ ,  $b_t$  should be large (see Table 2) to minimize penetrations of feet below ground level ( $k_n$ ) and provide fast damping of contact velocity ( $b_n$ ,  $b_t$ ).

The contact between a supporting foot and ground was modeled by four contact points shown in Fig. 8. The number of contact points can be increased, but it will increase computational cost. We performed simulations with four and eight contact points for each foot but did not detect any significant differences. Therefore, for flat surfaces and rectangular foot shape we used four contact points at the foot's corners that should be enough for suitable foot-ground interaction modelling of a biped robot locomotion.

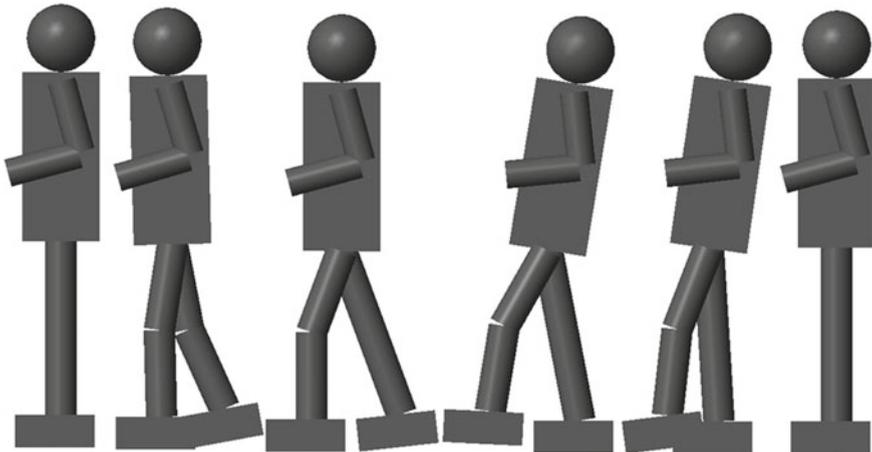
**Fig. 8** Nonlinear spring-damper model between a foot and ground



### 5 Simulation Results

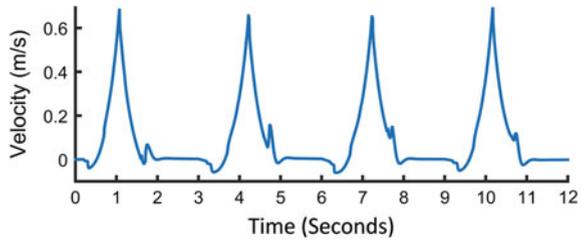
**Table 2** The ground parameters for the foot-ground interaction model

Coefficient	Label	Value (units)
Penetration	$k_n$	1000000 N
Normal damping	$b_n$	10000 (N*s)/m
Tangential damping	$b_t$	1000 (N*s)/m
Friction	$\mu$	10

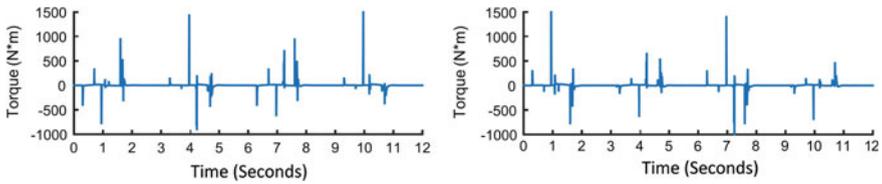


**Fig. 9** A single step of the biped robot. The robot moves from *right* to *left* as the time passes

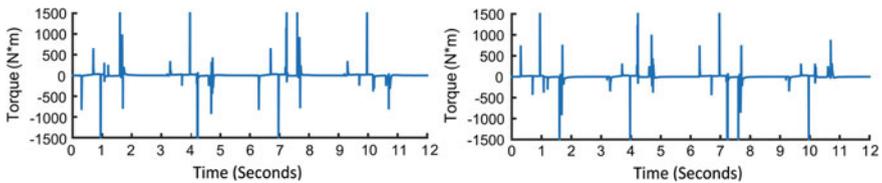
**Fig. 10** Robot torso velocity in sagittal plane



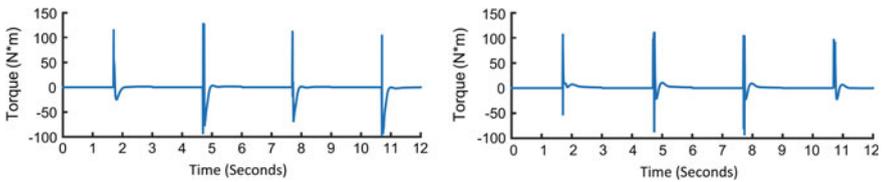
**Results for the 6 DoFs Model.** To solve differential equations described above we used Simulink solver method “ODE 23t”. Figure 9 shows a sequence of the biped robot frames at different time during a single step. The smooth walking of the simulated model illustrates reasonable settings of the angle values for the joints and contact forces within the model. Robot torso center velocity of 0.3–0.4 m/s in sagittal plane is shown in Fig. 10, demonstrating the periodical body movement with the 3 s time period. At the beginning of each period the body velocity is zero. Torque values in the hip and knee joints from the simulation are shown in Figs. 11 and 12, which can help to estimate motor characteristics. The simulation shows that maximum torque



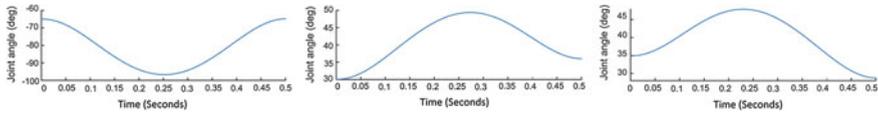
**Fig. 11** Torques in the left (on the *left*) and the right (on the *right*) knee



**Fig. 12** Torques in the left (on the *left*) and the right (on the *right*) hip



**Fig. 13** Torques in the left (on the *left*) and the right (on the *right*) ankle



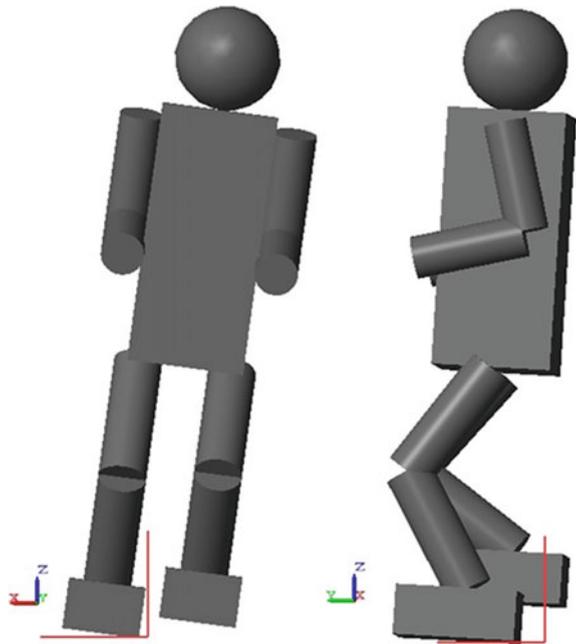
**Fig. 14** Angle functions for knee (*left*), hip (*center*), and ankle (*right*) joints

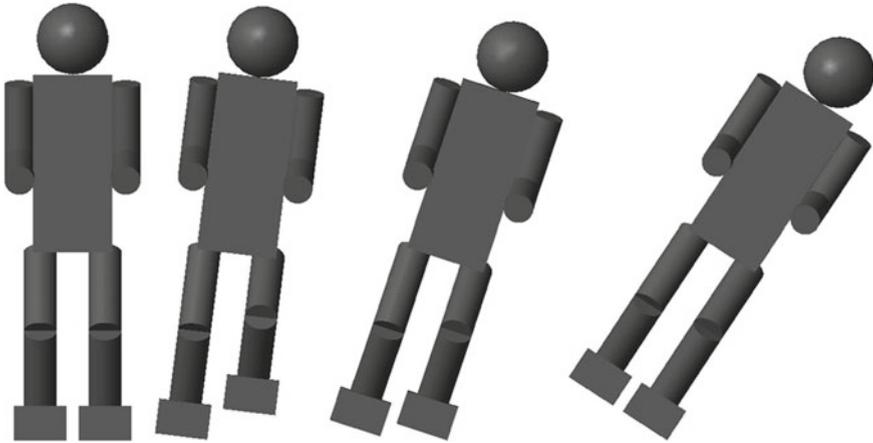
should be approximately 1500 N\*m in the thigh joints and the knee joints. At the same time discontinuous peaks in calculated torque values were detected, which are caused by foot-ground collisions. Torque values in ankles are shown in Fig. 13. As we expected, the torque was applied to damp robot’s inertia only in the last step phase. In our work the duration of one walking primitive is 3 s, the time between stepping phases  $\Delta t = 1.3$  s.

**Results for the 12 DoFs Model.** At first, angle functions for each joint were obtained by solving inverse kinematics equations. Figure 14 shows angle functions for a knee, a hip and an ankle joints for their rotations around X axis. The step parameters were defined as step length  $a = 0.1$  m, the maximum height of the moving foot  $h = 0.1$  m, the step period  $T = 0.5$  s. The time between stepping phases  $\Delta t$  was chosen equal to 1 s.

Figure 15 shows frontal and side views of the robot stepping. The supporting foot rotates around its inner edge because robot’s CoM leaves supporting foot’s

**Fig. 15** Frontal and side view of robot stepping





**Fig. 16** Robot falling down in the frontal plane with the step period of 1 s

**Table 3** Maximum motor characteristics for AR-601M robot leg joints

	Maximum torque (N*m)	Maximum acceleration (deg/s <sup>2</sup> )
Hip joints	50	500
Knee joint	50	700
Ankle joints	20	500

Courtesy of Android Technics company

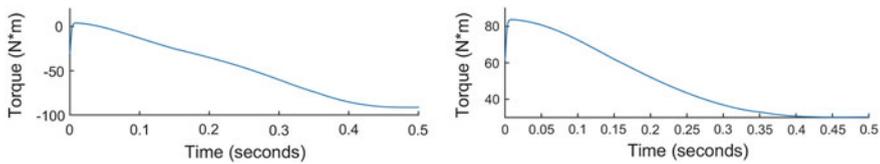
area. Therefore, at the end of step we have impact of stepping foot with ground. The same happens at the third phase, when the other foot moves forward to its initial position. Notice that if we set larger values for step period  $T$  and height of torso movement  $z_0$ , the robot falls down (Fig. 16). Therefore, robot should make steps reasonably fast. From AR-601M robot motor characteristics (see Table 3 for maximum rotation acceleration and torques for leg joint motors) we can estimate if walking primitive pattern would guarantee the real AR-601M locomotion stability or not. For this reason, we simulated the 12 DoFs robot model locomotion with different step parameters, presenting the simulation results in Table 4: stable robot gait or falling. Table 4 shows that the robot walking can be stable at the maximum step period  $T = 0.7$  s with step length  $a = 0.15$  m, whereas the maximum height of stepping foot  $h$  is achieved at 0.1 m with the height of torso movement  $z_0 = 0.85$  m.

For the robot stable locomotion case (the first row in Table 4) the maximum torque values are applied to ankle and knee joints of the supporting leg (Fig. 17). However, for the case of fast walking the robot needs maximum torque of nearly 90 N\*m in ankle joint, which is much higher than the motor can provide.

Thus, the simulation results with 12 DoFs robot model clearly demonstrated that walking primitive approach could not support fast biped locomotion of the real AR-601M robot due to leg’s motor torque limitations.

**Table 4** Influence of walking primitive pattern parameters on the AR-601M robot model locomotion stability

$\alpha$ (m)	$h$ (m)	$z_0$ (m)	$T$ (s)	Results
0.1	0.1	0.85	0.5	Stable gait
0.2	0.1	0.85	0.5	Failure
0.15	0.1	0.85	0.5	Stable gait
0.2	0.1	0.8	0.5	Failure
0.2	0.1	0.75	0.5	Failure
0.2	0.1	0.75	0.25	Failure
0.15	0.1	0.85	0.75	Failure
0.15	0.1	0.75	0.75	Failure
0.15	0.05	0.75	0.7	Stable gait

**Fig. 17** Torques in ankle (on the *left*) and knee (on the *right*) joints of the supporting leg

## 6 Conclusions and Future Work

In this paper we presented Russian biped robot AR-601M and its locomotion modelling in Simulink environment using walking primitives approach. We considered two robot models: with 6 and 12 Degrees of Freedom (DoFs) per legs, using the same walking strategies with robot inertia damping and a pause between the steps. While the simplified 6 DoFs model is constrained to sagittal plane motions, the 12 DoFs model precisely reflects the hardware of AR-601M robot legs. The locomotion algorithm utilizes position control and involves inverse kinematics computations for the joints. The resulting biped robot locomotion was dynamically stable for both models at the step length up to 0.15 m and the step time of 0.7 s with relatively long damping pauses between the steps.

The simulation allowed us to adjust the models with proper parameters' selection which then have demonstrated good results of the walking primitives approach. However, this approach could not be further applied for the real AR-601M robot as AR-601M cannot provide sufficient actuator torques for such fast steps. This points at a non one-to-one correspondence of successful simulations and reality, emphasizing crucial necessity of adequate simulations prior to real world experimentation. Our further research aims to complete a 41 DoFs precise model of AR-601M robot in Simulink and Gazebo environments with preview control methods and genetic algorithms techniques, and to perform exhaustive testing under ROS framework.

Next, the successful solutions will be transferred onto a real robot hardware in order to verify their performance in the real world.

**Acknowledgments** This research has been supported by Russian Ministry of Education and Science as a part of Scientific and Technological Research and Development Program of Russian Federation for 2014-2020 years (research grant ID RFMEFI60914X0004) and by Android Technics company, the industrial partner of the research.

## References

1. Wright, J., Jordanov, I.: Intelligent approaches in locomotion. In: International Joint Conference on Neural networks, pp. 1–8 (2012)
2. Khusainov, R., Shimchik, I., Afanasyev, I., Magid, E.: Toward a human-like locomotion: modelling dynamically stable locomotion of an anthropomorphic robot in simuLink environment. In: 12th International Conference on Informatics in Control, Automation and Robotics, pp. 141–148 (2015)
3. Afanasyev, I., Sagitov, A., Magid, E.: ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment. In: Advanced Concepts for Intelligent Vision Systems, Springer International Publishing, pp. 273–283 (2015)
4. Stephens, B., Atkeson, C.: Modeling and control of periodic humanoid balance using the linear biped model. In: 9th IEEE-RAS International Conference on Humanoid Robots, pp. 379–384 (2009)
5. Erbatur, K., Seven, U.: An inverted pendulum based approach to biped trajectory generation with swing leg dynamics. In: 7th IEEE-RAS International Conference on Humanoid Robots, pp. 216–221 (2007)
6. Siciliano, B., Khatib, O.: Biped robots in the ZMP scheme. In: Springer Handbook of Robotics, pp. 371–377. Springer, Berlin (2008)
7. Garofalo, G., Ott, C., Albu-Schaffer, A.: Walking control of fully actuated robots based on the bipedal SLIP model. In: IEEE International Conference on Robotics and Automation, pp. 1456–1463 (2012)
8. Ketelaar, J., Visser, L., Stramigioli, S., Carloni, R.: Controller design for a bipedal walking robot using variable stiffness actuators. In: IEEE International Conference on Robotics and Automation, pp. 5650–5655 (2013)
9. Suzuki, S., Furuta, K., Pan, Y., Hatakeyama, S.: Biped walking robot control with passive walker model by new VSC servo. In: IEEE American Control Conference, vol. 1, pp. 107–112 (2001)
10. Iribe, M., Osuka, K.: A designing method of the passive dynamic walking robot via analogy with the phase locked loop circuits. In: IEEE International Conference on Robotics and Biomimetics, pp. 636–641 (2006)
11. Erbatur, K., Kurt, O.: Humanoid walking robot control with natural ZMP references. In: IEEE Annual Conference on Industrial Electronics, pp. 4100–4106 (2006)
12. Vukobratovich, M., Borovac, B.: Zero-moment point: thirty five years of its life. *Int. J. Hum. Robot.* **01**, 157–173 (2004)
13. Magid, E., Tsubouchi, T., Koyanagi, E., Yoshida, T.: Building a search tree for a pilot system of a rescue search robot in a discretized random step environment. *J. Robot. Mechatron.* **23**, 567–581 (2011)
14. Denk, J., Schmidt, G.: Synthesis of walking primitive databases for biped robots in 3D-environments. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 1343–1349 (2003)

15. Magid, E., Ozawa, K., Tsubouchi, T., Koyanagi, E., Yoshida, T.: Rescue robot navigation: static stability estimation in random step environment. In: *Simulation, Modeling, and Programming for Autonomous Robots*, pp. 305–316 (2008)
16. Atmeh, G., Ranatunga, I., Popa, D., Subbarao, K., Lewis, F., Rowe, P.: Implementation of an adaptive, model free, learning controller on the Atlas robot. In: *IEEE American Control Conference*, pp. 2887–2892 (2014)
17. Pinto, C.A., Superior, I., Porto, E., Golubitsky, M.: Central pattern generators for bipedal locomotion. *J. Math. Biol.* **53**, 474–489 (2006)
18. Fukuoka, Y., Kimura, H., Cohen, A.H.: Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concept. *Int. J. Robot. Res.* **22**, 187–202 (2003)
19. Feng, K., Chew, C.M., Hong, G.S., Zielinska, T.: Bipedal locomotion control using a four-compartmental central pattern generator. In: *IEEE International Conference on Mechatronics and Automation*, vol. 3, pp. 1515–1520 (2005)
20. Wu, Q., Liu, C., Zhang, J., Chen, Q.: Survey of locomotion control of legged robots inspired by biological concept. In: *Science in China Series F: Information Sciences*, vol. 52, pp. 1715–1729 (2009)
21. Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H.: Motion planning for humanoid robots under obstacle and dynamic balance constraints. In: *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 692–698 (2001)
22. Goldsmith, W.: *Impact: The Theory and Physical Behaviour of Colliding Solids*. Courier Dover Publications (2001)
23. Hunt, K.H., Crossley, F.R.E.: Coefficient of restitution interpreted as damping in vibroimpact. *ASME J. Appl. Mech.* **2**, 440–445 (1975)

**Part IV**  
**Signal Processing, Sensors, Systems**  
**Modelling and Control**

# Freezing Method Approach to Stability Problems

Artur Babiarz, Adam Czornik and Michał Niezabitowski

**Abstract** In the work we present the stability criteria for difference linear equations obtained by the freezing method. The results are presented for discrete version of damped oscillator equation. We also report stability criteria for another type of equations obtained by this method. Our results are illustrated by numerical examples.

**Keywords** Difference equation · Linear system · Oscillator equation · Stability · Freezing method

## 1 Introduction

In the literature, there are known research work describing a model called the simple harmonic oscillator [1, 2]. The simple harmonic oscillator is expressed by the formula:

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = 0, \quad (1)$$

where:

- $y(t)$  is a measure of the displacement from the equilibrium point at a given time;
- $m$  is the mass;

---

A. Babiarz · A. Czornik · M. Niezabitowski (✉)  
Faculty of Automatic Control, Electronics and Computer Science,  
Institute of Automatic Control, Silesian University of Technology,  
Akademicka 16 Street, 44-100 Gliwice, Poland  
e-mail: [michal.niezabitowski@polsl.pl](mailto:michal.niezabitowski@polsl.pl)  
URL: <http://www.ia.polsl.pl>

A. Babiarz  
e-mail: [artur.babiarz@polsl.pl](mailto:artur.babiarz@polsl.pl)

A. Czornik  
e-mail: [adam.czornik@polsl.pl](mailto:adam.czornik@polsl.pl)

- $k$  is the spring parameters;
- $c$  is the friction parameter.

The generalization of the equation (1) is a damped linear oscillator defined by the following formula:

$$\ddot{y}(t) + a(t)\dot{y}(t) + \omega^2 y(t) = 0, \quad (2)$$

where:

- the spring constant  $\omega$  is positive;
- the damping coefficient  $a(t)$  is continuous and nonnegative for  $t \geq 0$ .

The Eq. (2) is one of the most famous model using to describe many physical phenomena [3]. Moreover, this equation is also known [4] as:

$$\ddot{y}(t) + a(t)\dot{y}(t) + y(t) = 0. \quad (3)$$

In [5], the damped impulsive equation is analysed and authors studied the stability of this model. Instead, authors of [6] researched the stability of oscillator equation with delays. In this case, the mathematical model was described by the first-order linear delay impulsive differential equation.

In [7] researchers focuses on several oscillation criteria for a related neutral first-order difference equation with delay. Authors of [8–10] consider stability criteria of the first-order difference equation with various kind of delays. They use, inter alia, the Lyapunov function method.

In our research, we concentrate on stability of second-order difference equation which we obtain by a certain discretization of the equation (3).

For the discretization of the equation (3), it has been used the so-called forward difference operator and denoted by  $\Delta$ . Its formal definition is as follows [11]:

**Definition 1** The first forward difference operator is expressed by following formula

$$\Delta x(k) = x(k + 1) - x(k) \quad (4)$$

and the second forward difference operator  $\Delta^2$  is defined as

$$\Delta^2 x(k) = \Delta(\Delta x(k)) = \Delta x(k + 1) - \Delta x(k). \quad (5)$$

Using the Definition 1, the Eq.(3) can be discretized in the following way (see [11], p. 3):

$$\Delta(\Delta(y(n))) + a(n)\Delta(y(n)) + y(n) = 0. \quad (6)$$

Subsequently, the forward difference is used and we get:

$$y(n + 2) + y(n + 1)(a(n) - 2) - y(n)(a(n) - 2) = 0. \quad (7)$$

$$y(n + 1) + y(n)(a(n - 1) - 2) - y(n - 1)(a(n - 1) - 2) = 0. \tag{8}$$

Substituting

$$q(n) = a(n - 1) - 2, \tag{9}$$

the Eq. (8) will obtained the following form:

$$y(n + 1) = -q(n)y(n) + q(n)y(n - 1). \tag{10}$$

The main objective of our further consideration is the stability problem for the Eq. (10) with initial conditions  $y(0) = y_0, y(1) = y_1$ , where  $(q(n))_{n \in \mathbb{N}}$  is a sequence of real numbers. We will say that Eq. (10) is asymptotically stable if for all initial conditions  $(y_0, y_1)$  the corresponding solution tends to zero. In the paper we will use the following notation:

- $\mathbb{R}^{s \times s}$  is the set of all  $s$  by  $s$  real matrices;
- $I_s$  is the identity matrix of size  $s$ ;
- $\|\cdot\|$  is the Euclidean norm in  $\mathbb{R}^s$  and the induced operator norm;
- $A^T$  means the transposition of the matrix  $A$ ;
- $\rho(A)$  is the spectral radius of matrix  $A$ .

The idea, which we use to obtain stability criteria for the Eq. (10) is called in the literature [12–16] the freezing method. For the discrete linear time-varying system given by

$$x(n + 1) = D(n)x(n) \tag{11}$$

the main conception of the freezing method is to freeze the matrices  $D(n)$  and consider system

$$x(l + 1) = D(n)x(l). \tag{12}$$

The stability of all Eq. (12) usually does not imply the stability of the system (11) as shows below example.

*Example 1* Let us consider the following discrete linear time-varying system with

$$D(n) = \begin{cases} B & \text{for } n \text{ even} \\ C & \text{for } n \text{ odd} \end{cases},$$

where

$$B = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & \frac{1}{3} \end{bmatrix}, \quad C = \begin{bmatrix} \frac{1}{2} & \frac{3}{4} \\ \frac{3}{4} & -\frac{1}{4} \end{bmatrix}.$$

Then the systems (12) has the following form

$$x(l + 1) = B x(l) \quad \text{or} \quad x(l + 1) = C x(l). \tag{13}$$

The eigenvalues of the matrices  $B$  and  $C$  are equal to  $\frac{1}{2}$ ,  $\frac{1}{3}$  and  $\frac{1}{8} + \frac{3}{8}\sqrt{5}$ ,  $\frac{1}{8} - \frac{3}{8}\sqrt{5}$ , respectively. Each of systems in (13) is stable because their spectral radii are less than one, i.e.

$$\rho(B) = \frac{1}{2} < 1, \quad \rho(C) = \frac{1}{8} + \frac{3}{8}\sqrt{5} < 1.$$

On the other hand the spectral radius of matrices product  $BC$  is greater than one, i.e.

$$\rho(BC) = \frac{\sqrt{187}}{24} + \frac{11}{24} > 1.$$

Thus

$$\|D(2n)D(2n-1)\dots D(2)D(1)\| = \|(BC)^n\| \xrightarrow{n \rightarrow \infty} \infty$$

and therefore the system (11) is unstable.

The stability criteria obtained for system (11) by the freezing method are usually a combination of the assumption of the stability of the system (12) and constraints on variation of the parameter  $D(n)$  of (11). The above-mentioned approach is presented in [13–15].

## 2 Preliminaries

Let us introduce the following notation

$$A(n) = \begin{bmatrix} -q(n) & q(n) \\ 1 & 0 \end{bmatrix} \quad (14)$$

and

$$B(q) = \begin{bmatrix} -q & q \\ 1 & 0 \end{bmatrix}. \quad (15)$$

Moreover for the solution  $(y(n))_{n \in \mathbb{N}}$  of (10) denote

$$x(n) = \begin{bmatrix} y(n) \\ y(n-1) \end{bmatrix} \quad (16)$$

for  $n = 1, 2, \dots$ . With this notation Eq. (10) may be rewritten in the following form

$$x(n+1) = A(n)x(n) \quad (17)$$

with initial condition

$$x(0) = \begin{bmatrix} y(1) \\ y(0) \end{bmatrix}. \quad (18)$$

If for all initial conditions  $x(0)$  the solution of (17) tends to zero, then we will call (17) asymptotically stable. In the paper we will use the following known facts.

**Lemma 1** [17]. *If for (17) there exists a function  $V : \mathbb{N} \times \mathbb{R}^2 \rightarrow [0, \infty)$  such that*

1.

$$\|x\|^2 \leq V(n, x) \leq C_1 \|x\|^2$$

2.

$$V(n, x(n)) - V(n, x(n+1)) \leq -C_2 \|x(n)\|^2$$

for all  $x \in \mathbb{R}^2$ ,  $n \in \mathbb{N}$  and certain positive  $C_1, C_2$ , then (17) is asymptotically stable. The function  $V$  is called the Lyapunov function.

**Lemma 2** [18]. *For a matrix  $A \in \mathbb{R}^{s \times s}$  the following conditions are equivalent:*

1. *matrix  $A \in \mathbb{R}^{s \times s}$  has all eigenvalues in the open unit circle;*
2. *for each positive definite matrix  $Q \in \mathbb{R}^{s \times s}$  there exists a positive definite matrix  $P \in \mathbb{R}^{s \times s}$  such that the following Lyapunov equation is satisfied*

$$A^T P A - P = -Q; \quad (19)$$

3. *there are positive definite matrices  $P, Q \in \mathbb{R}^{s \times s}$  such that (19) is satisfied. Moreover if  $A \in \mathbb{R}^{s \times s}$  has all eigenvalues in the unit circle then the solution of (19) is given by*

$$P = \sum_{k=0}^{\infty} (A^T)^k Q A^k. \quad (20)$$

**Lemma 3** [19]. *For any matrix  $A \in \mathbb{R}^{s \times s}$ ,  $A = [a_{ij}]_{i,j=1,\dots,s}$  we have*

$$\|A\| \leq s \max_{i,j=1,\dots,s} |a_{ij}|. \quad (21)$$

The next lemma provides a stability condition for matrices  $B(q)$  and gives us an estimation of  $\|B(q)\|$ .

**Lemma 4** *All the eigenvalues of  $B(q)$  lies in the open unit circle if and only if  $q \in (-1, \frac{1}{2})$  and then*

$$\|B(q)\| < \frac{\sqrt{5} + 1}{2}. \quad (22)$$

*Proof* Consider the Lyapunov equation (19) with  $A = B(q)$  and  $Q = I_2$  then the solution is given by

$$P = \frac{1}{2q^2 + q - 1} \begin{bmatrix} 2q - 2 & 2q^2 \\ 2q^2 & 2q^3 + q - 1 \end{bmatrix}. \quad (23)$$

It is easy to check that the solution is positive definite if and only if  $q \in (-1, \frac{1}{2})$ . The first conclusion follows now from point 3 of Lemma 2. We have

$$\|B(q)\| = \sqrt{q^2 + \frac{1}{2}\sqrt{4q^4 + 1} + \frac{1}{2}} \tag{24}$$

and using the standard method of calculus we may verify that the greatest value of the function  $f : [-1, \frac{1}{2}] \rightarrow R$ ,

$$f(x) = \sqrt{x^2 + \frac{1}{2}\sqrt{4x^4 + 1} + \frac{1}{2}} \tag{25}$$

is

$$f(-1) = \frac{\sqrt{5} + 1}{2}. \tag{26}$$

The proof is completed.

The below presented result includes an estimation for norm of the solution of Lyapunov equation (19) for  $A = B(q)$ .

**Lemma 5** *If*

$$\varepsilon \in \left(0, \frac{3}{4}\right), \tag{27}$$

$$q \in \left(-1 + \varepsilon, \frac{1}{2} - \varepsilon\right) \tag{28}$$

and  $P(q)$  is the solution of the Lyapunov equation (19) with  $A = B(q)$  and  $Q = I_2$ , then

$$1 \leq \|P(q)\| \leq \frac{8 - 4\varepsilon}{-2\varepsilon^2 + 3\varepsilon}. \tag{29}$$

*Proof* The most left inequality follows from the formula (20). As we noticed in the proof of Lemma (4)  $P(q)$  is given by (23).

Since for

$$q \in \left(-1 + \varepsilon, \frac{1}{2} - \varepsilon\right), \tag{30}$$

$$|2q^2 + q - 1| > -2\varepsilon^2 + 3\varepsilon, \tag{31}$$

therefore

$$\|P(q)\| \leq \frac{1}{-2\varepsilon^2 + 3\varepsilon} \left\| \begin{bmatrix} 2q - 2 & 2q^2 \\ 2q^2 & 2q^3 + q - 1 \end{bmatrix} \right\|. \tag{32}$$

Using Lemma 3 we have

$$\|P(q)\| \leq \frac{2}{-2\varepsilon^2 + 3\varepsilon} \cdot \left[ \max_{-1+\varepsilon < q < \frac{1}{2}-\varepsilon} \{|2q - 2|, 2q^2, |2q^3 + q - 1|\} \right] = \frac{8 - 4\varepsilon}{-2\varepsilon^2 + 3\varepsilon} \quad (33)$$

what ends the proof.

### 3 Main Result

In this section we apply results of Lemmas 4 and 5 in freezing method and in this way we obtain the main result of our paper which is contained in the next theorem.

**Theorem 1** *If for certain  $\varepsilon \in (0, \frac{3}{4})$  and  $\eta > 0$  the sequence  $(q(n))_{n \in \mathbb{N}}$  satisfies the following two conditions:*

1.

$$q(n) \in \left(-1 + \varepsilon, \frac{1}{2} - \varepsilon\right) \quad (34)$$

2.

$$|q(n) - q(n - 1)| \leq \frac{(1 - \eta)(-2\varepsilon^2 + 3\varepsilon)^2}{\sqrt{2}(\sqrt{5} + 1)(8 - 4\varepsilon)^2} \quad (35)$$

then the Eq. (10) is asymptotically stable.

*Proof* Consider the solution  $P(n)$  of (19) with  $A = A(n - 1)$  and  $Q = I_2$ . We will show that

$$V(n, x) = x^T P(n)x \quad (36)$$

is the Lyapunov function for (17). From Lemma 5 we know that

$$\|x\|^2 \leq V(n, x) \leq \alpha \|x\|^2, \quad (37)$$

where

$$\alpha = \frac{8 - 4\varepsilon}{-2\varepsilon^2 + 3\varepsilon}. \quad (38)$$

Let us estimate

$$R(n) = P(n + 1) - P(n). \quad (39)$$

We have

$$A^T(n)R(n)A(n) - R(n) = -Q(n), \quad (40)$$

where

$$Q(n) = (A^T(n) - A^T(n-1))P(n)A(n) + A^T(n-1)P(n)(A(n) - A(n-1)) \quad (41)$$

and according to (20)

$$R(n) = Q(n) + \sum_{k=1}^{\infty} (A^T(n))^k Q(n) A^k(n). \quad (42)$$

Because

$$(A^T(n))^k Q(n) A^k(n) \leq \|Q(n)\| (A^T(n))^k A^k(n) \quad (43)$$

and by Lemmas 4 and 5

$$\|Q(n)\| \leq 2 \|A(n) - A(n-1)\| \alpha \frac{\sqrt{5} + 1}{2}, \quad (44)$$

then from (42) we have

$$\begin{aligned} R(n) &\leq Q(n) + \|Q(n)\| \sum_{k=1}^{\infty} (A^T(n))^k A^k(n) \\ &\leq \|Q(n)\| I + \|Q(n)\| \sum_{k=1}^{\infty} (A^T(n))^k A^k(n) \\ &= \|Q(n)\| \left[ I + \sum_{k=1}^{\infty} (A^T(n))^k A^k(n) \right]. \end{aligned} \quad (45)$$

By the definition of  $P(n)$  we know that

$$P(n) = I + \sum_{k=1}^{\infty} (A^T(n))^k A^k(n). \quad (46)$$

From (45) and (46) we obtain

$$\|R(n)\| \leq \|Q(n)\| \|P(n)\|. \quad (47)$$

Using (29) and (44) we may estimate  $R(n)$  as follows

$$\begin{aligned} \|R(n)\| &\leq \|A(n) - A(n-1)\| \alpha^2 (\sqrt{5} + 1) \\ &= \sqrt{2} |q(n) - q(n-1)| \alpha^2 (\sqrt{5} + 1) \leq 1 - \eta. \end{aligned} \tag{48}$$

The last inequality implies that

$$V(n+1, x(n+1)) - V(n, x(n)) \leq -\eta \|x(n)\|^2. \tag{49}$$

Inequalities (37) and (49) show that  $V(n, x)$  is the Lyapunov function for (17). By Lemma 1 we conclude that (17) is asymptotically stable what implies that (10) is asymptotically stable. The proof is completed.

Next example is a numerical illustration of our theoretical result.

*Example 2* Consider Eq.(10) with

$$q(n) = \frac{\sin(\ln(\ln(n+12)))}{r}, \tag{50}$$

where  $r > 0$ . Using Theorem 1 we will find the values of  $r$  such that the conditions (34) and (35) are satisfied with  $\varepsilon = 0.39$  and certain  $\eta > 0$ , i.e.

$$q(n) \in (-0.61, 0.11) \tag{51}$$

and

$$|q(n) - q(n-1)| \leq (1 - \eta) \cdot 3.9494 \times 10^{-3} \tag{52}$$

with  $\eta > 0$ . According to Lagrange theorem

$$|q(n) - q(n-1)| = |f'(c)|, \tag{53}$$

where  $c \in (n-1, n)$  and

$$f(x) = \frac{\sin(\ln(\ln(x+12)))}{r}. \tag{54}$$

Since

$$\frac{d}{dx} \frac{\sin(\ln(\ln(x+12)))}{r} = \frac{1}{\ln(x+12)} \frac{\cos(\ln(\ln(x+12)))}{12r + rx}, \tag{55}$$

then

$$|f'(c)| \leq \frac{1}{(12r + rc) \ln(c+12)}. \tag{56}$$

It is easy to verify that for all  $c > 0$  and  $r > 8.5$  there exists  $\eta > 0$  such that

$$\frac{1}{(12r + rc) \ln(c + 12)} < (1 - \eta) \cdot 3.9494 \times 10^{-3}. \tag{57}$$

It is also clear that for  $r > 10$  the condition (51) is satisfied. Finally we conclude from Theorem 1 that Eq. (10) with  $q(n)$  given by (50) is asymptotically stable.

### 4 Possible Extensions of the Presented Approach

The general idea of freezing method is to obtain exact condition for stability of the system (11) by the easily analyzable analogous properties of solutions of frozen systems (12). Historically the first paper devoted to this method is [20]. For the general discrete linear time-varying systems this method was applied by Desoer [14]. In this paper we applied this method to the system with coefficient in particular form given by (14). Using this special form of the coefficients we obtained less restrictive condition for the stability than the condition coming from the general form of the method. Even though, our approach maybe applied to another type of equations and an example taken from [21] is given below.

**Theorem 2** [21]. *If for certain  $\varepsilon \in (0, \frac{1}{2})$  and  $\eta > 0$  the sequence  $(q(n))_{n \in \mathbb{N}}$  satisfies the following two conditions:*

1.

$$q(n) \in (\varepsilon, 1 - \varepsilon)$$

2.

$$|q(n) - q(n - 1)| \leq \frac{5(1 - \eta)}{2\sqrt{5} + 10} \frac{(\varepsilon^3 - 4\varepsilon^2 + 3\varepsilon)^2}{(4 - 2\varepsilon)^2}$$

then the equation

$$y(n + 1) = y(n) - q(n) y(n - 1)$$

is asymptotically stable.

Our approach may be also applied to obtain condition for stability of discrete linear inclusions. Let  $\Sigma = \{A_i : i \in I\}$  be a bounded set or real  $s$ -by- $s$  matrices. Moreover,  $D$  is the set of all sequences of elements of  $I$ , i.e.

$$D = \{d = (d(0), d(1), \dots) : d(i) \in I\}.$$

By a discrete linear inclusion  $DLI(\Sigma)$ , which will be denoted by

$$x(j+1) \in A_{d(j)}x(j), \quad j = 0, 1, 2, \dots,$$

we will understand the set of all sequences  $(x(j))_{j \in \mathbb{N}}$ ,  $x(j) \in \mathbb{R}^s$  such that

$$x(j+1) = A_{d(j)}(x(j)), \quad j = 0, 1, 2, \dots$$

for certain  $d \in D$ . Each such a sequence  $(x(j))_{j \in \mathbb{N}}$  will be called trajectory of  $DLI(\Sigma)$  and  $x(0)$  will be called initial value of this trajectory. Furthermore,  $DLI(\Sigma)$  is called stable if and only if for all trajectory  $(x(j))_{j \in \mathbb{N}}$  we have

$$\lim_{j \rightarrow \infty} x(j) = 0.$$

**Theorem 3** [22]. *Let us fix  $r > 6.5$  and define*

$$q(n) = \frac{\sin(\ln(\ln(n+12)))}{r}$$

and

$$\Sigma = \left\{ A_n = \begin{bmatrix} -q(n) & q(n) \\ 1 & 0 \end{bmatrix} : n \in \mathbb{N} \right\},$$

then  $DLI(\Sigma)$  is stable.

## 5 Conclusions

In this paper we applied the freezing method to obtain condition for the asymptotic stability for discrete version of damped linear oscillator. Typical situation in this approach is that the stability condition is a combination of requirements about eigenvalues of certain matrices and variation of the parameters. We were able to present the conditions in the terms of the original parameters of the Eq. (10) only. Moreover, we pointed out two other examples, where our approach may be used.

The objective of further works will be estimation of different numerical characteristic of the discrete linear time-varying systems [23–29] using the freezing method.

**Acknowledgments** The research presented here were done by the authors as parts of the projects funded by the National Science Centre granted according to decisions DEC-2014/13/B/ST7/00755, DEC-2012/07/N/ST7/03236 and DEC-2012/05/B/ST7/00065, respectively. The calculations were performed with the use of IT infrastructure of GeCONiI Upper Silesian Centre for Computational Science and Engineering (NCBiR grant no POIG.02.03.01-24-099/13).

## References

1. Buonomo, A., di Bello, C.: Asymptotic formulas in nearly sinusoidal nonlinear oscillators. *IEEE Trans. Circ. Syst. I: Fundam. Theory Appl.* **43**, 953–963 (1996)
2. Hyland, D., Bernstein, D.: The majorant Lyapunov equation: a nonnegative matrix equation for robust stability and performance of large scale systems. *IEEE Trans. Autom. Control* **32**, 1005–1013 (1987)
3. Sugie, J., et al.: Global asymptotic stability for oscillators with superlinear damping. *J. Dyn. Diff. Equat.* **24**, 777–802 (2012)
4. Smith, R.: Asymptotic stability of  $x' + a(t)x' + x = 0$ . *Oxford J.: Q. J. Math.* **12**, 123–126 (1961)
5. Graef, J., Karsai, J.: On the asymptotic behavior of solutions of impulsively damped nonlinear oscillator equations. *J. Comput. Appl. Math.* **71**, 147–162 (1996)
6. Yan, J., Zhao, A.: Oscillation and stability of linear impulsive delay differential equations. *J. Math. Anal. Appl.* **227**, 187–194 (1998)
7. Zhang, G., Cheng, S.: Oscillation criteria for a neutral difference equation with delay. *Appl. Math. Lett.* **8**, 13–17 (1995)
8. Kulikov, A.: Stability of a linear nonautonomous difference equation with bounded delays. *Russ. Math.* **54**, 18–26 (2010)
9. Yu, J.: Asymptotic stability for a linear difference equation with variable delay. *Comput. Math. Appl.* **36**, 203–210 (1998)
10. Yu, Y., Cheng, S.: A stability criterion for a neutral difference equation with delay. *Appl. Math. Lett.* **7**, 71–74 (1994)
11. Agarwal, R., et al: *Discrete Oscillation Theory*. Hindawi Publishing Corporation (2005)
12. Bylov, B., et al.: *Theory of Lyapunov Exponents and Its Applications to Stability Theory*. Nauka, Moscow (1966)
13. Czornik, A., Nawrat, A.: On new estimates for Lyapunov exponents of discrete time varying linear systems. *Automatica* **46**, 775–778 (2010)
14. Desoer, C.: Slowly varying discrete system  $x_{i+1} = A_i x_i$ . *Electron. Lett.* **6**, 339–340 (1970)
15. Gil, M., Medina, R.: The freezing method for linear difference equations. *J. Differ. Equ. Appl.* **8**, 485–494 (2001)
16. Medina, R.: Absolute stability of discrete-time systems with delay. *Adv. Differ. Equ.* 1–14 (2008)
17. Khalil, H.: *Nonlinear Systems*. Prentice Hall (1995)
18. Gajic, Z., et al: *Lyapunov Matrix Equation in System Stability and Control: Mathematics in Science and Engineering*. Academic Press (1995)
19. Horn, R., Johnson, C.: *Matrix Analysis*. Cambridge University Press (1985)
20. Bohl, P.: Über differentialgleichungen. *J. Reine Angew. Math.* **144**, 284–313 (1914)
21. Babiarz, A.: An approach to stability problem of the second order difference equation. In: 20th International Conference on Methods and Models in Automation and Robotics, pp. 99–103 (2015)
22. Czornik, A., Jurgaś, P., Niezabitowski, M.: Estimation of the joint spectral radius. In: Gruca, D.A., Czachórski, T., Kozielski, S. (eds.) *Man-Machine Interactions 4*. Volume 391 of *Advances in Intelligent Systems and Computing*. Springer International Publishing (2015)
23. Babiarz, A., Czornik, A., Niezabitowski, M.: On the number of upper Bohl exponents for diagonal discrete time-varying linear system. *J. Math. Anal. Appl.* **429**, 337–353 (2015)
24. Czornik, A., Mokry, P., Nawrat, A.: On the sigma exponent of discrete linear systems. *IEEE Trans. Autom. Control* **55**, 1511–1515 (2010)
25. Czornik, A., Mokry, P., Nawrat, A.: On the exponential exponents of discrete linear systems. *Linear Algebra Appl.* **433**, 867–875 (2010)
26. Czornik, A., Nawrat, A., Niezabitowski, M.: On the Lyapunov exponents of a class of second-order discrete time linear systems with bounded perturbations. *Dyn. Syst.* **28**, 473–483 (2013)
27. Czornik, A., Niezabitowski, M.: Lyapunov exponents for systems with unbounded coefficients. *Dyn. Syst.* **28**, 140–153 (2013)

28. Czornik, A., Niezabitowski, M.: On the spectrum of discrete time-varying linear systems. *Nonlinear Anal.: Hybrid Syst.* **9**, 27–41 (2013)
29. Czornik, A., Niezabitowski, M.: Alternative formulae for lower general exponent of discrete linear time-varying systems. *J. Franklin Inst.* **352**, 399–419 (2015)

# Scanning Techniques with Low Bandwidth Radar for Robotic Mapping and Localization

Paul Fritsche and Bernardo Wagner

**Abstract** This article presents two methods for setting up a scanning unit with low bandwidth radar sensors with a wide opening angle of the main beam and evaluates their suitability for robotic mapping. Both approaches, namely the lateration and the ASR technique, base upon a rotary joint and provide a two-dimensional scan. The relevant theory behind both methods and considerations on erroneous influences is described in the first part of this paper. The focus of the second part is laying on application in occupancy grid and feature mapping, which will be presented through experiments.

**Keywords** FMCW radar · Lateration · ASR · Robotic mapping · SLAM

## 1 Introduction

The following article evaluates two scanning methods, for mapping purposes in mobile robotics. The first method is based on a rotating mono-static radar network, which determines the positions of objects inside the scanning area via a continuously running lateration algorithm. The second method is based on rotating radar sensors with an angle offset and a determination of the positions of objects through the Amplitude Sensing Ratio (ASR) technique, which is also called lobe switching.

A precise model of the environment is essential in many areas of mobile robotics and builds the fundament for localization and navigation. Commonly, popular sensors like laser-scanners, sonar-sensors and stereo-cameras have established themselves as state of the art for most tasks in mobile robotics. Nevertheless, radar sensors frequently appear in field and rescue robotics [1, 18], but are seldom used to perform

---

P. Fritsche (✉) · B. Wagner  
RTS, Leibniz Universität Hannover, Appelstr. 9A, 30167 Hannover, Germany  
e-mail: fritsche@rts.uni-hannover.de  
URL: <http://www.rts.uni-hannover.de/>

B. Wagner  
e-mail: [wagner@rts.uni-hannover.de](mailto:wagner@rts.uni-hannover.de)

tasks like mapping and localization. Radar can penetrate certain materials, basically non-conductors, which provides advantages in dusty, foggy, rainy or other harsh environments. But, limited resolution, noisy data, influence of optical effects like refraction and reflection make the application in mobile robotics challenging.

The use of radar sensors in mobile robotics is challenging but not impossible. The first appearance of radar sensors in the robotic community is tracing back to the Australian Centre for Field Robotics in the early nineties, where fundamental work on feature based SLAM algorithms (EKF-SLAM) in combination with radar was explored [4]. Because of their limited resolution and other aforementioned drawbacks, radar sensors are not very suitable to use in indoor environments. Nevertheless, [6] were investigating the use of radar sensors in an industrial environment and [12] in an office. As far as we can see, most radar sensor principles in mobile robotics are based on mechanical beam-forming. Usually, the radar beam is focussed via a parabolic antenna and panned mechanically over the environment. Electrical beam-forming through phased array antennas is not seen very often in mobile robotics rather in automotive systems of the car industry.

Besides beam-forming techniques, position estimation can be achieved through lateration, which is a common technique in radar networks for aircraft surveillance. Lateration is a measurement method, where the position of a point target is calculated of distance information from  $n$  Sensors with known locations. The term trilateration refers to the measurement of three distances to define the position of an object (in contrast to triangulation, where three angles are used to calculate an object's position). The estimation of surfaces with ultra-wide band (UWB) radar networks has been studied experimentally in lab environments, especially with lateration by [13], envelopes of spheres by [10] and inverse boundary scattering algorithms by [15]. But, we can not see a link to the field of occupancy grid mapping in mobile robots, where laser scanners are dominating. ASR techniques on a rotary joint are common techniques in ground-based radar systems for air-traffic control [2] and are used for beam sharpening.

For our experiments, we use frequency modulated continuous wave (FMCW) radar sensors, which provide distance measurements but no information of the angle between objects and sensor. The sensors work in 24 GHz ISM band and accordingly are limited in Germany to a bandwidth  $B$  of 250 MHz, which corresponds to a theoretical distance resolution  $\Delta d$  of 0.6m in dependency of the speed of light  $c_0$  (See Eq. 1).

$$\Delta d = \frac{c_0}{2B} \quad (1)$$

The resolution  $\Delta d$  of a radar sensor is equal to its minimum detection range. The availability of sensors with a high resolution depends on national and international bandwidth regulations. For example, an UWB channel between 22–26,65 GHz has been closed in 2013, but is moved to 79 GHz for automotive purposes recently [16, p. 20].

A radar's resolution is its capability to distinguish objects. If the difference between the radial distances of two or more objects to the sensor is less than its resolution, then the sensor merges the two or more distance information to one. Additionally, the detection of objects depends on their radar cross section (RCS) and the background noise of the environment.

This article is organized as follows. In Sect. 2, we present a short overview, how position estimation via lateration and ASR techniques are solved. Besides the theory and introduction to common terms, we explain the ghost target and non-point target problems in radar networks which are based on lateration. Additionally, we describe influences of errors related to the power and range measurement accuracy of radar sensors. The reader who is familiar to these topics might go directly to Sect. 3, where the sensor principles are described in the beginning. Further in this section, we will present our experiments, which results are discussed in Sect. 4. A brief summation of the obtained knowledge is given in Sect. 5.

## 2 Materials and Methods

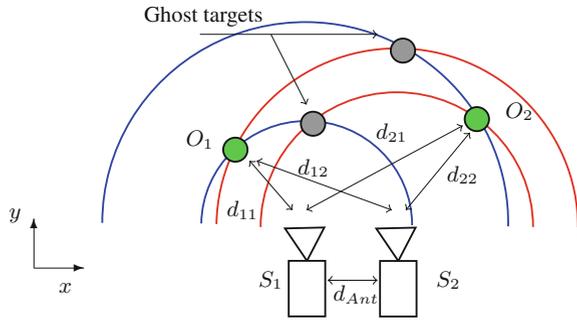
Estimating the position of an object with a radar network can be solved by standard lateration methods or ASR techniques. In order to define an object's position in two-dimensional space, at least two sensors are necessary. In case of lateration, two radii from two range measurements at different positions can break down the object's position to two possible locations. Although two radii have two intersections, only one location is plausible due to the antenna's direction. In case of the ASR technique, the position of an object can be estimated through the difference in the receiver power at two antennas which are located at the same position but pointing into different directions. In this paper, we investigate the usability of both methods in combination with rotating scanning unit to generate occupancy grid maps.

### 2.1 Principle of the Lateration Technique

Lateration is a measurement principle to estimate the positions of points with distance informations to a known locations. If the distance to an unknown point is given, then this point must be laying on a radius (two-dimensional case) around the location. If two locations are known, then two radii result in an intersection, which is the position of the point. Figure 1 demonstrates the basic of operation of the lateration principle.

For a two-dimensional space,  $n$  objects  $O_i$  ( $i = 1..n$ ) and  $m$  sensors  $S_j$  ( $j = 1..m$ ) result in  $m \cdot n$  equations of circle. The euclidean distances between the sensor positions  $S_j(x_{S_j}, y_{S_j})$  and the object positions  $O_i(x_{O_i}, y_{O_i})$  are given by the following equations:

**Fig. 1** The position of objects  $O_i$  (green) can be estimated via the distances  $d_{ij}$ , which are measured from sensors at the locations  $S_j$ . Ghost targets (grey) represent a geometrical ambiguity



$$\begin{aligned}
 (x_{S1} - x_{O_i})^2 + (y_{S1} - y_{O_i})^2 &= d_{i1}^2 \\
 (x_{Sj} - x_{O_i})^2 + (y_{Sj} - y_{O_i})^2 &= d_{ij}^2 \\
 &\vdots \\
 (x_{Sm} - x_{O_i})^2 + (y_{Sm} - y_{O_i})^2 &= d_{im}^2
 \end{aligned} \tag{2}$$

The distance between an Object  $O_i$  and an Sensor  $S_j$  is defined as  $d_{ij}$ . The general description of the linear system of equations can be achieved through subtracting the last equation ( $j = m$ ) of Eq. 2 from all other equations ( $j = 1 \dots m - 1$ ) [17, p. 8].

$$\begin{aligned}
 &\begin{pmatrix} 2 \cdot (x_{S1} - x_{Sm}) & 2 \cdot (y_{S1} - y_{Sm}) \\ 2 \cdot (x_{S2} - x_{Sm}) & 2 \cdot (y_{S2} - y_{Sm}) \\ \vdots & \vdots \\ 2 \cdot (x_{Sm-1} - x_{Sm}) & 2 \cdot (y_{Sm-1} - y_{Sm}) \end{pmatrix} \cdot \begin{pmatrix} x_{O_i} \\ y_{O_i} \end{pmatrix} \\
 &= \begin{pmatrix} x_{S1}^2 - x_{Sm}^2 + y_{S1}^2 - y_{Sm}^2 - d_{i1}^2 - d_{im}^2 \\ x_{S2}^2 - x_{Sm}^2 + y_{S2}^2 - y_{Sm}^2 - d_{i2}^2 - d_{im}^2 \\ \vdots \\ x_{Sm-1}^2 - x_{Sm}^2 + y_{Sm-1}^2 - y_{Sm}^2 - d_{i,m-1}^2 - d_{im}^2 \end{pmatrix} \\
 &A \cdot \mathbf{o}_i = \mathbf{d}_i
 \end{aligned} \tag{3}$$

In reality, every sensor outputs measurement values with errors. The difference from the true value occur due to limited accuracy and resolution. Hence, the system of equations does not result in one single solution if it is overdetermined. The system of equation gets overdetermined if the radar network has at least two more sensor than the dimension of its measurement space. Commonly, overdetermined systems of equation with no single solution get resolved through regression. The most popular solution is the least mean square method [17, p. 8] [8, p. 39].

Like it can be seen in Fig. 1, ghost targets can appear in radar networks. Ghost targets represent a wrong data association. As shown in Eq. 2, four objects can be calculated if two objects are placed in front of two sensors. Ghost target appear if the sensor's resolution is smaller than the half antenna distance ( $d_{Ant}$ ). A detailed derivation of the ghost target cases can be seen in [14].

In order to resolve the ghost target problem, [9] present the button-up data association method. In two-dimensional space, at least three radar sensors are required. In order to distinguish ghost objects from real objects, the observation area in front of the sensor network is discretized into a finite set of possible object positions. Then, a simple minimum distance calculation is done. For each point, an error value  $E(x, y)$  can be calculated from the square of the minimum distance of the point to the sensor  $S_j$  minus the distance  $d_{ij}$  between object  $O_i$  and sensor  $S_j$ , summarized over all  $m$  sensors (See Eq. 4).

$$E(x, y) = \sum_{j=1}^m \left[ \min_{d_{ij} \in OL_i} (d_{ij} - d(x, y)) \right]^2 \quad (4)$$

This calculation results in the lowest error values at points that are closest to the real objects. Afterwards, a threshold distinguishes likely ghost target from real objects.

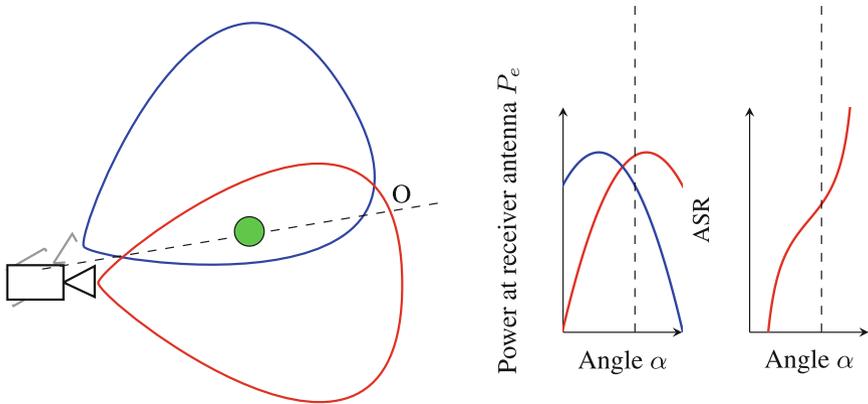
## 2.2 Principle of the ASR Technique

Every location in front of an antenna, is connected to a different antenna gain factor. For example, if we walked on a radius around a loudspeaker with closed eyes, we would have a feeling when we would be walking directly in front of it, because then the sound appears to be louder. A similar effect is used for the ASR technique. If we point two radar antennas in slightly different directions, then the power at the receiver antennas would not be equal, due to different antenna gains.

The measured power at the receiver antenna  $P_e$  of an object depends on the angle  $\alpha$  between antenna and object. Two sensors, which are facing into different directions, but placed at the same location, measure the same distance but different power  $P_e$ . If a function for the power in dependency of the angle position of the object is given, then the object's position can be estimated via the ASR function (See Fig. 2).

The angle of an object inside the observation area can be estimated through the Amplitude Sensing Ratio (ASR). In a two-dimensional case, the ASR for the azimuth is defined by a delta signal ( $\Delta$ ) and a sum signal ( $\Sigma$ ), which are derived from the powers at the receiver antennas [3, pp. 34–35].

$$ASR = \frac{\Delta}{\Sigma} = \frac{P_{e1} - P_{e2}}{P_{e1} + P_{e2}} \quad (5)$$



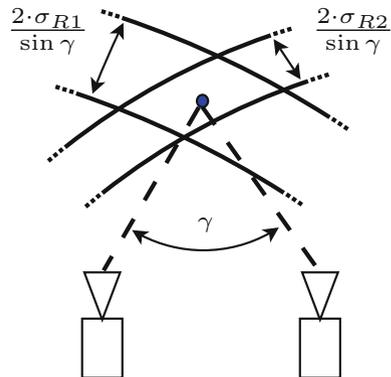
**Fig. 2** The position of an object O inside the observation area can be estimated via the difference of power of into different direction facing receiver antennas

In order to obtain information in three-dimensional space, the elevation needs to be estimated through an additional sensor beam.

### 2.3 Consideration on Erroneous Influences

Estimating the position of objects with lateration and ASR requires sensors with very high accuracy regarding the range and power measurement. Nevertheless, every sensor has a measurement error. In case of lateration, the maximum position measurement error  $\sigma_{PD}$  can be approximated by the maximal range measurement error  $\sigma_{R_{max}}$  of all sensors and the angle  $\gamma$  (See Eq. 6). The range measurement accuracy is defined by the root-mean-square (rms) measurement error  $\sigma_R$  [5, p. 167]. Figure 3 clarifies the relation between the range measurement error  $\sigma_{Ri}$ , the angle  $\gamma$  and the position measurement error  $\sigma_{PD}$  graphically.

**Fig. 3** Area of ambiguity, which is caused by the two rms-errors  $\sigma_{R1}$  and  $\sigma_{R2}$



If the ratio between distance of the object and the antenna distance gets higher, then the rhomb-shaped area of ambiguity gets wider. In Fig. 5, it can be seen that the area of ambiguity of a point target appears almost to be a line, if the distance to the object is much larger comparing to the antenna distance. From Eq. 6, it can be seen that the accuracy of a lateration based radar network is getting worse at the sides of the sensor network, where  $\gamma$  is approaching zero.

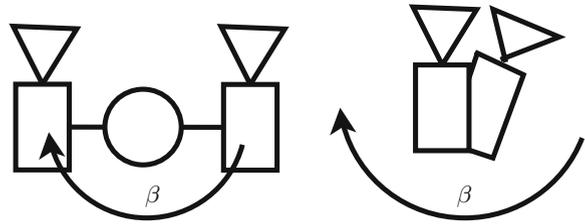
$$\sigma_{PD} \approx \frac{\sigma_{R_{max}}}{\sin \gamma} \quad (6)$$

The rms error  $\sigma_R$  depends basically on the signal-to-noise ratio of the received signal. The signal-to-noise ratio is higher and results accordingly in a better accuracy, if the RCS of an object is higher. Consequently, our radar principle results in better position estimations for objects with high RCS. But, objects with a high RCS are entering the observation area of a rotating scan earlier from the sides than objects with a lower RCS, hence an object with high RCS suffers more from the position estimation error  $\sigma_{PD}$ .

Besides the range measurement accuracy, the resolution of a radar sensor has important impact on the reliability of the scan results. A radar sensor will not distinguish two point targets, if they are inside a so called resolution cell. For example, a radar with a resolution of 1 m, can not differ between two or more objects which are inside a band, with the wide of 1 m, around the sensor and would output the detection of only one range value somewhere between those two objects. The lateration technique results only in correct position estimation if all sensors of the radar network are measuring the same distance to the same point. But, single point targets are very rare in standard environments. Usually, every sensor of a radar network measures the distance to a different point, which results in wrong position estimations.

The precondition for the ASR technique is the placement of sensors at the same location. The closer the antennas are placed to each others, the more reliable is the result and the same centres of reflection of a target can be assumed in a ASR radar network. Unfortunately, the accuracy and resolution of the receiver power  $P_e$  can not be defined, because it is depending on the RCS of the object as well (See radar equation). There is a fluctuation of the RCS, which can be explained by the Swerling Models. A. Ludloff explains in [11, pp. 3–14] how the fluctuation can be modeled. The model is based on the idea that one radar target exists of multiple reflector elements, which are distributed over the volume of the target. The model assumes the reflector elements to be isotropic and with the same RCS and neglects the effects of reflection or shadows among themselves. Through overlapping of reflected radar waves on this multiple isotropic reflector elements, phase differences result in complex interferences. This model explains the appearance of high fluctuation of the RCS (and accordingly the receiver power), even if the aspect angle is changed only slightly. To sum it up, an exact estimation of the RCS, even of standard geometries, is difficult and fluctuation effects disturb the reliability of the position estimation results via ASR techniques.

**Fig. 4** The *left* drawing shows the scanner with the lateration setup. The *right* drawing presents the ASR setup. Both setups are based on a rotating joint and its angle position  $\beta$



### 3 Experiments

The following section describes first experiments with both sensor principles. Therefore, two scans at the same location but with different sensor principle have been performed. Both sensor principles, the lateration technique and the ASR technique, have been set up on a rotating platform. Our radar sensors work with a center frequency of 24 GHz with a bandwidth of 250 MHz, which is the reason for the low resolution. Furthermore, our radar's beam has a wide opening angle of approximately  $40^\circ$ . The setup of both experiments can be seen in Fig. 4.

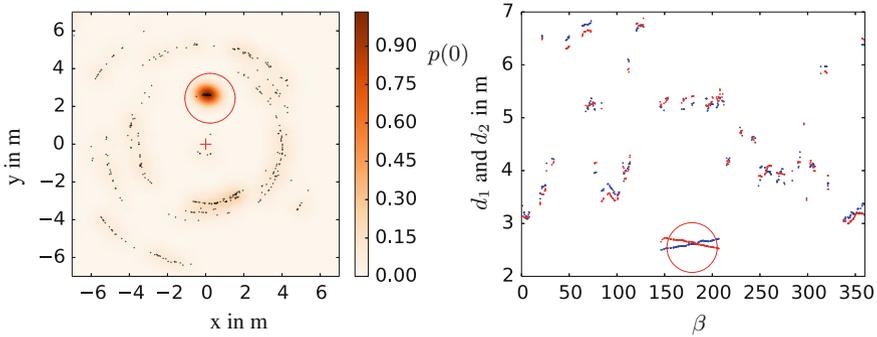
Our first scans have been performed in an indoor environment. In order to have a landmark, we were placing a point target (an aluminium corner reflector), at the same height like our sensor unit, inside our scan area. A serial of scans of a hallway with both principles has been performed for further evaluation on occupancy grid mapping. A feature based mapping approach is investigated in an outdoor environment.

#### 3.1 Single Scan with the Lateration Technique

In order to evaluate the sensor principle, we were performing first scans of standard objects in an indoor environment. The goal of the first experiment is to find out about error influences in our sensor principle. As mentioned before, a limited resolution can be problematic in an indoor environment, like our office. There are metallic radiators, steal-beams behind the walls, computer towers and many other objects that can have a RCS huge enough of being detected by the radar sensor. The first scan results are presented in Fig. 5.

The measurement contains the accumulation of five  $360^\circ$  scans with a step size of  $0.7^\circ$ . Not every measurement cycle leads to a successful position estimation. A successful position estimation can be processed if both sensors detect an object.

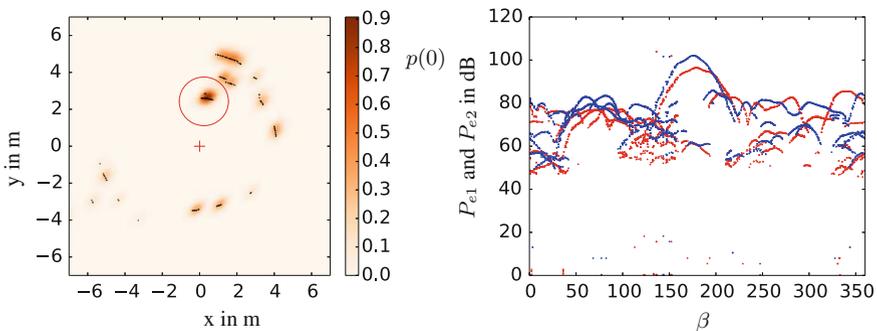
The probability of occurrence of two objects, with a smaller difference of their radial distances to the sensor than the radar's resolution, is high, hence we can rarely trust our scan results, if performed in an indoor environment. For fundamental research, our radar sensors with an resolution of approximately 0.6 m are sufficient.



**Fig. 5** *Left* This diagram depicts a *top view* on our scanning area. The *red cross* represents the location of the sensor unit. The point target results in an accumulation of distance values. The remaining spread is caused by the sensor measurement error  $\sigma_R$ . Accordingly, the point target has the highest probability  $p(0)$  for its correct position estimation. *Right* This diagram presents the distance informations of the two sensors during a  $360^\circ$  scan of the rotating platform. It can be seen that the corner reflector is the only trustful point target in our office environment (See *red circle*). The characteristics of the distances of the two sensors  $d_1$  and  $d_2$  is caused by the rotation of the platform, where one sensor is approaching and the other sensor gets more far away during a rotation. The distances of non point target do not have a symmetric characteristic

### 3.2 Single Scan with the ASR Technique

For our investigation on the ASR technique, we were using exactly the same scene and same sensors. Instead of placing the sensor at two different positions, the ASR technique works the best if both sensors are placed close as possible, but with a small shift regarding the antenna direction. Our scan results are presented in Fig. 6. The



**Fig. 6** *Left* This diagram presents a 2D scan of our office environment via the ASR technique. The *red cross* shows the location of the sensor unit. *Right* This diagram depicts the power at the receiver antennas  $P_{e1}$  and  $P_{e2}$  during a  $360^\circ$  scan. The non-point target effect does not effect the results. The main error influence is caused by the low resolution and the wide beam size of the radar sensor. An angle estimation of objects through the ASR in an traditional way is impossible, because of the non-uniform distribution of the power values

measurement contains the accumulation of five 360° scans with a step size of 0.7° as well. The distribution of the powers at the receiver antennas is not suitable for a traditional ASR based position estimation, because the high amount of objects does not allow to develop regression functions for the ASR of our experiment. Nevertheless, we can assume to have an object perpendicular in front of the sensor unit if the ASR is close to zero. The ASR method requires, besides calibration of the range measurement of the sensors an calibration of the antenna directions.

### 3.3 Gridmapping with Low Bandwidth Radar

In order to compare the suitability for robotic grid mapping of both sensor principles, a serial of scans has been recorded of a hallway. To avoid influences of control and odometry errors of our robots, all scans have been performed at known poses (See Fig. 7).

The locations of the sensor unit have been chosen under the consideration of the minimal detectable distance of the radar sensors, which is equal to their resolution. The minimal detectable distance is a reason, why radar sensors with low resolution are only suitable for outdoor environments with larger scale.

Like mentioned before, a 2D experiment is performed in a 3D environment, hence metallic objects with rectangular shape elements, like office lamps, get layered into the map as well. Occupancy grid maps, which are obtained via classical inverse sensor model [7], are presented in Fig. 8.

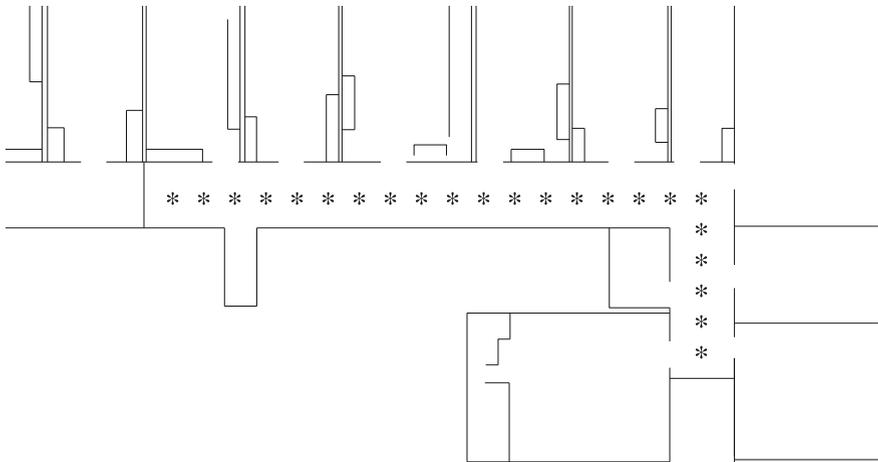
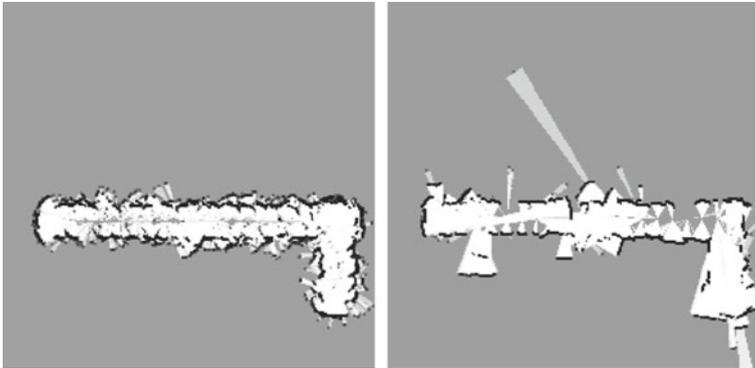


Fig. 7 Ground truth of the office environment with scan positions of the sensor unit



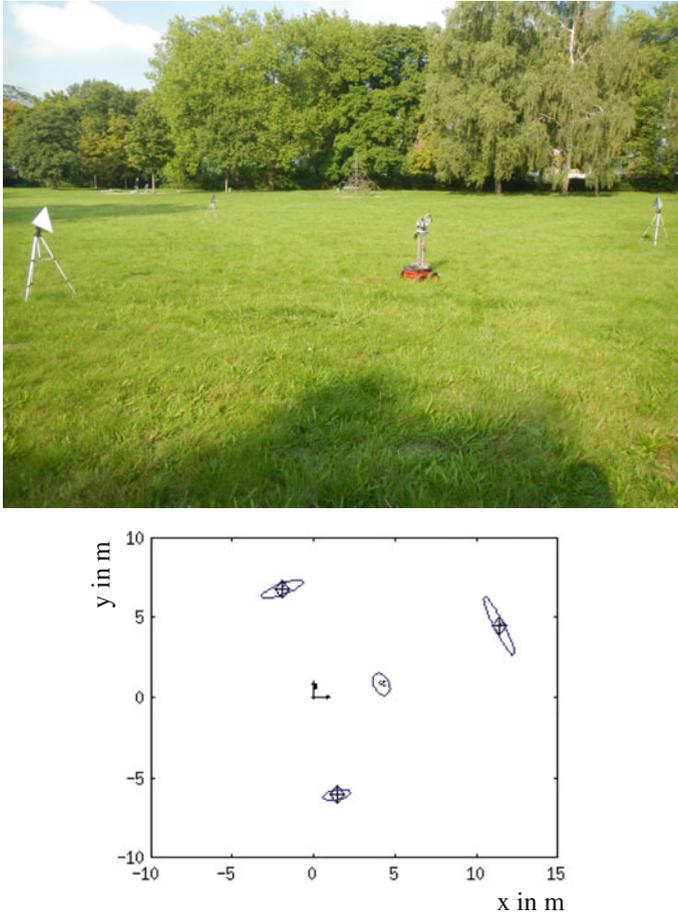
**Fig. 8** *Left* This occupancy grid map is built from raw data of the lationation technique. The wide spread of the sensor data is caused by non point target situations. *Right* This map is obtained from of the ASR technique

### 3.4 Feature Mapping with Low Bandwidth Radar

In Sect. 3.3, first mapping approaches with known poses of an indoor environment were investigated. But, the estimation of the current position of a robot via dead reckoning suffers under the summation of errors, which are caused by slip between the ground and the wheels of the robot. Even a well calibrated robot odometry with the best error model will have a not acceptable displacement after a while. Accordingly, a simultaneous localisation during the map building is necessary. Since localisation in a grid map seems to be not solvable with our scanner setups, because of not enough measurement points of a full scan and many measurement errors, a feature map based solution is the only working way for a full SLAM cycle.

We were performing an experiment on a grassland (See Fig. 9), where we placed three corner reflector in order to have landmarks. As a scanner setup we used the lationation technique. The landmark extraction is done via a simple nearest neighbour tracking, which detects a landmark after it has been tracked a certain number of times, in order to reduce outliers.

The limited number of landmarks and the slow scan speed makes the localization via a rigid transform to three landmarks not suitable. Accordingly, the extended kalman filter slam, which is well known in robotics, is chosen, because it can provide an enhancement of the position estimation of the robot and the landmarks, even if only one landmark is observed.



**Fig. 9** *Left* Grassland with three corner reflectors. *Right* Three detected landmarks and the estimated robot position with their error ellipses

### 4 Results and Discussion

This section will give an interpretation of the obtained results of the experiments, which have been described in Sect. 3.

Figure 5 presents the results of one single scan inside an office environment, where a point target has been placed. The position of the point target gets estimated very well and the remaining spread of the estimated points is caused by the range measurement error  $\sigma_{R_i}$  of the sensors. Tests with different distances between corner reflector and sensor unit have proven, that a rhomb-shaped area of ambiguity is achieved. This has been explained for a static case in Fig. 3. The position of non point targets gets not estimated very well. The sensors measure distances to different points. This results

in a non symmetric characteristics of the distance values  $d_1$  and  $d_2$ . Consequently, the lateration algorithm calculates the wrong positions.

Figure 6 presents the result of a single scan with the ASR technique. The corner reflector gets the highest accumulation of detected object locations, like in case of the lateration technique. Theoretically, the characteristics of the receiver powers  $P_{e1}$  and  $P_{e2}$  supposed to have a phase difference equal to the angle shift of the antenna directions. But, the fact that we can not place both sensors in exactly the same point leads to the fact that again we can not measure exactly the same point target. Furthermore, we can not guarantee both antenna diagrams to be exactly the same due to fabrication tolerances. Nevertheless, we can assume a position of an object, if the ASR is close to zero.

In order to evaluate if the lateration and ASR technique are suitable for robotic mapping, we built two occupancy grid maps with an inverse sensor model [20, pp. 279–300] which we applied on the raw data that has been recorded during a scan of a hallway (See Fig. 7). The wide of the hallway is approximately 2 m and it has a curve at 20 m. Figure 8 displays both results. The ASR technique results in a quite good map. Consequently, we see possibilities to map even indoor environments with radar sensors of low resolution, but a smaller opening angle of the main beam would enhance the result. The minimum detection range, which is equal to the resolution of the sensor, should be considered (See Eq. 1). To enhance the result of the lateration technique, more sensors should be used. In general, both principles suffer from bad resolution of the radar sensors. Optical effects like double reflections inside a narrow hallway have a negative effect on the methods as well.

In Fig. 9, the three extracted landmarks and the current robot position during a drive on a grassland can be seen. The EKF-SLAM can still keep the robot's position, even when the position estimation of the odometry is drifting away.

## 5 Conclusion

Robust localization and navigation in hazardous and tough environments are still a difficult issue in field robotics research. Dust, rain, fog or inadequate illumination are conditions, which make popular sensors, such as laser scanners or cameras, not suitable. Radar overcomes the aforementioned difficulties.

In this article, we were investigating two new scanning methods for mobile robotics and took a closer look on failure influences. We were focusing on three influences. First, the range measurement error of the sensor itself. Second, the influence of wrong position estimation due to non point targets regarding the lateration technique. Third, we investigated if the received power of the receiver antenna is reliable for position estimation, in an environment with multiple targets. We discovered that the influence of non point targets has a huge influence, especially in a setup with only two sensors.

There exists several mapping algorithms. An overview is given by Thrun in [19, p. 7]. Thrun introduces algorithms, which are suitable for mapping with unknown robot poses (SLAM). In this article, we focus on occupancy grid mapping with known

poses, and feature based SLAM. Our proposed scanning methods are suitable for occupancy grid mapping with a classical inverse sensor model, but are not suitable for localisation in a grid map.

The proposed radar-based scanning methods are an alternative to mechanical and electrical beam-forming methods. Mechanical beam-forming techniques require an antenna and electrical beam-forming techniques need phase array radars. Although no antenna construction is required, our methods needs more than one sensor.

From one single 360°—scan of a radar-scanner, which pivots mechanically a focused beam over a surrounding, a more continues distribution of the measurement can be expected. Our proposed methods base are suitable for antennas with a very large beam width. The lateration technique is recording more than one measurement of an object during one scan rotation, which raises the possibility of a correct detection of a landmark. An advantage over techniques with focused beams is the possibility to perform 3D scans as well, which would be mechanically complicated in case of mechanical beam-forming techniques and is only known in combination with electrical beam-forming radars. Unfortunately, the lateration technique suffers more from bad accuracy and resolution, wrong calibration or asynchronism of measurements than traditional techniques. The detection of different centres of reflection is the main problem of the lateration technique. The ASR technique results in pretty well raw data, although a traditional ASR curve approximation is not possible in an environment with multiple objects. In this article we propose the simple solution of filtering all data, with a threshold close to a ASR of zero.

**Acknowledgments** This work has partly been supported within H2020-ICT by the European Commission under grant agreement number 645101 (SmokeBot).

## References

1. Adams, M., Mullane, J., Jose, E.: *Robotic Navigation and Mapping with Radar*. Artech House (2012)
2. Agilent. *Techniques for Precision Validation of Radar System Performance in the Field*. Agilent Technologies (2014)
3. Bühren, M.: *Simulation und Verarbeitung von Radarziellisten im Automobil*. PhD thesis, Universität Stuttgart (2008)
4. Clark, S., Dissanayake, G.: Simultaneous localisation and map building using millimetre wave radar to extract natural features. In: *International Conference on Robotics and Automation* (1999)
5. Curry, G.R.: *Radar System Performance Modeling*. Artech House (2005)
6. Detlefsen, J., Rozmann, M., Lange, M.: 94 HGZ 3-D imaging radar sensor for industrial environments. *EARSeL Adv. Remote Sens.* (1993)
7. Elfes, A.: *Using occupancy grid mapping for mobile robot perception and navigation*. IEEE Comput. (1989)
8. Fölster, F.: *Erfassung ausgedehnter Objekte durch ein Automobil-Radar*. PhD thesis, Technische Universität Hamburg-Harburg (2006)
9. Fölster, F., Rohling, H.: Data association and tracking for automotive radar networks. *IEEE Trans. Intell. Transp. Syst.* (2005)

10. Kidera, S., Sakamoto, T., Toru, S.: High-resolution and real-time three-dimensional imaging algorithm with envelopes of spheres for uwb radars. *IEEE Trans. Geosci. Remote Sens.* (2008)
11. Ludloff, A.: *Praxiswissen Radar und Radarsignalverarbeitung*. vieweg (1998)
12. Marck, J.W., Mohamoud, A., Houwen, R.V., Hejster, Eric vd qnd.: Indoor radar SLAM. A radar application for vision and gps denied environments. In: *10th European Radar Conference* (2013)
13. Mirbach, M., Menzel, W.: A simple surface estimation algorithm for uwb pulse radas based on trilateration. *IEEE International Conference on Ultra-Wideband* (2011)
14. Rabe, H., Denicke, E., Armbrrecht, G., Musch, T., Rolfes, I.: Considerations on radar localization in multi-target environments. *Adv. Radio Sci.* (2009)
15. Sakamoto, T.: A fast algorithm for 3-dimensional imaging with uwb pulse radar systems. *IEICE Trans. Commun.* (2007)
16. Schmid, G., Neubauer, G.: *Bestimmung der exposition durch ultra-wideband technologien*. Technical report, Bundesamt für Strahlenschutz (2007)
17. Schneider, M.: *LSB-Methode Bestimmung von Distanzunterschieden mittels parametrierter Schwebungen*. PhD thesis, Universität Rostock (2013)
18. Tadokoro, S.: *Rescue Robotics—DDT Project on Robots and Systems for Uraban Search and Rescue*. Springer (2009)
19. Thrun, S.: *Robotic Mapping: A Survey*. School of Comuputer Science (2002)
20. Thrun, S., Burgard, W., Fox, D.: *Probalistic Robotics*. The MIT Press (2005)

# Pole Placement of Linear Time-Varying Discrete Systems and Its Application to Trajectory Tracking Control of Nonlinear Systems

Yasuhiko Mutoh, Masakatsu Kemmotsu and Lisa Awatsu

**Abstract** In this paper, the trajectory tracking controller is designed for nonlinear systems, based on the linear time-varying approximate model around some desired trajectory. The idea is very simple and basic, but this technique has not been commonly used so far, because the approximate model becomes linear time-varying and design method of linear time-varying controller is not necessarily well-developed. The authors proposed the simple design method of the pole placement controller for linear time-varying discrete systems. To show the applicability of this technique to the trajectory tracking control problem of nonlinear systems, we apply this controller to actual 2-link robot manipulator and present the experimental results.

**Keywords** Trajectory tracking control · Linear time-varying system · Discrete system · Time-varying pole placement control

## 1 Introduction

For the control problem of nonlinear systems, it is a classical and standard approach to design a linear controller for the linear approximate model around some equilibrium point or some desired operating point. Since there are variety of control strategies for linear systems, this approach should be the first choice to control nonlinear systems among a lot of controllers including some nonlinear control methods.

If the problem is to control nonlinear systems in wide range, for example, trajectory tracking control, then it is necessary to borrow the idea of gain scheduling control or robust control using approximate models around multiple operating points along some desired trajectory.

On the other hand, it is also classical and simple idea for this problem to apply a linear controller to a linear approximate model around entire trajectory. This method

---

Y. Mutoh (✉) · M. Kemmotsu · L. Awatsu  
Department of Engineering and Applied Sciences, Sophia University, 7-1 Kioicho,  
Chiyoda-ku, Tokyo, Japan  
e-mail: y\_mutou@sophia.ac.jp

can be applied to any type of nonlinear systems. However, in this case, an approximate model becomes a linear time-varying system, and since the controller design method for linear time-varying systems is not necessarily well-developed, this type of approach has not commonly used. Nguyen [5], Valášek et al. [7, 8] extended the linear time-invariant pole placement controller design method to the linear time-varying case. This design method uses the time-varying state transformation matrix and is not necessarily simple. The author et.al. proposed the simple pole placement controller design method for linear time-varying systems [2–4], using the concept of a full relative degree of the system. Such controller is obtained by finding a new output signal so that the total relative degree from the input to this new output is equal to the system degree.

In this paper, we apply this linear time-varying discrete pole placement technique to the trajectory tracking control of an actual 2-link robot manipulator, to show the applicability of linear time-varying controller to practical nonlinear systems.

In the following, some basic properties of linear time-varying discrete systems are stated in Sect. 2. Section 3 summarizes the design procedure of a pole placement controller for linear time-varying discrete systems. In Sect. 4, this control method is applied to the trajectory tracking control problems of practical 2-link robot manipulator and experimental results are presented to show the validity of this control system.

## 2 Basic Properties of Linear Time-Varying Discrete Systems

In this section, some basic properties of linear time varying multi variable discrete systems are presented. Consider the following system.

$$x(k + 1) = A(k)x(k) + B(k)u(k) \quad (1)$$

Here,  $x(k) \in R^n$  and  $u(k) \in R^m$  are the state variable and the input signal.  $A(k) \in R^{n \times n}$  and  $B(k) \in R^{n \times m}$  are time-varying bounded coefficient matrices. The state transition matrix of the system (1) from  $k = j$  to  $k = i$ ,  $\Phi(i, j)$ , is defined as follows.

$$\Phi(i, j) = A(i - 1)A(i - 2) \dots A(j) \quad i > j \quad (2)$$

**Definition 1** System (1) is called “completely reachable within  $n$  steps” if and only if, for any state  $x_1 \in R^n$  there exists a bounded input  $u(l)$  ( $l = k, \dots, k + n - 1$ ) such that  $x(k) = 0$  and  $x(k + n) = x_1$  for all  $k$ .

**Lemma 1** System (1) is completely reachable within  $n$  steps if and only if the rank of the reachability matrix defined below is  $n$  for all  $k$ .

$$U_R(k) = [ B_0(k), B_1(k), \dots, B_{n-1}(k) ] \tag{3}$$

where,

$$\begin{aligned} B_0(k) &= B(k + n - 1) \\ B_1(k) &= \Phi(k + n, k + n - 1)B(k + n - 2) \\ &\vdots \\ B_{n-1}(k) &= \Phi(k + n, k + 1)B(k) \end{aligned} \tag{4}$$

Let  $b_i^j(k)$  be the  $i$ -th column of  $B_j(k)$ , then, the reachability matrix  $U_R(k)$  can be written as

$$U_R(k) = [ b_1^0(k) \dots b_m^0(k) | \dots | b_1^{n-1}(k) \dots b_m^{n-1}(k) ]. \tag{5}$$

$b_i^r(k)$  also satisfies the same equations as (4), i.e.,

$$\begin{aligned} b_i^0(k) &= b_i(k + n - 1) \\ b_i^1(k) &= \Phi(k + n, k + n - 1)b_i(k + n - 2) \\ &\vdots \\ b_i^{n-1}(k) &= \Phi(k + n, k + 1)b_i(k) \quad (i = 1, \dots, m) \end{aligned} \tag{6}$$

where  $b_i(k)$  is the  $i$ -th column of  $B(k)$ .

Suppose that the system (1) is completely reachable within  $n$  steps. And, it is also supposed that the reachability indices,  $\mu_i (i = 1, \dots, m)$ , can be defined, so that

$$\sum_{i=1}^m \mu_i = n \tag{7}$$

and the  $n \times n$  truncated reachability matrix

$$R(k) = [ b_1^0(k), \dots, b_1^{\mu_1-1}(k) | \dots | b_m^0(k), \dots, b_m^{\mu_m-1}(k) ] \tag{8}$$

is non-singular. It is assumed that  $\mu_1 \geq \mu_2 \geq \dots \geq \mu_m$  without loss of generality. (The reachability indices are defined by checking linear independency of columns of  $U_R(k)$  from the left to the right as the same manner as for the linear time-invariant case.)

Next, we define “a full relative degree” of linear MIMO systems.

**Lemma 2** *If the total relative degree from input signal to output signal of a linear MIMO system is equal to the system degree,  $n$ , it is said that this system has a full relative degree.*

Let the following  $y(k) \in R^m$  be an output vector of the system (1),

$$y(k) = C(k)x(k), \quad C(k) \in R^{m \times n} \quad (9)$$

where

$$y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_m(k) \end{bmatrix} \in R^m, \quad C(k) = \begin{bmatrix} c_1^T(k) \\ c_2^T(k) \\ \vdots \\ c_m^T(k) \end{bmatrix} \in R^{m \times n}. \quad (10)$$

We define  $c_i^{jT}(k)$  by the following recursive equations.

$$\begin{aligned} c_i^{0T}(k) &= c_i^T(k) \\ c_i^{(j+1)T}(k) &= c_i^{jT}(k+1)A(k) \end{aligned} \quad (11)$$

for  $i = 1, \dots, m$ .

**Lemma 3** *The system (1), (9) has the full relative degree  $n$  from  $u(k)$  to  $y(k)$  if the following equations hold.*

$$\begin{aligned} c_i^{lT}(k+1)b_j(k) &= 0, & \begin{cases} l = 0, 1, \dots, \mu_i - 2 \\ j = 1, 2, \dots, m \end{cases} \\ c_i^{\mu_i-1T}(k+1)b_j(k) &= 0, & j = 1, 2, \dots, i-1 \\ c_i^{\mu_i-1T}(k+1)b_i(k) &= 1 \end{aligned} \quad (12)$$

for  $i = 1, \dots, m$ . (Here, 1 can be replaced by a nonzero scalar function  $\lambda_i(k) \neq 0$ .)

*Proof* Using (1), (9), (11) and (12),  $y_i(k+1)$ ,  $y_i(k+2)$ , ... can be calculated to obtain the following equations.

$$\begin{aligned} y_i(k+1) &= c_i^{0T}(k+1)A(k)x(k) + c_i^{0T}(k+1)B(k)u(k) \\ &= c_i^{1T}(k)x(k) \\ y_i(k+2) &= c_i^{1T}(k+1)A(k)x(k) + c_i^{1T}(k+1)B(k)u(k) \\ &= c_i^{2T}(k)x(k) \\ &\vdots \\ y_i(k+\mu_i) &= c_i^{\mu_i-1T}(k+1)A(k)x(k) + c_i^{\mu_i-1T}(k+1)B(k)u(k) \\ &= c_i^{\mu_i T}(k)x(k) + u_i(k) + \gamma_{i(i+1)}(k)u_{i+1}(k) + \dots + \gamma_{im}(k)u_m(k) \end{aligned} \quad (13)$$

Here,

$$\gamma_{ij}(k) = c_i^{\mu_i-1T}(k+1)b_j(k) \quad (14)$$

This implies that the system (1), (9) has a full relative degree, because the vector relative degree of this system is  $(\mu_1, \dots, \mu_m)$ , and hence, the total relative degree from  $u(k)$  to  $y(k)$  is  $n$  from (7).

The following is a simple example of Eq. (12).

*Example 1* Suppose that  $n = 5, m = 2, \mu_1 = 3,$  and  $\mu_2 = 2.$  Then,  $C(k)$  and  $B(k)$  are written as follows.

$$C(k) = \begin{bmatrix} c_1^T(k) \\ c_2^T(k) \end{bmatrix}, \quad B(k) = [b_1(k) \ b_2(k)]$$

Then, the condition (12) in this case is as follows.

$$\begin{aligned} c_1^{0T}(k+1) [b_1(k) \ b_2(k)] &= [0 \ 0] \\ c_1^{1T}(k+1) [b_1(k) \ b_2(k)] &= [0 \ 0] \\ c_1^{2T}(k+1) [b_1(k) \ b_2(k)] &= [1 \ \gamma_{12}(k)] \\ c_2^{0T}(k+1) [b_1(k) \ b_2(k)] &= [0 \ 0] \\ c_2^{1T}(k+1) [b_1(k) \ b_2(k)] &= [0 \ 1] \end{aligned}$$

Let  $\alpha^i(z)$  be a polynomial of degree  $\mu_i$ , i.e.,

$$\alpha^i(z) = z^{\mu_i} + \alpha_{\mu_i-1}^i z^{\mu_i-1} + \dots + \alpha_1^i z + \alpha_0^i \tag{15}$$

Here,  $z$  is the shift operator.

Multiplying  $y_i(k+l)$  by  $\alpha_l^i(l = 0, 1, \dots, \mu_i)$  and then summing them up, the following equation is obtained ( $\alpha_{\mu_i}^i = 1$ ).

$$\alpha^i(z)y_i(k) = D_i^T(k)x(k) + \Lambda_i^T(k)u(k) \tag{16}$$

where,  $D_i^T(k)$  and  $\Lambda_i^T(k)$  are defined by

$$\begin{aligned} D_i^T(k) &= [\alpha_0^i, \alpha_1^i, \dots, \alpha_{\mu_i-1}^i, 1] \begin{bmatrix} c_i^{0T}(k) \\ c_i^{1T}(k) \\ \vdots \\ c_i^{\mu_i T}(k) \end{bmatrix} \\ \Lambda_i^T(k) &= [0, \dots, 0, 1, \gamma_{i(i+1)}(k), \dots, \gamma_{im}(k)]. \end{aligned} \tag{17}$$

Define  $D(k)$  and  $\Lambda(k)$  by

$$D(k) = \begin{bmatrix} D_1^T(k) \\ D_2^T(k) \\ \vdots \\ D_m^T(k) \end{bmatrix}, \quad \Lambda(k) = \begin{bmatrix} \Lambda_1^T(k) \\ \Lambda_2^T(k) \\ \vdots \\ \Lambda_m^T(k) \end{bmatrix} \tag{18}$$

then, we have the following equation.

$$\begin{bmatrix} \alpha^1(z) & & \\ & \ddots & \\ & & \alpha^m(z) \end{bmatrix} y(k) = D(k)x(k) + \Lambda(k)u(k) \tag{19}$$

where  $\Lambda(k)$  is nonsingular. This is another input-output representation of the given linear time-varying system (1), (9).

### 3 Design of Discrete Time-Varying Pole Placement Controller

In this section, the design procedure of the pole placement controller for linear time-varying multi input discrete systems is summarized. Suppose that the system (1) is completely reachable with its reachability indices,  $\mu_1, \dots, \mu_m$ . The problem is to design a state feedback for the system (1) so that the resulting time-varying closed-loop system becomes equivalent to some linear time-invariant system with desired stable eigenvalues. For this purpose, we first define a new output signal  $y(k) \in R^m$  for the system (1) by

$$y(k) = C(k)x(k) \tag{20}$$

so that the system (1), (20) has a full relative degree  $n$  from  $u(k)$  to  $y(k)$ . Here,  $y(k)$  and  $C(k)$  are represented in (10). Such  $C(k)$  can be obtained by solving (12). Equation (12) can be rewritten as follows, using (5) and (6).

$$\begin{aligned} c_i^T(k+1)b_j^l(k) &= 0, & \begin{cases} l = 0, 1, \dots, \mu_i - 2 \\ j = 1, 2, \dots, m \end{cases} \\ c_i^T(k+1)b_j^{\mu_i-1}(k) &= 0, & j = 1, 2, \dots, i - 1 \\ c_i^T(k+1)b_i^{\mu_i-1}(k) &= 1 \end{aligned} \tag{21}$$

for  $i = 1, \dots, m$ . These equations contain only  $c_j^T(k)$ , and which makes it possible to derive  $C(k)$  directly from (12). From which, we have the following Theorem.

**Theorem 1** *If the system is completely reachable in  $n$  steps and has the reachability indices,  $\mu_1, \dots, \mu_m$ , there exists a new output signal  $y(k)$  given by (20) such that the system (1), (20) has a full relative degree  $n$  from  $u(k)$  to  $y(k)$ . And, such  $C(k)$  can be calculated by the following equation.*

$$C(k) = WR^{-1}(k - n) \tag{22}$$

where

$$\begin{aligned} W &= \text{diag} (w_1, w_2, \dots, w_m) \\ w_i &= [0 \dots 0 \ 1] \in R^{1 \times \mu_i} \end{aligned} \tag{23}$$

and  $R(k)$  is the truncated reachability matrix in (8).

The following Example shows the procedure to obtain  $C(k)$  using the simple Example 1.

*Example 2* Since (12) can be rewritten as (21), the condition (12) of Example 1 is shown as follows.

$$\begin{aligned} c_1^T(k) [b_1^0(k - n) \ b_2^0(k - n)] &= [0 \ 0] \\ c_1^T(k) [b_1^1(k - n) \ b_2^1(k - n)] &= [0 \ 0] \\ c_1^T(k) [b_1^2(k - n) \ b_2^2(k - n)] &= [1 \ \gamma_{12}(k - 1)] \\ c_2^T(k) [b_1^0(k - n) \ b_2^0(k - n)] &= [0 \ 0] \\ c_2^T(k) [b_1^1(k - n) \ b_2^1(k - n)] &= [0 \ 1] \end{aligned}$$

The reachability indices are  $\mu_1 = 3$  and  $\mu_2 = 2$ , which implies that the nonsingular truncated reachability matrix is

$$R(k) = [b_1^0(k), \ b_1^1(k), \ b_1^2(k), \ b_2^0(k), \ b_2^1(k)]$$

Using these linearly independent vectors,  $c_1^T(k)$  and  $c_2^T(k)$  satisfying the following equation can be determined.

$$\begin{bmatrix} c_1^T(k) \\ c_2^T(k) \end{bmatrix} R(k - n) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that since  $c_2^T(k)b_1^2(k)$  is arbitrary, we chose it as 0.

In the sequel, using the new output,  $y(k)$ , we will obtain the pole placement state feedback gain.

Let  $\alpha^i(z)$  be defined as the desired stable characteristic polynomial of degree  $\mu_i$  for the closed loop system from  $u_i(k)$  to  $y_i(k)$  in (15), i.e.,

$$\alpha^i(z) = z^{\mu_i} + \alpha_{\mu_i-1}^i z^{\mu_i-1} + \dots + \alpha_1^i z + \alpha_0^i. \quad (i = 1, \dots, m) \quad (24)$$

Since, the system (1), (20) has a full relative degree  $n$  with its vector relative degrees  $\mu_1, \mu_2, \dots, \mu_m$ ,  $y(k)$ ,  $x(k)$  and  $u(k)$  satisfy another input output representation, (19), i.e.,

$$\begin{bmatrix} \alpha^1(z) & & \\ & \ddots & \\ & & \alpha^m(z) \end{bmatrix} y(k) = D(k)x(k) + \Lambda(k)u(k) \quad (25)$$

Here,  $D(k)$  and  $\Lambda(k)$  are defined by (17) and (18). Note that  $\Lambda(k)$  is nonsingular. Then, by the state feedback

$$u(k) = -\Lambda^{-1}(k)D(k)x(k) \quad (26)$$

the closed loop system becomes as follows.

$$\begin{bmatrix} \alpha^1(z) & & \\ & \ddots & \\ & & \alpha^m(z) \end{bmatrix} y(k) = 0 \quad (27)$$

This system is time-invariant and has the following state representation.

$$w(k + 1) = A^*w(k) \quad (28)$$

where  $w(k) \in R^n$  is a new state variable. The matrices  $A^* \in R^{n \times n}$  is written by

$$A^* = \begin{bmatrix} A_1^* & & 0 \\ & \ddots & \\ 0 & & A_m^* \end{bmatrix} \quad (29)$$

and  $A_i^* \in R^{\mu_i \times \mu_i}$  is defined as follows.

$$A_i^* = \begin{bmatrix} 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ \vdots & & & 1 \\ -\alpha_0^i & \dots & \dots & -\alpha_{\mu_i-1}^i \end{bmatrix} \quad (i = 1, \dots, m) \quad (30)$$

From this, the characteristic polynomial of  $A^*$  is

$$\alpha(z) = \prod_{i=1}^m \alpha^i(z). \tag{31}$$

(9) and (13) imply that  $w(k)$  is written as follows.

$$\begin{aligned} w(k) &:= \begin{bmatrix} y_1(k) \\ \vdots \\ y_1(k + \mu_1 - 1) \\ \vdots \\ y_m(k) \\ \vdots \\ y_m(k + \mu_m - 1) \end{bmatrix} = \begin{bmatrix} c_1^0(k) \\ \vdots \\ c_1^{\mu_1-1}(k) \\ \vdots \\ c_m^0(k) \\ \vdots \\ c_m^{\mu_m-1}(k) \end{bmatrix} x(k) \\ &= P(k)x(k) \end{aligned} \tag{32}$$

On the other hand, from (1) and (26), the time-varying state equation of the closed loop system becomes

$$x(k + 1) = (A(k) - B(k)\Lambda^{-1}D(k))x(k). \tag{33}$$

Thus, the system (33) is equivalent to the system (28), with the transformation matrix  $P(k)$ . It is then obvious that the following equation holds.

$$P(k + 1)(A(k) - B(k)\Lambda^{-1}D(k))P^{-1}(k) = A^* \tag{34}$$

This implies that the state feedback (26) makes the closed loop system equivalent to the system (29) that has the desired stable characteristic polynomial,  $\alpha(z)$ .

Note that the transformation matrix  $P(k)$  and  $P^{-1}(k)$  must be bounded functions, in other words,  $P(k)$  must be a Lyapunov transformation [1], to ensure the stability of the closed-loop system.

The procedures to obtain the state feedback gain is summarized below.

**[Pole Placement Design Procedure]**

- Step 1.** Calculate the reachability matrix  $U_R(k - n)$  and the reachability indices  $\mu_i$ .
- Step 2.** Calculate  $C(k) = WR^{-1}(k - n)$  for the new output signal,  $y(k)$ , using the truncated reachability matrix  $R(k)$ .

**Step 3.** Determine the desired stable closed-loop characteristic polynomials as follows for  $i = 1, \dots, m$ .

$$\alpha^i(z) = z^{\mu_i} + \alpha_{\mu_i-1}^i z^{\mu_i-1} + \dots + \alpha_1^i z + \alpha_0^i$$

**Step 4.** Using (14)–(18), calculate  $D(k)$  and  $\Lambda(k)$ . Then, the state feedback for the pole placement is

$$u(k) = -\Lambda^{-1}(k)D(k)x(k)$$

## 4 Trajectory Tracking Control of 2-Link Manipulators

In this section, discrete time-varying pole placement technique is applied to the trajectory tracking control of a two-link robot manipulator.

### 4.1 The Model of the Manipulator

Figures 1 and 2 show the picture and the model of the 2-link robot manipulator for the experiment. All links rotate in the horizontal plane.

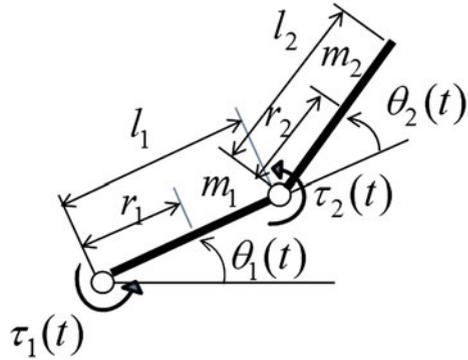
Its motion equation is described as follows.

$$M(\theta(t))\ddot{\theta}(t) + C(\theta(t), \dot{\theta}(t))\dot{\theta}(t) + D(\dot{\theta}(t)) = \tau(t) \quad (35)$$

**Fig. 1** Two-link manipulator (SR-402DDII)



**Fig. 2** Two-link manipulator model



where,

$$\begin{aligned}
 \theta(t) &= \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \end{bmatrix} \\
 M(\theta(t)) &= \begin{bmatrix} J_1 + J_2 + 2m_2r_2l_1 \cos \theta_2(t), & J_2 + m_2r_2l_1 \cos \theta_2(t) \\ J_2 + m_2r_2l_1 \cos \theta_2(t), & J_2 \end{bmatrix} \\
 C(\theta(t), \dot{\theta}(t)) &= \begin{bmatrix} -2m_2r_2l_1\dot{\theta}_2(t) \sin \theta_2(t), & -m_2r_2l_1\dot{\theta}_2(t) \sin \theta_2(t) \\ m_2r_2l_1\dot{\theta}_1(t) \sin \theta_2(t), & 0 \end{bmatrix} \\
 D(\dot{\theta}(t)) &= \begin{bmatrix} 2\text{sgn}(\dot{\theta}_1(t)) \\ 0.25\text{sgn}(\dot{\theta}_2(t)) \end{bmatrix} \\
 J_i &= J_{l_i} + m_i r_i^2 \quad (i = 1, 2).
 \end{aligned} \tag{36}$$

Here,  $\theta_i(t)$  and  $\tau_i(t)$  are joint angle and input torque of  $i$ -th joint,  $l_i$  and  $r_i$  are length of the  $i$ -th link and the distance between the  $i$ -th joint and the center of gravity of the  $i$ -th link, and  $J_{l_i}$  is the moment of inertia of the  $i$ -th link about its center of gravity ( $i = 1, 2$ ).  $D(\dot{\theta}(t))$  is a friction term which is estimated from the experimental data.

In the above, the values of the physical parameters are shown in Table 1.

**Table 1** Parameter of manipulator

Variable ( $i = 1, 2$ )	Unit	Link1 $i = 1$	Link2 $i = 2$
$m_i$	[kg]	3.43	1.55
$l_i$	[m]	0.2	0.2
$r_i$	[m]	0.1	0.1
$J_{l_i}$	[kgm <sup>2</sup> ]	0.208	0.03

## 4.2 Experimental Results

In this section, we show the experimental result of the trajectory tracking control of the 2-link robot manipulator using the time-varying discrete pole placement controller.

To design the discrete controller, we discretize the manipulator system (35) by Euler method as follows. Here,  $T_s$  is the sampling time.

$$\begin{aligned} x(k+1) &= x(k) + \begin{bmatrix} 0 & T_s I_2 \\ 0 & T_s \Gamma(x(k)) \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ T_s \Phi(x(k)) \end{bmatrix} u(k) \\ &= f(x(k), u(k)) \end{aligned} \quad (37)$$

where

$$\begin{aligned} x(k) &= \begin{bmatrix} \theta(k) \\ \dot{\theta}(k) \end{bmatrix} \in R^4 \\ u(k) &= \begin{bmatrix} \tau_1(k) \\ \tau_2(k) \end{bmatrix} \in R^2 \\ I_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \Gamma(x(k)) &= -M(\theta(k))^{-1} C(\theta(k), \dot{\theta}(k)) \in R^{2 \times 2} \\ \Phi(x(k)) &= M(\theta(k))^{-1} \in R^{2 \times 2} \end{aligned} \quad (38)$$

The sampling time  $T_s$  is 10 [msec] for the experiment. Let the desired continuous trajectory of the end portion of this manipulator be the circle in the horizontal  $X$ - $Y$  work space as presented by the following equation,

$$X(t) = 0.08 \cos \frac{\pi}{5} t + 0.3 \quad (39)$$

$$Y(t) = 0.08 \sin \frac{\pi}{5} t + 0.05 \quad (40)$$

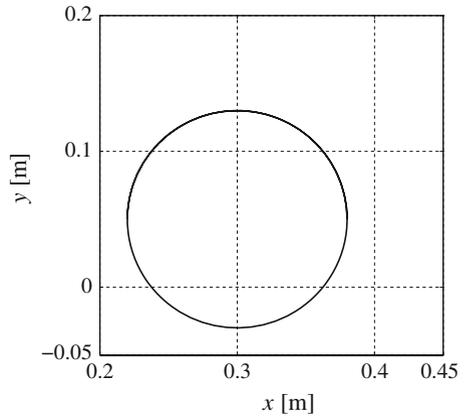
which is described in Fig. 3.

From the desired trajectory of the end portion, the desired trajectory of the joint angles,  $\theta^*(t)$ , and joint angular velocities,  $\dot{\theta}^*(t)$ , can be calculated using the inverse kinematics. This gives the desired continuous state variable  $x^*(t)$  as follows.

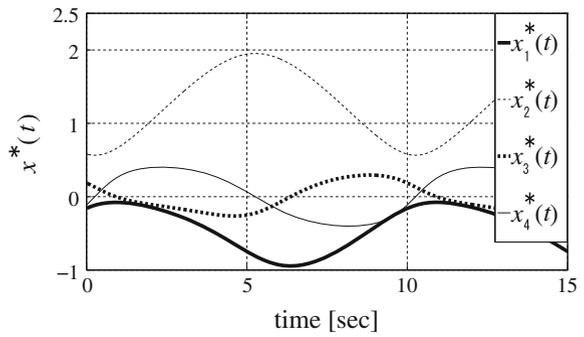
$$x^*(t) = \begin{bmatrix} \theta^*(t) \\ \dot{\theta}^*(t) \end{bmatrix} \quad (41)$$

The desired continuous input signal  $u^*(t)$  is obtained from  $\theta^*(t)$  and  $\dot{\theta}^*(t)$  using (35). The symbolic calculation software MAXIMA is used to calculate the explicit function representations for  $x^*(t)$  and  $u^*(t)$ , which are not shown here because of the space limitation. Instead of this, the graphs of  $x^*(t)$  and  $u^*(t)$  are shown in Figs. 4 and 5.

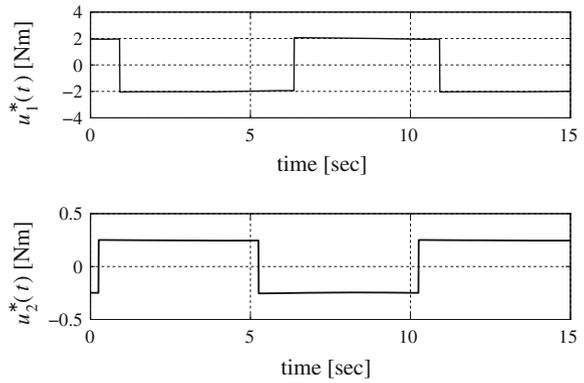
**Fig. 3** Desired trajectory of the end portion



**Fig. 4** Desired trajectory  $x^*(t)$



**Fig. 5** Desired input  $u^*(t)$



By discretizing these signals using the sampling time  $T_s$ , the discrete desired state trajectory  $x^*(k)$  and the discrete desired input  $u^*(k)$  are obtained. Note that we use the same variable for continuous space and discrete space, i.e.,  $x(t)$  is continuous variable,  $x(k)$  is a discrete variable of  $k$ -th step and  $x(kT_s)$  is a sampling variable in the  $t$ -axis.

To obtain the linear time-varying approximate model around the desired trajectory,  $x^*(k)$  and  $u^*(k)$ , define  $\Delta x(k)$  and  $\Delta u(k)$  by

$$\begin{cases} \Delta x(k) = x(k) - x^*(k) \\ \Delta u(k) = u(k) - u^*(k). \end{cases} \tag{42}$$

Then we have the following approximate model from (37) and (38).

$$\begin{aligned} \Delta x(k + 1) &= \frac{\partial}{\partial x} f(x^*(k), u^*(k)) \Delta x(k) \\ &\quad + \frac{\partial}{\partial u} f(x^*(k), u^*(k)) \Delta u(k) \\ &= A(k) \Delta x(k) + B(k) \Delta u(k) \end{aligned} \tag{43}$$

where,

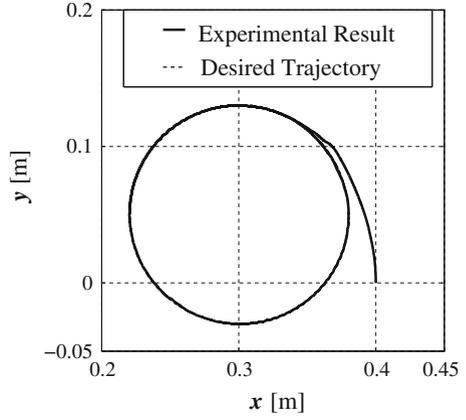
$$A(k) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_{32}(k) & A_{33}(k) & A_{34}(k) \\ 0 & A_{42}(k) & A_{43}(k) & A_{44}(k) \end{bmatrix} \tag{44}$$

$$B(k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_{31}(k) & B_{32}(k) \\ B_{41}(k) & B_{42}(k) \end{bmatrix} \tag{45}$$

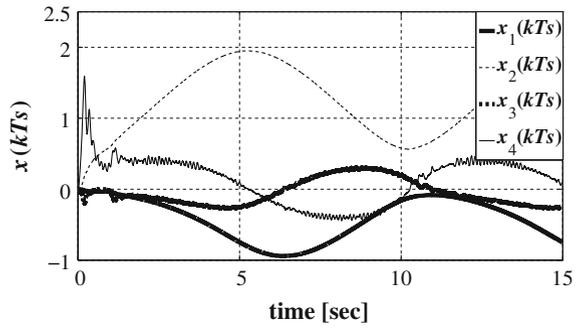
Here, the explicit function representation of  $A(k)$  and  $B(k)$  are also obtained by using MAXIMA, which are described in Appendix A, for reference.

Figure 6 shows the closed loop response of the manipulator end portion in the horizontal work space. The initial position of the end portion is (0.4, 0) in the coordinate of the horizontal work space. This initial condition corresponds to the initial condition of state variable vector,  $x_1(0) = x_2(0) = x_3(0) = x_4(0) = 0$ . The response of the manipulator state variable  $x(kT_s)$  (joint angles and joint angular velocities), the state error  $\Delta x(kT_s) = x(kT_s) - x^*(kT_s)$ , are respectively shown in Figs. 7 and 8. The control input  $u(kT_s) = u^*(kT_s) + \Delta u(kT_s)$  is shown in Fig. 9. The desired stable poles of the closed loop system are chosen as (0.942, 0.407, 0.942, 0.407).

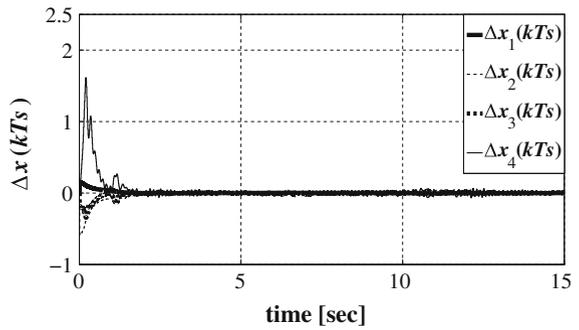
**Fig. 6** Response of end portion

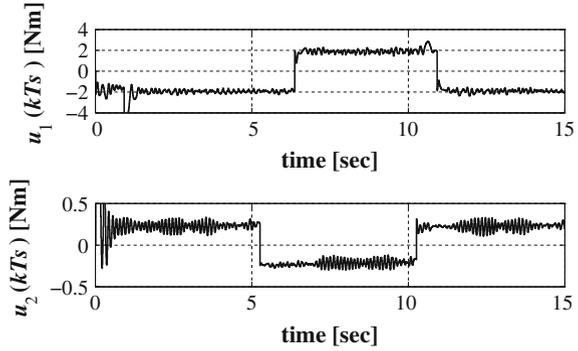


**Fig. 7** Response of state variable  $x(k)$



**Fig. 8** State error  $\Delta x(k) = x(k) - x^*(k)$



**Fig. 9** Input torque  $u(k)$ 

## 5 Conclusions

In this paper, the trajectory tracking control of nonlinear systems was considered. For this purpose, we applied the pole placement technique to linear time-varying approximate model around some desired trajectory of a nonlinear system. To simplify the design procedure of the time-varying controller, the concept of the full relative degree of the system plays a important role. This method was applied to the trajectory tracking control of the 2-link manipulator. The experimental results showed good applicability of this control strategy to the control problem of practical nonlinear systems.

## Appendix: The Explicit Form of $A(k)$ and $B(k)$

The followings are the explicit form of functions of the elements of  $A(k)$  and  $B(k)$  in Eqs. (44) and (45) calculated by MAXIMA. We used this result with  $T = T_s$ .

$$\begin{aligned}
 A_{32} = & \frac{31 (28830 \cos(x_2(k))^3 x_4(k)^2 + 185340 \cos(x_2(k)) x_4(k)^2) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 & + \frac{31 (57660 \cos(x_2(k))^3 x_3(k) x_4(k) + 370680 \cos(x_2(k)) x_3(k) x_4(k)) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 & + \frac{31 (28830 \cos(x_2(k))^3 x_3(k)^2 + 472409 \cos(x_2(k))^2 x_3(k)^2)}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 & + \frac{31 (185340 \cos(x_2(k)) x_3(k)^2 - 251100 x_3(k)^2) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 A_{33} = & 1 - \frac{62 \sin(x_2(k)) (30 x_4(k) + 31 \cos(x_2(k)) x_3(k) + 30 x_3(k)) T}{(31 \cos(x_2(k)) - 90) (31 \cos(x_2(k)) + 90)}
 \end{aligned}$$

$$\begin{aligned}
 A_{34} &= - \frac{1860 \sin(x_2(k)) (x_4(k) + x_3(k)) T}{(31 \cos(x_2(k)) - 90) (31 \cos(x_2(k)) + 90)} \\
 A_{42} &= - \frac{31(28830 \cos(x_2(k))^3 x_4(k)^2 + 472409 \cos(x_2(k))^2 x_4(k)^2) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 &\quad - \frac{31(185340 * \cos(x_2(k)) x_4(k)^2 - 251100 x_4(k)^2) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 &\quad - \frac{31(57660 \cos(x_2(k))^3 * x_3(k) x_4(k) + 944818 \cos(x_2(k))^2 x_3(k) x_4(k)) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 &\quad - \frac{31(370680 \cos(x_2(k)) x_3(k) x_4(k) - 502200 x_3(k) x_4(k)) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 &\quad - \frac{31(288300 \cos(x_2(k))^3 x_3(k)^2 + 944818 \cos(x_2(k))^2 x_3(k)^2) T}{(31 \cos(x_2(k)) - 90)^2 (31 \cos(x_2(k)) + 90)^2} \\
 &\quad + \frac{31(1853400 \cos(x_2(k))^2 x_3(k) - 502200 x_3(k)^2) T}{(31 \cos(x_2(k)) - 90)^2 (31 * \cos(x_2(k)) + 90)^2} \\
 A_{43} &= \frac{62 \sin(x_2(k)) (31 \cos(x_2(k)) x_4(k) + 30 x_4(k) + 62 \cos(x_2(k)) x_3(k)) T}{(31 \cos(x_2(k)) - 90) (31 \cos(x_2(k)) + 90)} \\
 &\quad + \frac{31 (300 x_3(k)) T}{(31 \cos(x_2(k)) - 90) (31 \cos(x_2(k)) + 90)} \\
 A_{44} &= \frac{62 (31 \cos(x_2(k)) + 30) \sin(x_2(k)) (x_4(k) + x_3(k)) T}{(31 \cos(x_2(k)) - 90) (31 \cos(x_2(k)) + 90)} + 1 \\
 B_{31} &= - \frac{30000 T}{961 \cos(x_2(k))^2 - 8100} \\
 B_{32} &= \frac{(31000 \cos(x_2(k)) + 30000) T}{961 \cos(x_2(k))^2 - 8100} \\
 B_{41} &= \frac{(31000 \cos(x_2(k)) + 30000) T}{961 \cos(x_2(k))^2 - 8100} \\
 B_{42} &= - \frac{(62000 \cos(x_2(k)) + 300000) T}{961 \cos(x_2(k))^2 - 8100}.
 \end{aligned}$$

## References

1. Chen, C.T.: Linear System Theory and Design (3rd edn). Oxford University Press (1999)
2. Mutoh, Y., Hara, T.: A New Method for Pole Placement of Linear Time-Varying Discrete Multivariable Systems. 30th CCC (2011)
3. Mutoh, Y., Kimura, N.: Observer-based pole placement for non-lexicographically-fixed linear time-varying systems. In: 50th IEEE CDC and ECC (2011)
4. Mutoh, Y.: A new design procedure of the pole-placement and the state observer for linear time-varying discrete systems. Informatics in Control, Automation and Robotics, pp. 321–334. Springer (2011)

5. Nguyen, C.C.: Arbitrary eigenvalue assignments for linear time-varying multivariable control systems. *Int. J. Control.* **45-3**, 1051–1057 (1987)
6. Rugh, W.J.: *Linear System Theory* (2nd edn). Prentice Hall (1993)
7. Valášek, M.: Efficient eigenvalue assignment for general linear MIMO systems. *Automatica.* **31-11**, 1605–1617 (1995)
8. Valášek, M., Olgaç, N.: Pole placement for linear time-varying non-lexico-graphically fixed MIMO systems. *Automatica.* **35-1**, 101–108 (1999)

# Bayesian Quadrature Variance in Sigma-Point Filtering

Jakub Průher and Miroslav Šimandl

**Abstract** Sigma-point filters are algorithms for recursive state estimation of the stochastic dynamic systems from noisy measurements, which rely on moment integral approximations by means of various numerical quadrature rules. In practice, however, it is hardly guaranteed that the system dynamics or measurement functions will meet the restrictive requirements of the classical quadratures, which inevitably results in approximation errors that are not accounted for in the current state-of-the-art sigma-point filters. We propose a method for incorporating information about the integral approximation error into the filtering algorithm by exploiting features of a Bayesian quadrature—an alternative to classical numerical integration. This is enabled by the fact that the Bayesian quadrature treats numerical integration as a statistical estimation problem, where the posterior distribution over the values of the integral serves as a model of numerical error. We demonstrate superior performance of the proposed filters on a simple univariate benchmarking example.

**Keywords** Nonlinear filtering · Sigma-point filter · Gaussian filter · Integral variance · Bayesian quadrature · Gaussian process

## 1 Introduction

Dynamic systems are widely used to model behaviour of real processes throughout the sciences. In many cases, it is useful to define a state of the system and consequently work with a state-space representation of the dynamics. When the dynamics exhibits

---

J. Průher (✉) · M. Šimandl  
European Center of Excellence - New Technologies for the Information Society,  
Faculty of Applied Sciences, University of West Bohemia, Univerzita 18,  
Pilsen, Czech Republic  
e-mail: jacobnzw@gmail.com

© Springer International Publishing Switzerland 2016  
J. Filipe et al. (eds.), *Informatics in Control, Automation and Robotics 12th International Conference, ICINCO 2015 Colmar, France, July 21-23, 2015 Revised Selected Papers*,  
Lecture Notes in Electrical Engineering 383, DOI 10.1007/978-3-319-31898-1\_20

355

stochasticity or can only be observed indirectly, we are faced with the problem of state estimation. Estimating a state of the dynamic system from noisy measurements is a prevalent problem in many application areas such as aircraft guidance, GPS navigation [12], weather forecast [9], telecommunications [14] and time series analysis [2]. When the state estimator is required to produce an estimate using only the present and past measurements, this is known as the filtering problem.

For a discrete-time linear Gaussian systems, the best estimator in the mean-square-error sense is the much-celebrated Kalman filter (KF) [16]. First attempts to deal with the estimation of nonlinear dynamics can be traced to the work of [29], which resulted in the extended Kalman filter (EKF). The EKF algorithm uses the Taylor series expansion to approximate the nonlinearities in the system description. A disadvantage of the Taylor series is that it requires differentiability of the approximated functions. This prompted further development [20, 28] resulting in the derivative-free filters based on the Stirling's interpolation formula. Other approaches that approximate nonlinearities include the Fourier-Hermite KF [27], special case of which is the statistically linearized filter [7, 18].

Instead of explicitly dealing with nonlinearities in the system description, the unscented Kalman filter (UKF) [15] describes the densities by a finite set of deterministically chosen sigma-points, which are then propagated through the nonlinearity. Other filters, such as the Gauss-Hermite Kalman filter (GHKF) [13], the cubature Kalman filter (CKF) [1] and the stochastic integration filter [6], utilize numerical quadrature rules to approximate moments of the relevant densities. These filters can be seen as representatives of a more general sigma-point methodology.

A limitation of classical integral approximations, such as the Gauss-Hermite quadrature (GHQ), is that they are specifically designed to perform with zero error on a narrow class of functions (typically polynomials up to a given degree). It is also possible to design rules, that have best average-case performance on a wider range of functions at the cost of permitting small non-zero error [19]. In recent years, the Bayesian quadrature (BQ) has become a focus of interest in probabilistic numerics community [22]. The BQ treats numerical integration as a problem of Bayesian inference and thus it is able to provide an additional information—namely, uncertainty in the computation of the integral itself. In [26], the authors work with the concept of BQ, but the algorithms derived therein do not make use of the uncertainty in the integral computations. The goal of this paper is to augment the current sigma-point algorithms so that the uncertainty associated with the integral approximations is also reflected in their estimates.

The rest of the paper is organized as follows. Formal definition of the Gaussian filtering problem is outlined in Sect. 2, followed by the exposition of the basic idea of Bayesian quadrature in Sect. 3. The main contribution, which is the design of the Bayes-Hermite Kalman filter (BHKF), is presented in Sect. 4. Finally, comparison of the BHKF with existing filters is made in Sect. 5.

## 2 Problem Formulation

The discrete-time stochastic dynamic system is described by the following state-space model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}, \quad \mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \quad \mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \quad (2)$$

with initial conditions  $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$ , where  $\mathbf{x}_k \in \mathbb{R}^n$  is the system state evolving according to the known nonlinear dynamics  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  perturbed by the white state noise  $\mathbf{w}_{k-1} \in \mathbb{R}^n$ . Measurement  $\mathbf{z}_k \in \mathbb{R}^p$  is a result of applying known nonlinear transformation  $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^p$  to the system state and white additive measurement noise  $\mathbf{r}_k \in \mathbb{R}^p$ . The mutual independence is assumed between the state noise  $\mathbf{w}_k$ , the measurement noise  $\mathbf{r}_k$  and the system initial condition  $\mathbf{x}_0$  for all  $k \geq 1$ .

The filtering problem is concerned with determination of the probability density function  $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ . The shorthand  $\mathbf{z}_{1:k}$  stands for the sequence of measurements  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k$ . The general solution to the filtering problem is given by the Bayesian recursive relations in the form of density functions

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}, \quad (3)$$

with predictive density  $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$  given by the Chapman-Kolmogorov equation

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}. \quad (4)$$

In this paper, the integration domain is assumed to be the support of  $\mathbf{x}_{k-1}$ . The likelihood term  $p(\mathbf{z}_k | \mathbf{x}_k)$  in (3) is determined by the measurement model (2) and the transition probability  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$  in (4) by the dynamics model (1).

For tractability reasons, the Gaussian filters make simplifying assumption, that the joint density of state and measurement  $p(\mathbf{x}_k, \mathbf{z}_k | \mathbf{z}_{1:k-1})$  is of the form

$$\mathcal{N} \left( \begin{bmatrix} \mathbf{x}_{k|k-1} \\ \mathbf{z}_{k|k-1} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{m}_{k|k-1}^x \\ \mathbf{m}_{k|k-1}^z \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k-1}^x & \mathbf{P}_{k|k-1}^{xz} \\ \mathbf{P}_{k|k-1}^{zx} & \mathbf{P}_{k|k-1}^z \end{bmatrix} \right). \quad (5)$$

Knowledge of the moments in (5) is fully sufficient [4] to express the first two moments,  $\mathbf{m}_{k|k}^x$  and  $\mathbf{P}_{k|k}^x$ , of the conditional density  $p(\mathbf{x}_k | \mathbf{z}_{1:k})$  using the conditioning formula for Gaussians as

$$\mathbf{m}_{k|k}^x = \mathbf{m}_{k|k-1}^x + \mathbf{K}_k (\mathbf{z}_k - \mathbf{m}_{k|k-1}^z), \quad (6)$$

$$\mathbf{P}_{k|k}^x = \mathbf{P}_{k|k-1}^x - \mathbf{K}_k \mathbf{P}_{k|k-1}^z \mathbf{K}_k^\top, \quad (7)$$

with the Kalman gain defined as  $\mathbf{K}_k = \mathbf{P}_{k|k-1}^{xz} \left( \mathbf{P}_{k|k-1}^z \right)^{-1}$ .

The problem of computing the moments in (5) can be seen, on a general level, as a computation of moments of a transformed random variable

$$\mathbf{y} = \mathbf{g}(\mathbf{x}), \tag{8}$$

where  $\mathbf{g}$  is a nonlinear vector function. This invariably entails evaluation of the integrals of the following kind

$$\mathbb{E}[\mathbf{y}] = \int \mathbf{g}(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x} \tag{9}$$

with Gaussian  $p(\mathbf{x})$ . Since the integral is typically intractable, sigma-point algorithms resort to the approximations based on weighted sum of function evaluations

$$\int \mathbf{g}(\mathbf{x}) p(\mathbf{x}) \, d\mathbf{x} \approx \sum_{i=1}^N w_i \mathbf{g}(\mathbf{x}^{(i)}). \tag{10}$$

The evaluation points  $\mathbf{x}^{(i)}$  are also known as the sigma-points, hence the name. Thus, for instance, to compute  $\mathbf{m}_{k|k-1}^x$ ,  $\mathbf{P}_{k|k-1}^x$  and  $\mathbf{P}_{k|k-1}^{xz}$ , we would use the following expressions given in matrix notation

$$\mathbf{m}_{k|k-1}^x \simeq \mathbf{F}^\top \mathbf{w}, \tag{11}$$

$$\mathbf{P}_{k|k-1}^x \simeq \tilde{\mathbf{F}}^\top \mathbf{W} \tilde{\mathbf{F}}, \tag{12}$$

$$\mathbf{P}_{k|k-1}^{xz} \simeq \tilde{\mathbf{X}}^\top \mathbf{W} \tilde{\mathbf{H}}, \tag{13}$$

where  $\mathbf{w} = [w_1, \dots, w_N]^\top$ ,  $\mathbf{W} = \text{diag}([w_1, \dots, w_N])$  are the quadrature weights. The remaining matrices are defined as

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}(\mathbf{x}_{k-1}^{(1)})^\top \\ \vdots \\ \mathbf{f}(\mathbf{x}_{k-1}^{(N)})^\top \end{bmatrix}, \quad \tilde{\mathbf{F}} = \begin{bmatrix} \left( \mathbf{f}(\mathbf{x}_{k-1}^{(1)}) - \mathbf{m}_{k|k-1}^x \right)^\top \\ \vdots \\ \left( \mathbf{f}(\mathbf{x}_{k-1}^{(N)}) - \mathbf{m}_{k|k-1}^x \right)^\top \end{bmatrix} \tag{14}$$

and

$$\tilde{\mathbf{X}} = \begin{bmatrix} \left( \mathbf{x}_{k-1}^{(1)} - \mathbf{m}_{k|k-1}^x \right)^\top \\ \vdots \\ \left( \mathbf{x}_{k-1}^{(N)} - \mathbf{m}_{k|k-1}^x \right)^\top \end{bmatrix}, \quad \tilde{\mathbf{H}} = \begin{bmatrix} \left( \mathbf{h}(\mathbf{x}_{k-1}^{(1)}) - \mathbf{m}_{k|k-1}^z \right)^\top \\ \vdots \\ \left( \mathbf{h}(\mathbf{x}_{k-1}^{(N)}) - \mathbf{m}_{k|k-1}^z \right)^\top \end{bmatrix}. \tag{15}$$

All the information a quadrature rule has about the function behaviour is conveyed by the  $N$  function values  $\mathbf{g}(\mathbf{x}^{(i)})$ . Conversely, this means that any quadrature is uncertain about the true function values in between the sigma-points. The importance of quantifying this uncertainty becomes particularly pronounced, when the function is not integrated exactly due to the inherent design limitations of the quadrature (such as the choice of weights and sigma-points). All sigma-point filters thus operate with the uncertainty, which is not accounted for in their estimates. The classical treatment of the quadrature does not lend itself nicely to the quantification of the uncertainty associated with a given rule. On the other hand, the Bayesian quadrature, which treats the integral approximation as a problem in Bayesian inference, is perfectly suited for this task.

The idea of using Bayesian quadrature in the state estimation algorithms was already treated in [26]. The derived filters and smoothers, however, do not utilize the full potential of the Bayesian quadrature. Namely, the integral variance is not reflected in their estimates. In this article, we aim to remedy this issue by making use of familiar expressions for GP prediction at uncertain inputs [3, 10].

### 3 Gaussian Process Priors and Bayesian Quadrature

In this section, we introduce the key concepts of Gaussian process priors and Bayesian quadrature, which are crucial to the derivation of the filtering algorithm in Sect. 4.

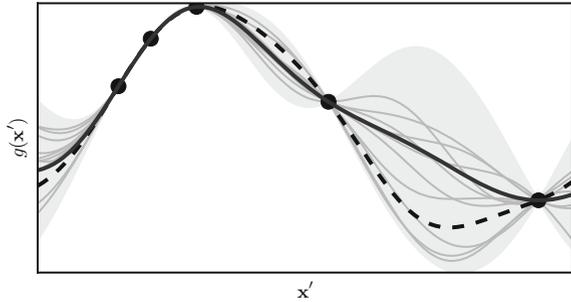
#### 3.1 Gaussian Process Priors

Uncertainty over functions is naturally expressed by a stochastic process. In Bayesian quadrature, Gaussian processes (GP) are used for their favourable analytical properties. Gaussian process is a collection of random variables indexed by elements of an index set, any finite number of which has a joint Gaussian density [23]. That is, for any finite set of indices  $\mathbf{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_m\}$ , it holds that

$$[g(\mathbf{x}'_1), g(\mathbf{x}'_2), \dots, g(\mathbf{x}'_m)]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{16}$$

where the kernel (covariance) matrix  $\mathbf{K}$  is made up of pair-wise evaluations of the kernel function, thus  $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ . Choosing a kernel, which in principle can be any symmetric positive definite function of two arguments, introduces assumptions about the underlying function we are trying to model. Bayesian inference allows to combine the GP prior  $p(g)$  with the data,  $\mathcal{D} = \{\mathbf{x}_i, g(\mathbf{x}_i)\}, i = 1, \dots, N$  comprising the evaluation points  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$  and the function evaluations  $\mathbf{y}_g = [g(\mathbf{x}_1), \dots, g(\mathbf{x}_N)]^\top$ , to produce a GP posterior  $p(g|\mathcal{D})$  with moments given by [23]

**Fig. 1** True function (dashed), GP posterior mean (solid), observed function values (dots) and GP posterior samples (grey). The shaded area represents GP posterior predictive uncertainty ( $\pm 2\sigma_g(\mathbf{x}')$ ). Uncertainty is collapsed around the observations



$$\mathbb{E}_g[g(\mathbf{x}')] = m_g(\mathbf{x}') = \mathbf{k}^\top(\mathbf{x}')\mathbf{K}^{-1}\mathbf{y}_g, \tag{17}$$

$$\nabla_g[g(\mathbf{x}')] = \sigma_g^2(\mathbf{x}') = k(\mathbf{x}', \mathbf{x}') - \mathbf{k}^\top(\mathbf{x}')\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}'), \tag{18}$$

where  $\mathbf{k}(\mathbf{x}') = [k(\mathbf{x}', \mathbf{x}_1), \dots, k(\mathbf{x}', \mathbf{x}_N)]^\top$ . Thus, for any test input  $\mathbf{x}'$ , we recover a Gaussian posterior predictive density over the function values  $g(\mathbf{x}')$ . Figure 1 depicts predictive moments of the GP posterior density. Notice, that in between the evaluations, where the true function value is not known, the GP model is uncertain.

### 3.2 Bayesian Quadrature

The problem of numerical quadrature pertains to the approximate computation of the integral

$$\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})] = \int g(\mathbf{x})p(\mathbf{x}) \, d\mathbf{x}. \tag{19}$$

The key distinguishing feature of the BQ is that it “treats the problem of numerical integration as the one of statistical inference.” [21]. This is achieved by placing a prior density over the integrated functions themselves. Consequence of this is that the integral itself is then a random variable as well. Concretely, if GP prior density is used, then the value of the integral of the function will also be Gaussian distributed. This follows from the fact that integral is a linear operator acting on the GP distributed random function  $g(\mathbf{x})$ .

Following the line of thought of [24] we take expectation (with respect to  $p(g|\mathcal{D})$ ) of the integral (19) and obtain

$$\begin{aligned} \mathbb{E}_{g|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] &= \iint g(\mathbf{x})p(\mathbf{x}) \, d\mathbf{x} p(g|\mathcal{D}) \, dg \\ &= \iint g(\mathbf{x})p(g|\mathcal{D}) \, dg p(\mathbf{x}) \, d\mathbf{x} = \mathbb{E}_{\mathbf{x}}[\mathbb{E}_{g|\mathcal{D}}[g(\mathbf{x})]]. \end{aligned} \tag{20}$$

From (20) we see, that taking the expectation of integral is the same as integrating the GP posterior mean function, which effectively approximates the integrated function  $g(x)$ . A posteriori integral variance is [24]

$$\mathbb{V}_{g|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \iint [k(\mathbf{x}, \mathbf{x}') - \mathbf{k}^\top(\mathbf{x})\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}')]p(\mathbf{x})p(\mathbf{x}') d\mathbf{x} d\mathbf{x}'. \quad (21)$$

A popular choice of kernel function, that enables the expressions (20) and (21) to be computed analytically is an Exponentiated Quadratic (EQ)

$$k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}) = \alpha^2 \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \boldsymbol{\Lambda}^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right), \quad (22)$$

where the vertical length scale  $\alpha$  and the horizontal length scales on diagonal of  $\boldsymbol{\Lambda} = \text{diag}([\ell_1^2, \dots, \ell_n^2])$  are kernel hyper-parameters, collectively denoted by the symbol  $\boldsymbol{\theta}$ . By using this particular kernel the assumption of smoothness (infinite differentiability) of the integrand is introduced [23]. Given the kernel function in the form (22) and  $p(\mathbf{x}) = \mathcal{N}(\mathbf{m}, \mathbf{P})$ , the expressions for the integral posterior mean and variance reduce to

$$\mathbb{E}_{g|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbf{I}^\top \mathbf{K}^{-1} \mathbf{y}_g, \quad (23)$$

$$\mathbb{V}_{g|\mathcal{D}}[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \alpha^2 |2\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}|^{-1/2} - \mathbf{I}^\top \mathbf{K}^{-1} \mathbf{I}, \quad (24)$$

with  $\mathbf{I} = [l_1, \dots, l_N]^\top$ , where

$$\begin{aligned} l_i &= \int k(\mathbf{x}, \mathbf{x}_i; \boldsymbol{\theta}_g) \mathcal{N}(\mathbf{x} | \mathbf{m}, \mathbf{P}) d\mathbf{x} \\ &= \alpha^2 |\boldsymbol{\Lambda}^{-1}\mathbf{P} + \mathbf{I}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{m})^\top (\boldsymbol{\Lambda} + \mathbf{P})^{-1}(\mathbf{x}_i - \mathbf{m})\right). \end{aligned} \quad (25)$$

Notice that we could define weights as  $\mathbf{w} = \mathbf{I}^\top \mathbf{K}^{-1}$ . Then the expression (23) is just a weighted sum of function evaluations, formally conforming to the general sigma-point method as described by (11). As opposed to classical quadrature rules, that prescribe the precise locations of sigma-points, BQ makes no such restrictions. In [19], the optimal placement is determined by minimizing the posterior variance of the integral (21). In the next section, we show how the integral variance (21) can be reflected in the current nonlinear filtering quadrature-based algorithms.

## 4 Bayes-Hermite Kalman Filter

In this section, we show how the integral variance can be incorporated into the moment estimates of the transformed random variable. Parallels are drawn with existing GP-based filters and the Bayes-Hermite Kalman filter algorithm is outlined.

## 4.1 Incorporating Integral Uncertainty

Uncertainty over the function values is introduced by a GP posterior  $p(g | \mathcal{D})$ , whose mean function (17) acts effectively as an approximation to the deterministic function  $g$ . Note that the equations (17), (18) can only be used to model single output dimension of the vector function  $\mathbf{g}$ . For now, we will assume a scalar function  $g$  unless otherwise stated. To keep the notation uncluttered, conditioning on  $\mathcal{D}$  will be omitted. Treating the function values  $g(\mathbf{x})$  as random leads to the joint density  $p(g, \mathbf{x})$  and thus, when computing the moments of  $g(\mathbf{x})$ , the expectations need to be taken with respect to both variables. This results in the following approximation of the true moments

$$\mu = \mathbb{E}_{\mathbf{x}}[g(\mathbf{x})] \approx \mathbb{E}_{g, \mathbf{x}}[g(\mathbf{x})], \quad (26)$$

$$\sigma^2 = \mathbb{V}_{\mathbf{x}}[g(\mathbf{x})] \approx \mathbb{V}_{g, \mathbf{x}}[g(\mathbf{x})]. \quad (27)$$

Using the law of iterated expectations, we get

$$\mathbb{E}_{g, \mathbf{x}}[g(\mathbf{x})] = \mathbb{E}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]] = \mathbb{E}_{\mathbf{x}}[\mathbb{E}_g[g(\mathbf{x})]]. \quad (28)$$

This fact was used to derive weights for the filtering and smoothing algorithms in [26], where the same weights were used in computations of means and covariances. Our proposed approach, however, proceeds differently in derivation of weights used in the computation of covariance matrices.

Note, that the term for variance can be written out using the decomposition formula  $\mathbb{V}_{g, \mathbf{x}}[g(\mathbf{x})] = \mathbb{E}_{g, \mathbf{x}}[g(\mathbf{x})^2] - \mathbb{E}_{g, \mathbf{x}}[g(\mathbf{x})]^2$  and the Eq. (28) either as

$$\mathbb{V}_{g, \mathbf{x}}[g(\mathbf{x})] = \mathbb{E}_{\mathbf{x}}[\mathbb{V}_g[g(\mathbf{x})]] + \mathbb{V}_{\mathbf{x}}[\mathbb{E}_g[g(\mathbf{x})]], \quad (29)$$

or as

$$\mathbb{V}_{g, \mathbf{x}}[g(\mathbf{x})] = \mathbb{E}_g[\mathbb{V}_{\mathbf{x}}[g(\mathbf{x})]] + \mathbb{V}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]], \quad (30)$$

depending on which factorization of the joint density  $p(g, \mathbf{x})$  is used. The terms  $\mathbb{V}_g[g(\mathbf{x})]$  and  $\mathbb{V}_g[\mathbb{E}_{\mathbf{x}}[g(\mathbf{x})]]$  can be identified as variance of the *integrand* and variance of the *integral* respectively. In case of deterministic  $g$ , both of these terms are zero. With EQ covariance (22), the expression (28) for the first moment of a transformed random variable takes on the form (23).

Since the variance decompositions in (29) and (30) are equivalent, both can be used to achieve the same goal. The form (29) was utilized in derivation of the Gaussian process—assumed density filter (GP-ADF) [5], which relies on the solution to the problem of prediction with GPs at uncertain inputs [10]. So, even though these results were derived to solve a seemingly different problem, we point out, that by using the

form (29), the uncertainty of the integral (as seen in the last term of (30)) is implicitly reflected in the resulting covariance. To conserve space, we only provide a summary of the results in [3] and point reader to the said reference for detailed derivations. The expressions for the moments of transformed variable were rewritten into a form, which assumes that a single GP is used to model all the output dimensions of the vector function (8)

$$\boldsymbol{\mu} = \mathbf{G}^\top \mathbf{w}, \tag{31}$$

$$\boldsymbol{\Sigma} = \mathbf{G}^\top \mathbf{W} \mathbf{G} - \boldsymbol{\mu} \boldsymbol{\mu}^\top + \text{diag}(\alpha^2 - \text{tr}(\mathbf{K}^{-1} \mathbf{L})), \tag{32}$$

with matrix  $\mathbf{G}$  being defined analogously to  $\mathbf{F}$  in (11)–(13). The weights are given as

$$\mathbf{w} = \mathbf{K}^{-1} \mathbf{1} \text{ and } \mathbf{W} = \mathbf{K}^{-1} \mathbf{L} \mathbf{K}^{-1}, \tag{33}$$

where

$$\mathbf{L} = \int k(\mathbf{X}, \mathbf{x}; \boldsymbol{\theta}_g) k(\mathbf{x}, \mathbf{X}; \boldsymbol{\theta}_g) \mathcal{N}(\mathbf{x} | \mathbf{m}, \mathbf{P}) \, d\mathbf{x}. \tag{34}$$

The elements of the matrix  $\mathbf{L}$  are given by

$$[\mathbf{L}]_{ij} = \frac{k(\mathbf{x}_i, \mathbf{m}; \boldsymbol{\theta}_g) k(\mathbf{x}_j, \mathbf{m}; \boldsymbol{\theta}_g)}{|2\mathbf{P}\boldsymbol{\Lambda}^{-1} + \mathbf{I}|^{1/2}} \times \exp\left(\left(\mathbf{z}_{ij} - \mathbf{m}\right)^\top \left(\mathbf{P} + \frac{1}{2}\boldsymbol{\Lambda}\right)^{-1} \mathbf{P}\boldsymbol{\Lambda}^{-1} \left(\mathbf{z}_{ij} - \mathbf{m}\right)\right), \tag{35}$$

where  $\mathbf{z}_{ij} = \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$ . The Eqs. (31) and (32) bear certain resemblance to the sigma-point method in (11), (12); however, in this case matrix  $\mathbf{W}$  is not diagonal.

## 4.2 BHKF Algorithm

The filtering algorithm based on the BQ can now be constructed utilizing (31) and (32). The BHKF uses two GPs with the EQ covariance—one for each function in the state-space model (1) and (2), which means that the two sets of hyper-parameters are used;  $\boldsymbol{\theta}_f$  and  $\boldsymbol{\theta}_h$ . In the algorithm specification below, the lower index of  $\mathbf{q}$  and  $\mathbf{K}$  specifies the set of hyper-parameters used to compute these quantities.

### Algorithm 1 (Bayes-Hermite Kalman Filter)

In the following, let system initial conditions  $\mathbf{x}_{0|0} \sim \mathcal{N}(\mathbf{m}_{0|0}, \mathbf{P}_{0|0})$ , the sigma-point index  $i = 1, \dots, N$  and time step index  $k = 1, 2, \dots$ .

*Initialization:*

Choose unit sigma-points  $\boldsymbol{\xi}^{(i)}$ . Set hyper-parameters  $\boldsymbol{\theta}_f$  and  $\boldsymbol{\theta}_h$ . For all time steps  $k$ ,

proceed from the initial conditions  $\mathbf{x}_{0|0}$ , by alternating between the following prediction and filtering steps.

*Prediction:*

1. Form the sigma-points  $\mathbf{x}_{k-1}^{(i)} = \mathbf{m}_{k-1|k-1}^x + \sqrt{\mathbf{P}_{k-1|k-1}^x} \boldsymbol{\xi}^{(i)}$ .
2. Propagate sigma-points through the dynamics model  $\mathbf{x}_k^{(i)} = \mathbf{f}(\mathbf{x}_{k-1}^{(i)})$ , and form  $\mathbf{F}$  as in (11)–(13).
3. Using  $\xi_k^{(i)}$  and hyper-parameters  $\boldsymbol{\theta}_f$ , compute weights  $\mathbf{w}^x$  and  $\mathbf{W}^x$  according to (33) and (34) (with  $m = 0$ , and  $P = I$ ).
4. Compute predictive mean  $\mathbf{m}_{k|k-1}^x$  and predictive covariance  $\mathbf{P}_{k|k-1}^x$

$$\begin{aligned} \mathbf{m}_{k|k-1}^x &= \mathbf{F}^\top \mathbf{w}^x, \\ \mathbf{P}_{k|k-1}^x &= \mathbf{F}^\top \mathbf{W}^x \mathbf{F} - \mathbf{m}_{k|k-1}^x (\mathbf{m}_{k|k-1}^x)^\top \\ &\quad + \text{diag}(\alpha^2 - \text{tr}(\mathbf{K}_f^{-1} \mathbf{L})) + \mathbf{Q}. \end{aligned}$$

*Filtering:*

1. Form the sigma-points  $\mathbf{x}_k^{(i)} = \mathbf{m}_{k|k-1}^x + \sqrt{\mathbf{P}_{k|k-1}^x} \boldsymbol{\xi}^{(i)}$ .
2. Propagate the sigma-points through the measurement model  $\mathbf{z}_k^{(i)} = \mathbf{h}(\mathbf{x}_k^{(i)})$ , and form  $\mathbf{H}$  as in (11)–(13).
3. Using  $\xi_k^{(i)}$  and hyper-parameters  $\boldsymbol{\theta}_h$ , compute weights  $\mathbf{w}^z$  and  $\mathbf{W}^z$  according to (33) and (35) (with  $m = 0$ , and  $P = I$ ) and  $\mathbf{W}^{xz} = \text{diag}(\mathbf{l}_h) \mathbf{K}_h^{-1}$ .
4. Compute measurement mean, covariance and state-measurement cross-covariance

$$\begin{aligned} \mathbf{m}_{k|k-1}^z &= \mathbf{H}^\top \mathbf{w}^z, \\ \mathbf{P}_{k|k-1}^z &= \mathbf{H}^\top \mathbf{W}^z \mathbf{H} - \mathbf{m}_{k|k-1}^z (\mathbf{m}_{k|k-1}^z)^\top \\ &\quad + \text{diag}(\alpha^2 - \text{tr}(\mathbf{K}_h^{-1} \mathbf{L})) + \mathbf{R}, \\ \mathbf{P}_{k|k-1}^{xz} &= \mathbf{P}_{k|k-1}^x (\mathbf{P}_{k|k-1}^x + \boldsymbol{\Lambda})^{-1} \tilde{\mathbf{X}} \mathbf{W}^{xz} \mathbf{H}, \end{aligned}$$

where the  $i$ -th row of  $\tilde{\mathbf{X}}$  is  $\mathbf{x}_k^{(i)} - \mathbf{m}_{k|k-1}^x$

5. Compute the filtered mean  $\mathbf{m}_{k|k}^x$  and filtered covariance  $\mathbf{P}_{k|k}^x$

$$\begin{aligned} \mathbf{m}_{k|k}^x &= \mathbf{m}_{k|k-1}^x + \mathbf{K}_k (\mathbf{z}_k - \mathbf{m}_{k|k-1}^z), \\ \mathbf{P}_{k|k}^x &= \mathbf{P}_{k|k-1}^x - \mathbf{K}_k \mathbf{P}_{k|k-1}^z \mathbf{K}_k^\top, \end{aligned}$$

with Kalman gain  $\mathbf{K}_k = \mathbf{P}_{k|k-1}^{xz} (\mathbf{P}_{k|k-1}^z)^{-1}$ .

## 5 Numerical Illustration

In the numerical simulations the performance of the filters was tested on a univariate non-stationary growth model (UNGM) [11]

$$x_k = \frac{1}{2}x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8 \cos(1.2k) + q_{k-1}, \tag{36}$$

$$z_k = \frac{1}{20}x_{k-1}^2 + r_k, \tag{37}$$

with the state noise  $q_{k-1} \sim \mathcal{N}(0, 10)$ , measurement noise  $r_k \sim \mathcal{N}(0, 1)$  and initial conditions  $x_{0|0} \sim \mathcal{N}(0, 5)$ .

Since the BHKF does not prescribe the sigma-point locations, they can be chosen at will. The GHKF based on the  $r$ -th order Gauss-Hermite (GH) quadrature rule uses sigma-points, which are determined as the roots of the  $r$ -th degree univariate Hermite polynomial  $H_r(x)$ . When it is required to integrate function of a vector argument ( $n > 1$ ), a multidimensional grid of points is formed by the Cartesian product, leading to their exponential growth ( $N = r^n$ ). The GH weights are computed according to [25] as

$$w_i = \frac{r!}{[r H_{r-1}(x^{(i)})]^2}. \tag{38}$$

The Unscented Transform (UT) is also a simple quadrature rule [13], that uses  $N = 2n + 1$  deterministically chosen sigma-points,

$$\mathbf{x}^{(i)} = \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}^{(i)} \tag{39}$$

with unit sigma-points defined as columns of the matrix

$$\left[ \boldsymbol{\xi}^{(0)}, \boldsymbol{\xi}^{(1)}, \dots, \boldsymbol{\xi}^{(2n+1)} \right] = \left[ \mathbf{0}, c\mathbf{I}_n, -c\mathbf{I}_n \right], \tag{40}$$

where  $\mathbf{I}_n$  denotes  $n \times n$  identity matrix. The corresponding weights are defined by

$$w_0 = \frac{\kappa}{n + \kappa}, \quad w_i = \frac{1}{2(n + \kappa)}, \quad i = 1, \dots, 2n \tag{41}$$

with scaling factor  $c = \sqrt{n + \kappa}$ . Very similar to UT is the spherical-radial (SR) integration rule, which is a basis of the cubature Kalman filter (CKF) [1]. The SR rule uses  $2n$  sigma-points given by

$$\left[ \boldsymbol{\xi}^{(1)}, \dots, \boldsymbol{\xi}^{(2n)} \right] = \left[ c\mathbf{I}_n, -c\mathbf{I}_n \right] \tag{42}$$

with  $c = \sqrt{n}$  and weights  $w_i = 1/2n$ ,  $i = 1, \dots, 2n$ . All of the BHKFs used the same vertical lengthscale  $\alpha = 1$ . The horizontal lengthscale was set to  $\ell = 3.0$  for UT,  $\ell = 0.3$  for SR and GH-5, and  $\ell = 0.1$  for all higher-order GH sigma-point sets. BHKFs that used UT and GH sigma-points of order 5, 7, 10, 15 and 20 were compared with their classical quadrature-based counterparts, namely, UKF and GHKF of order 5, 7, 10, 15 and 20. UKF operated with  $\kappa = 0$ .

We performed 100 simulations, each for  $K = 500$  time steps. Root-mean-square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{K} \sum_{k=1}^K (\mathbf{x}_k - \mathbf{m}_{k|k}^x)^2} \tag{43}$$

was used to measure the overall difference of the state estimate  $\mathbf{m}_{k|k}^x$  from the true state  $\mathbf{x}_k$  across all time steps. The negative log-likelihood of the filtered state estimate  $\mathbf{m}_{k|k}^x$  and covariance  $\mathbf{P}_k^x$

$$\text{NLL} = -\log p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{1}{2} [\log |2\pi \mathbf{P}_k^x| + (\mathbf{x}_k - \mathbf{m}_{k|k}^x)^\top (\mathbf{P}_k^x)^{-1} (\mathbf{x}_k - \mathbf{m}_{k|k}^x)] \tag{44}$$

was used to measure the overall model fit [8]. As a metric that takes into account the estimated state covariance, the non-credibility index (NCI) [17] given by

$$\text{NCI} = \frac{10}{K} \sum_{k=1}^K \log_{10} \frac{(\mathbf{x}_k - \mathbf{m}_{k|k}^x)^\top \mathbf{P}_{k|k}^{-1} (\mathbf{x}_k - \mathbf{m}_{k|k}^x)}{(\mathbf{x}_k - \mathbf{m}_{k|k}^x)^\top \Sigma_k^{-1} (\mathbf{x}_k - \mathbf{m}_{k|k}^x)} \tag{45}$$

was used, where  $\Sigma_k$  is the mean-square-error matrix. The filter is said to be optimistic if it underestimates the actual error, which is indicated by  $\text{NCI} > 0$ . Perfectly credible filter would provide  $\text{NCI} = 0$ , that is, it would neither overestimate nor underestimate the actual error.

Tables show average values of the performance criteria across simulations with estimates of  $\pm 2$  standard deviations (obtained by bootstrapping [30]). As evidenced by the results in Table 1, the BQ provides superior RMSE performance for all sigma-

**Table 1** The average root-mean-square error

Sigma-pts.	N	Bayesian	Classical
SR	2	6.157 ± 0.071	13.652 ± 0.253
UT	3	7.124 ± 0.131	7.103 ± 0.130
GH-5	5	8.371 ± 0.128	10.466 ± 0.198
GH-7	7	8.360 ± 0.043	9.919 ± 0.215
GH-10	10	7.082 ± 0.098	8.035 ± 0.193
GH-15	15	6.944 ± 0.048	8.224 ± 0.188
GH-20	20	6.601 ± 0.058	7.406 ± 0.193

**Table 2** The average negative log-likelihood

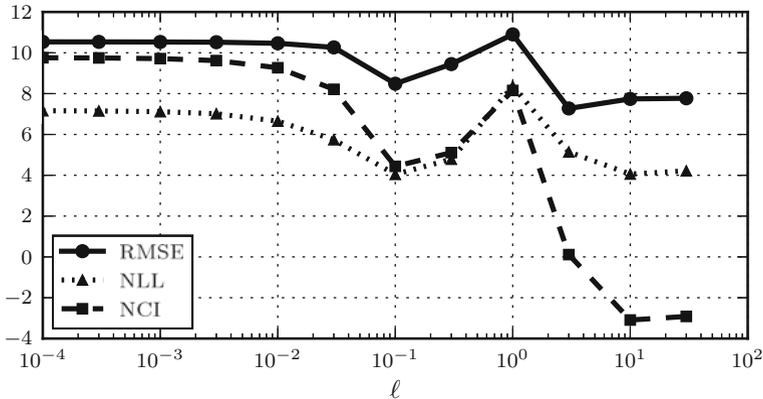
Sigma-pts.	N	Bayesian	Classical
SR	2	3.328 ± 0.026	56.570 ± 2.728
UT	3	4.970 ± 0.343	5.306 ± 0.481
GH-5	5	4.088 ± 0.064	14.722 ± 0.829
GH-7	7	4.045 ± 0.017	12.395 ± 0.855
GH-10	10	3.530 ± 0.012	7.565 ± 0.534
GH-15	15	3.468 ± 0.014	7.142 ± 0.557
GH-20	20	3.378 ± 0.017	5.664 ± 0.488

**Table 3** The average non-credibility index

Sigma-pts.	N	Bayesian	Classical
SR	2	1.265 ± 0.010	18.585 ± 0.045
UT	3	0.363 ± 0.108	0.897 ± 0.088
GH-5	5	4.549 ± 0.013	9.679 ± 0.068
GH-7	7	4.368 ± 0.006	8.409 ± 0.076
GH-10	10	2.520 ± 0.006	5.315 ± 0.058
GH-15	15	2.331 ± 0.008	5.424 ± 0.059
GH-20	20	1.654 ± 0.007	4.105 ± 0.055

point sets. In the classical quadrature case the performance improves with increasing number of sigma-points used. Table 2 shows that the performance of BHKF is clearly superior in terms of NLL, which indicates that the estimates produced by the BQ-based filters are better representations of the unknown true state development. The self-assessment of the filter performance is less optimistic in the case of BQ, as indicated by the lower NCI in the Table 3. This indicates that the BQ based filters are more conservative in their covariance estimates. This is a consequence of including additional uncertainty (integral variance), which the classical quadrature-based filters do not utilize. Also note, that the variance of all the evaluated criteria for Bayesian quadrature based filters is mostly an order of magnitude lower.

To achieve competitive results, the kernel lengthscale  $\ell$  had to be manually set for each filter separately. This was done by running the filters with increasing lengthscale, plotting the performance metrics and choosing the value which gave the smallest RMSE and the NCI closest to 0. Figure 2 illustrates the effect of changing length scale on the overall performance of the BHKF with UT sigma-points.



**Fig. 2** Sensitivity of BHKF performance (using UT sigma-points) to changes in the length scale hyper parameter  $\ell$ . The choice  $\ell = 3$  minimizes RMSE and provides nearly optimal NCI.

## 6 Conclusions

In this paper, we proposed a way of utilizing uncertainty associated with integral approximations in the nonlinear sigma-point filtering algorithms. This was enabled by the Bayesian treatment of the quadrature as well as by making use of the previously derived results for the GP prediction at uncertain inputs.

The proposed Bayesian quadrature based filtering algorithms were tested on a univariate benchmarking example. The results show that the filters utilizing additional quadrature uncertainty show significant improvement in terms of estimate credibility and overall model fit.

We also showed, that proper setting of the hyper-parameters is crucially important for achieving competitive results. Further research should be concerned with development of principled approaches for dealing with the kernel hyper-parameters. Freedom of choice of the sigma-points in the BQ offers a good opportunity for developing adaptive sigma-point placement techniques.

**Acknowledgments** This work was supported by the Czech Science Foundation, project no. GACR P103-13-07058J.

## References

1. Arasaratnam, I., Haykin, S.: Cubature kalman filters. *IEEE Trans. Autom. Control* **54**(6), 1254–1269 (2009)
2. Bhar, R.: *Stochastic filtering with applications in finance*. World Scientific (2010)

3. Deisenroth, M. P., Huber, M. F., Hanebeck, U. D.: Analytic moment-based Gaussian process filtering. In: Proceedings of the 26th Annual International Conference on Machine Learning—ICML '09, pp. 1–8. ACM Press (2009)
4. Deisenroth, M. P., Ohlsson, H.: A general perspective on gaussian filtering and smoothing: explaining current and deriving new algorithms. In: IEEE (June 2011) American Control Conference (ACC), pp. 1807–1812 (2011)
5. Deisenroth, M.P., Turner, R.D., Huber, M.F., Hanebeck, U.D., Rasmussen, C.E.: Robust filtering and smoothing with gaussian processes. *IEEE Trans. Autom. Control* **57**(7), 1865–1871 (2012)
6. Dunik, J., Straka, O., Šimandl, M.: Stochastic integration filter. *IEEE Trans. Autom. Control* **58**(6), 1561–1566 (2013)
7. Gelb, A.: *Applied Optimal Estimation*. The MIT Press (1974)
8. Gelman, A.: *Bayesian Data Analysis*. Chapman and Hall/CRC, 3rd edn (2013)
9. Gillijns, S., Mendoza, O., Chandrasekar, J., De Moor, B., Bernstein, D., Ridley, A.: What is the ensemble kalman filter and how well does it work? In: American Control Conference, 2006. p. 6 (2006)
10. Girard, A., Rasmussen, C. E., Quiñero Candela, J., Murray-Smith, R.: Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In: Becker, S., Thrun, S., Obermayer, K. (eds.) *Advances in Neural Information Processing Systems 15*, pp. 545–552. MIT Press (2003)
11. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)* **140**(2), 107–113 (1993)
12. Grewal, M. S., Weill, L. R., Andrews, A. P.: *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley (2007)
13. Ito, K., Xiong, K.: Gaussian filters for nonlinear filtering problems. *IEEE Trans. Autom. Control* **45**(5), 910–927 (2000)
14. Jiang, T., Sidiropoulos, N., Giannakis, G.: Kalman filtering for power estimation in mobile communications. *IEEE Trans. Wireless Commun.* **2**(1), 151–161 (2003)
15. Julier, S.J., Uhlmann, J.K., Durrant-Whyte, H.F.: A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control* **45**(3), 477–482 (2000)
16. Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(1), 35–45 (1960)
17. Li, X. R., Zhao, Z.: Measuring estimator’s credibility: noncredibility index. In: 2006 9th International Conference on Information Fusion, pp. 1–8 (2006)
18. Maybeck, P. S.: *Stochastic Models, Estimation and Control: Volume 2*. Academic Press (1982)
19. Minka, T.P.: *Deriving Quadrature Rules from Gaussian Processes*. Tech. rep., Statistics Department, Carnegie Mellon University, Tech. Rep (2000)
20. Nørgaard, M., Poulsen, N.K., Ravn, O.: New developments in state estimation for nonlinear systems. *Automatica* **36**, 1627–1638 (2000)
21. O’Hagan, A.: Bayes-Hermite quadrature. *J. Stat. Plann. Infer.* **29**(3), 245–260 (1991)
22. Osborne, M. A., Rasmussen, C. E., Duvenaud, D. K., Garnett, R., Roberts, S. J.: Active learning of model evidence using bayesian quadrature. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 46–54 (2012)
23. Rasmussen, C. E., Williams, C. K.: *Gaussian Processes for Machine Learning*. The MIT Press (2006)
24. Rasmussen, C. E., Ghahramani, Z.: Bayesian monte carlo. In: Becker, S.T., Obermayer, K. (eds.) *Advances in Neural Information Processing Systems 15*, pp. 489–496. MIT Press, Cambridge, MA (2003)
25. Särkkä, S.: *Bayesian Filtering and Smoothing*. Cambridge University Press, New York (2013)
26. Särkkä, S., Hartikainen, J., Svensson, L., Sandblom, F.: Gaussian process quadratures in nonlinear sigma-point filtering and smoothing. In: 2014 17th International Conference on Information Fusion (FUSION), pp. 1–8 (2014)

27. Sarmavuori, J., Särkkä, S.: Fourier-Hermite Kalman filter. *IEEE Trans. Autom. Control* **57**(6), 1511–1515 (2012)
28. Šimandl, M., Duník, J.: Derivative-free estimation methods: new results and performance analysis. *Automatica* **45**(7), 1749–1757 (2009)
29. Smith, G. L., Schmidt, S. F., McGee, L. A.: Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Tech. rep., NASA Tech. Rep. (1962)
30. Wasserman, L.: *All of Nonparametric Statistics*. Springer (2007)

# Adaptive Nonlinear Information Filters for Multiple Sensor Fusion

Aritro Dey, Smita Sadhu and Tapan Kumar Ghoshal

**Abstract** This paper addresses the problem of multiple sensor fusion in situations when the system dynamics is affected by unknown parameter variation and proposes a set of adaptive nonlinear information filters. For the above estimation problem complete knowledge of the process noise covariance ( $\mathbf{Q}$ ) remains unavailable due to unknown parameter variation. The proposed varieties of adaptive nonlinear information filters are so designed that they can present satisfactory estimation performance in the face of parametric uncertainty by online adaptation of unknown  $\mathbf{Q}$ . The adaptation steps incorporated in the algorithms have been formulated using Maximum Likelihood Estimation method. Superiority of the adaptive information filters over their non adaptive counterparts is demonstrated in simulation considering a case study where a maneuvering aircraft is to be tracked using multiple radars. Additionally, comparison of performance of proposed alternative adaptive filters is also carried out to appreciate the relative advantages of the proposed variants of adaptive information filters for multiple sensor fusion.

**Keywords**  $\mathbf{Q}$  adaptation · Information filter · Parametric uncertainty · Multiple sensor fusion

## 1 Introduction

Sensor fusion is a conventional process of integration of information from multiple measurements and has widespread use in real life applications, like, target tracking in collaborative sensor networks [1], sensor fusion in the fields of robot navigation,

---

A. Dey (✉) · S. Sadhu · T.K. Ghoshal  
Department of Electrical Engineering, Jadavpur University, Kolkata 700032, India  
e-mail: adeyju@gmail.com; no.words.acts@gmail.com

S. Sadhu  
e-mail: ssadhuju@gmail.com

T.K. Ghoshal  
e-mail: tkghoshal@gmail.com

intelligent vehicle, surveillance [2]. Sensor fusion using homogeneous as well as heterogeneous measurements is advantageous as this can present sufficiently reliable and enhanced estimate of the unmeasured states of the system under investigation. The success of sensor fusion depends on the formulation of an estimation problem followed by a satisfactory solution method. For multiple sensor estimation the Information filter configuration of estimators are usually preferred over the conventional error covariance form because of their simple computation methodology and easy initialization [3]. Recent works on sensor fusion which demonstrate continued interest in this specific configuration have been reported as follows.

For multiple sensor estimation with nonlinear signal model a class of nonlinear information filters viz., Unscented information filters [2], Central Difference information filters [4], Cubature and higher order cubature information filters [5–7] have been reported in literature where multi sensor estimation is found to be satisfactory only when the tuning parameters, i.e., the process and the measurement noise covariances are chosen accurately. Improper tuning of noise covariance causes suboptimal estimation performance and occurrence of divergence may also be observed at the worst case. In the present paper, a contingent situation has been addressed where the system dynamics is perturbed with parametric uncertainty. Incomplete knowledge of process noise covariance ( $\mathbf{Q}$ ) due to unknown parameter variation is, therefore, a critical concern for multiple sensor fusion which needs attention. This paper presents a set of new algorithms for adaptive nonlinear information filters which are intended for the critical situation as described above. In such situations some elements of  $\mathbf{Q}$  remain unknown and therefore have to be assigned with some assumed values. With the use of the adaptation algorithm, which has been mathematically formulated in this paper, the unknown elements can be adapted online and subsequently a satisfactory estimation performance can be obtained even in the face of parametric uncertainty.

The proposed algorithms for adaptive nonlinear information filters use several derivative free non adaptive information filters as the underlying framework. The adaptation algorithm is then incorporated in the structure of these non adaptive information filters. Non adaptive versions of 3rd degree Cubature Information filter (CIF) [5], Divided Difference Information filter (DDIF), an alias of Central Difference Information filter [4], Gauss Hermite Information filter (GHIF), Cubature Quadrature Information filter (CQIF) have been considered as an underlying structure to formulate their corresponding Adaptive information filter versions.

The adaptation algorithm for  $\mathbf{Q}$  is derived based on 'Maximum Likelihood Estimation' which is inspired from the early works on adaptive filters [8, 9] with linear signal models. Adaptation of  $\mathbf{Q}$  has been addressed in adaptive nonlinear filters as well with conventional error covariance form, e.g., Adaptive Unscented Kalman filter (AUKF) [10–12], Adaptive Divided Difference filter (ADDF) [11], Adaptive Gauss Hermite filter (AGHF) [13]. The concept of  $\mathbf{Q}$  adaptive nonlinear filter has been extended in this paper with information filter configuration. This paper is an extended version of [14] wherein Adaptive Divided Difference Information filter was proposed. The information filter configuration for adaptive nonlinear filter has not yet received much attention except a recent work by [15]. In this work instead

of  $\mathbf{Q}$  adaptation tuning of error covariance has been done using an adaptive factor. However, formulation of a set of  $\mathbf{Q}$  adaptive nonlinear information filters has not yet been reported in literature to the best knowledge of the authors.

The proposed  $\mathbf{Q}$  adaptive nonlinear information filters intended for sensor fusion has the following advantages: (i) Unlike Extended Information Filter and its higher order relatives computation of complex Jacobian and Hessian matrices are not required. (ii) Proposed algorithms do not require tuning parameters like the Unscented information filter. (iii) The adaptation algorithm ensures positive definiteness of the adapted  $\mathbf{Q}$ . (iv) The proposed filters have the flexibility of multiple sensor estimation even in the face of unknown noise covariance because of its dual aspect of information filter framework and  $\mathbf{Q}$  adaptation algorithm.

Like other existing adaptive nonlinear filters in conventional error covariance form, formal proof for optimality and convergence are yet to be available. The validation of the proposed filters is, therefore, done on the basis of extensive Monte Carlo simulation and involving a nonlinear estimation problem. For this estimation problem superiority of adaptive information filters over their non adaptive versions is demonstrated and the relative advantages of the alternative versions of adaptive nonlinear information filters are also presented.

## 2 Problem Statement

We consider the nonlinear dynamic equation of the system as presented below where  $\boldsymbol{\theta}_k$  is the state vector and  $\boldsymbol{\varsigma}_{k-1}$  is the vector of unknown time varying parameters.

$$\boldsymbol{\theta}_k = \boldsymbol{\lambda}(\boldsymbol{\theta}_{k-1}, \boldsymbol{\varsigma}_{k-1}) + \mathbf{w}_k^\theta \quad (1)$$

The parameters are augmented with the state vector as  $\mathbf{x}_k = [\boldsymbol{\theta}_k \ \boldsymbol{\varsigma}_k]^T$  and the augmented system dynamics is given by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_k \quad (2)$$

$$\mathbf{y}_k^\zeta = \mathbf{g}^\zeta(\mathbf{x}_k) + \mathbf{v}_k^\zeta \quad (3)$$

Here  $\mathbf{x}_k \in \mathbf{R}^n$  is an augmented state vector. As the dynamics for the parameter variation remains unknown difference equations corresponding to these typical unknown parameter  $\boldsymbol{\varsigma}_k$  are considered to obey the random walk model, i.e.,  $\boldsymbol{\varsigma}_k = \boldsymbol{\varsigma}_{k-1} + \mathbf{w}_k^\varsigma$ , where  $\mathbf{w}_k^\varsigma$  is the noise term. The covariance of  $\mathbf{w}_k^\varsigma$  remains unavailable due to the uncertainty in the nature of parameter variation.  $\mathbf{w}_k$  is the noise term linked with augmented state vector such that  $\mathbf{w}_k = [\mathbf{w}_k^\theta \ \mathbf{w}_k^\varsigma]^T$ . It is considered that  $\mathbf{w}_k$  is white noise (Gaussian) and  $\mathbf{w}_k \in \mathbf{R}^n \sim (\boldsymbol{\theta}, \mathbf{Q}_k)$ . The elements of  $\mathbf{Q}_k$  which are related to  $\mathbf{w}_k^\zeta$  remain unknown for this estimation problem.  $\mathbf{y}_k^\zeta \in \mathbf{R}^m$  is the measurement available from the  $\zeta$ th sensor among  $M$  different sensors where  $\zeta = 1, \dots, M$ . The

measurement noise of each sensor is considered to be white noise (Gaussian) and denoted as,  $v_k^\xi \in \mathbf{R}^m \sim (\mathbf{0}, \mathbf{R}_\zeta)$ . It is considered that the sensors are well characterized and their covariances are known to the designer.

### 3 Algorithm for Adaptive Nonlinear Information Filters

This section presents variants of adaptive nonlinear information filters. The algorithms are presented with the help of two subsections. The first subsection presents the existing non adaptive nonlinear filters followed by the other wherein the  $\mathbf{Q}$  adaptation algorithms have been mathematically derived. Although the latter is the central contribution of this work it depends on the former subsection for its completeness. In this paper the following adaptive nonlinear filters have been proposed (a) Adaptive Divided Difference Information filter (ADDIF), (b) Adaptive Gauss Hermite Information filter (AGHIF), (c) Adaptive Cubature Information filter (ACIF), (d) Adaptive Cubature Quadrature Information filter (ACQIF). Among the non adaptive information filtering algorithms it is important to note that non adaptive CQIF is proposed in this paper and has not yet been reported in literature to the best knowledge of the authors.

#### 3.1 Basic (Non Adaptive) Information Filters

##### Non-adaptive DDIF

(i) *Initialization*: Initialize  $\hat{\mathbf{x}}_0, \hat{\mathbf{P}}_0, \bar{\mathbf{Q}}, \mathbf{R}_\zeta$

(ii) *Time Update Steps*

$$\text{Calculate } \hat{\mathbf{S}}_x(\mathbf{k} - \mathbf{1}) = \text{CholeskyFactor}(\hat{\mathbf{P}}_{\mathbf{k} - \mathbf{1}}) \tag{4}$$

The a-priori estimate of state can be propagated as

$$\bar{\mathbf{x}}_k = \frac{h^2 - n}{h^2} \mathbf{f}(\hat{\mathbf{x}}_{k-1}) + \frac{1}{2h^2} \sum_{p=1}^n \{ \mathbf{f}(\hat{\mathbf{x}}_{k-1} + h\hat{\mathbf{s}}_{x,p}) + \mathbf{f}(\hat{\mathbf{x}}_{k-1} - h\hat{\mathbf{s}}_{x,p}) \} \tag{5}$$

where  $\hat{\mathbf{s}}_{x,p}$  is  $p$ th column of  $\hat{\mathbf{S}}_x(\mathbf{k} - \mathbf{1})$  and the interval length is chosen as,  $h = \sqrt{3}$  for Gaussian distribution following [16].

The a-priori error covariance can be obtained as:

$$\bar{\mathbf{P}}_k = \begin{bmatrix} S_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{(1)}(\mathbf{k}) & S_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{(2)}(\mathbf{k}) \end{bmatrix} \begin{bmatrix} S_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{(1)}(\mathbf{k}) & S_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{(2)}(\mathbf{k}) \end{bmatrix}^T + \bar{\mathbf{Q}} \tag{6}$$

where

$$\left\{ S_{\hat{x}\hat{x}}^{(I)}(k)_{(i,j)} \right\} = \left\{ \frac{1}{2h} (f_i(\hat{x}_{k-1} + h\hat{s}_{x,j}) - f_i(\hat{x}_{k-1} - h\hat{s}_{x,j})) \right\} \quad (7)$$

$\left\{ S_{\hat{x}\hat{x}}^{(I)}(k)_{(i,j)} \right\}$  indicates the element ( $e_{ij}$ ) of  $S_{\hat{x}\hat{x}}^{(I)}(k)$ , where  $S_{\hat{x}\hat{x}}^{(I)}(k)$  is first order approximation of the square root of a-priori error covariance based on interpolation formulae [16].  $S_{\hat{x}\hat{x}}^{(I)}(k)$  is to be computed using (7) for  $i = 1, \dots, n$  and  $j = 1, \dots, n$ .

$$\left\{ S_{\hat{x}\hat{x}}^{(2)}(k)_{(i,j)} \right\} = \left\{ \frac{\sqrt{h^2 - 1}}{2h^2} ((f_i(\hat{x}_{k-1} + h\hat{s}_{x,j}) + f_i(\hat{x}_{k-1} - h\hat{s}_{x,j}) - 2f_i(\hat{x}_{k-1}))) \right\} \quad (8)$$

$S_{\hat{x}\hat{x}}^{(2)}(k)$  is the 2nd order approximation to be computed in a similar way using (7).

The a-priori information matrix  $\bar{Z}_k$  and information vector  $\bar{z}_k$  are related with the a-priori estimate and error covariance as :

$$\bar{Z}_k = \bar{P}_k^{-1} \quad (9)$$

$$\bar{z}_k = \bar{Z}_k \bar{x}_k \quad (10)$$

Compute  $\bar{S}_x(k)$  such that it is the Cholesky factor of  $\bar{P}_k$ . This factor has been involved for measurement update steps.

$$\bar{S}_x(k) = \text{CholeskyFactor}(\bar{P}_k) \quad (11)$$

### (iii) Measurement Update Steps

For  $\zeta = 1, \dots, M$  the following steps are to be executed:

The a-priori estimate of measurement can be propagated as

$$\bar{y}_k^\zeta = \frac{h^2 - n}{h^2} \mathbf{g}^\zeta(\bar{x}_k) + \frac{1}{2h^2} \sum_{p=1}^n \{ \mathbf{g}^\zeta(\bar{x}_k + h\bar{s}_{x,p}) + \mathbf{g}^\zeta(\bar{x}_k - h\bar{s}_{x,p}) \} \quad (12)$$

The cross covariance is

$$P_k^{xy} = [\bar{S}_x(k)] \left[ S_{y\bar{x}}^{(I)}(k) \right]^T \quad (13)$$

where

$$\left\{ S_{y\bar{x}}^{(I)}(k)_{(i,j)} \right\} = \left\{ \frac{1}{2h} \left( \mathbf{g}_i^\zeta(\bar{x}_k + h\bar{s}_{x,j}) - \mathbf{g}_i^\zeta(\bar{x}_k - h\bar{s}_{x,j}) \right) \right\} \quad (14)$$

$\left\{ S_{y\bar{x}}^{(I)}(k)_{(i,j)} \right\}$  in the similar way of (7) for  $i = 1, \dots, m$  and  $j = 1, \dots, n$

The Pseudo Measurement Matrix can be computed as follows. To make the information contribution equations compatible to those of the EIF, a pseudo-measurement

matrix is defined by (15) following the approach of [2].

$$\Psi_k^\zeta = \left( \bar{P}_k^{-1} P_k^{xy} \right)^T \tag{15}$$

Each sensor presents local information state contribution and its associated information matrix as

$$\phi_k^\zeta = \left( \Psi_k^\zeta \right)^T \left( R_k^\zeta \right)^{-1} \left( y_k^\zeta - \bar{y}_k^\zeta + \Psi_k^\zeta \bar{x}_k \right) \tag{16}$$

$$\Phi_k^\zeta = \left( \Psi_k^\zeta \right)^T \left( R_k^\zeta \right)^{-1} \Psi_k^\zeta \tag{17}$$

**(iv) Multi Sensor Estimation**

For reliable estimation the information regarding the measurements obtained from all the sensors are combined using the Divided Difference information filter. The decentralized approach has been followed for multiple sensor estimation to economize computational effort [2].

As described in the problem statement, measurements are available from  $\zeta$ th sensor where  $\zeta = 1, \dots, M$ . The local information state contribution and its associated information matrix from each sensor can be obtained by (16) and (17).

The measurement update for the information vector and information matrix after fusion is simply expressed as a linear combination of these local information contribution terms by:

$$\hat{z}_k = \bar{z}_k + \sum_{\zeta=1}^M \phi_k^\zeta \tag{18}$$

$$\hat{Z}_k = \bar{Z}_k + \sum_{\zeta=1}^M \Phi_k^\zeta \tag{19}$$

The a-posteriori estimates of systems state and error covariance matrix are extracted using the formula by:

$$\hat{x}_k = \hat{Z}_k^{-1} \hat{z}_k \tag{20}$$

where,

$$\hat{P}_k = \hat{Z}_k^{-1} \tag{21}$$

**Non-adaptive GHIF**

*(i) Initialization:* Initialize  $\hat{x}_0, \hat{P}_0, \bar{Q}, R_k^\zeta$

*(ii) Selection of Gauss Hermite Quadrature Points*

The Gauss Hermite quadrature rule can be used to numerically evaluate the Bayesian integral. For single dimensional systems the rule becomes

$$\int_{-\infty}^{\infty} f(s) \frac{1}{\sqrt{2\pi}} e^{-s^2/2} ds \approx \sum_{i=1}^N f(q_i) w_i \quad (22)$$

For obtaining the quadrature points,  $q_i$  and respective weights,  $w_i$ , a symmetric tri-diagonal matrix  $\mathbf{J}$  is computed, which is defined as  $\mathbf{J}_{i,i} = 0$  and  $\mathbf{J}_{i,i+1} = \sqrt{\frac{i}{2}}$  for  $1 \leq i \leq N - 1$  for  $N$ -quadrature points.

The quadrature points are then chosen as  $q_i = \sqrt{2}x_i$  where  $x_i$  are the eigen values of  $\mathbf{J}$  matrix.

The corresponding weights,  $w_i$  is computed as  $|(v_i)_1|^2$  where  $(v_i)_1$  is the first element of the  $i$ th normalized eigenvector of  $\mathbf{J}$ . The base paper is referred in [13, 17].

The single dimensional GH quadrature rule is extended for approximating integral of Gaussian filters of higher order with the help of product rule. The  $n$ th order Gaussian integral  $\mathbf{I}_N = \int_{\mathbb{R}^n} \tilde{\mathbf{F}}(s) \frac{1}{(2\pi)^{n/2}} e^{-(1/2)|s|^2} ds$  can be equivalently expressed as

$$\mathbf{I}_N \approx \sum_{i_1=1}^N \dots \sum_{i_n=1}^N \tilde{\mathbf{F}}(q_{i_1}, q_{i_2}, \dots, q_{i_n}) w_{i_1} w_{i_2} \dots w_{i_n} \quad (23)$$

In order to evaluate  $\mathbf{I}_N$  for  $n$ th order system with the help of GH quadrature rule,  $N^n$  number of quadrature points and weights are required. This indicates that the number of quadrature points rises exponentially with the order of system.

### (ii) Time Update Steps

Compute Cholesky Factor such that

$$\hat{\mathbf{P}}_{k-1} = \hat{\mathbf{S}}_{k-1} \left( \hat{\mathbf{S}}_{k-1} \right)^T \quad (24)$$

The points selected for propagation of mean and covariance are given below as

$$\hat{\chi}_i = \hat{\mathbf{S}}_{k-1} \mathbf{q}_i + \hat{x}_{k-1} \quad (25)$$

Compute a-priori estimate of state as

$$\bar{\mathbf{x}}_k = \sum_{i=1}^N f(\hat{\chi}_i) w_i \quad (26)$$

and respective a-priori error covariance is obtained as

$$\bar{\mathbf{P}}_k = \bar{\mathbf{Q}} + \sum_{i=1}^N (f(\hat{\chi}_i) - \bar{\mathbf{x}}_k) (f(\hat{\chi}_i) - \bar{\mathbf{x}}_k)^T w_i \quad (27)$$

The a-priori information matrix is obtained as

$$\bar{\mathbf{Z}}_k = \bar{\mathbf{P}}_k^{-1} \quad (28)$$

The a-priori information vector becomes

$$\bar{\mathbf{z}}_k = \bar{\mathbf{Z}}_k \bar{\mathbf{x}}_k \quad (29)$$

**(iii) Measurement Update Steps**

For  $\zeta = 1, \dots, M$  the following steps are to be executed:

Compute the Cholesky Factor such that

$$\bar{\mathbf{P}}_k = \bar{\mathbf{S}}_k (\bar{\mathbf{S}}_k)^T \quad (30)$$

Select sigma points as

$$\bar{\mathbf{x}}_i = \bar{\mathbf{S}}_k \mathbf{q}_i + \bar{\mathbf{x}}_k \quad (31)$$

The a-priori estimate of measurement becomes

$$\bar{\mathbf{y}}_k^\zeta = \sum_{i=1}^N \mathbf{g}^\zeta(\bar{\mathbf{x}}_i) w_i \quad (32)$$

The cross covariance can be computed as

$$\mathbf{P}_k^{\mathbf{xy}} = \sum_{i=1}^N (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_k) (\mathbf{g}^\zeta(\bar{\mathbf{x}}_i) - \bar{\mathbf{y}}_k^\zeta)^T w_i \quad (33)$$

The pseudo measurement matrix becomes

$$\boldsymbol{\Psi}_k^\zeta = \left( (\bar{\mathbf{P}}_k)^{-1} \mathbf{P}_k^{\mathbf{xy}} \right)^T \quad (34)$$

The a-posteriori of information matrix is  $\hat{\mathbf{Z}}_k = \hat{\mathbf{P}}_k^{-1}$  where

$$\hat{\mathbf{P}}_k^{-1} = \bar{\mathbf{P}}_k^{-1} + \left( \boldsymbol{\Psi}_k^\zeta \right)^T \left( \mathbf{R}_k^\zeta \right)^{-1} \boldsymbol{\Psi}_k^\zeta \quad (35)$$

The update estimate of information vector

$$\hat{\mathbf{z}}_k = \bar{\mathbf{z}}_k + \left( \boldsymbol{\Psi}_k^\zeta \right)^T \left( \mathbf{R}_k^\zeta \right)^{-1} \left( \mathbf{y}_k^\zeta - \bar{\mathbf{y}}_k^\zeta + \boldsymbol{\Psi}_k^\zeta \bar{\mathbf{x}}_k \right) \quad (36)$$

Therefore, the updated state estimate becomes

$$\hat{\mathbf{x}}_k = \hat{\mathbf{Z}}_k^{-1} \hat{\mathbf{z}}_k \quad (37)$$

(iv) **Multi Sensor Estimation**: same as that of DDIF

### Non-adaptive CQIF

The steps of non adaptive CQIF are same as that of non adaptive GHIF expect the step (ii) where the quadrature points are to be selected following [17]. The rationale for the choice of quadrature points is reported in [17]. Below are given the steps for generation cubature quadrature points. The points are selected considering zero mean unity variance Gaussian noise which is scaled accordingly in the algorithm.

$2nn'$  number of cubature quadrature points are to be selected as

$$\boldsymbol{\gamma}_i = \sqrt{2\lambda_j} \mathbf{e}_k \quad (38)$$

where  $\lambda_j$  is the solution of  $n'$ th order Chebyshev-Laguerre polynomial (denoted by  $L_{n'}^\alpha$  as given below) with  $\alpha = n/2 - 1$ .

$$L_{n'}^\alpha = \lambda^{n'} - \frac{n'}{1!} (n' + \alpha) \lambda^{n'-1} + \frac{n' (n' - 1)}{2!} (n' + \alpha) (n' + \alpha - 1) \lambda^{n'-2} - \dots = 0 \quad (39)$$

Here,  $i = 1, 2, \dots, 2nn'$ ,  $j = 1, 2, \dots, n'$  and  $k = 1, 2, \dots, 2n$ .

The weights are selected as

$$w_i = \frac{1}{2n\Gamma(n/2)} \frac{n'!\Gamma(\alpha + n' + 1)}{\lambda_j [L_{n'}^\alpha(\lambda_j)]^2} \quad (40)$$

### Non-adaptive CIF

The Cubature Information filter (3rd degree) can also be formulated following the algorithm of non adaptive GHIF after replacing the step (ii) where instead of quadrature points cubature points are to be selected following the 3rd degree Cubature rule [5].

$2n$  number of cubature points have been selected as

$$\mathbf{q}_i = \mathbf{e}_i \sqrt{n} \text{ for } i = 1, \dots, n \quad (41)$$

$$\mathbf{q}_{n+i} = -\mathbf{e}_i \sqrt{n} \text{ for } i = 1, \dots, n \quad (42)$$

Here,  $\mathbf{e}_i$  is the  $i$ th unit vector. The weights are selected as

$$w_i = \frac{1}{2n} \text{ for all } i \quad (43)$$

### 3.2 $Q$ -Adaptation

#### Algorithmic Steps

For adaptation of  $Q$  replace the assumed value of  $\bar{Q}$  in the expression of a-priori error covariance by the adapted  $Q$  of previous instant, i.e.,  $\hat{Q}_{k-1}$ , where the expression of adapted  $Q$  is given as follows

The state residual is defined as

$$\rho_k = \hat{x}_k - \bar{x}_k \quad (44)$$

The covariance from the sliding window of state residual is computed as

$$\hat{C}_k^\rho = \frac{1}{N} \sum_{j=k-N+1}^k \rho_k \rho_k^T \quad (45)$$

Using the estimated state residual covariance from the sliding window (size  $N$ ) the adapted  $Q$  can be expressed as

$$\hat{Q}_k = \hat{C}_k^\rho \quad (46)$$

#### Derivation

The  $Q$  adaptation formula used in the proposed algorithm is derived using MLE method. The steps followed for derivation of adapted  $Q$  are inspired from the work of [8, 9] for linear signal models. The probability density function of the measurements conditioned on adaptive parameter,  $\alpha$ , at specific epoch  $k$  is chosen on the basis of innovation sequence. For simplification of derivation in presence of multiple sensor measurements we augment all the available measurements to get a single measurement vector as  $y_k = [y_k^1 \ y_k^2 \ \dots \ y_k^M]^T$  with dimension  $mM$ . Therefore, the corresponding measurement noise covariance becomes  $R_k = \text{diag}(R_k^1, R_k^2, \dots, R_k^M)$  and the pseudo measurement matrix is also augmented as  $\Psi_k = \text{diag}(\Psi_k^1, \Psi_k^2, \dots, \Psi_k^M)$ . The probability density function of measurement is expressed in terms of innovation sequence which is defined as  $\vartheta_k = y_k - \bar{y}_k$ . The pdf for measurement is presented as

$$P_{(z|\alpha)_k} = \frac{1}{\sqrt{(2\pi)^{mM} |C_{\vartheta_k}|}} \exp\left(-\frac{1}{2} \vartheta_k^T C_{\vartheta_k}^{-1} \vartheta_k\right) \quad (47)$$

or, taking logarithm we get

$$\ln(P_{(z|\alpha)_k}) = -\frac{1}{2} \{mM \ln(2\pi) + \ln |C_{\vartheta_k}| + \vartheta_k^T C_{\vartheta_k}^{-1} \vartheta_k\} \quad (48)$$

Multiplying both sides with  $-2$  and neglecting the constant term we get  $E = \ln |\mathbf{C}_{\vartheta k}| + \vartheta_k^T \mathbf{C}_{\vartheta k}^{-1} \vartheta_k$ . Note that the innovation sequence has been considered inside a window (a fixed length memory of size  $N$ ). Therefore, the innovation inside the window will be summed and the condition for Maximum Likelihood becomes:

$$\min \left\{ \sum_{j=j_0}^k \left( \ln |\mathbf{C}_{\vartheta j}| + \vartheta_j^T \mathbf{C}_{\vartheta j}^{-1} \vartheta_j \right) \right\} \quad (49)$$

Differentiating the expression with respect to the adaptive parameter  $\alpha$  we get

$$\sum_{j=j_0}^k \left[ \text{tr} \left\{ \mathbf{C}_{\vartheta j}^{-1} \left( \frac{\partial \mathbf{C}_{\vartheta j}}{\partial \alpha_k} \right) \right\} - \vartheta_j^T \mathbf{C}_{\vartheta j}^{-1} \left( \frac{\partial \mathbf{C}_{\vartheta j}}{\partial \alpha_k} \right) \mathbf{C}_{\vartheta j}^{-1} \vartheta_j \right] = 0 \quad (50)$$

$$\Rightarrow \sum_{j=j_0}^k \left[ \text{tr} \left\{ \mathbf{C}_{\vartheta j}^{-1} \left( \frac{\partial \mathbf{C}_{\vartheta j}}{\partial \alpha_k} \right) \right\} - \text{tr} \left\{ \mathbf{C}_{\vartheta j}^{-1} \vartheta_j \vartheta_j^T \mathbf{C}_{\vartheta j}^{-1} \left( \frac{\partial \mathbf{C}_{\vartheta j}}{\partial \alpha_k} \right) \right\} \right] = 0 \quad (51)$$

$$\Rightarrow \sum_{j=j_0}^k \left[ \text{tr} \left\{ \left[ \mathbf{C}_{\vartheta j}^{-1} - \mathbf{C}_{\vartheta j}^{-1} \vartheta_j \vartheta_j^T \mathbf{C}_{\vartheta j}^{-1} \right] \left( \frac{\partial \mathbf{C}_{\vartheta j}}{\partial \alpha_k} \right) \right\} \right] = 0 \quad (52)$$

The formulae for matrix operation are given in [8]. Here,  $\text{tr}$  indicates trace of matrix and  $j_0 = k - N + 1$ . The deduction of the relation between innovation covariance,  $\mathbf{C}_{\vartheta k}$  and  $\mathbf{Q}_k$  necessitates the augmented pseudo measurement matrix of the nonlinear measurement equation. The use of the pseudo measurement matrix is justified as reported in [2, 12]. Using the pseudo measurement matrix the innovation covariance can be represented as  $\mathbf{C}_{\vartheta k} = \mathbf{R}_k + \Psi_k \mathbf{P}_k \Psi_k^T$ . For adaptation of  $\mathbf{Q}_k$ , the adaptive parameter  $\alpha$  is chosen as  $\alpha_k = \mathbf{Q}_{kk}$  where  $\mathbf{Q}_{kk}$  is the diagonal elements of  $\mathbf{Q}_k$ .

Hence,

$$\frac{\partial \mathbf{C}_{\vartheta k}}{\partial \mathbf{Q}_{kk}} = \Psi_k \frac{\partial}{\partial \mathbf{Q}_{kk}} (\overline{\mathbf{P}}_k) \Psi_k^T \quad (53)$$

$$\frac{\partial \mathbf{C}_{\vartheta k}}{\partial \mathbf{Q}_{kk}} = \Psi_k \frac{\partial}{\partial \mathbf{Q}_{kk}} (\mathbf{A}_k + \mathbf{Q}_k) \Psi_k^T \quad (54)$$

The term  $\mathbf{A}_k$  is the a-priori error covariance where  $\mathbf{Q}$  is a zero matrix i.e., equal with  $[S_{\hat{x}\hat{x}}^{(1)}(\mathbf{k}) S_{\hat{x}\hat{x}}^{(2)}(\mathbf{k})] [S_{\hat{x}\hat{x}}^{(1)}(\mathbf{k}) S_{\hat{x}\hat{x}}^{(2)}(\mathbf{k})]^T$  for DDIF and  $\sum_{i=1}^N (f(\hat{\chi}_i) - \bar{x}_k) (f(\hat{\chi}_i) - \bar{x}_k)^T w_i$  for GHIF and others. It is assumed following the work of [9] that the within the estimation window the a-priori error covariance is in steady state. Hence the derivative of this term may be ignored. The ML equation gets modified as

$$\sum_{j=j_0}^k \left[ \text{tr} \left\{ \left[ \mathbf{C}_{\vartheta j}^{-1} - \mathbf{C}_{\vartheta j}^{-1} \vartheta_j \vartheta_j^T \mathbf{C}_{\vartheta j}^{-1} \right] (\Psi_j \mathbf{I} \Psi_j^T) \right\} \right] = 0 \quad (55)$$

Alternatively,

$$\sum_{j=j_0}^k \left[ \text{tr} \left\{ \Psi_j^T \left[ C_{\vartheta_j}^{-1} - C_{\vartheta_j}^{-1} \vartheta_j \vartheta_j^T C_{\vartheta_j}^{-1} \right] \Psi_j \right\} \right] = 0 \quad (56)$$

$$\Rightarrow \sum_{j=j_0}^k \left[ \text{tr} \left\{ \left[ \Psi_j^T C_{\vartheta_j}^{-1} \Psi_j - \Psi_j^T C_{\vartheta_j}^{-1} \vartheta_j \vartheta_j^T C_{\vartheta_j}^{-1} \Psi_j \right] \right\} \right] = 0 \quad (57)$$

$$\Rightarrow \sum_{j=j_0}^k \left[ \text{tr} \left\{ \bar{P}_j^{-1} \mathbf{K}_j \Psi_j - \bar{P}_j^{-1} \mathbf{K}_j \vartheta_j \vartheta_j^T \mathbf{K}_j^T \bar{P}_j^{-1} \right\} \right] = 0 \quad (58)$$

$$\Rightarrow \sum_{j=j_0}^k \left[ \text{tr} \left\{ \bar{P}_j^{-1} \left( \mathbf{K}_j \Psi_j \bar{P}_j - \mathbf{K}_j \vartheta_j \vartheta_j^T \mathbf{K}_j^T \right) \bar{P}_j^{-1} \right\} \right] = 0 \quad (59)$$

The term  $\mathbf{K}_k \vartheta_k$  can also be represented as  $\mathbf{K}_k \vartheta_k = \hat{\mathbf{x}}_k - \bar{\mathbf{x}}_k = \boldsymbol{\rho}_k$

$$\sum_{j=j_0}^k \left[ \text{tr} \left\{ \bar{P}_j^{-1} \left( \mathbf{K}_j \Psi_j \bar{P}_j - \boldsymbol{\rho}_j \boldsymbol{\rho}_j^T \right) \bar{P}_j^{-1} \right\} \right] = 0 \quad (60)$$

The expression of  $\bar{P}_k$  ensures that  $\bar{P}_k$  is positive definite. Therefore, above expression vanishes only when the following expression becomes zero.

$$\sum_{j=j_0}^k \left[ \text{tr} \left\{ \left( \mathbf{K}_j \Psi_j \bar{P}_j - \boldsymbol{\rho}_j \boldsymbol{\rho}_j^T \right) \right\} \right] = 0 \quad (61)$$

$$\sum_{j=j_0}^k \left[ \text{tr} \left\{ \left( \bar{P}_j - \hat{P}_j - \boldsymbol{\rho}_j \boldsymbol{\rho}_j^T \right) \right\} \right] = 0 \quad (62)$$

$$\bar{P}_k - \hat{P}_k = \frac{1}{N} \sum_{j=j_0}^k \left[ \boldsymbol{\rho}_j \boldsymbol{\rho}_j^T \right] \quad (63)$$

$$\Lambda_k + \hat{Q}_k - \hat{P}_k = \frac{1}{N} \sum_{j=j_0}^k \left[ \boldsymbol{\rho}_j \boldsymbol{\rho}_j^T \right] \quad (64)$$

$$\hat{Q}_k = \frac{1}{N} \sum_{j=j_0}^k \left[ \boldsymbol{\rho}_j \boldsymbol{\rho}_j^T \right] + \left( \hat{P}_k^- \Lambda_k \right) \quad (65)$$

During steady state the term  $(\hat{P}_k - \Lambda_k)$  becomes often low and may negligible as recommended in [9]. Hence, adapted  $Q$  is

$$\hat{Q}_k = \frac{1}{N} \sum_{j=j_0}^k [\rho_j \rho_j^T] \quad (66)$$

### Notes on Adaptation

The expression (66) as above has been presented in a simplified approach. The a-posteriori error covariance  $\hat{P}_k$  and the term  $\Lambda_k$  which is implicitly dependent on  $\hat{P}_{k-1}$ , acquire a steady value (often low) as the filter approaches steady state. Therefore, their effect can be ignored from the adapted  $Q$  so that the symmetry and the positive definiteness of  $Q$  can be ensured. Otherwise, singularity cannot be avoided. The same approach has been followed in [9] for linear signal model. For this assumption the accuracy of adaptation performance increases when the filter achieves steady state.

It is to be noted that the expression (66) is effective only when the time step index  $k$  is greater than or equal to the window length  $N$ . When the time step index  $k$  is less than  $N$ , adaptation begins with available size of state residual. The window length is gradually increased till it reaches the preferred window length  $N$ . Afterwards the sliding window concept becomes applicable.

The window size  $N$ , should be chosen considering several factors. A smaller window size generally ensures better tractability but reportedly may be prone to divergence with a size less than the dimension of state [9]. A larger window can size ensures unbiased estimates. However, cannot track the short term variation in  $Q$ .

If desired, the filtering algorithm can be recomputed after obtaining the current value of adapted  $Q$  on the basis of the availability of adequate computation power.

## 4 Case Study

The performance of adaptive nonlinear information filter has been demonstrated using a problem of multi radar tracking of an aircraft which is executing a maneuvering turn with unknown turn rate. The trajectory of the maneuvering aircraft is presented in 2D space as given in [6]. As the turn rate of the aircraft is considered to be unknown and time varying the kinematic model of the system becomes significantly nonlinear. The aircraft while executing such maneuvering turn may escape radar stations and consequently the estimators may lose the track of the aircraft as would be explained in the following subsections.

### 4.1 Kinematic Model

The kinematic equation of the motion of the aircraft is presented below. The turn rate of the aircraft being unknown it is modeled as a state and augmented with the state vector of the kinematic model [6].

$$\xi_k = \begin{bmatrix} 1 & \frac{\sin(\omega_{k-1}\tau)}{\omega_{k-1}} & 0 & \frac{\cos(\omega_{k-1}\tau)-1}{\omega_{k-1}} & 0 \\ 0 & \cos(\omega_{k-1}\tau) & 0 & -\sin(\omega_{k-1}\tau) & 0 \\ 0 & \frac{1-\cos(\omega_{k-1}\tau)}{\omega_{k-1}} & 1 & \frac{\sin(\omega_{k-1}\tau)}{\omega_{k-1}} & 0 \\ 0 & \sin(\omega_{k-1}\tau) & 0 & \cos(\omega_{k-1}\tau) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \xi_{k-1} + w_k \quad (67)$$

where the state vector  $\xi_k = [p_{x_k} \ v_{x_k} \ p_{y_k} \ v_{y_k} \ \omega_k]^T$ ;  $p_{x_k}$  and  $p_{y_k}$  are the position in  $x$  and  $y$  coordinate;  $v_{x_k}$  and  $v_{y_k}$  are the corresponding velocities at the instant  $k$ .  $\tau$  indicates the time interval between two consecutive measurements.  $w_k$  is zero mean Gaussian noise (white) which indicates the modeling error. The process noise for this noise sequence is considered as

$$Q_k = \begin{bmatrix} \frac{0.1\tau^3}{3} & \frac{0.1\tau^2}{2} & 0 & 0 & 0 \\ \frac{0.1\tau^2}{2} & 0.1\tau & 0 & 0 & 0 \\ 0 & 0 & \frac{0.1\tau^3}{3} & \frac{0.1\tau^2}{2} & 0 \\ 0 & 0 & \frac{0.1\tau^2}{2} & 0.1\tau & 0 \\ 0 & 0 & 0 & 0 & q\tau \end{bmatrix} \quad (68)$$

Note, that the element  $Q_k(5, 5)$  is the noise covariance of respective augmented parameter, i.e., turn rate of the aircraft. As the turn rate is unknown and time varying the accurate knowledge of  $q$  is unavailable and has to be assumed arbitrarily for traditional non adaptive information filter. However, the value,  $q = (1.323 \times 10^{-2} \text{ rads}^{-1})^2$ , is selected to generate the true state trajectories for this case study and henceforth would be termed as  $q_{true}$ .

The trajectories of the aircraft are tracked by the fusion of the bearing angle measurements from two tracking radars which are positioned in two different locations of the atmosphere. The measurement equations can be represented as

$$\theta_k^\zeta = \tan^{-1} \left( \frac{p_{y_k} - p_{y_{ref}}^\zeta}{p_{x_k} - p_{x_{ref}}^\zeta} \right) + v_k^\zeta \quad \zeta = 1, 2 \quad (69)$$

$\zeta$  indicates position of the  $\zeta$ th radar.  $p_{y_{ref}}^1 = -10^4 \text{ m}$ ;  $p_{x_{ref}}^1 = -10^4 \text{ m}$ ;  $p_{y_{ref}}^2 = 10^4 \text{ m}$ ;  $p_{x_{ref}}^2 = 10^4 \text{ m}$ . The zero mean measurement noise (Gaussian) sequences have covariances  $R_1 = (\sqrt{30} \text{ m rad})^2$  and  $R_2 = (\sqrt{40} \text{ m rad})^2$ . The interval between two successive measurements is,  $\tau = 1 \text{ s}$ .

## 4.2 Simulation Procedure and Parameters

The proposed filtering algorithms are validated with the help of Monte Carlo simulation with 10000 runs. For generation of true state trajectories an initial choice of state is made as  $x_0 = [1000 \text{ m } 300 \text{ ms}^{-1} \ 1000 \text{ m } 0 \text{ ms}^{-1} \ -0.05235 \text{ rad s}^{-1}]^T$ . The unknown element of  $\mathbf{Q}$  is chosen as  $q_{true}$  to generate the true trajectories. The filters are initialized with a Gaussian prior with mean  $x_0$  and  $\hat{P}_0$ , where  $\hat{P}_0 = \text{diag}([100 \ 10 \ 100 \ 10 \ 10^{-4}])$ . As the element  $q_{true}$  is practically unknown to the filters, we assume that the filters with an assumed value of the that unknown element which may be termed  $q_{filter}$ . To induce uncertainty we assume  $q_{filter} = 20 * q_{true}$ . Such a choice is reasonable as it signifies high degree of uncertainty about the nature of variation of unknown time varying turn rate.

Note that only the element of  $\mathbf{Q}$  which is associated with the turn rate is unknown while the other elements are known. Therefore, we need to adapt the unknown element of  $\mathbf{Q}$  only keeping others frozen at their known truth value. This is justified as the derivation of adapted  $\mathbf{Q}$  is done by partial differentiation with respect to each diagonal element of  $\mathbf{Q}$ . For further discussions readers are referred to [13].

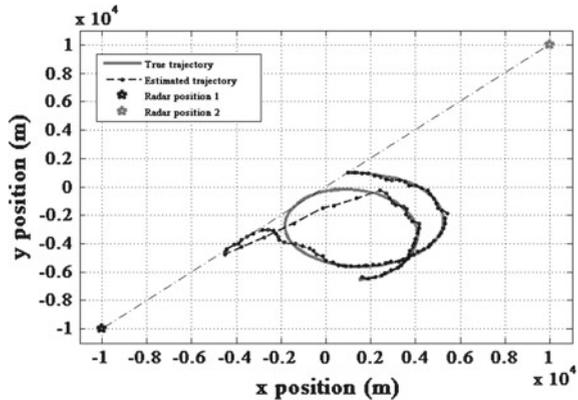
Root means square error (RMSE) of position, velocity and turn rate estimation are calculated for performance analysis of the proposed filter. The RMSE for position and velocity and turn rate are computed using the formula given in [6].

$$RMSE = \sqrt{\frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} \left( \left( \xi_{k,i} e_j - \hat{\xi}_{k,i} e_j \right)^2 + \left( \xi_{k,i} e_l - \hat{\xi}_{k,i} e_l \right)^2 \right)} \quad (70)$$

where  $j = 1$  and  $l = 3$  for RMSE of position estimation. For RMSE of velocity estimation  $j = 2$  and  $l = 4$ . RMSE for turn rate estimation is obtained with  $j = 5$  and replacing the unit vector  $e_l$  by a zero vector.

Further investigation with this tracking problem revealed the fact that this particular bearing only tracking problem is susceptible to track losses because of the non unique solution of the measurement equations. The true trajectories of aircraft have been randomly generated as a consequence of time varying turn rate as discussed in the previous section. For some time instants the trajectory of the aircraft become such that the difference between the bearing angle from two different radars may either be negligibly small or become closer to  $\pi$ . Practically, at this moment the line of sight of two radars does not intersect each other and the aircraft can not be precisely located. It is to be noted that in [6] which considers the same tracking problem the track loss phenomenon has not been discussed. Figure 1 has been presented for illustration where track loss has occurred for the non adaptive filter in the ideal situation. The performance of each of the proposed adaptive filters has also been compared with its respective non adaptive counter part as well as the other alternative adaptive filters in the context of RMSE performance and their susceptibility to the occurrence of track loss. To detect the occurrence of track loss the following condition has been

**Fig. 1** A representative run to illustrate track loss for the ideal case when the non adaptive DDIF has known  $Q$

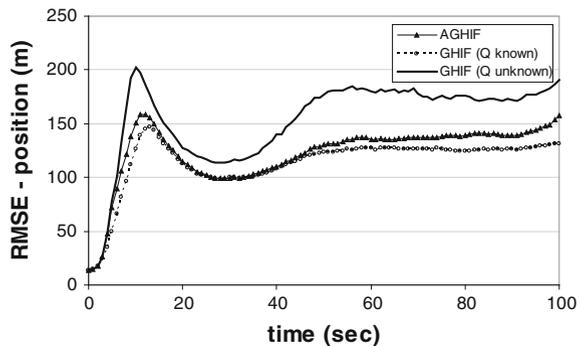


considered. When the condition  $\left\| \sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2} \right\|_{\infty} \geq 800 \text{ m}$  is true for any instance of Monte Carlo runs, it is understood that the estimated trajectory has failed to track the true trajectory of the aircraft.

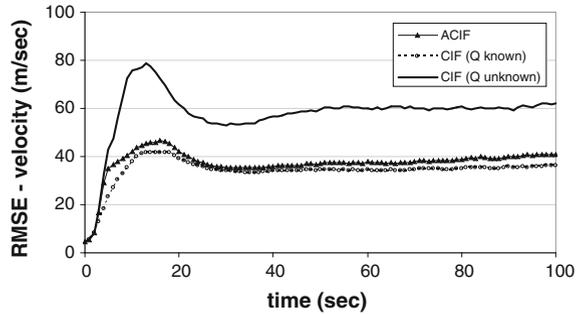
### 4.3 Simulation Results

From the Monte Carlo simulation with 10000 runs, performance of each of the proposed adaptive filter is compared with that of its non adaptive version in the situation when the turn rate of the aircraft is unknown. In each case it is observed that estimation performance of adaptive filter is superior compared to the non adaptive version. Some of the RMSE plots of such comparison are presented for illustration in Figs. 2, 3 and 4. The RMSE plots are presented excluding the cases where track loss occurs.

**Fig. 2** Comparison of RMSE (position estimation) of AGHIF & GHIF for 10000 MC runs



**Fig. 3** Comparison of RMSE (velocity estimation) of ACIF & CIF for 10000 MC runs



It has been observed from Figs. 2, 3 and 4 that the performance of adaptive filters are substantially superior to that of non adaptive ones as the RMSE for all three states converged to a lower steady state value within reasonably less time. Figure 2 presents RMSE of adaptive and non adaptive GHIF for position estimation. Figure 3 presents the RMSE of adaptive and non adaptive CIF (3rd degree) for velocity estimation and Fig. 4 depicts the RMSE of adaptive and non adaptive CQIF for turn rate estimation. The results for ADDIF are provided in [14]<sup>1</sup> and, therefore, are not presented here. It is also to be noted that although the elements of  $\mathbf{Q}$  related to position and velocity are known RMSE of position and velocity for the non adaptive filters are deteriorated because of the implicit effect of unsatisfactorily estimated turn rate.

As the proposed filters are validated in simulation, it is also possible to compare the RMSE performance of proposed filters with their non adaptive counterpart in the ideal situation when  $q$  is known only to the non adaptive version. Though this comparison may sound impractical, this comparison illumines how far the performance of adaptive filters (without complete knowledge of  $\mathbf{Q}$ ) is close to that of conventional non adaptive filter in ideal situation with known  $\mathbf{Q}$ . It is demonstrated that the RMSE of adaptive filters for all the states are nearly closed to the nature of RMSE of non adaptive filter in ideal condition. The initial mismatch in RMSE is because of the time taken for adapted element of  $\mathbf{Q}$  to converge on the truth value which has been shown in Fig. 5.

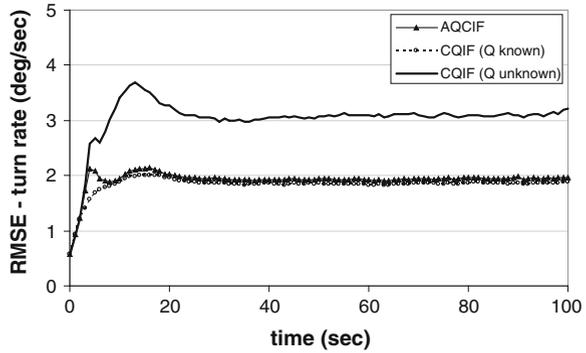
Figure 5 illustrates the  $\mathbf{Q}$  adaptation performance of the adaptive filters. For all of the proposed filters the adapted value of  $\mathbf{Q}$  converged to the truth value even though initialized with an assumed value with large error. The time to converge is near about 30 s. The adapted value keeps tracking the truth value after convergence.

In this particular case study the present authors have also compared the performances of ADDIF, ACIF (3rd degree), AGHIF, and ACQIF on the basis of RMS errors and percentage of track loss out of 10000 Monte Carlo runs.

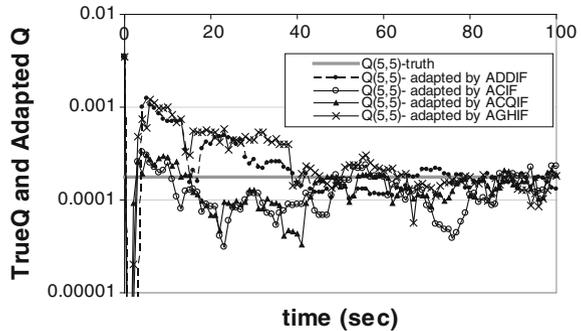
It has been observed from the RMSE plot of position estimation in Fig. 6 (excluding the track loss case) that the RMSE of proposed adaptive filters are performance wise comparable. Same trend is observed for velocity and turn rate estimation and

<sup>1</sup>The present paper is an extended version of the conference paper [14] presented in 12th International Conference on Informatics in Control, Automation and Robotics, 2015.

**Fig. 4** Comparison of RMSE (turn rate estimation) of ACQIF & CQIF for 10000 MC runs



**Fig. 5** Plot of estimated process noise covariance ( $Q_{5,5}$ ) for a representative run

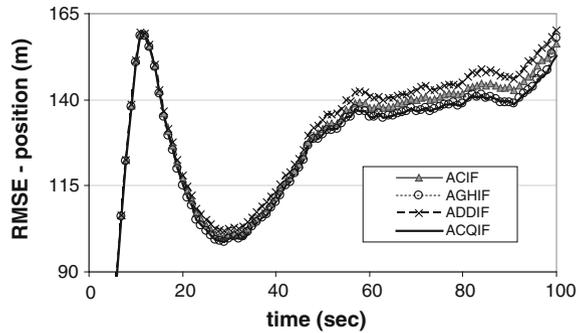


**Table 1** I: Comparative study of percentage of track loss

Underlying framework used	Percentage of track loss for		
	Adaptive filter with unknown Q (%)	Non adaptive filter with unknown Q (%)	Non adaptive filter with known Q (ideal) (%)
CIF	1.02	4.10	0.91
DDIF	1.14	4.17	0.97
GHIF	0.84	3.67	0.64
CQIF	0.86	3.69	0.64

are not presented to avoid repetition. However, on the basis of track loss performance (Table 1) it may be commented that AGHIF and ACQIF are better alternatives than ADDIF and ACIF as they indicate less percentage of track loss. Therefore, for this case study AGHIF and ACQIF may be preferred over ADDIF and ACIF. It is also to be noted here that though AGHIF and ACQIF are performance wise equivalent the latter is computationally less expensive. It is mentioned that the quadrature points for AGHIF rises exponentially with the system dimension. On contrary, the quadrature points for ACQIF rises linearly with the system dimension.

**Fig. 6** Comparison of RMS error (position estimation) of proposed filters for 10000 MC runs



## 5 Concluding Discussions

A number of Adaptive Nonlinear Information filters have been proposed for multiple sensor fusion in the situation with parametric uncertainty in the system dynamics and exemplified with the help of an aircraft tracking problem. The proposed filters are observed to present satisfactory estimation performance by online adaptation of  $Q$  where the complete knowledge of  $Q$  remains unavailable because of unknown parameter variation. For each of the proposed adaptive filters it is observed that the adapted value of the unknown element of  $Q$  converges on its truth value and subsequently tracks it. The results from Monte Carlo study demonstrate that the RMS error performance of each of the proposed filters is superior to its non adaptive counterpart. The performance comparison of proposed alternative adaptive information filters revealed that for the case study addressed in this paper the estimation performance of ACQIF and AGHIF is superior to ADDIF and ACIF. Although the estimation performance of ACQIF and AGHIF are comparably same use of the former is preferable because of the advantage of economic computation. As the analytical proof of optimality and convergence of proposed adaptive information filters are not yet available like other adaptive filters, these algorithms need to be thoroughly evaluated in simulation involving various nonlinear signal models for appropriate selection for real time application with confidence. Towards this objective some significant findings have been enumerated in this paper. The aspect of the adaptation of the unknown  $Q$ , flexibility for multiple sensor estimation framework and satisfactory estimation performance makes the proposed filters suitable candidates for multiple sensor fusion where the system dynamics is perturbed with unknown parameter variation.

**Acknowledgments** The first author thanks Council of Scientific and Industrial Research, New Delhi, India for financial support and expresses his gratitude to Centre for Knowledge Based System, Jadavpur University, Kolkata, India for infrastructural support.

## References

1. Vercauteren, T., Wang, X.: Decentralized sigma-point information filters for target tracking in collaborative sensor networks. *IEEE Trans. Signal Process.* **53**(8), 2997–3009 (2005)
2. Lee, D.J.: Nonlinear estimation and multiple sensor fusion using unscented information filtering. *IEEE Signal Process. Lett.* **15**, 861–864 (2008)
3. Anderson, B.D.O., Moore, J.B.: *Optimal Filtering*, 1st edn. Prentice Hall, Englewood Cliffs (1979)
4. Liu, G., Worgotter, F., Markelic, I.: Nonlinear estimation using central difference information filter. In: *Proceedings IEEE Workshop Statistical Signal Processing*, pp. 593–596. (2011)
5. Chandra, K.B., Gu, D.W., Postlethwaite, I.: Square root cubature information filter. *IEEE Sens. J.* **13**(2), 750–758 (2013)
6. Jia, B., Xin, M., Pham, K., Blasch, E., Chen, G.: Multiple sensor estimation using a high-degree cubature information filter. In: *Proceedings SPIE Defense, Security, and Sensing. International Society for Optics and Photonics*, 87390T–87390T (2013)
7. Ge, Q., Daxing, X., Chenglin, W.: Cubature information filters with correlated noises and their applications in decentralized fusion. *J Signal Process.* **94**, 434–444 (2014)
8. Maybeck, P.S.: *Stochastic Models, Estimation, and Control*, 1st edn. Academic Press, New York (1982)
9. Mohamed, A.H., Schwarz, K.P.: Adaptive Kalman filtering for INS/GPS. *J. Geodesy* **73**(4), 193–203 (1999)
10. Hajiyeve, C., Soken, H.E.: Robust adaptive unscented Kalman filter for attitude estimation of pico satellites. *Int. J. Adapt. Control Signal Process.* **28**(2), 107–120 (2014)
11. Lee, D.J.: *Nonlinear Bayesian filtering with applications to estimation and navigation*, In: Ph.D. thesis, Texas A&M University (2005)
12. Soken, H., Sakai, S.: Adaptive tuning of the UKF for satellite attitude estimation. *J. Aerosp. Eng.* **28**(3) (2015)
13. Dey, A., Sadhu, S., Ghoshal, T.K.: Adaptive Gauss Hermite filter for parameter varying nonlinear systems. In: *Proceedings International Conference on Signal Processing and Communication*, pp. 1–5 (2014). doi:[10.1109/SPCOM.2014.6983948](https://doi.org/10.1109/SPCOM.2014.6983948)
14. Dey, A., Sadhu, S., Ghoshal, T.K.: Multiple sensor fusion using adaptive divided difference information filter. In: *Proceedings 12th International Conference on Informatics in Control, Automation and Robotics*, Colmar, France, pp. 398–406 (2015)
15. Ge, Q.B., Wen, C.L., Chen, S.D., Sun, R.Y.: Adaptive cubature strong tracking information filter using variational Bayesian method. In: *Proceedings 19th World Congress of the International Federation of Automatic Control*, Cape Town, South Africa, pp. 24–29 (2014)
16. Nørgaard, M., Poulsen, N.K., Ravn, O.: New developments in state estimation for nonlinear systems. *Automatica* **36**(11), 1627–1638 (2000)
17. Bhaumik, S., Swati: Cubature quadrature Kalman filter. *IET Signal Process.* **7**(7), 533–541 (2013)

# Decentralized Control: Application to a Turboprop Engine

Christophe Le Brun, Emmanuel Godoy, Dominique Beauvois,  
Bogdan Liacu and Ricardo Noguera

**Abstract** This paper proposes a straightforward and systematic way of designing a decentralized control. The main setup steps are the interaction analysis, the decoupling of the interactions, and the design of the controllers. The proposed procedure is applied on identified models of a turboprop engine. The interaction analysis leads to the choice of a decentralized strategy with a full compensator. On each operating point, an inverted decoupler is selected in order to reduce interactions, and PI controllers are designed using an analytical method. The control laws robustness is then ensured using the structured singular value approach. Finally, control laws performances are validated using the non-linear simulation model of turboprop engine.

**Keywords** Decentralized control · Interaction analysis · Decoupling techniques · Robustness analysis · Multivariable processes · PID controllers

## 1 Introduction

In multivariable systems, each manipulated variable may affect several controlled variables, causing interaction between the loops. In many practical situations, the design of a full MIMO (Multiple-Input Multiple-Output) controller is cumbersome

---

C. Le Brun (✉) · E. Godoy · D. Beauvois  
Automatic Control Department, CentraleSupélec, Gif-sur-Yvette, France  
e-mail: christophe.lebrun@supelec.fr

E. Godoy  
e-mail: emmanuel.godoy@supelec.fr

D. Beauvois  
e-mail: dominique.beauvois@supelec.fr

C. Le Brun · B. Liacu  
Systems Division, SNECMA (SAFRAN), Villaroche, France  
e-mail: bogdancristian.liacu@sneema.fr

C. Le Brun · R. Noguera  
DynFluid Laboratory, Arts Et Métiers ParisTech, Paris, France  
e-mail: ricardo.noguera@ensam.eu

and high-order controllers are generally obtained. The decentralized strategy consists in dividing the MIMO process into a combination of several SISO (Single-Input Single-Output) processes and to design monovariable controllers in order to drive the MIMO process. Due to important benefits, such as flexibility as well as design and tuning simplicity, decentralized control design techniques are largely preferred in industry.

This paper is an extension of [1], which presents a preliminary study of an alternative control solution for a turboprop engine. This paper is organized as follows: Sect. 2 introduces the turboprop engine and its functioning. The interaction analysis is then presented in Sect. 3. Sections 4 and 5 expose the decoupling techniques and the PID tuning. Robustness analysis and simulation results demonstrate the efficiency of the control laws in Sects. 6 and 7 before presenting conclusions and perspectives in Sect. 8.

## 2 Functioning of a Turboprop Engine

### 2.1 Turboprop Overview

Basically, a turboprop engine [2] includes an intake, 1–2 compressors, a combustor, 1–3 turbines, a reduction gearing and a variable pitch propeller. Air is drawn into the intake and compressed until it reaches the desired pressure, speed and temperature. Fuel is then injected in the combustion chamber, where the fuel-air mixture is combusted. The hot combustion gases expand through the turbine. The power generated by the turbine is transmitted through the reduction gearing to the propeller, which generates the thrust of the turboprop engine. Thanks to its variable pitch, the propeller turns at constant speed. From the control point of view, the turboprop engine is a TITO (Two-Input Two-Output) process. Usually, the fuel flow  $WF$  is used to control the shaft power  $SHP$ , while the blade pitch angle  $\beta$  is used to control the propeller speed  $XNP$ . The propeller speed is impacted in case of fuel flow changes and similarly, when varying the blade pitch angle to change the propeller speed to another level, the shaft power is affected, particularly during the transient states. Fast transitions may generate over-torques with damaging mechanical impacts [3].

### 2.2 Technical Specifications

Technical specifications are described in Table 1. Beside these technical specifications, couplings between inputs and outputs have to be reduced as much as possible and control laws have to be robust to model uncertainties. Moreover, if modifications are required following bench tests or objectives updates, the control laws have

**Table 1** Technical requirements

Output	Bandwidth	Stability margin (dB)	Overshoot (%)
<i>SHP</i>	$\omega_{c1} = 2 \times 10^{-2}$	45–6	1
<i>XNP</i>	$\omega_{c2} = 5 \times 10^{-2}$	45–6	5

to be easily tunable. According to the introduction, a decentralized strategy has been chosen in order to respect these last objectives.

*Note:* in the paper, all the frequency characteristics are given in a normalized way. The following notations are used:  $G_o$  is the static gain matrix of the process  $G$ , and  $G^*$  is the matrix composed of the diagonal elements of  $G$ . The methods are presented with discrete-time models but the transposition in the continuous domain is straightforward.

### 2.3 Plant Identification

The behavior of the turboprop engine depends on the altitude, the Mach number and the engine rotation speed. A numerical identification (least squares method) has been done, based on step responses at different operating points obtained from a complete non-linear simulator of the turboprop engine. Linear discrete models of second order have been determined to represent the behavior of the turboprop (1), (2). The sampling time  $T_e$  has been taken in agreement with the digital electronic unit of the engine. Bode diagrams of the identified models are represented in Fig. 1.

$$\begin{bmatrix} SHP[k + 1] \\ XNP[k + 1] \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} SHP[k] \\ XNP[k] \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} WF[k] \\ \beta[k] \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} SHP(z) \\ XNP(z) \end{bmatrix} = \frac{1}{(z - p_1)(z - p_2)} \begin{bmatrix} K_{11}(z - z_{11}) & K_{12}(z - z_{12}) \\ K_{21}(z - z_{21}) & K_{22}(z - z_{22}) \end{bmatrix} \begin{bmatrix} WF(z) \\ \beta(z) \end{bmatrix} \quad (2)$$

## 3 Interaction Analysis

### 3.1 Objectives of the Analysis

For significant interactions, a decentralized control may not be adapted due to its limited structure. The interaction level strongly depends on the loop configuration, i.e. the manner in which the manipulated variables and the controlled variables have been associated.

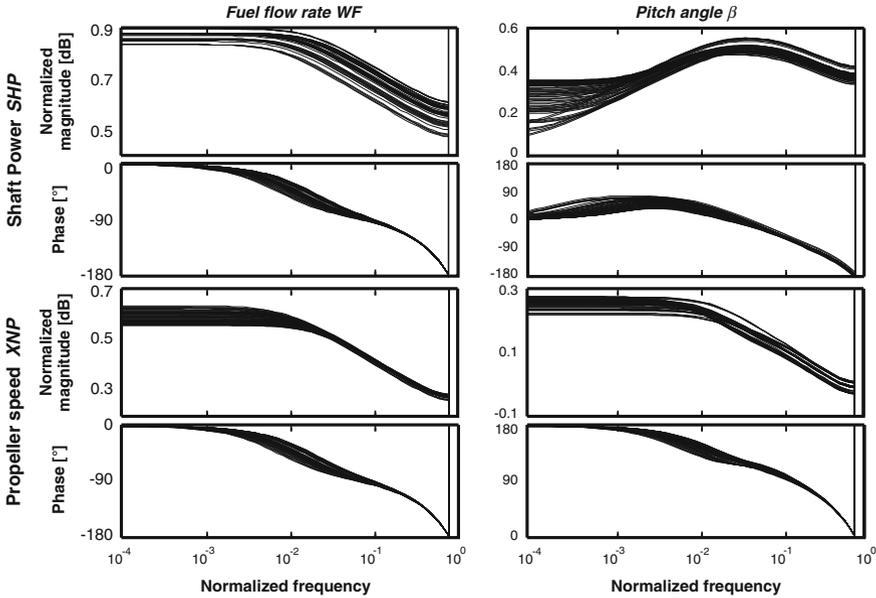


Fig. 1 Bode diagrams of the identified models of the turboprop engine

If a decentralized control seems not appropriate, a decoupling network can be used to reduce the existing process interactions before designing a decentralized controller. The choice of the structure and the computation of the decoupling network depend on the level of interaction.

The second step is to design the monovariate controllers for each loop. It is possible to use single-loop or multi-loop design methods. The first ones do not take into account the interactions and do not guarantee the performances of the multivariable closed-loop system. The second ones take the interactions into account but are more cumbersome. It can thus be interesting to have a metric to evaluate if a multi-loop tuning method is necessary or not.

### 3.2 Proposed Procedure

Interaction measures have been benchmarked in [4] on such academic processes, as three-tank and four-tank systems. It has been noted through this study that the interaction measures are not adapted to the set of previous objectives (choice of a: control strategy, loop pairing, tuning method). According to their performances and ease of use, some interaction measures have been put forward. The proposed procedure includes four complementary interaction measures for TITO processes. Some advices are given for MIMO plants as well.

**Relative Gain Array.** The Relative Gain Array (RGA) developed by [5] gives a suggestion on how to solve the pairing problem. By denoting  $\otimes$  the element-wise multiplication, the matrix RGA is given by:

$$RGA = G_0 \otimes (G_0^{-1})^T \quad (3)$$

The element  $RGA_{ij}$  can be seen as the quotient between the static gain of the transfer between input  $j$  and output  $i$  when all other loops are open, and the static gain of the same input/output pairing when all other loops are closed. The input/output pairings leading to elements close to 1 should be selected.

The RGA has been selected because it provides a quite simple way of choosing the pairing configuration. This index is pairing-independent, that means the RGA is computed only one time and gives information for every loop configuration, unlike some other indexes that have to be computed and analyzed for each loop configuration.

**Column Diagonal Dominance.** The column diagonal dominance (DD) [6] is defined as the ratio between the gain of the diagonal element and the sum of the gains of the off-diagonal column elements of the process transfer matrix (4). Values of  $DD_i$  greater than 1 will indicate weak interactions.

$$DD_i(G(z)) = \frac{|G_{ii}(z)|}{\sum_{j \neq i} |G_{ji}(z)|}, z = e^{j\omega T_e} \quad (4)$$

The DD has been chosen since it allows to study the interactions specifically for each loop using a dynamic approach.

**Performance Relative Gain Array.** In the frequency region where the control is effective, the true sensitivity matrix  $S$  (5) can be estimated using the decentralized sensitivity matrix  $S^*$  (6) and the Performance Relative Gain Array PRGA (8) [7]. The following equations resume the PRGA theory:

$$S(z) = (I + G(z)K(z))^{-1} \quad (5)$$

$$S^*(z) = (I + G^*(z)K(z))^{-1} \quad (6)$$

$$PRGA(z) = G^*(z)G^{-1}(z) \quad (7)$$

$$S^*(z) \approx 0 \Rightarrow S(z) \approx S^*(z)PRGA(z) \quad (8)$$

The PRGA has been chosen because it evaluates the difference between the performances of the single-loops and the real performances of the closed-loop multivariable process (for instance, large elements of the PRGA indicate that fast control is needed to get acceptable reference tracking).

**Index  $\Sigma_2$ .** The Index  $\Sigma_2$  [8] is obtained by an approach based on the  $H_2$ -norm. The  $H_2$ -norm can be interpreted as the transmitted energy between the inputs and the outputs. Hence, the matrix  $\Sigma_2$  (9) is suitable for quantifying the importance of the input-output channels. Each element describes the normalized impact of the

corresponding input signal on the specific output signal. For greater values of  $\Sigma_{2,ij}$ , the corresponding controllers  $K_{ij}$  have to be introduced. The control laws decoupling performances are directly linked to the sum of the selected  $\Sigma_{2,ij}$ , which should be close to 1 for optimal results.

$$\Sigma_{2,ij} = \frac{\|G_{ij}\|_2}{\sum_{k,l} \|G_{kl}\|_2} \tag{9}$$

Other similar Gramians based approaches (HIIA [9], PM [10]) could be featured but the  $\Sigma_2$  Index has been chosen because of its energy interpretation, and it was found in [11] that it reveals more clearly the high-frequency interactions.

**Procedure.** The proposed procedure for TITO normalized processes is described in Fig. 2a. The RGA is firstly used to choose an input/output pairing. Due to some limitations of the RGA, the diagonal dominance is used to corroborate the choice of the loop configuration. If RGA and DD have highlighted that a decentralized control can be used with a specific control configuration, the PRGA indicates if a mono-loop or a multi-loop tuning method has to be used. On the other case, the index  $\Sigma_2$  can be used to determine the choice of the controller structure. For a non TITO process, the procedure is described in Fig. 2b. The Niederlinsky Index NI [12] can be used to eliminate some configurations. Moreover, the RGA can be replaced by the Decomposed Relative Interaction Analysis (DRIA) [13], which gives important insight into cause-effects of loop interactions. Indeed, the DRIA is more adapted than the RGA because it examines the interactions between each loop and the others. The other indexes (DD, PRGA and  $\Sigma_2$ ) can be used as in the TITO case.

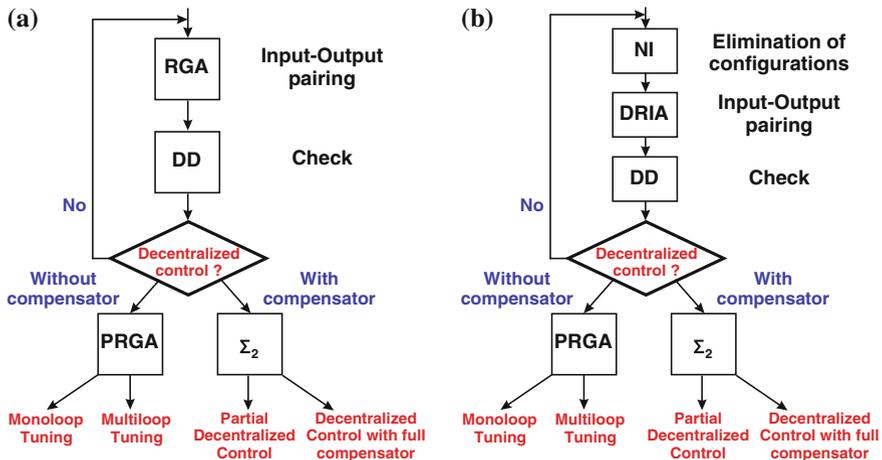


Fig. 2 Procedure of interaction analysis. a Procedure for TITO plants, b procedure for MIMO plants

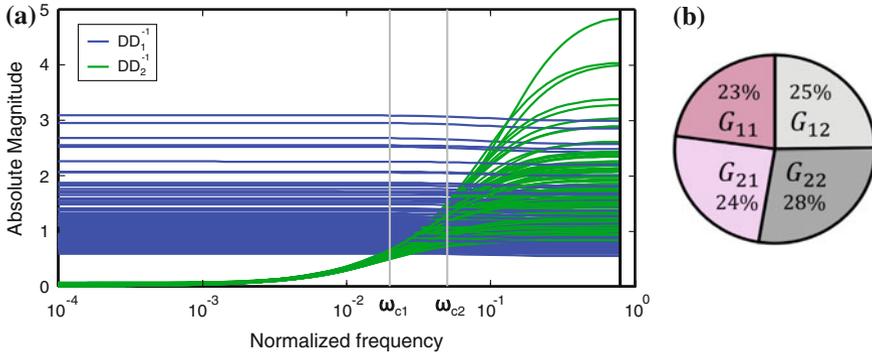


Fig. 3 Interaction analysis. **a** Column DD, **b**  $\Sigma_2$

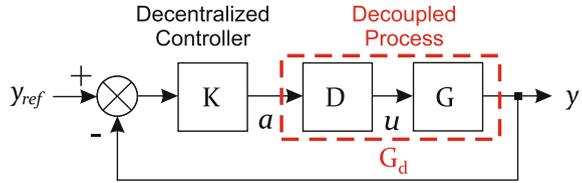
### 3.3 Interaction Analysis of the Turboprop Engine

The proposed procedure is applied to the turboprop engine model after the scaling and a first ordering of its inputs ( $WF$  and  $\beta$ ) and outputs ( $SHP$  and  $XNP$ ). The RGA is first computed on each operating point. The diagonal elements are contained between 0.9 and 1.1 and the off-diagonal elements between  $-0.1$  and  $0.1$ . The diagonal configuration is thus selected for pairing inputs and outputs. In order to evaluate more precisely interactions in the turboprop engine, the inverse of the column DD of identified models is plotted in Fig. 3a. The study of the column DD allows to notice that interactions are important from  $WF$  to  $XNP$  on the whole frequency domain. Interactions from  $\beta$  to  $SHP$  are neglectable at low frequencies (which mislead the RGA) and become important around the desired bandwidth and in high frequencies. A decentralized control is thus not viable. The  $\Sigma_2$  index is calculated to determine the structure of the desired compensator. The mean of the  $\Sigma_2$  matrices for various operating points is presented in Fig. 3b. It indicates that each transfer represents the same energy and cannot be neglected. A full compensator is thus required.

## 4 Decoupling Methods

The basic idea behind the control design based on decoupling is to find a compensator  $D$  in order to obtain a *near* diagonal process  $G_d$ , as shown in Fig. 4. The compensator can be static (i.e. constant matrix) or dynamic (i.e. transfer matrix). The advantage of the static approach is that the compensator is easier to be computed and to be implemented, whereas the dynamic approach allows leading to a better decoupling accuracy in a wider range of frequencies.

**Fig. 4** Decentralized controller with compensator



### 4.1 Proposed Procedure

The choice of a decoupling method is a relatively complex task. Decoupling methods have been benchmarked in [4] according to their complexity, the complexity of the obtained compensators and their performances. Performances have been compared by characterizing the interactions with the “decoupling index”  $\rho_i$  (10), defined as a trade-off between the frequency weighted mean  $DD_i^{-1}(j\omega)$  and the worst  $DD_i^{-1}(j\omega)$  over  $\omega$ .  $W$  is a frequency dependent weighting function that allows to emphasize the frequency band of interest around the desired bandwidth  $w_d$  (11).

The proposed procedure includes methods with different levels of complexity and performances. The methods are described in this part, before exposing the decoupling procedure.

$$\rho_i = \max_k(DD_i^{-1}(G(e^{j\omega_k T_e}))) + \frac{\sum_k W(\omega_k)DD_i^{-1}(G(e^{j\omega_k T_e}))}{\sum_k W(\omega_k)} \tag{10}$$

$$W(\omega_k) = \frac{\ln(|\frac{\omega_k + w_d}{\omega_k - w_d}|)}{\max_k(\ln(|\frac{\omega_k + w_d}{\omega_k - w_d}|))} \tag{11}$$

**Optimization Decoupling.** It is possible to search the elements of the optimal compensator that minimize the decoupling index  $\rho_i$ . Let  $L_i(G)$  be the  $i$ th row of  $G$  and  $C_j(D)$  the  $j$ th column of  $D$ . The elements of the compensated open-loop matrix  $G_d$  are given as follows:

$$G_{d,ij}(z) = L_i(G(z))C_j(D(z)) \tag{12}$$

The index  $DD_i$  of  $G_d$  depends only on the process and on the  $i$ th column of the compensator:

$$DD_i(G_d(z)) = \frac{|L_i(G(z))C_i(D(z))|}{\sum_{j \neq i} |L_j(G(z))C_j(D(z))|} \tag{13}$$

It is thus possible to maximize each index  $\rho_i$  independently. Moreover, it can be noticed that multiplying one column of  $D$  by a scalar does not affect the column  $DD$  nor the index  $\rho_i$ . A static compensator can thus be obtained in solving two optimization problems with one parameter for TITO plants (14), and a first order polynomial compensator is determined by solving two optimization problems with three parameters (15).

$$D = \begin{bmatrix} 1 & \beta \\ \alpha & 1 \end{bmatrix} \quad (14)$$

$$D(z) = \begin{bmatrix} 1 + \alpha_1 z & \beta_2 + \beta_3 z \\ \alpha_2 + \alpha_3 z & 1 + \beta_1 z \end{bmatrix} \quad (15)$$

The simulated annealing optimization algorithm is chosen, since it avoids being trapped in local minima in early iterations and is able to globally explore for better solutions. The resulting compensator is a polynomial matrix and hence improper. Multiplying every column of  $D$  with a transfer function of appropriate order makes the compensator proper and does not influence the decoupling accuracy.

The main advantage of this procedure is that it allows to find the optimal solution but it is time consuming and is thus difficult to implement practically when the order of the compensator or the size of the plant is important.

**Pseudo-diagonalization.** Ford and Daly [14] proposes to search a compensator by minimizing another cost-function  $v_i$  (16).  $v_i$  represents the ratio between the weighted sum of the diagonal element frequency gains and the weighted sum of the off-diagonal frequency gains. The advantage of this formulation lies in the optimization problem that can be solved analytically, using the multi-ALIGN algorithm [6, 15].

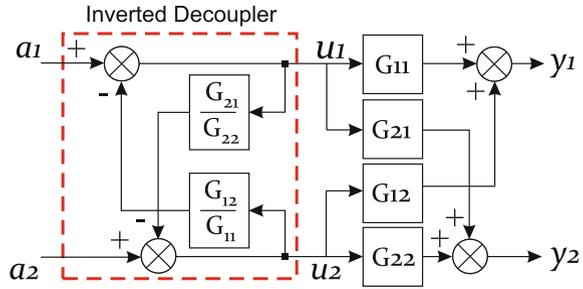
$$v_i(G) = \frac{\sum_k W_t(\omega_k) |L_i(G(j\omega_k)) C_i(D(j\omega_k))|^2}{\sum_k W_t(\omega_k) \sum_{j \neq i} |L_j(G(j\omega_k)) C_i(D(j\omega_k))|^2} \quad (16)$$

It is natural to choose  $W_t = W$ , but the weighting function  $W_t$  can also be determined to minimize the previously introduced index  $\rho_i$ . Assuming that this function has the same form as  $W$  (11), three parameters can be optimized: the desired bandwidth and the two limits of the frequency grid (the number of points of the frequency grid could also be optimized but the optimization problem would become complex).

The main advantage of this solution compared to the previous one is that the number of optimization parameters is constant, whatever the size of the plant and the order of the computed compensator. However, the optimization of these parameters is less efficient and the analytical solution provided by the pseudo-diagonalization could be based on the inversion of ill-conditioned matrices.

**Inverse-Based Decoupling.** The principle of the inverted decoupling (described in [16]) is to compute the decoupler  $D$  in order to get an open-loop diagonal transfer matrix with the same diagonal elements as the original plant (17), (18). After some algebraic manipulations (19), the structure of the inverted decoupler is presented in Fig. 5. The realizability problems can be solved by solutions proposed in [17], based on the addition of extra dynamics or time delays.

**Fig. 5** Scheme of the inverted decoupler



$$D(z) = G^{-1}(z)G^*(z) \tag{17}$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}^{-1} \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} a_1 - \frac{G_{12}}{G_{11}}u_2 \\ a_2 - \frac{G_{21}}{G_{22}}u_1 \end{bmatrix} \tag{19}$$

The ideal decoupling and the simplified decoupling are also based on this inverse-based approach. Ideal decoupling corresponds to the first part of the inverted decoupling described in (17). The main inconveniences of this method are the complexity of the decoupler elements and realizability problems. On the other hand, the simplified decoupling consists in selecting the decoupler as shown in (20). Decoupler network design is easier than the inverted decoupler; however the complexity of the apparent process  $GD$  is greater since its elements are the sum of transfer functions. The inverted decoupling has been chosen since it regroups the advantages of the two first approaches.

$$D(z) = \begin{bmatrix} 1 & -\frac{G_{12}(z)}{G_{11}(z)} \\ -\frac{G_{21}(z)}{G_{22}(z)} & 1 \end{bmatrix} \tag{20}$$

$$G(z)D(z) = \begin{bmatrix} G_{11}(z) - \frac{G_{21}(z)G_{12}(z)}{G_{22}(z)} & 0 \\ 0 & G_{22}(z) - \frac{G_{12}(z)G_{21}(z)}{G_{11}(z)} \end{bmatrix} \tag{21}$$

**Decoupling Procedure.** The first step of the procedure for TITO plants (Fig. 6a) is to compute the elements of the inverted decoupler in order to evaluate the complexity of a compensator that achieves perfect decoupling. The optimization of a static compensator and then a first order compensator are then applied. The pseudo-diagonalization is then applied. The order of the compensator can be increased until it reaches the complexity of the inverted decoupler. Finally, the inverted decoupler is chosen if the previous compensators do not lead to an acceptable decoupling.

For systems with a large number of inputs/outputs, an alternative procedure is presented in Fig. 6b: the pseudo-diagonalization can replace the optimization decoupling methods because of computation time constraints. Indeed, as presented before,

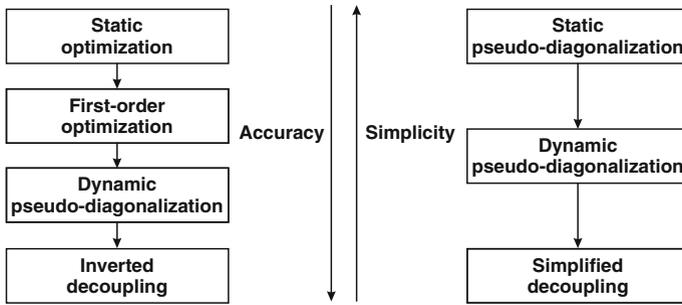


Fig. 6 Procedure of decoupling

the number of optimization parameters increases with the size of the plant in the case of the optimization decoupling method, whereas it stays constant for the pseudo-diagonalization method. Moreover, the simplified decoupling can be an interesting alternative to the inverted one when structures become complex.

### 4.2 Decoupling of the Turboprop Engine

The two elements of the inverted decoupler are computed in (22) and are composed of one zero and one pole. The requirements for the realizability of the inverted decoupler are respected. Thus, the dynamic compensator computed by optimization or pseudo-diagonalization must not exceed a full first order matrix transfer. An average normalized model  $G_n$  is considered in this part. The DD of this model is represented in Fig. 7. A static compensator is first researched under the form (14). A simulated annealing optimization leads to the results presented in Table 2 and Fig. 7. Couplings being too important, a first order compensator (15) is computed. Interactions are highly reduced, but they still remain important. The inverted decoupler is thus chosen.

$$\frac{G_{12}}{G_{11}} = \frac{K_{12}(z - z_{12})}{K_{11}(z - z_{11})} \quad \frac{G_{21}}{G_{22}} = \frac{K_{21}(z - z_{21})}{K_{22}(z - z_{22})} \quad (22)$$

## 5 Decentralized Control

The goal of this part is to design the monovariate controllers for a multivariable process or an apparent one (including a compensator). The interaction procedure, presented in Fig. 2, can be applied in order to choose between a single-loop or a multi-loop tuning method.

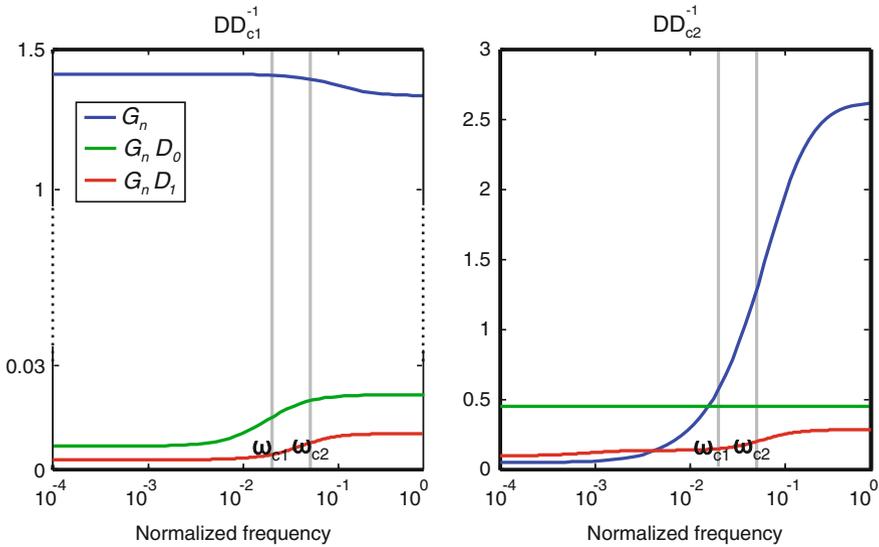


Fig. 7  $DD^{-1}$  of the process and decoupled processes

Table 2 Decoupling results

Compensator	$\rho_1$	$\rho_2$
Normalized process $G_n$	1.4	1.4
Static optimization	0.02	0.45
Dynamic optimization	0.007	0.21

### 5.1 Single-Loop Tuning Method

The single-loop tuning methods consist in designing each controller of the system using the corresponding diagonal element while ignoring the interactions with other loops. The classical methods can thus be used. The main advantage of single-loop tuning methods is the simplicity. However, due to the process interactions in MIMO systems, this approach cannot guarantee stability and desired performances of the multivariable closed-loop system.

### 5.2 Multi-loop Tuning Method

The multi-loop tuning methods allow to cope with the interaction effect, in order to improve multivariable closed-loop performances. Most of the existing design methods are based on the extension of single-loop tuning methods, and are used

to design PID controllers. The main categories of methods are briefly recalled here (for more details, some methods are described and benchmarked in [4]).

**Detuning Methods.** The detuning methods consist in applying first a single-loop method, and then in detuning the controllers to take into account interactions. The simplicity of these methods is their major advantage. However, the drawback results from the fact that loop performances and stability are not clearly ensured through the detuning procedures.

**Sequential/Iterative Methods.** The sequential and iterative methods propose to tune each loop one by one, taking into account the controllers already tuned. Unlike the sequential methods, the iterative methods propose to re-tune the controllers one after the other when all other loops are closed. These methods are relatively easy to implement and the final design ensures the stability and the desired performances for the multivariable system. However, the decentralized controller depends on the order chosen for the controller design, and the iterative procedure can raise convergence issues.

**Simultaneous Methods.** The simultaneous methods prescribe to tune simultaneously the controllers, using heuristic methods, or solving equations analytically or by optimization. Although these methods are less conservative than the previous ones, solving multi-loop controllers is not obvious to set up and is a numerically difficult problem.

**Independent Methods.** The independent methods consist in translating the interactions effects and desired multivariable closed-loop performances into constraints on sensibility functions of each loop. Controllers are then designed one by one while respecting the established constraints. These methods allow to ensure the multivariable and the monovariate performances but they are conservative (because the other controller dynamics are not specifically taking into account).

### 5.3 *Decentralized Control of the Turboprop Engine*

As previously mentioned, the loops are perfectly decoupled on each operating point thanks to the inverted decoupler. A mono-loop design method can thus be used. Considering the dynamics of diagonal elements of the system (Fig. 1), PI controllers can be used. Since the number of operating points is important, an analytical tuning method has been chosen. The IMC-PID (Internal Model Controller) [18] method has been selected since it provides a suitable framework for satisfying the desired objectives. For the process structure given in (23), the controller in (25) can reach the desired transfer function given in (24). Using MacLaurin series approximations, the obtained controller can be approximate by a PI controller (26), with the parameters defined in (27), (28).

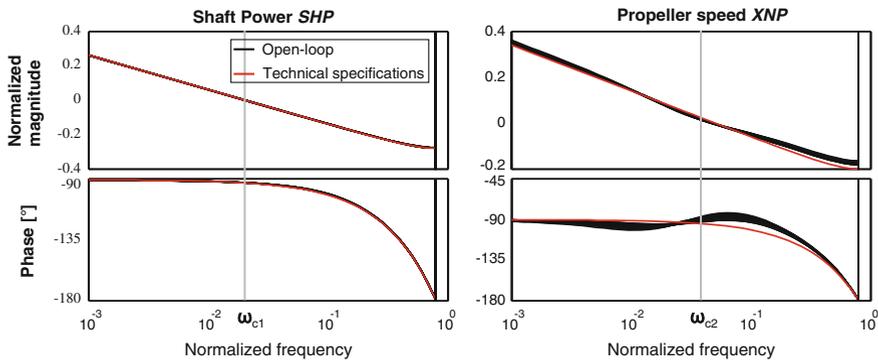


Fig. 8 Bode diagrams of the open-loop system

$$G_{ii} = k_{ii} \frac{(z - z_{ii})}{(z - p_1)(z - p_2)} \quad (23)$$

$$BO_i = \frac{a}{z - 1} \quad (24)$$

$$K_i(z) = G_{ii}^{-1}(z) BO_i(z) \quad (25)$$

$$K_p \left( 1 + \frac{1}{\tau_i} \frac{T_e}{z - 1} \right) \quad (26)$$

$$K_p = a \frac{((1 - p_1) + (1 - p_2))(1 - z_{ii}) - (1 - p_1)(1 - p_2)}{k_{ii}(1 - z_{ii})^2} \quad (27)$$

$$\tau_i = \frac{((1 - p_1) + (1 - p_2))(1 - z_{ii}) - (1 - p_1)(1 - p_2)}{(1 - z_{ii})(1 - p_1)(1 - p_2)} T_e \quad (28)$$

The Bode diagrams of each open-loop system (for the different operating points) are presented in Fig. 8 and compared to the desired frequency characteristics. It can be seen that the PI tuning allows obtaining a behavior close to the technical specifications.

## 6 Robustness Analysis

Even if the nominal performances meet the specifications with the plant model, it is still necessary to validate that the control laws ensure stability and *some* minimum performance specifications in presence of uncertainties. These last correspond, either to parameters' uncertainties of the model or to neglected dynamics. The structured singular value approach has been selected because it provides a general framework to robustness analysis problem [19].

### 6.1 Uncertain Turboprop Engine Under an LFT Form

The main issue of this part is to transform the closed-loop subject to model uncertainties, into the standard interconnection structure. Uncertainties can be considered on each of the eight parameters of the identified model (under state-space representation). In order to have meaningful uncertainties, the maximum variation range for each parameter is chosen as percentage of its possible range determined from a set of identified models (29), (30):

$$A_{ij} = A_{ij0} + x\% \times (A_{ij\text{sup}} - A_{ij\text{inf}})\delta_{A_{ij}} \tag{29}$$

$$B_{ij} = B_{ij0} + x\% \times (B_{ij\text{sup}} - B_{ij\text{inf}})\delta_{B_{ij}} \tag{30}$$

Moreover, some dynamics could have been neglected during the modelling or the identification steps. Neglected dynamics are thus introduced at the plant inputs: first order filters (with bandwidth five times greater than the desired bandwidths) are considered for each loop. The turboprop engine under LFT (Linear Fractional Transformation) form is presented in Fig. 9.

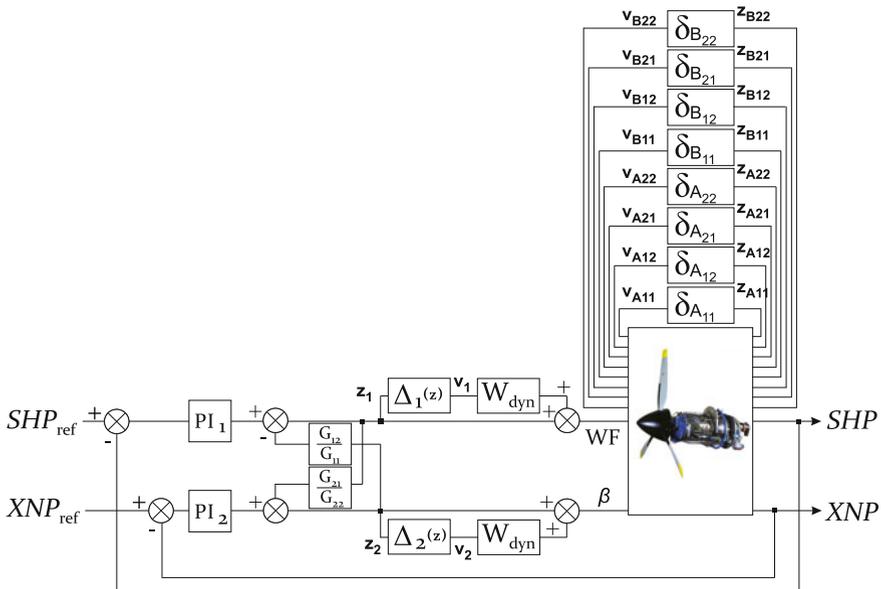
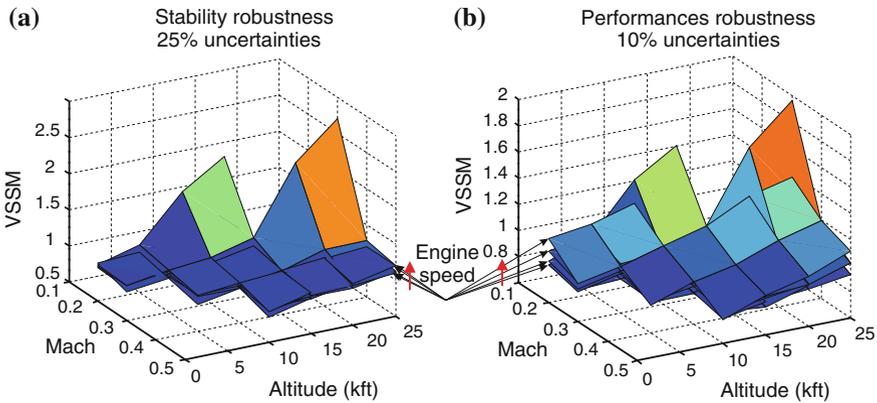


Fig. 9 Turboprop engine under LFT form



**Fig. 10** Upper bounds of the singular values. **a** VSSM stability, **b** VSSM performances

## 6.2 Results

For various operating points defined by a Mach number, an altitude and an engine speed, uncertainties of 25% on the parameters' ranges are considered in order to evaluate the robustness of the stability with the corresponding controller. The maximum of the upper bounds of the singular values (noted VSSM) are represented in the Fig. 10a. Except for three points that present VSSM greater than 1.5, stability is guaranteed for an uncertainty level of approximately 25%.

In order to test the performances robustness, an additional (fictitious) performance block is added to the model perturbation. This last allows ensuring modulus margins of 0.4 and half of the desired bandwidths. Uncertainties of 10% of the ranges of the parameters are considered. The VSSM are represented in the Fig. 10b. Except for the same three points of the previous case, the control laws maintain the chosen performances in terms of set-point tracking and stability margin with the uncertainty range of approximately 10%.

The control laws for the three operating points previously mentioned have been re-designed, with poorer nominal performances but better robust performances. Even if these results are satisfactory, the control laws designed for a particular operating point are not able to ensure the desired performances on the whole flight envelope, hence an interpolation strategy is needed.

## 6.3 Interpolation

A gain scheduling technique has been adopted. Moreover, an incremental algorithm (also called velocity algorithm) has been used to ensure bumpless parameter changes.

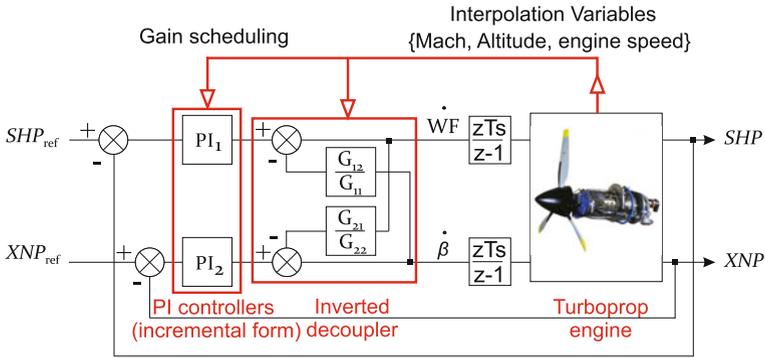


Fig. 11 Control laws implemented with an incremental algorithm

The algorithm first computes the change rate of the control signal, which is then fed to an integrator [20]. Finally, Fig. 11 presents the control laws in their final configuration.

### 7 Simulation Results

Control laws have been implemented on the non-linear model of the turboprop engine. The validation scenario includes successive reference steps, perturbations and noise. Simulation results are plotted in Figs. 12, 13. The time responses are in agreement with the specified bandwidths (considering the limitations on commands and their derivatives). Moreover, overshoots are not important and there are no steady-state errors. Some peaks are noticed on the propeller speed when there are important steps on the shaft power, but they are quickly corrected. Technical specifications are thus respected.

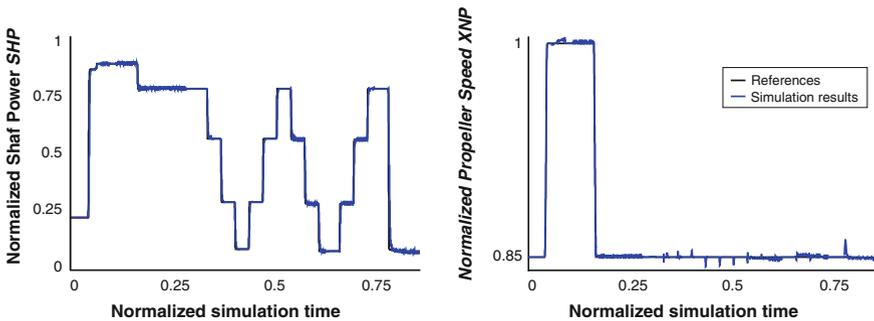


Fig. 12 Simulation results

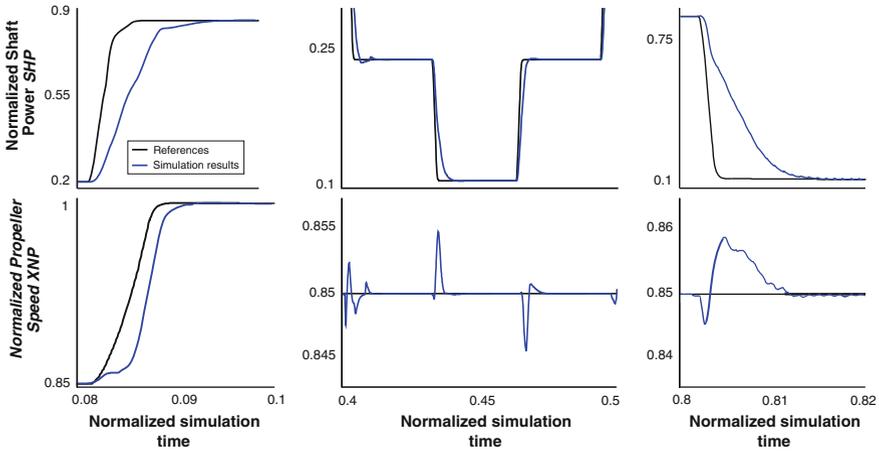


Fig. 13 Simulation results (zoom on some transient states)

## 8 Conclusions

This paper describes the different steps of the decentralized control design. The first step consists in analyzing the interactions of the process. Some decoupling methods are then proposed in order to reduce the existing interactions. Subsequently, the monovaryable controllers have to be tuned using mono-loop or multi-loop tuning methods, depending on the level of residual interactions. The different exposed methods have been applied on identified models of a turboprop engine. Given the high couplings of the process, a decentralized control with decoupler has been chosen. An inverted decoupler has been used in order to respect severe constraints on the couplings. PI controllers have then been tuned using an IMC-PID mono-loop tuning method. Control laws have finally been interpolated using a gain scheduling method in order to ensure the desired performances on the flight envelope. Robustness analysis and simulation results finally illustrate the good performances of the control laws. Future works will focus on the adaptation of the proposed methodology in order to take into account the uncertainties during the interaction analysis and the decoupling steps.

## References

1. Le Brun, C., Godoy, E., Beauvois, D., Liacu, B., Noguera, R.: Control laws design of a turboprop engine. *Appl. Mech. Mater.* **704**, 362–367 (2014)
2. Soares, C.: *Gas Turbines: A Handbook of Air, Land and Sea Application*. Butterworth-Heinemann (2008)
3. High, G.T., Prevallet, L.C., Free, J.W.: Apparatus for decoupling a torque loop from a speed loop in a power management system for turboprop engines. US 5274558 A (1991)

4. Le Brun: Analyse et commande de systèmes multivariables - Application à un turbopropulseur. PhD CentraleSupélec (2015)
5. Bristol, E.: On a new measure of interactions for multivariable process control. *IEEE Autom. Control* **11**(1), 133–134 (1966)
6. Maciejowski, J.M.: *Multivariable Feedback Design*. Addison Wiley (1989)
7. Hovd, M., Skogestad, S.: Simple frequency-dependent tools for control system analysis, structure selection and design. *Automatica* **28**(5), 989–996 (1992)
8. Birk, W., Medvedev, A.: A note on gramian-based interaction measures. In: *Proceedings of European Control Conference*, Cambridge, UK (2003)
9. Wittenmark, B., Salgado, M.E.: Hankel-norm based interaction measure for input-output pairing. *IFAC*, Barcelona, Spain (2002)
10. Salgado, M.E., Conley, A.: MIMO interaction measure and controller structure selection. *Int. J. Control* **77**(4), 367–383 (2004)
11. Halvarsson, B.: *Interaction Analysis in Multivariable Control Systems*. PhD, Uppsala University (2010)
12. Niederlinski, A.: A heuristic approach to the design of linear multivariable interacting control systems. *Automatica* **7**, 691–701 (1971)
13. He, M.J., Cai, W.-J.: New criterion for control-loop configuration of multivariable processes. *Ind. Eng. Chem. Res.* **43**(22), 7057–7064 (2004)
14. Ford, M.P., Daly, K.C.: Dominance improvement by pseudodecoupling. In: *Proceedings of the Institution of Electrical Engineers*, vol. 126, no. 12, pp. 1316–1320 (1979)
15. MacFarlane, A.G.J., Kouvaritakis, B.: A design technique for linear multivariable feedback systems. *Int. J. Control* **25**(6), 837–874 (1977)
16. Gagnon, E., Pomerleau, A., Desbiens, A.: Simplified, ideal or inverted decoupling? *ISA Trans.* **37**, 265–276 (1998)
17. Garrido, J., Vasquez, F., Morilla, F.: Generalized Inverted Decoupling for TITO processes. *IFAC*, Milano, Italy (2011)
18. Rivera, D.E., Morari, M., Skogestad, S.: Internal model control. 4. PID controller design. *Ind. Eng. Chem. Process Des. Dev.* **25**, 252 (1986)
19. Ferreres, G.: *A Practical Approach to Robustness Analysis with Aeronautical Applications*. Kluwer Academic Publishers (1999)
20. Åström, K.J., Häggglund, T.: *PID Controllers: Theory, Design, and Tuning*, 2nd edn. ISA (1995)

# Author Index

## A

Ababsa, Fakhreddine, 229  
Afanasyev, Ilya, 287  
Akhmedova, Shakhnaz, 79  
Amar, Faïz, 29  
Andreff, N., 157  
Andryushenko, Eugene, 79  
Angerer, Andreas, 209  
Awatsu, Lisa, 337

## B

Babiarz, Artur, 307  
Beauvois, Dominique, 391  
Berenguer, Yerai, 115  
Bouyakhf, El Houssine, 229  
Brun, Christophe, 391  
Burakov, Sergey, 79

## C

Chugo, Daisuke, 137  
Czornik, Adam, 307

## D

Dey, Aritro, 371  
Dormido, Raquel, 97  
Dormido, Sebastian, 97  
Dubinin, Petr, 79

## E

Ezin, Eugène, 63

## F

Fabregas, Ernesto, 97  
Fedorovici, Lucian-Ovidiu, 51  
Fritsche, Paul, 321

## G

Garstka, Jens, 267  
Ghoshal, Tapan, 371  
Gil, Arturo, 115  
Gil, Pablo, 197  
Godoy, Emmanuel, 391  
Gurko, Alexander, 177

## H

Hashimoto, Hiroshi, 137  
Hernandez, Wilmar, 177  
Hipólito, Juan, 177  
Hiyadi, Hajar, 229  
Hoffmann, Alwin, 209

## K

Kemmotsu, Masakatsu, 337  
Khusainov, Ramil, 287  
Kirichenko, Igor, 177  
Kowalczyk, Wojciech, 3  
Kozłowski, Krzysztof, 3  
Krid, Mohamed, 29

**L**

Laleye, Fréjus, [63](#)  
Liacu, Bogdan, [391](#)

**M**

Magid, Evgeni, [287](#)  
Marchand, E., [157](#)  
Marín, José, [115](#)  
Mateo, Carlos, [197](#)  
Montagne, Christophe, [229](#)  
Motamed, Cina, [63](#)  
Muramatsu, Satoshi, [137](#)  
Mutoh, Yasuhiko, [337](#)

**N**

Nicolai, Philip, [247](#)  
Niezabitowski, Michał, [307](#)  
Noguera, Ricardo, [391](#)

**O**

Ourak, M., [157](#)

**P**

Peidró, Adrián, [115](#)  
Peters, Gabriele, [267](#)  
Petriu, Emil, [51](#)  
Piksina, Oksana, [79](#)  
Precup, Radu-Emil, [51](#)  
Prüher, Jakub, [355](#)

**R**

Raczkowsky, Jörg, [247](#)  
Radac, Mircea-Bogdan, [51](#)  
Regragui, Fakhita, [229](#)

Reif, Wolfgang, [209](#)

Reinoso, Óscar, [115](#)

**S**

Sadhu, Smita, [371](#)  
Schierl, Andreas, [209](#)  
Semenkin, Eugene, [79](#)  
Sergiyenko, Oleg, [177](#)  
Shimchik, Ilya, [287](#)  
Šimandl, Miroslav, [355](#)  
Socas, Rafael, [97](#)

**T**

Tamadazte, B., [157](#)  
Torres, Fernando, [197](#)  
Tyrsa, Vera, [177](#)

**V**

Vistein, Michael, [209](#)  
Voisan, Emil-Ioan, [51](#)

**W**

Wagner, Bernardo, [321](#)  
Wörn, Heinz, [247](#)

**Y**

Yakimov, Igor, [79](#)  
Yokota, Sho, [137](#)

**Z**

Zaloga, Aleksandr, [79](#)  
Zamzami, Ziad, [29](#)