Advances in Intelligent Systems and Computing 253

Manuel A. Armada
Alberto Sanfeliu
Manuel Ferre *Editors*

ROBOT2013: First Iberian Robotics Conference

Advances in Robotics, Volume 2

Springer

Advances in Intelligent Systems and Computing

Volume 253

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/11156

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

Manuel A. Armada · Alberto Sanfeliu
Manuel Ferre
Editors

ROBOT2013: First Iberian Robotics Conference

Advances in Robotics, Volume 2

Springer

Editors
Manuel A. Armada
State Agency National Research Council (CSIC)
Madrid
Spain

Manuel Ferre
University Polytechnic of Madrid (UPM)
Madrid
Spain

Alberto Sanfeliu
University Technical of Catalonia (UPC)
Barcelona
Spain

ISSN 2194-5357 ISSN 2194-5365 (electronic)
ISBN 978-3-319-03652-6 ISBN 978-3-319-03653-3 (eBook)
DOI 10.1007/978-3-319-03653-3
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013952937

Preface

These Proceedings are a collection of selected papers presented at ROBOT 2013: FIRST IBERIAN ROBOTICS CONFERENCE, organised by the Sociedad Española para la Investigación y Desarrollo en Robótica (SEIDROB) and by the Centre for Automation and Robotics (CAR, a joint research centre from Universidad Politécnica de Madrid (UPM) and Consejo Superior de Investigaciones Científicas (CSIC)), along with the co-operation of "Grupo Temático de Robótica CEA"-GTRob, "Sociedade Portuguesa de Robotica" (SPR), "Asociación Española de Promoción de la Investigación en Agentes Físicos" (RedAF), and partially supported by "Comunidad de Madrid under RoboCity2030 Programme". This Conference, that was held in Madrid (28–29 November 2013), builds upon the highly successful previous three biannual Workshops that started in Zaragoza (2007), and continued in Barcelona (2009) and Sevilla (2011). Those previous and fruitful events, both from the standpoint of their scientific and technical quality, and for the important number of attending delegates, have motivated in organizing ROBOT 2013. ROBOT 2013's main goal was to continue the precedent efforts in presenting the most recent robotic research and the development of new applications. While previous events were focused on the Spanish activity, in this edition the core was extended to the Iberian Peninsula, although we welcome to delegates from other countries. The interest in robotics has remarkably augmented over recent years. Novel solutions for complex and very diverse application fields (exploration/intervention in severe environments, assistive, social, personal services, emergency rescue operations, transportation, entertainment, unmanned aerial vehicles, medical, etc.), has been anticipated by means of a large progress in this area of robotics. Moreover, the amalgamation of original ideas and related innovations, the search for new potential applications and the use of state of the art supporting technologies permit to foresee an important step forward and a significant socio-economic impact of advanced robot technology in the forthcoming years. In response to the technical challenges in the development of these sophisticated machines, a significant research and development effort has yet to be undertaken. It concerns embedded technologies (for power sources, actuators, sensors, information systems), new design methods, adapted control techniques for highly redundant systems, as well as operational and decisional autonomy and human/robot co-existence. Confirming this situation greatest awareness has been received

to ROBOT2013, and after a careful reviewing procedure the conference finally accommodate 106 papers of high quality, where the number of authors goes over 300.

As a summary of the conference, it can be said that included both state of the art and more practical presentations dealing with implementation problems, support technologies and future applications. A growing interest in Assistive Robotics, Agricultural Robotics, Field Robotics, Grasping and Dexterous Manipulation, Humanoid Robots, Intelligent Systems and Robotics, Telerobotics, Marine Robotics, has been demonstrated by the very relevant number of contributions. Moreover, ROBOT2013 incorporates a special session on Legal and Ethical Aspects in Robotics that is becoming a topic of key relevance. All in all, ROBOT 2013 has been an excellent forum for mutual exchange of knowledge among major stakeholders in the field of robotics and we are sure it will foster a better cooperation to face the challenges posed by Horizon 2020.

We would like to take this opportunity to thank all those involved in organizing ROBOT2013. To the Co-Chairs (Pedro J. Sanz, U. Jaume I, Spain, Vicente Matellán, U. León, Spain, Luis P. Reis, Universidade do Minho, Portugal), to the Plenary Speakers, to the Programme Committee, to the Consejo Superior de Investigaciones Científicas, and to the Escuela Superior de Ingenieros Industriales -UPM, who host the Conference, our acknowledgement for their invaluable help and kind assistance. Particularly, thanks are extensive to the Centre for Automation and Robotics - CAR (CSIC-UPM) colleagues, to its Technical and Administrative Staff and, with special mention to: Hector Montes Franceschi, Javier F. Sarria Paz, Roemi Emilia Fernández Saavedra and Fernando Delgado Medrano, because without their invaluable assistance ROBOT2013 would never been a sound reality.

To end this preface, special thanks to our editors, Springer, that are in charge of this Conference Proceedings edition, and in particular to Dr. Thomas Ditzinger (Springer, Applied Sciences and Engineering).

November 2013

Manuel Armada
Alberto Sanfeliu
Manuel Ferre

Organization

ROBOT 2013 is organized by La Sociedad para la Investigación y Desarrollo en Robótica SEIDROB and the Centre for Automation and Robotics CAR (UPM-CSIC).

Organizing Committee

Manuel A. Armada	CAR CSIC-UPM, Spain
Alberto Sanfeliu	IRI UPC-CSIC, Spain
Pedro J. Sanz	U. Jaume I, Spain
Vicente Matellán	U. León, Spain
Luis P. Reis	Universidade do Minho, Portugal

Program Committee

Mohamed Abderrahim	U. Carlos III de Madrid, Spain
Jon Agirre	Tecnalia, Spain
Eugenio Aguirre	U. Granada, Spain
Rachid Alami	CNRS-LAAS, France
José R. Alique	CAR CSIC-UPM, Spain
Luis Almeida	U. of Porto, FEUP, Portugal
Josep, Amat	U. Politécnica de Cataluña, Spain
Rafael, Aracil	CAR UPM-CSIC, Spain
Juan R., Astorga	Airbus Military, Spain
Carlos Balaguer	U. Carlos III de Madrid, Spain
Guilherme Barreto	U. Federal do Ceára, Brazil
Antonio Barrientos	CAR CSIC-UPM, Spain
Luis Basañez	U. Politécnica de Cataluña, Spain
Luis M. Bergasa	U. Alcalá de Henares, Spain
Reinaldo Bianchi	Centro universitário da FEI, Brazil
Fernando Caballero	U. Sevilla, Spain
João Calado	Ins. Sup. de Engenharia de Lisboa, Portugal

Pascual Campoy	CAR CSIC-UPM, Spain
José M. Cañas	U. Rey Juan Carlos, Spain
Carlos Cardeira	Lisbon Technical University/IDMEC, Portugal
Alícia Casals	U. Politécnica de Cataluña, Spain
José A. Castellanos	U. Zaragoza, Spain
Miguel A. Cazorla	U. de Alicante, Spain
Carlos Cerrada	UNED, Spain
Bernardo Cunha	U. of Aveiro, Portugal
Javier de Lope	CAR UPM-CSIC, Spain
Teresa de Pedro	CAR CSIC-UPM, Spain
Jorge Dias	U. of Coimbra - ISR, Portugal
Rüdiger Dillmann	Karlsruhe Institute for Technology, Germany
Sergio Domínguez	CAR UPM-CSIC, Spain
Vicente Feliú	U. Castilla la Mancha, Spain
Roemi E. Fernandez	CAR CSIC-UPM, Spain
Manuel Ferre	CAR UPM-CSIC, Spain
Ernesto Gambao	CAR UPM-CSIC, Spain
Elena García	CAR UPM-CSIC, Spain
Nicolás García-Aracil	U. Miguel Hernández, Spain
Alfonso García-Cerezo	U. Málaga, Spain
Juan C. García-Prada	U. Carlos III de Madrid, Spain
Antonio Giménez	U. Almería, Spain
Fernando Gómez	U. Huelva, Spain
Pablo González	CAR CSIC-UPM, Spain
Antonio González	U. Granada, Spain
Javier González	U. Málaga, Spain
Carlos González	CAR UPM-CSIC, Spain
Rodolfo Haber	CAR UPM-CSIC, Spain
Agustín Jiménez	CAR CSIC-UPM, Spain
Antonio R. Jiménez	CAR CSIC-UPM, Spain
Nuno Lau	U. of Aveiro, Portugal
Pedro U. Lima	Lisbon Technical Univ., Portugal
Maria R. Llácer	U. Barcelona, Spain
Juan López-Coronado	U. Politécnica Cartagena, Spain
Darío Maravall	CAR UPM-CSIC
Lino Marques	ISR-U. of Coimbra, Portugal
Humberto Martínez	U. Murcia, Spain
José R. Martínez	U. Sevilla, Spain
Fernando Matía	CAR UPM-CSIC, Spain
Luis Merino	U. Pablo Olavide, Spain
Luis Montano	U. Zaragoza, Spain
Hector Montes	CAR CSIC-UPM, Spain
Antonio P. Moreira	U. of Porto, FEUP, Portugal
Luis E. Moreno	U. Carlos III de Madrid, Spain
Víctor Muñóz	U. Málaga, Spain

Urbano Nunes	U. of Coimbra - ISR, Portugal
Aníbal Ollero	U. Sevilla, Spain
Gonzalo Pajares	U. Complutense de Madrid, Spain
Domenec Puig	U. Rovira i Virgili, Spain
Oscar Reinoso	U. Miguel Hernández, Spain
Fernando Ribeiro	U. of Minho, Portugal
Angela Ribeiro	CAR CSIC-UPM, Spain
Pere Ridao	U. Girona, Spain
Laura Roa	U. Sevilla, Spain
Rui Rocha	U. of Coimbra - ISR, Portugal
Claudio Rossi	CAR UPM-CSIC, Spain
Miguel A. Salichs	U. Carlos III de Madrid, Spain
Roque Saltarén	CAR UPM-CSIC, Spain
Vitor Santos	U. of Aveiro, Portugal
José Santos Victor	Lisbon Technical Univ., Portugal
Ricardo Sanz	CAR CSIC-UPM, Spain
Rafael Sanz	U. Vigo, Spain
José M. Sebastián	CAR UPM-CSIC, Spain
Fernando Seco	CAR UPM-CSIC, Spain
Bruno Siciliano	U. degli Studi di Napoli Federico II, Italy
Eduardo Silva	Inst. Sup. de Engenharia do Porto, Portugal
Alexandre Silva Simões	UNESP - São Paulo, Brazil
Armando Sousa	U. of Porto, FEUP, Portugal
Fernando Torres	U. Alicante, Spain
Rodrigo Ventura	Lisbon Technical Univ., Portugal
Eduardo Zalama	U. Valladolid, Spain

Co-organised by

Grupo Temático de Robótica CEA-GTRob
Sociedad Portuguesa de Robotica SPR

Contents

Part X: Social Robots and Robot Competitions

Part XI: Perception for UAV

Part XIV: Perception for Mobile Robots

Part XV: Field Robotics

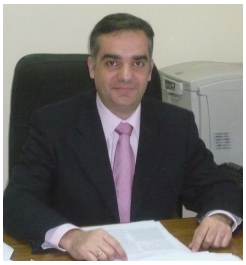
Part XVI: Robot Networks and Telerobotics

About Editors

Manuel A. Armada received his Ph. D. in Physics from the University of Valladolid (Spain) in 1979. He has been involved since 1976 in research activities related to Automatic Control (singular perturbations, bilinear systems, adaptive and non-linear control, multivariable systems in the frequency domain, and digital control) and Robotics (kinematics, dynamics, tele-operation, locomotion). He has been working in more than one hundred RTD projects (including international ones like EUREKA, ESPRIT, BRITE/EURAM, GROWTH, INCO-COPERNICUS, NMP, and others abroad the EU, especially with Latin America (CYTED/AECID) and Russia. Prof. Armada owns several patents and has published over 200 papers (including contributions to several books, monographs, journals, international congresses and workshops). He is currently Professor of Research at the Consejo Superior de Investigaciones Científicas (CSIC) and Director of the Centre for Automation and Robotics - CAR (CSIC-UPM). Prof. Armada been Presented as Doctor Honoris Causa by the State Technical University-MADI (Moscow) and by the Kursk State Technical University (Kursk).

Alberto Sanfeliu received the BSEE and PhD degrees from the Universitat Politècnica de Catalunya (UPC), Spain, in 1978 and 1982 respectively. He joined the faculty of UPC in 1981 and is full professor of Computational Sciences and Artificial Intelligence. He is director of the Institut de Robòtica i Informàtica Industrial, UPC-CSIC, director of the Artificial Vision and Intelligent System Group (VIS), former director of the UPC's Automatic Control department and past president of AERFAI, (Spanish Association for Pattern Recognition). He has worked on various theoretical aspects on pattern recognition, computer vision and robotics and on applications on vision defect detection, tracking, object recognition, robot vision, SLAM, robot navigation and urban robots. He has several patents on quality control based on computer vision. He has authored books in pattern recognition and SLAM, and

published more than 260 papers in international and national journals and conferences. He has lead and participated in 37 R&D projects, 13 of them funded by the European Commission, and he has been the coordinator of the European project URUS (Ubiquitous Networking Robotics in Urban Areas). He is (or has been) member of editorial boards of several top scientific journals in computer vision and pattern recognition. He received the prize to the Technology given by the Generalitat de Catalonia and is Fellow of the International Association for Pattern Recognition.



Manuel Ferre received the Laurea degree in control engineering and electronics and the Ph.D. degree in automation and robotics from the 'Universidad Politécnica de Madrid' (UPM), Madrid, Spain, in 1992 and 1997, respectively. He is currently a Professor Titular at UPM, and Vicedirector of Centre for Automation and Robotics UPM-CSIC. He has participated and coordinated several research projects in robotics, telerobotics and automatic control, at national and international programs. His research interest is focused on automatic control, advanced telerobotics, haptics and human-robot interaction. He has published over 100 papers in these fields, and has three patents related to human interfaces for teleoperation. He has also edited many publications, and currently is editor of the 'Springer Series on Touch and Haptic Systems'.

Part X

Social Robots and Robot Competitions

Sacarino, a Service Robot in a Hotel Environment

Eduardo Zalama[1], Jaime Gómez García-Bermejo[1], Samuel Marcos[2], Salvador Domínguez[3], Raúl Feliz[2], Roberto Pinillos[2], and Joaquín López[3]

[1] Instituto Tecnologías de la Producción (ITAP), Universidad de Valladolid
{ezalama,jaigom}@eii.uva.es
[2] Fundación Cartif, Valladolid, Spain
{sammar,saldom,raufel,robpin}@cartif.es
[3] Departamento de Ingeniería de Sistemas y Automática, Universidad de Vigo
joaquin@uvigo.es

Abstract. This paper presents Sacarino, a service robot whose purpose is to work in a hotel providing information for guests about the hotel services and accompanying them through the hotel spaces. The article describes the three levels of the development of Sacarino: the hardware level, with the sensors, actuators and robot performance; the architecture level, which describes the different functional blocks; and the application level which describes the services offered by Sacarino. Finally, we show some preliminary results of the evaluation of Sacarino in the hotel environment and the way these results have been used to improve the robot.

Keywords: Service Robot, Social Robot, Robot Architecture.

1 Introduction

Robots with social skills have seen a boom in recent years, although their application has not yet made the leap from research centers to the consumer society. These robots must be able to provide services to citizens in a natural way and interact with them like humans do. Applications that are expected of these robots include luggage or mail transportation and help for disabled and elderly people. According to Dauthenhahn [6], social robots are customized agents that are part of a heterogeneous group (a society of robots or humans), able to recognize each other and engage in social interactions. They perceive and interpret the world in terms of its own experience and communicate by learning from each other. To accomplish these goals, a robot has to be able to display certain characteristics: embodiment, emotion, dialogue, personality, human-oriented perception, environment modeling, social learning and intentionality. Furthermore, it is very important that the robot has a natural interface with some degree of anthropomorphic representation, such as a head (personification) to produce facial expressions to convey moods and emotions [2, 4].

Social robotics requires the integration of a variety of techniques related to the fields of electronics, computers, mechanics, communications, localization, navigation and the study of the psychological, sociological and biological aspects. Many of the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_1, © Springer International Publishing Switzerland 2014

techniques required for the successful completion of a social robot have still not been fully developed and require further research and experimentation, especially in the fields of environment perception, visual and voice recognition and knowledge representation of mind models, among others.

The solution to addressing real problems in the context of social robotics passes through simplification, i.e. limiting the functionality of each subsystem while ensuring proper integration of all subsystems. For example, the solution to a complex problem, such as maintaining a conversation with the robot, can be bounded by limiting the actual dialog context. This contextual clustering can be applied to other aspects such as visual recognition or emotional control.

The ultimate goal of the robot is to serve people by providing information or helping at home, in hospitals, hotels or industrial environments. Losing sight of this ultimate goal is the main reason why many robotic research projects do not reach the market. Often, technical objectives are proposed but social or practical objectives are set aside, therefore the robot has no real use that people demand. Keeping this in mind would allow social robotics to be gradually integrated into society and become more widespread and accepted, while pending technological problems are solved.

The present paper presents Sacarino, a robot with social skills whose mission is to provide bellboy service in hotels. This mission includes such features as providing information about the hotel services and accompanying guests through the hotel spaces.

2 Architecture

The infrastructure of a service robot can be defined in three levels (Figure 1). The first level is the hardware and mechanical design, including perception systems (sensors) and movement systems (actuators). The second level is the robot control architecture. This level has received major attention during recent years. Several architectures have been proposed: deliberative and hybrid architectures [11] which include navigation techniques, map location and simultaneous generation SLAM [8], planning [5], human-machine interfacing and communication which include dialogue systems, automatic recognition systems (voice, face), cognitive models and knowledge representation, etc.

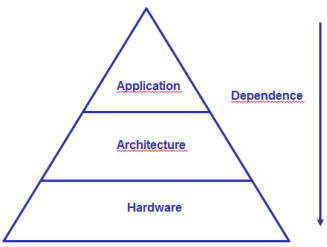


Fig. 1. Infrastructure levels of a service robot

3 Hardware Level

Sacarino's base (Figure 2) is controlled by four double wheels arranged in a synchrodrive configuration. This configuration has greater mechanical complexity than others (e.g. differential or tricycle), but allows the linear and angular velocities to be decoupled. The four wheels pull and rotate at the same time, driven by two motors: one responsible for the traction and the other one for the turn. A platform on the drive system, which turns synchronously with the wheels, supports Sacarino's body in such a way that the social part of the robot faces the direction of motion. The base is responsible for housing the control electronics and the robot navigation sensors.

The base of the robot includes the following elements (figure 2):

- Two Longway brushless motors (80W/24V, 6.2N / m) (one motor for rotation and the other one for translation).
- LiFePo4 26V/40Ah battery providing 3 to 5 hours power autonomy
- Doors damped with contact system to detect collisions.
- Ring of 16 ultrasonic sensors to detect obstacles.
- Sick LMS100 Laser for localization, obstacle avoidance and map building.
- Drawer slides with motorized door for carrying snacks, newspapers and other items.

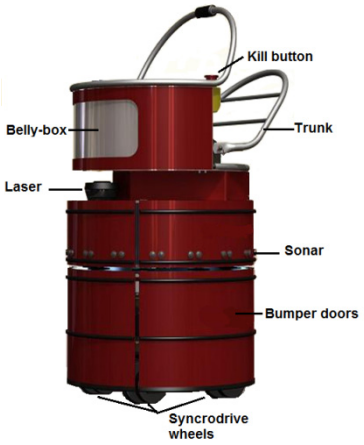


Fig. 2. Sacarino's base

The body is in charge of interactive social communication. This body has a nice, humanoid aspect that confers personality to the robot. The body system combines expressiveness and Human-Machine Interaction. Also, the body can be easily detached from the rest of the robot so that it can be used as an independent system. The robot body includes the following elements (figure 3):

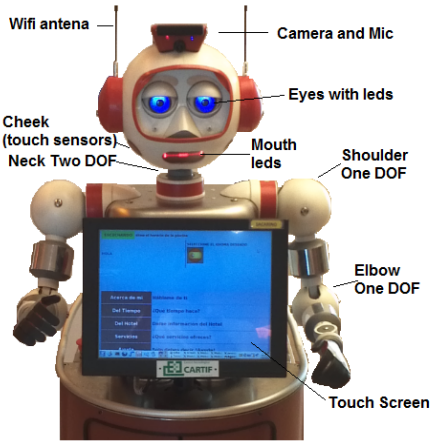


Fig. 3. Bellboy body

- Torso. The torso of robot includes two arms with two degrees of freedom (shoulder and elbow) that are driven by four servomotors. A touch screen in the front of the torso provides information and user interaction. The torso also holds the computer system used for social behavior.
- Expressive Head. The head is the component that provides more expressiveness to the system. It is responsible for supporting the HMI interaction sensors and transmitting the robot "emotions" to the users. The head, jointly with voice and arms, provides the robot with personality for a pleasant interaction. The head has two direct-coupled servomotors (providing pan and tilt movements) in order to look at the user in a natural way.
- Camera and microphone. A (Playstation-3) camera is placed at the top of the head. This camera provides face recognition and tracking for a comfortable face to face interaction. A microphone array is included for noise filtering and voice recognition.
- Eyes. The eyes can be lit up and the brightness can be adjusted by pulse width modulation. The eyelids are controlled by two servomotors for robot blinking and for adding expressiveness.
- Mouth. The mouth is shaped by an array of LEDs that can show different gestures according to the emotional state of the robot and simulate the lip movement during robot speaking.

3.1 Electronic Design

The electronics is also divided into two parts: one for controlling the robot navigation systems (motors and sensors used exclusively for navigation) and the other one for controlling the social functionality of the robot. Figure 4 shows a diagram of the electronic components of the system and their connections.

Sacarino's base is controlled by a specific microcontroller board called GPMRC (General Purpose Mobile Robot Controller). This controller has two dsPIC microcontrollers dsPIC30F6011A in a master-slave for closed-loop motor control and security device management (e.g. emergency buttons or bumpers). The GPRMC includes an LCD display for status notification, a game port for connecting a wireless playstation joystick and several buses (I2C, CAN and RS232) for connecting different elements. The board provides a number of functions:

- Monitors the system power. To do this, the GPMRC continuously measures the battery level and notifies the navigation computer. If the level falls under a critical threshold, the board executes an emergency stop and produces an alarm sound.
- Run-Stop. The board runs emergency stops and activates various alarms to detect the activation of emergency buttons, critical battery level and watchdog timer.
- Reads the bumper signals.
- Monitors the robot state and shows information on the LCD display.
- Performs motor PID control loops according to the linear and/or angular speed references provided by the computer system (or the joystick).
- Monitors the computer system. The board is in charge of the computer on/off switching. Also, an alarm is activated in case of a board-to-PC communication loss.
- Computes odometry estimation. Based on the information provided by the motor encoders, the robot's pose is estimated by dead reckoning.
- Centralizes all the information regarding the robot status and auxiliary boards (e.g. sonars) and transmits it to the computer.

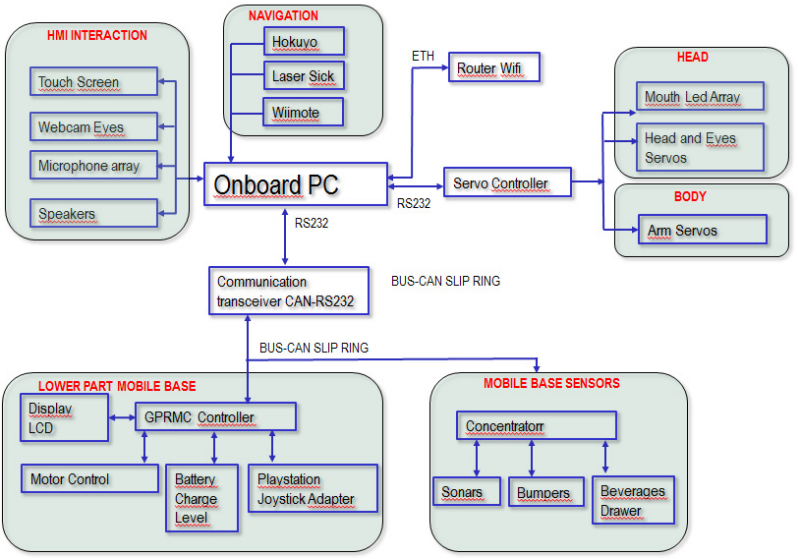


Fig. 4. Component structure and communication

The concentrator board is a specifically designed board used for configuring and reading the sonar ring and the bumpers. This board communicates with the GPRMC through CAN Bus and with the computer through RS232, via a transceiver (see Figure 4). The bus allows the number of wires to be reduced. The GPRMC is located at the robot base (along with the batteries) so that it rotates with respect to the robot torso (housing the remaining elements). A slipping ring is used for connecting the two parts of the robot.

The body of Sacarino is driven by a servo controller that performs head, eyelid and eye movements. The controller receives the position references for each axis from the computer through an RS232/USB communication line.

Finally, the transceiver is a board for converting CAN bus protocol to RS232/USB serial protocol, in order to communicate the low level control system and the navigation computer.

4 Architecture Level

The development of Sacarino's architecture has been conducted under the principle of component-based integration (component-based robotics framework). The component-based approach has several advantages [3], such as modularity, reusability, scalability and especially, the existence of repositories of different implementations of modules that facilitate the integration of systems. There are several frameworks that are enjoying a great acceptance in the robotics community. OROCOS (Open Robot Control Software) is a free software project that includes an application framework. Microsoft Robotic Studio [13] is a framework aimed at robotic design and simulation. It provides support for accessing different types of sensors, but it is neither open source nor portable to different operating systems, which has greatly limited its applicability. CARMEN (Carnegie Mellon Robot Navigation Toolkit) [14] and Player [17,18] are two implementations of open source robotics projects that maintain repositories of verified algorithm implementations in robotics. However, ROS (Robotic Operating System) [15] is undoubtedly the most successful mobile robotics re-use project to date. It provides tools and libraries to help software developers create robot applications. ROS also provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and large repositories of well-tested re-useable implementations of algorithms.

Figure 5 shows a functional block diagram of Sacarino's architecture. There is a set of functional modules, each one in charge of a specific task. Module integration and communication has been done using the ROS framework. Modules are grouped into two major functional subsystems. The sub-navigation subsystem, as the name suggests, is responsible for all on-robot navigation tasks. This subsystem includes the control modules that communicate with the GPRMC board and perform an abstraction of the hardware level (sensors and actuators) as well as the other navigation modules (localization, reactive navigation and planning). The other subsystem is the interaction and behavior one, which includes modules for gesture

control, (body control), visual perception, chatbot to generate dialogue and Automatic Speech Generation and Recognition [12] for a more versatile communication.

5 Application Level

The main mission of Sacarino is to inform the guests, show them the hotel facilities and accompany them to the different facilities. Currently, Sacarino can provide the following services:

- Give information about the hotel facilities. This includes audio and visual information about the hotel, meal times and restaurant services. Information about the city, shopping, museums, restaurants, etc. is also given. All interactions are bidirectional and are carried out in a multimodal way (using voice and/or touch screen). Sacarino presents information through spoken and written messages (displayed on the screen), images and maps. Moreover, the user can request information using voice or the touch screen, thus easing the interaction when the environment is noisy or the voice recognition system cannot correctly recognize the user.

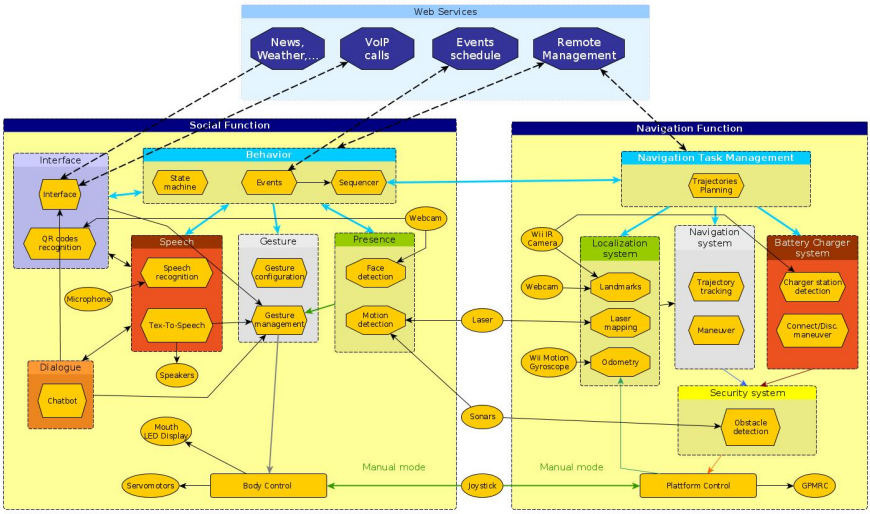


Fig. 5. Sacarino Architecture

- Provide online information from Internet, like weather forecasts, news of the day and other entertainment information such as jokes and proverbs. Sacarino currently has a database of over 5000 jokes and proverbs.
- Give information on events. Sacarino informs about upcoming events (congresses, conferences, presentations, etc.) to be held at the hotel. A dedicated website allows the staff to enter event information (description, meeting room, schedule, etc.).

- Taxi service. When a guest requests a taxi, Sacarino completes a web form to call a taxi. In order to avoid bad practices, the robot requests guest identification, asking the user to show a QR code printed on the hotel guest card, to the head camera.
- Breakfast control. Control of guests who have breakfast every day. It is intended that Sacarino performs this task. The robot will be placed at the entrance of the restaurant and when a guest is detected by the laser scanner, he/she will be prompted to say (or input through the touch screen) his/her room number.
- Videoconference service to the hotel desk. When Sacarino is unable to provide feedback for a user request, it can establish a video Skype conference with desk staff.
- Accompanying guests. Sacarino can navigate through the hotel spaces to accompany guests. Sacarino's navigation is currently restricted to the ground floor (hall, dining room, bar and meeting rooms), but, in the future, it will be able to travel to the rooms in order to guide guests or deliver items (newspapers, snacks, etc.). This task will be addressed as soon as the electronics for calling the elevator remotely have been set up.
- Satisfaction survey. The user answers a simple questionnaire about the interaction (appearance, expressiveness, ease of use) and the services offered by Sacarino.

6 Experimental Results

Sacarino has been tested initially for a two-month period at the Hotel Novotel (Valladolid, Spain). The robot usually stood in front of the reception desk, and moved regularly through the different hotel spaces on the ground floor, main lobby, coffee shop and meeting rooms, according to a certain scheduled agenda. Some services, such as taxi call, breakfast and video conferencing, have been implemented and will be evaluated in a second phase (scheduled to begin shortly). After interaction, users were asked to complete a questionnaire about the features and functionality of Sacarino. The survey questions are shown in Table 1. Visitors were asked to rate the performance on a scale 0-7 and also to fill in an open-ended form to express the reasons for their answers, the constraints encountered and suggestions for possible improvements.

Table 1 shows the mean and standard deviation of the ratings corresponding to 53 visitors who answered the questionnaire.

Table 1. Sacarino assessment survey (scale 0-7)

Question	Mean	Std
1.-Rate 0 to 7 how Sacarino is able to display information clearly.	5.05	1.18
2.- Rate 0 to 7 the ease of use of the touch interface	6.36	0.58
3.- Rate 0 to 7 the voice communication with Sacarino	3.48	1.88
4.- Rate 0 to 7 the overall experience with Sacarino	5.10	0.94
5.- Rate 0 to 7 the Sacarino movement (if you have seen it moving).	4.94	1.36
6.- Rate 0 to 7 the Sacarino usefulness as a helper and guide in a hotel.	5.43	1.10

The evaluation of the survey has been generally good. The highest rated aspects have been the touch interface, which is able to display information clearly, and Sacarino's usefulness as a helper and guide in a hotel. The worst rated question has been the voice recognition. In general, guests report some difficulties to be understood by the robot. This problem has happened due to the ambient noise reaching the microphone during recognition. The second worst rated question concerns Sacarino's movement. This question was answered by only 17 users who were present when one of the scheduled tours was performed. Some of the comments about this issue were that Sacarino should travel at a higher speed.

Regardless of the numerical scores, the qualitative information provided by users has been a very important feedback element for improvement. In order to improve the speech recognition, a more directional microphone has been set up on the touch screen (on a vibration absorbing mount). This reduces the incoming noise from the motors at the robot head. Also, when Sacarino is continually on alert listening, ambient noise may eventually be recognized as a user request, thus making the robot provide unsolicited information. One solution could consist in forcing the user to start each interaction with the word "Sacarino". However, this would result in a less natural interaction, especially when very short sentences are required. Moreover, this solution does not prevent the recognizer from being continuously active. So, the chosen solution has been a button on the touch screen. The user touches this button to start the voice recognition. This method is providing satisfactory results and is familiar to users, given that it is commonly implemented in smartphones. We are also evaluating the possibility of placing an infrared sensor to detect that a person approaches the microphone to speak, in order to activate the speech recognizer automatically (and inform the user that he/she is being listened to).

Another recurrent observation is that the user does not know what to say to Sacarino. Unfortunately, there are still no universal speech recognition systems able to deal with fluent conversation. Aware of this, users generally do not know how to restrict their speech to be understood by one of these systems. This problem appears for example when interacting with common telephonic voice recognition systems. We tend to express ourselves using monosyllables until we reach a human operator. This is coupled in many cases with the shame that people feel when talking to a machine out loud. We have added the multimodal interface to deal with these circumstances. The touch screen also includes contextual menus which provide the user with information about the discourse universe that can be recognized at the moment. Dialogue and discourse are managed using the chatbot RebecaAIML [16], which provides flexible dialogue and eases user interaction, requesting Sacarino's services (such as hotel information screen, taxi reservation or guiding through hotel spaces).

Regarding robot mobility, the robot may occasionally be unable to move due to the need of recharging its batteries. The limited autonomy of mobile robots is a recurring problem that cannot be solved by simply increasing the battery capacity (especially when robots must provide continuous service). The key idea is to take advantage of times when the robot is idle, for recharging the batteries. We have designed an automatic recharging system (Figure 6) that uses the IR sensor of the Nintendo Wiimote controller for accurate robot positioning in the vicinity of the charging

station [9]. The charging station includes four LED emitters and a QR code (bidi). This allows the robot to identify and locate the station at a distance of 3 to 4 meters, and subsequently engage it in an accurate way. The system not only enables automatic recharge but avoids the degradation of the robot localization when operating for long periods of time. This degradation, common in navigation systems based on maps, has happened occasionally in the hotel, preventing the robot from reaching the correct destination. This situation has been mitigated by forcing the robot to couple with the charge station each time a tour is completed in order to reset the robot's position. This has been tested intensively in our laboratory, resulting in more than 100 successful couplings over 100 consecutive times (scheduled every half hour tour, for a week, 8 hours a day), and then in the hotel.

Fig. 6. Detail of Sacarino coupled to the charge station

7 Conclusions

This paper presents Sacarino, a service robot whose mission is to provide added value services in a hotel environment. The goal has been to completely design the robot, from the hardware to the application level, with the ultimate aim of assessing the potential of future technology transfer. At a hardware level, a specific control board has been set up for low-level control and perception. A hierarchical control structure has been developed to achieve an easy integration and maintenance. For example, Sacarino may be manually controlled (joystick) even when the high control system (the computer board) is unavailable. Furthermore, pose estimation based on dead reckoning is performed at a low level, thus releasing the high-level computer system for other activities. The use of the ROS framework has allowed us to generate a component-based scalable architecture and reuse modules already verified and widely used by the scientific community, such as localization [7], navigation and route planning [8].

The application layer should be given the greatest transcendence in a service robot. Robust robot planning and navigation algorithms are of low interest if adequate services have not been defined at the application level. However, it is necessary for this level to be supported by solid principles in the lower levels. A good application

cannot be built onto a low-robustness or inefficient hardware architecture. At the application level, special attention must be paid to services that can be offered by the robot. This requires a correct definition of specifications in coordination with the hotel staff, always being aware of the current technological limitations.

Finally, intensive experimentation is needed, not only in the laboratory but also in the place where the robot has to operate. Continuous improvement based on continuous assessment is also required. This paper has made a first assessment of the opinions of guests. The future goal is to continue evaluating and improving the robot by incorporating a set of metrics [1] for the evaluation of navigation (distance traveled, localization faults, most requested destinations), interaction (recognition rates, most requested services) and maintenance (number of assistances, failure rate, etc.).

Acknowledgements. This work has been partially supported by the Ministry of Science and Innovation (Proyecto de Investigación Fundamental no Orientada Ref. DPI2011-25489) and the Junta de Castilla y León (Agencia de Inversiones y Servicios, y Prog. de Apoyo a Proys. de Investigación VA013A12-2).

References

1. Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., Goodrich, M.: Common metrics for human-robot interaction. In: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction (HRI 2006), pp. 33–40. ACM, New York (2006)
2. Breazeal, C., Scassellati, B.: How to build robots that make friends and influence people. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1999), Kyonju, Korea (1999)
3. Brooks, A., Kaupp, T., Makarenko, A., Williams, S., Oreback, A.: Towards component-based robotics. In: IEEE/RSJ International Conference on Robots and Intelligent Systems, pages 3567–72, Edmonton (August 2005)
4. Bruce, A., Nourbakhsh, I., Simmons, R.: The role of expressiveness and attention in human-robot interaction. In: IEEE Conference on Robotics and Automation (2002)
5. Choset, H., Burgard, W., Hutchinson, S., Kantor, G., Kavraki, L.E., Lynch, K., Thrun, S.: Principles of Robot Motion: Theory, Algorithms, and Implementation. MIT Press (April 2005)
6. Dautenhahn, K.: Embodiment and Interaction in Socially Intelligent Life-Like Agents. In: Nehaniv, C.L. (ed.) CMAA 1998. LNCS (LNAI), vol. 1562, pp. 102–142. Springer, Heidelberg (1999)
7. Dellaerty, F., Foxy, D., Burgardz, W., Thrun, S.: MonteCarlo localization for mobile robots. In: IEEE International Conference on Robotics and Automation (ICRA 1999) (1999)
8. Durrant-Whyte, H., Bailey, T.: Simultaneous Localization and Mapping (SLAM): Part I the Essential Algorithms. Robotics and Automation Magazine 13(2), 99–110 (2006)
9. Quijada, S.D., Casanova, E.Z., García-Bermejo, J.G.: Corrección de la posición mediante marcas planas utilizando el sensor IR del Wiimote. Actas de las XXXIII Jornadas de Automática, Universidad de Vigo Ed., página inicial 711 final 719, Vigo (2012) ISBN 978-84-8158-583-4

10. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots. Robotics and Autonomous Systems. Robotics and Autonomous Systems 42, 143–166 (2003)
11. Gat, E., et al.: On three-layer architectures. Artificial Intelligence and Mobile Robots, 195–210 (1998)
12. Speech Recognition System (ASR) Loquendo, http://www.loquendo.com (accessed: June 18, 2012)
13. Microsoft Developers Robotic Studio, http://www.microsoft.com/robotics/
14. Montemerlo, M., Roy, N., Thrun, S.: Perspectives on standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit. In: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), October 27-31, vol. 3, pp. 2436–2441 (2003)
15. Quigley, M., Conley, K., Gerkey, B.: ROS: an open-source Robot Operating System. In: International Conference on Robotics and Automation (2009), http://www.ros.org
16. RebeccaAIML, http://rebecca-aiml.sourceforge.net (last accessed May 14, 2013)
17. Player: Cross-platform Robot Device Interface and Server, http://playerstage.sourceforge.net/index.php?src=player
18. Vaughan, R., Gerkey, B., Howard, A.: On device abstractions for portable, reusable robot code. In: Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS (2003)

Animation of Expressions in a Mechatronic Head

Loza David, Marcos Pablos Samuel,
Zalama Casanova Eduardo, and Jaime Gómez García-Bermejo

Universidad de Valladolid, ITAP-DISA, and CARTIF Foundation
davidlozaing@hotmail.com
sammar@cartif.es
{ezalama,jaigom}@eii.uva.es
http://www.uva.es , http://www.cartif.es

Abstract. In this work, a method for animating a mechatronic head with realistic appearance is presented. The required actuators have been defined upon the Facial Action Coding System (FACS). The generation of the six basic emotions is addressed: happiness, disgust, sadness, anger, fear and surprise. These expressions are generated by interpolating movements through a sequence of key poses. The voice is integrated in a similar way, using a viseme-based scheme that allows synchronizing voice, lips and mouth movements. Implementation details and results showing suitability of the approach are also given.

Keywords: Human-robot interaction, mechatronic head, key poses.

1 Introduction

Gestures are one of the main communication ways of humans. Specifically, human face produces the most obvious and immediate signals in feeling transmission [2]. Therefore, robots should be able to provide visual information through face gestures in order humans to attain a full understanding of robot messages.

The basic elements required for designing and building a robotic, mechatronic head with realistic appearance are analyzed in [7]. The developed model is shown in figure 1. This model has 23 degrees of freedom and a large expressive aptitude.

However, controlling servomotors so that gestures are produced in a natural way is a challenging issue. The development of tools for parametrizing the process in an intuitive and effective way can be envisaged. Some standards have been proposed in Psychology for objectively coding and parametrizing the face movements, in such a way that emotions and facial expressions can be related upon specific face movement patterns. Good examples are EMFACS [3](Ekman and Friesen, 1982), Monadic Phases [12](Tronick, Als, and Brazelton, 1980), the Maximally Discriminative Facial Movement Coding System [5] (MAX: Izard, 1979) and more recently, Facial Action Coding System [1] (FACS: Ekman, Friesen and Hager, 2002).

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_2,

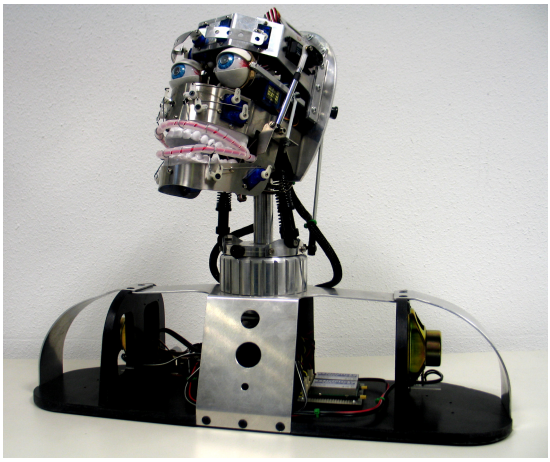


Fig. 1. Developed model

One of the most outstanding standard is the *Facial Action Coding System*, FACS [1]. The FACS is the most used and versatile method for measuring and describing facial behavior. Ekman and Friesen developed this system in the 70s by determining the effect of the contraction of each muscle or group of muscles in the face changes. The FACS has been widely used in different fields, such as animation [13], lie detection [6] and medical diagnosis [4], among others.

The FACS is structured in *Action Units*, AUs. An AU is the contraction or relax of some muscles that produces a change in the face appearance. The system is not based on the separate activity of individual muscles because a given muscle may actuate more than once in a given gesture. Morever, some face changes are produced by the coordinated activation of more than one muscle.

The present work is aimed to the production of realistic and visually effective gestures in a mechatronic head. A number of tools are presented, that allow different expressions and animations to be generated and tuned readily. This way, the robotic head can provide the user with the same amount of information than a human interlocutor does.

2 Integration of Expressions in the Mechanical Model

The CAD model of the mechatronic head and its expected aspect after adding an artificial skin are shown in figure 2. The mechanical model for the development of the different face expressions has been built upon this model.

Three main elements are considered for generating the different expressions: servomotors, action units and expressions. These elements are implemented through a Servomotor Control Module, an Action Unit Module and an emotion manager. The integration of this modules and the physical structure is shown in figure 3.

Fig. 2. CAD model, final appearance

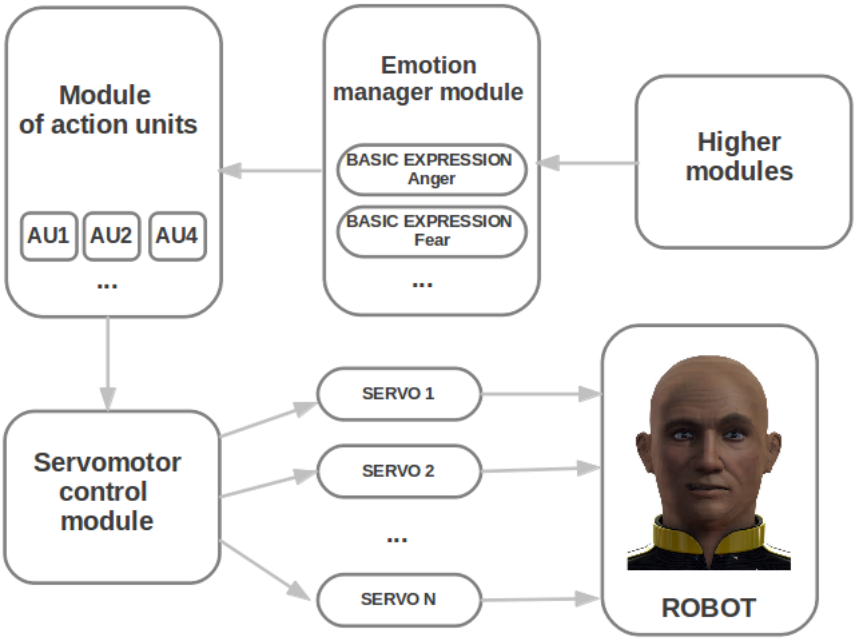


Fig. 3. Integration of expressions into the mechanical model

The servomotor control module activates the different servomotors to the consigned position and speed. This module is also assigned the observance of the physical limits of each servomotor. Moreover, the module manages with the mechanical design restrictions.

The action unit module has the codified AUs. According to the FACS, there are more than 70 AUs. Implementing such a large number of AUs would be an extremely difficult task. In the present work we have obtained a suitable reduction of this number by following previous works on virtual heads [8]. The number of significant AUs has been reduced to 26 (see table 1), used for both producing face gestures and moving eyes and neck.

Table 1. Action units

AU	**Action**
1	Inner Brow Raiser
2	Outer Brow Raiser
4	Brow Lowerer
5	Upper Lid Raiser
6	Cheek Raiser
7	Lid Tightener
9	Nose Wrinkler
10	Upper Lip Raiser
11	Nasolabial Deepener
12	Lip Corner Puller
15	Lip Corner Depressor
17	Chin Raiser
25	Lips Part
26	Jaw Drop
29	Jaw Thrust
51	Head Turn Left
52	Head Turn Right
53	Head Up
54	Head Down
61	Eyes Turn Left
62	Eyes Turn Right
63	Eyes Up
64	Eyes Down
65	Walleye

The emotion manager module allows the head performing the six basic emotions: happiness, disgust, sadness, anger, fear and surprise. This module determines AUs activation intensities required for performing these emotions as well as other expressions corresponding to robot status and moods.

3 Animation of Expressions

This section describes the animation of the different expressions using the control structure presented in section 2. The key pose structure and the tools developed for animating the head are presented.

3.1 Structure in Key Poses

The animations are structured into what we have called *key poses*. A key pose is a point within the animation sequence at which a given pose of the robot is requested. The key-pose based structure in presented in figure 4.

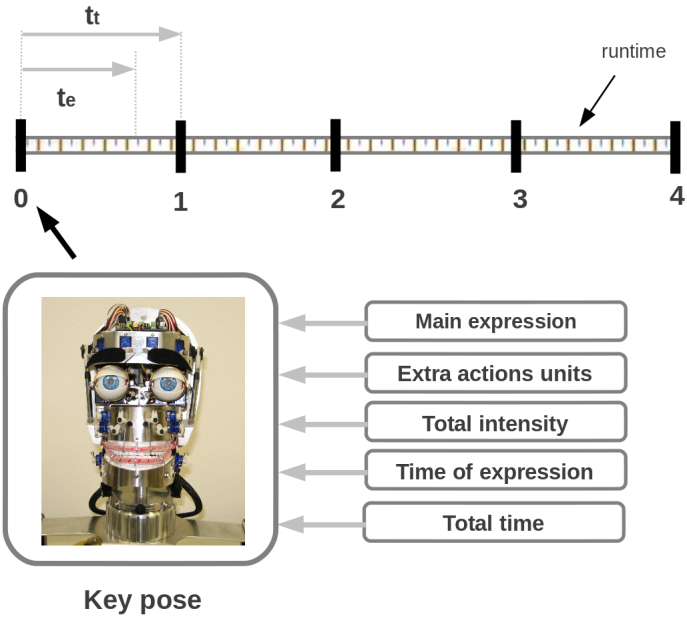


Fig. 4. Key-pose based structure

Each key pose contains: the main (or base) expression that is selected from a set of preconfigured expressions; the extra AUs added (or modified) with respect to the base expression; the 0 to 1 intensity of the expression, that affects all the AU intensities of the pose; the expression time required for performing the pose (t_e); and the total time (t_t) (larger or equal than the expression time).

3.2 Integration of Action Units and Basic Expressions

Complex movement sequences can be developed using the presented structure (section 2). However, from a practical point of view, the tuning time required for

attaining the desired results can be extremely large. Therefore, the implementation of a tool for easing work on sequences would be largely desirable. We have developed such a tool that allows real-time working on sequences, following [11] [9].

The tool integrates three main components: AU implementation, basic and complex expression implementation and viseme implementation (that will be addressed in section 3.3).

The AU implementation component is shown in figure 5. This component allows developing facial expressions in a detailed way upon AUs presented in table 1. The intensity and laterality of each AU can be readily adjusted.

Fig. 5. Tool for expression generation Unit actions

The second component allows dealing with basic and complex expressions (figure 6). The six basic expressions can me modified to a given strength that affects the intensity of every AU. Other predefined expressions can be added, such as sleeping expression, nods or specific performances of the basic expressions. The generated file can be also edited, thus allowing a detailed real-time control of the different key poses and the changes made.

A number of additional tools are provided for gesture composing, such as pose display, return to neutral pose, key pose addition or removing etc. Moreover, three different execution modes are supported: (i) *Entire sequence*, the robot repeats the complete sequence; (ii) *Separate expression*, the robot shows the selected key frame; and (iii) *Sequence pause*, the robot stops at an exact position of the animation.

3.3 Lip Syncronization

It is important that the robot gives correct visual information during communication. Humans learn from an early age the relation between some expressions

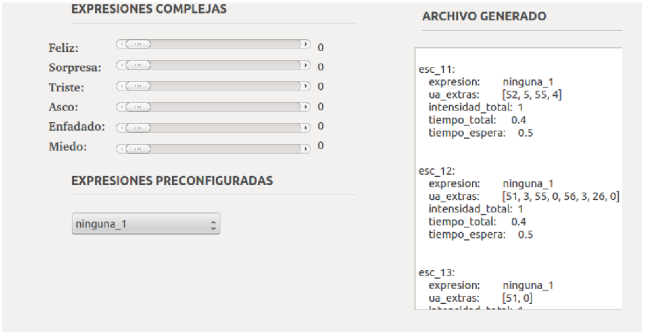


Fig. 6. Tool for expression generation Basic expressions, editable file

and the pose of lips and mouth. So, temporal and spatial coherence is required for the robot to correctly interact with humans.

Viseme Generation

There are several ways to define the relation between the voice signal and the mouth pose. The model implemented in this work is shown in figure 7. It is based on *visemes*. A viseme is the visual representation of a phoneme and is related directly to the mouth and lip pose. Good results can be obtained using a moderate number of visemes. In this work, we have implemented as basic expressions, the main visemes corresponding to the Spanish language [10]. However, generating a suitable sequence of visemes manually can be tedious, thus the automatic generation of visemes has been approached.

The process is described in figure 7. Both the phonemes (audio signal) and the visemes (visual signal) are generated from a phrase ("phrase to say", in this example). The audio signal is processed to obtain the phoneme sequence through time and the associate visemes are then calculated. The required visemes are saved as key poses. The expression time will be that corresponding to the phoneme. Two (or more) key poses are merged into a single one when the corresponding time is too small with respect to the mechanical dynamics. Finally, the visemes and the audio file are performed concurrently. This way a precise syncronization of mouth, lips and audio signal is obtained.

Integration of Visemes

A graphical interface is provided for adding AUs to the visemes generated automatically in the described way (see figure 8). This interface provides some fine tunning actions: adjust the time intervals in order to achieve improved results, modify the viseme execution times, change the phonemes generated automatically etc. Moreover, additional AUs can be added to the visemes for reaching an increased realism. For example, a nod or a neck movement can be added to certain words or phrases, for a greater naturality.

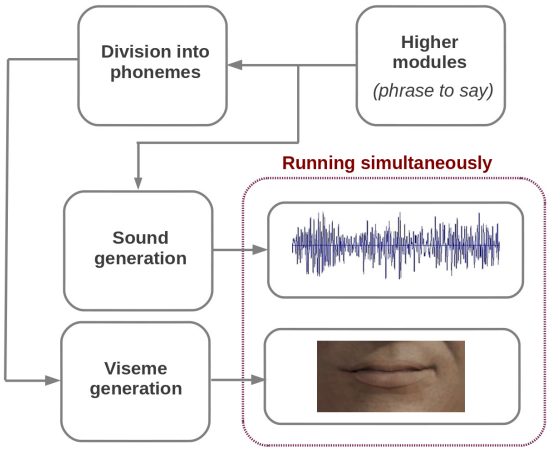


Fig. 7. Speaking process of the robot

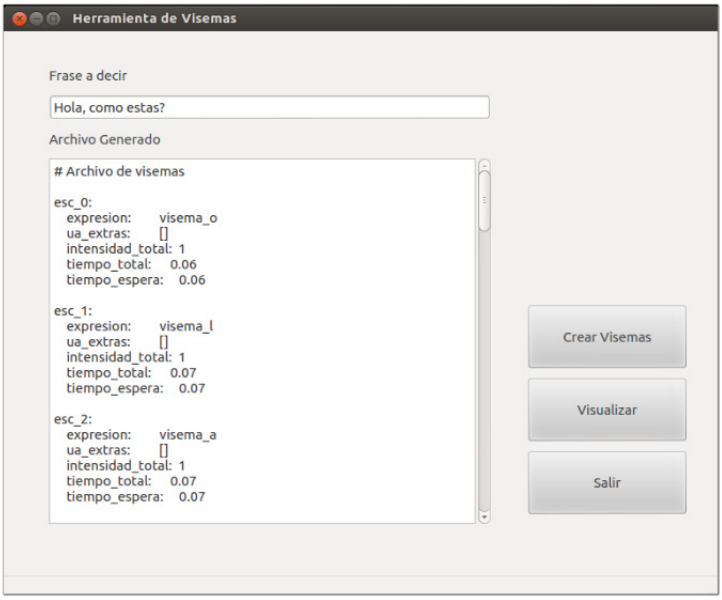


Fig. 8. Tool for expression generation Integration of visemes

4 Implementation and Results

The implementation process has been carried out considering that the AUs activated by the three main modules described in section 3.2 might overlap. A priority scheme has been used for avoiding any interference during animations.

The combinations of AUs that allow performing the six basic expressions defined by the FACS are shown in figure 9. The expressions have been generated by using the three said modules. For example, the disgust expression is generated by combining (AU4 + AU5 + AU10L) and the anger expression has been obtained by combining (AU2 + AU7 + AU10L + AU15L + AU54 + AU62). In turn, the happiness expression has been generated by combining (AU5 + AU6 + AU12 + AU26 + AU53).

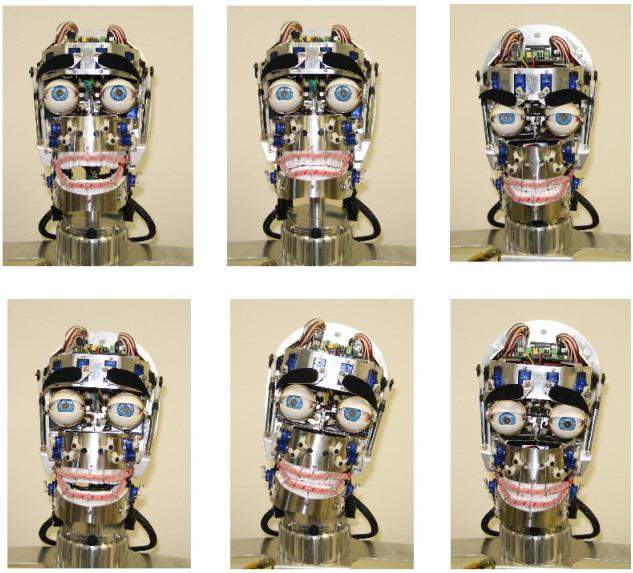


Fig. 9. Different basic expressions obtained with the mechatronic head. From left to right, top to bottom: Surprise, Fear, Anger, Happiness, Disgust and Sadness.

It can be seen that some expressions are clearly performed by the head and can be easily recognized: surprise, sadness and anger. This happens because they involve the activation of well-noticeable AUs that clearly identify the expressions. For example, this is the case of AU26 (jawbone aperture) that identifies the surprise expression. Moreover, some expressions would improve significantly when adding a synthetic skin. For example, this is the case of disgust expression, where AU9 (nose wrinkler) is activated.

The software tools described in section 3.2 allow interacting the robot directly. This helps obtaining realistic results quite straightforwardly, upon user feedback.

Moreover, the obtained animations are saved in a plain format so that they can be readily reused and edited.

Another example is presented in figure 10. Some poses corresponding to a given sequence are shown. Obtaining the animation sequence manually would have required an extremely tedious and time consuming process. However, the use of the presented methodology has shortened the required editing time to a few minutes, using 30 key poses.

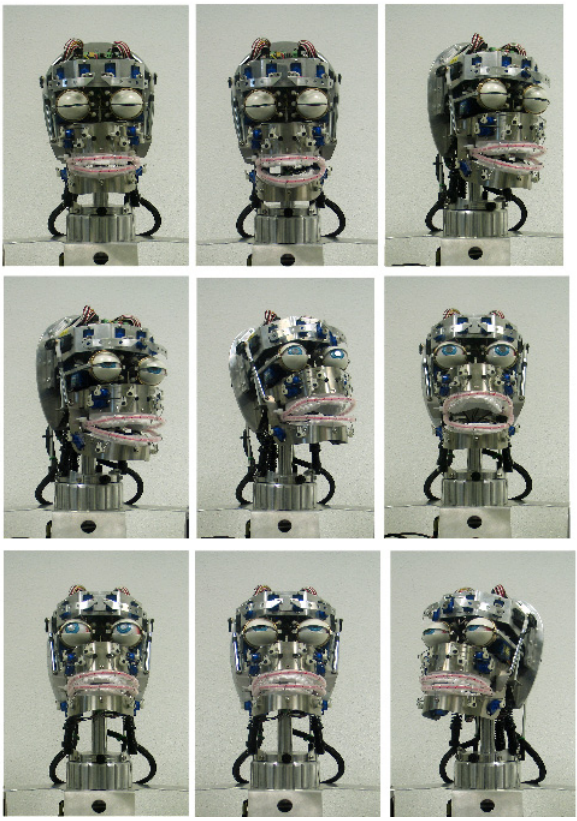


Fig. 10. Animated sequence

5 Conclusions and Future Work

In the present paper, a method for animating a mechatronic head has been presented. The method is based on the FACS and allows obtaining realistic animations of the head in a simple way.

Animations are generated by interpolating movements through a sequence of key poses. Some tools have been set up that allow parametrizing an animation in a few minutes editing work (while doing this process in a manual way would require a tedious effort).

The lip and mouth synchronization during the head voice-communication to humans has also been addressed. The required visemes are generated in an automatic way, from the text to be voiced. A fine-tunning process can then be carried out for obtaining an improved realism (by finely tunning some viseme execution time and adding eye blink or other gestures).

The proposed approach allows obtainning realistic results readily. The system is flexible and allows working in real-time, so that the results of the design actions can be analyzed immediately.

In the future, we plan to integrate an affective central module for handling the gesture and speech capabilities of the head, and to cover it with a synthetic skin. Also, experiments on the human perception of the head gestures are being conducted.

Acknowledgments. This work was supported by the Ministry of Science and Innovation, fundamental research project ref. DPI-2011-25489 and the Junta de Castilla y León, research project VA013A12-2.

References

1. Ekman, P., Friesen, W., Hager, J.C.: The Facial Action Coding System, 2nd edn. Weidenfeld and Nicolson, London (2002)
2. Freitas-Magalhes, A.: The psychology of emotions: The allure of human face. University Fernando Pessoa Press, Oporto (2007)
3. Friesen, W.V., Ekman, P.: Emotional facial action coding system (1983)
4. Hamm, J., Kohler, C.G., Gur, R.C., Vermaa, R.: Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. J. Neurosci. Methods 200(2), 237–256 (2011)
5. Izard, C.: Innate and universal facial expressions: Evidence from developmental and cross-cultural research. American Psychological Association 115, 288–299 (1994)
6. Ko, K.-E., Sim, K.-B.: Development of a facial emotion recognition method based on combining aam with dbn. In: International Conference on Cyberworlds (2010)
7. Loza, D., Marcos, S., Zalama, E., Gómez-García-Bermejo, J., González-Fraile, J.L.: Application of the FACS in the design and construction of a mechatronic head with realistic appearance. Journal of Physical Agents 7, 30–37 (2013)
8. Marcos, S., Gómez-García-Bermejo, J., Zalama, E.: A realistic, virtual head for human-computer interaction. Interact. Comput. 22, 176–192 (2010)
9. Nakaoka, S., Kanehiro, F., Miura, K., Morisawa, M., Fujiwara, K., Kaneko, K., Kajita, S., Hirukawa, H.: Creating facial motions of cybernetic human hrp-4c. In: 9th IEEE-RAS International Conference on Humanoid Robots, pp. 561–567 (2009)

10. Perales, F.: Inevai 3d: Agentes autnomos 3d, escenarios virtuales e interfaces inteligentes para aplicaciones de domtica y de realidad virtual. Simposio de Computacin - Ubicua e Inteligencia Ambiental (2005)
11. Saldien, J., Goris, K., Yilmazyildiz, S., Verhelst, W., Lefeber, D.: On the design of the huggable robot probo. Journal of Physical Agents 2, 2 (2008)
12. Tronick, E., Als, H., Berry Brazelton, T.: Monadic phases: A structural descriptive analysis of infant-mother face to face interaction. Merrill-Palmer Quarterly 26, 3–24 (1980)
13. Villagrasa, S., Susin, A.: Face! 3d facial animation system based on facs. In: IV Iberoamerican Symposium in Computer Graphic (2009)

Limited Resources Management in a RoboCup Team Vision System

Manuel Muñoz[1], Pau Muñoz[2], Eduardo Munera[1], J. Francisco Blanes[1], and José Simó[1]

[1] Institute of Control Systems and Industrial Computing
Polytechnic City of Innovation
Polytechnic University of Valencia, Spain
{mmunoz,emunera,pblanes,jsimo}@ai2.upv.es
www.ai2.upv.es
[2] Institute of Design and Manufacturing
Polytechnic City of Innovation
Polytechnic University of Valencia, Spain
pmunyoz@idf.upv.es
www.institutoidf.com

Abstract. The vision system provides to a humanoid robot the most complete information about the environment, having thus visual information extraction a great influence in the decision process of interaction with the environment. It represents one of the most resources consumer in the system, so it has to be efficiently designed without compromising the reliability of the provided information. Software implementation of an accurate vision system is one of the more distinctive elements in RoboCup competitions, especially in the RoboCup Standard Platform League (Robocup SPL), where normative is becoming more and more restrictive and no hardware modifications can be made. In embedded control systems, management of the available resources is crucial to achieve a good performance and get the most from them. Focusing on the Robocup SPL, Nao robot has the CPU as the main shared and limited resource and, to a lesser degree, wireless network and memory should also be considered. In this work, the design of the vision module and the ability of the Hidalgos Team architecture to cope with limited resources are presented. The goal is to maximise CPU utilisation by boosting the capacities of the vision system, while maintaining control properties such as reactivity and stability. A first strategy emphasises on reducing resources consumed by the vision system through adaptability to control requirements. A second strategy focuses on adapting the execution rates of the modules, and especially of the vision system, to the available resources.

Keywords: RoboCup Standard Platform League, Limited resources management, Embedded systems.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_3,

1 Introduction

RoboCup [1] is an international scientific initiative with the goal of advancing the state of the art of intelligent robots. The RoboCup Soccer competitions use the game of soccer to advance in research goals concerning cooperative multi-robot and multi-agent systems in dynamic adversarial environments. The proposed environment promotes the development and improvement on several scientific areas as robot kinematics and dynamics, computer vision, artificial intelligence, data fusion, communications, coordination, etc.

RoboCup Soccer competitions are structured in several leagues, among which the Robocup SPL. In this competition all teams use identical robots, offering hardware equality between opponents. Software implementation is thus the decisive factor. Additionally to the inherent difficulty, it is added the fact that the autonomy of the system is supported on a computer with relatively limited resources. From this perspective, the competitions are a good reference for research in embedded systems real-time control.

All these aspects make the design of an architecture with a good balance between flexibility, scalability (to easily adapt to changes in the rules of competition or hardware upgrades) and efficient use of resources is crucial in this application.

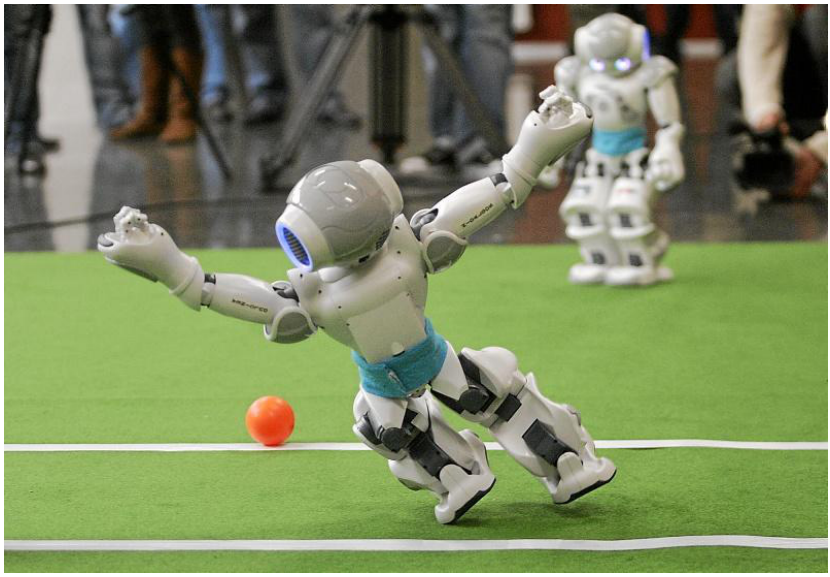


Fig. 1. RoboCup Standard Platform League

[1] `http://www.robocup.org`

1.1 Related Works

Many of the teams competing in the Robocup SPL maintain a continuous line of evolution of the vision system trying it improve year after year. Resource Management in control systems has been a highly contested issue for years, and remains, in response to the large number of publications that keep coming on this subject.

Lots of teams use original engineering solutions to improve its capabilities and then its performance during a Robocup SPL matches. Austin Villa Team[7,2] in addition to maximising information acquisition through the use of two cameras, improves object detection with the use of a higher resolution image as well as line and curve detection and differentiation via robot pose estimation. To optimise the computational cost, when accessing two neighbouring pixels, burdens the memory controller only once by retrieving a 32-bit value instead of querying four 8-bit pixel channel values independently. This retrieves information about two adjacent pixels together and improves the access time. In other approach the Portuguese team [11] has developed a reliable implementation of a modular vision system to be used in colour-coded environments, that achieve a compromise between complexity and processing times, allowing perform complex algorithms in real time. In the same line the Greek Team, Kouretes[5], proposes a object recognition called KVision, performing a light-weight image processing method for visual object recognition focusing on reliability and efficiency.

Other trend, not achieving a high performing in algorithms but obtaining good results from the recognition point of view and development. The UChile Robotics Team[13] implementation runs OpenSurf and OpenCV in the NAO robot successfully, with these software libraries it is possible to extract local descriptors in order to identify detected goals (own or opponent).

Resource optimisation techniques are not only applied on Vision system, RobotViz [1] explains that planning algorithms themselves are costly and thus hard to use in a demanding online scenario as robotic soccer, previously generated generic plans are provided to the agent who then can select the best plan w.r.t. some performance measure out of the set of plans that can be applied to a situation. As the pre-defined plans take into account multiagent settings it is possible to select a tactical move for a group of agents where different roles are assigned to various agents.

1.2 Outline

As outline of this paper: After the introduction section 1.1 describes a review of related works explaining other Teams solutions to the problem. A description about the problem is depicted in section 2 . The proposed solution is shown in section 3 and comment about the obtained results is reported in section 4. The paper ends with some conclusions about the work.

2 Framework

2.1 RoboCup Environment

The RoboCup environment is known a priori by participants. Concerning visual information, the main characteristic is that the objects are colour coded to facilitate their recognition (goals, ball, field, lines). Hence, colour recognition is meant to be the initial step in the process of visual interpretation.

Over the years, the normative of the Robocup SPL is becoming more and more restrictive. A huge change in the environment settings occurred in 2012, with the removal of the different colouring between own and opponent goals. That gives to the vision system even more relevance, since robots localisation relies also on visual information.

2.2 Nao Robot

The Nao humanoid robot, developed by Aldebaran Robotics [2], is the common platform used in the previously introduced Robocup SPL. This robot is equipped with several sensors and actuators in order to provide the capability of performing an accurate game play, bearing in mind that the hardware can not be modified in any way in this league.

Sensory information is provided by two CMOS cameras disposed on the head, Force Sensitive Sensors (FSR) and two bumpers in the feet, a 3 axis accelerometer, a 1 axis gyroscope and an array of sonar devices integrated inside the chest.

Computational units vary depending on the Nao robot version, but always based on x86 platform. V3 version is designed with the 32bits processor AMD Geode 500MHz, 256 MB of RAM and 2GB flash memory, while newer V4 version shows a more powerful configuration: Intel Atom Z530 with 1GHz CPU, 1GB RAM and 2GB of flash memory. Ethernet and Wi-Fi interfaces provide communication capabilities. Regarding the software features, disposes of a embedded Linux in which runs the NAOqi Middleware (NqMW), developed by the manufacturer, provides a full framework for managing the hardware resources.

2.3 Hidalgos Architecture

This subsection shortly describes the architecture used in Hidalgos team robots[3]. Some of its modules are evolved from the system used by the TeamChaos. The architecture is divided into five modules from the viewpoint of logic design, operation and implementation. These modules have been divided for performing specific functions within the system. Figure 2 shows an overview of the module organisation. Thus, the modules are:

– **PAM:** Perception Module. Provides sensing information internal and external to the system.

[2] `http://www.aldebaran-robotics.com/`

– **CTRL:** Control Module. Manages the robot's behaviour and team's strategy.
– **GM:** Location Module. Performs the location of the robot and the objects of interest in the world.
– **CMD:** Interface Module and locomotion. Implements the interface to the robot management middleware (NqMW) and contains the different engines of locomotion.
– **COMMS:** Communication Module. Manages the distribution of information among robots as well as outside communications (GameController - User).

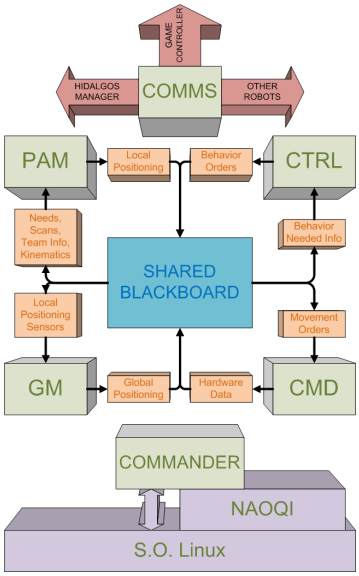
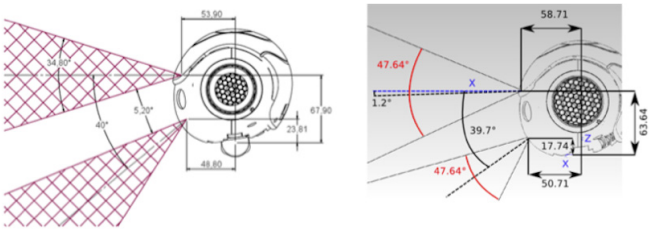


Fig. 2. Hidalgos Architecture Overview

2.4 Vision System Description

The vision system provides to the robot the most complete information about the environment, whereas other sensor systems are limited to simpler information, such as distance and forces. Hence, visual information extraction is a crucial factor in the RoboCup competitions.

Nao is equipped with two on-board CMOS cameras, disposed on the head as depicted in Fig. 3. The head is attached to the body through two servomotors, responsible of the pan and tilt movements. These movements allow the robot to perform visual scans in a big range and gives it the capacity to perform a visual



(a) Nao V3 (b) Nao V4

Fig. 3. Camera Arrangement

servoing for tracking objects, like the soccer ball, to keep them in its Field Of View (FOV).

The Nao robot, and specially in its V3 version, offers limited resources that have to be employed in several tasks, which range from kinematic control to high level tasks of team coordination. In this way, the vision system, being one of the most resources consumer in the system, has to be efficiently designed without compromising the reliability of the provided information. The adaptability of the vision system to cope with limited resources is explained in more detail in Section 3.

The process of extracting information from the images is not a direct procedure, since many techniques are involved. Our approach, consisting of the following sequential steps: colour segmentation, blobs characterisation and object recognition, is described now in this section.

Acquisition. Image acquisition is carried out through the Video For Linux (V4L) driver, instead of using the mechanisms offered by NqMW, due to its performance and configuration capabilities. To perform a more efficient game playing, it has been added a camera management system, which is meant to provide the image from the desired camera on each instant. That way, three operating modes are introduced: lower camera, upper camera and camera alternation, which is, in turn, the most used mode because of presenting the largest coverage area. The mode is determined by the resource manager.

Colour Segmentation. The thresholds used for segmenting colours are defined during a calibration process. A Look-Up Table (LUT) is built by means of a user interface to store the range of values that define each colour. Image is acquired in YUV colour space, but colour ranges are defined in HSV colour space, because it offers a faster and more accurate calibration. Therefore, a conversion between YUV and HSV colour space is performed during the calibration process. The input to the resulting LUT are the YUV values of each pixel in the image and the output is the corresponding calibrated colour.

Blobs Formation and Characterisation. Adjacent pixels of the same colour are grouped forming blobs. Several techniques are designed for blob forming such as Scale Gait Representation (SGR) or Run Length Encoding (RLE), being the last one used in this work. This method recognises trails of pixels with the same colour (runs) in each row of the image to, afterwards, cluster adjacent runs to form the blobs.

Blobs of the same colour are merged if they are close enough and, then, all the resulting blobs are characterised with a rectangular bounding box and the density of pixels surrounded by the bounding box. This information is used on the recognition step as is presented next.

Recognition. Once the blobs are spotted in the image, a classification of each one can be performed according to its characteristics. The very first step is to compute the horizon line for delimiting the bounds of the field. Starting from the segmented image, the horizon is computed taking into account the pixels associated with the green colour of the carpet. This line is used for discarding those elements located above the horizon, avoiding false detection of features out of the field (people, objects in the room, etc.).

Then, the information extracted from the blobs (size, density of pixels and bounding box aspect ratio) is used to recognise the desired objects: ball, goals and lines. The recognition algorithm works better when objects are fully represented in the image, but it is also prepared to deal with partial representation and partial occlusions.

Robot recognition is currently under development. It would allow the team to add advanced skills such as robust obstacle avoidance, opposite goalkeeper localisation to shoot at uncovered parts of the goal, and passing between teammates to improve team performance. All these abilities have to take the positions of teammates and opponents into account. Different approaches for robot recognition have been proposed based on: decision tree learning [12], neural networks [6], Kalman filter [4].

The last step in the recognition process consists of estimating the distance between the robot and the objects. This step is crucial to establish the environment model, used thereafter in localisation algorithms and to share information about global representation of the objects. Distance estimation process is detailed in [9].

3 Efficient Resource Management

In embedded control systems there are some resources, limited or not, that should be managed to achieve a good performance. Special mention whether this resources are shared among several components that require the most from them. Focusing on the Robocup SPL, Nao has the CPU as the main shared and limited resource. Likewise, but to a lesser degree, wireless network and memory should also be considered.

Next, the use of resources by the different modules is analysed and the global management capacity and strategies about the use of them are presented.

3.1 CPU

Over the Nao's embedded board, the main processor is the major source of system's computational power. It has an initial charge, barely controllable by the programmer, generated by the resident operating system (OS) and exclusive NqMW that manages the robot hardware. Therefore, the ability to manage the computational load is reduced to the modules that comprise the H-Team control architecture described above. Not all of these modules are susceptible to modify its operation, either because it is not possible, or because their influence on the total load is very poor and an efficient management does not significantly improve the computation. Therefore, this work mainly focuses on the vision module that is the major consumer of computing power.

The goal is to maximise CPU utilisation by boosting the capacities of the vision system, while maintaining control properties such as reactivity and stability. The equation governing the utilisation of CPU by the periodic tasks of the control system is shown in equation 1.

$$U = \frac{C_{CMD}}{T_{CMD}} + \frac{C_{CTRL}}{T_{CTRL}} + \frac{C_{PAM}}{T_{PAM}} + \frac{C_{GM}}{T_{GM}} + \dots \tag{1}$$

It is intended to maintain constant the CPU utilisation U at around 70%, which allows proper operation of the OS and NqMW. Therefore, any change in C (computational time), T (period) or C/T ratio in any of the modules, implies changes in one or more of the other modules (see section 2.3).

Two premises based on optimisation of the vision system's execution are defined to optimise overall system's computation.

- If computing time C of the vision system is reduced and its period T is kept constant, C/T ratio (i.e. partial CPU utilisation) in any of the remaining modules can be increased.
- If computing time C of the vision system is reduced, its period T can be decreased to maintain vision system C/T ratio.

Bearing in mind these premises, the following strategies have been implemented.

A first strategy focuses on reducing resources needed by the vision system when possible and is classified in two approaches: mission oriented visual interpretation and environment dependent visual interpretation.

Mission Oriented Visual Interpretation. The use of resources is optimised according to the requirements of the current state of the mission. As the information needed during a mission is not always the same, we have developed mechanisms by which image processing and object recognition techniques vary over time in response to the requirements imposed over the mission.

In the visual interpretation process, this strategy has been used in several steps, applying different techniques.

- **Selective image processing.** The image processing steps are parametrised to recognise selected objects. The processing time vary drastically depending on the type and amount of objects to recognise. Depending on the current state of the mission behaviours module information about different objects could be required. But in certain circumstances only some of them are required, thus not being necessary to process the information about the remainder objects.
- **Global subsampling + Local zoom.** Different image subsampling is applied depending on the size of the object to recognise and the precision needed. Optional local zoom could be applied to increase precision. Just as in the selective processing, if the mission requires a single object, and this has a small size (ball), the subsampling conditions are adjusted to optimise the image processing.

Environment Dependent Visual Interpretation. Knowledge of the environment and the state of the robot and the game is used to optimise the image analysis. According to that, the following techniques are used.

- **Various cameras processing.** The system is prepared to extract information from one or both of the two Nao on-board cameras. But depending on the angle of the robot head, and knowing the dimensions and configuration of the field, it could be determined when the analysis of the information from the upper camera does not provide any useful information. Even avoiding the analysis of many small random blobs from the ceiling.
- **Global subsampling.** In the same way, Nao's neck inclination is used as a guide to the subsampling, since the depth of the image and, consequently, the size of the objects in the image, depend on that value.

A second strategy focuses on decreasing the vision system period T, once the resources needed by the vision system have been reduced, to propitiate better robot reactive response.

Sometimes the circumstances of the game (eg. fine tuning of the ball's position to kick) require a really reactive performance of the control loop, while an exact localisation, exhaustive movements planning or team coordination are not needed. In these cases the saving of resources, previously exposed, do not suppose any advantage if vision system period T is kept constant. The possibility of using a variable period is thus implemented, making the vision system to use more CPU percentage and loop in a very high rate.

Control Scheduling. Control scheduling techniques could be used to adjust the period of the threads composing the vision system in the way that images are acquired and processed as fast as the analysis allows. Thus, if the computing time

C is maintained as low as possible and the utilisation is kept in a constant value as proposed above, period T could be really low so control loop, acquisition-planning-action, would have a higher reactivity.

Multirate Estimation of the Relative Position of the Objects. In several cases, vision system sampling rate may be much slower than the control rate, so the visual information in the control loop is often quite old. To overcome this problem, multi-rate approaches [8,10] can be used instead, where the aim is to reproduce, as ideal behaviour, the high-frequency response.

3.2 Network

During Robocup SPL games the bandwidth that each robot team could use in inter-communications is limited to 500 kbps. To carry out high-level matches, robots are required to share large amounts of information, being thus essential the proper use of the media. Robots use communication with two main objectives: sharing information about the moving objects in the field and collaborating on group strategy.

Shared Objects and Localisation. By sharing the position of moving objects in the field, in addition to the current own viewing area, the robots have virtual shared vision zones, expanding the knowledge of the environment around them. It is important to notice that robots should be reliably located in the field in order to reference the detected objects to a shared global coordinates system. Moreover, the location system is fed from the shared information, being reinforced with successes and invalidated with failures [9].

Team Cooperation. The Robocup SPL matches require robot collaboration to win games. On one hand, typical fateful game scenarios where all robots are simultaneously trying to reach the ball would be avoided. On the other hand, collaboration between robots enables dynamic allocation of roles, which greatly facilitates team's strategy.

Message Management. Two types of messages are transmitted, one for shared object positioning and another one with information about team coordination.

Messages related with team coordination, contain very light information and have fixed size, therefore no adjustment is made on them.

Messages that share information about the environment are more complex, so management is required to limit the consumed bandwidth. Concretely, the frequency at which the robots send that information is proportional to the confidence that the robots have on its correctness. In this way, robots better located and providing more reliable information transmit more messages, helping the rest to improve their location or even play using that information.

4 Experimental Results

To demonstrate the suitability of the adopted architecture, two opposite situations, regarding the amount of resources they require, are here described: situations where the reactivity of the robot is preferable (cases like goalkeeper saves) versus situations where more information is needed (robot localisation). Experiments were carried out with the Nao V4 version.

Table 1 shows the huge computational time reduction to recognise one object while reducing image resolution by using subsampling techniques.

Table 1. Average of processing times gathered during tests

RESOLUTION	COMPUTATIONAL TIME (ms)
VGA	20,22
QVGA	7,38
QQVGA	4,23
3QVGA	3,61
4QVGA	3,47

4.1 High Reactivity Use Case

Information from the vision system must be updated in a rate as close as possible to the control loop. Assuming the goalkeeper is in a correct and well known position in the field, only information regarding the ball (i.e. relative position to the robot and its variation) is needed. The proper visual information extraction would consist of taking the image from one of the cameras and segmenting only orange pixels (ball) and green pixels (horizon) with an image subsampling proportional to the distance (i.e. to the size of the ball in the image), thus being able to recognise the ball bellow the horizon line and to calculate its position relative to the robot. The high rate of the visual loop would allow the goalkeeper to take the decision of performing a save much faster.

Mean periods of the modules from different experiments are shown in table 2. PAM period can be reduced to half the acquisition time because of the two cameras running in parallel at 30 fps.

4.2 High Requirement of Environment Information Use Case

On the other hand, in cases such as robot localisation, much information as possible must be taken into account. An extreme case would be when the robot is completely lost, for example, after a goalkeeper save. Hence, images from both cameras are used to extract lines and landmarks to facilitate the localisation

Table 2. High reactivity case: minimum PAM mean period achieved

MODULE	MEAN PERIOD (ms)
CONTROL	15
CMD	10
PAM	16,67
GM	250

task. Since lines could be quite narrow in the image (i.e. few pixels), no subsampling would be applied. Therefore, working with high resolution images and segmenting and extracting all the objects in the field of view, would lead to a much higher vision rate, as shown in table 3.

Table 3. High requirement of environment information case: maximum PAM mean period achieved

MODULE	MEAN PERIOD (ms)
CONTROL	50
CMD	10
PAM	200
GM	250

5 Conclusions

Due to the fact that the vision system represents the most resources consumer in the Hidalgos Team architecture for Nao robots, the work has been focused on the design and adaptability of the vision system to achieve a good performance and get the most from the available resources. In that sense, two strategies have been followed: the first one emphasises on reducing resources consumed by the vision system, while the second one focuses on adapting the execution rates of the modules, and especially of the vision system, to the available resources. The suitability of the implemented vision module has been exemplified with diametrically opposed situations, regarding the amount of resources they require.

Acknowledgments. This work has been supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under the CICYT project Mission Based Control (COBAMI): DPI2011-28507-C02-01/02.

References

1. Abeyruwan, S., Härtl, A., Nath, P., Seekircher, A., Stoecker, J., Visser, U.: Robocup standard platform league team description paper 2013 (2013)
2. Barrett, S., Genter, K., He, Y., Hester, T., Menashe, J., Perez, J., Stone, P.: Ut austin villa 2013 team description paper for the standard platform league (2013)

3. Blanes, F., Muñoz, M., Simó, J., Martínez, H., José Alcaraz, J.: Arquitecturas de control sobre robots nao en la spl de robocup. In: III Workshop de Robótica: Robótica Experimental, ROBOT 2011, pp. 128–134. Universidad de Sevilla (2011) ISSN 978-84-615-6787-4
4. Fabisch, A., Laue, T., Röfer, T.: Robot recognition and modeling in the robocup standard platform league. In: Proc. of the Fifth Workshop on Humanoid Soccer Robots (2010)
5. Kargas, N., Kofinas, N., Michelioudakis, E., Pavlakis, N., Piperakis, S., Spanoudakis, N.I., Lagoudakis, M.G.: Kouretes 2013 spl team description paper (2013)
6. Kaufmann, U., Mayer, G., Kraetzschmar, G.K., Palm, G.: Visual robot detection in roboCup using neural networks. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 262–273. Springer, Heidelberg (2005)
7. Khandelwal, P., Hausknecht, M., Lee, J., Tian, A., Stone, P.: Vision calibration and processing on a humanoid soccer robot. In: The Fifth Workshop on Humanoid Soccer Robots (2010)
8. Lee, J.-J., Xu, Y.: A multiple rate control scheme of robot manipulators. Journal of Systems Engineering 3(1), 22–33 (1993)
9. Munera, E., Muñoz, M., Simó, J., Blanes, F.: Humanoid Robot Self-Locatio in SPL League. In: Comité Español de Automática (CEA), XXXIII Jornadas de Automatica, pp. 797–804 (2012)
10. Solanes, J.E., Armesto, L., Tornero, J., Benavent, P.M., Girbes, V.: Dual-rate non-linear high order holds for visual servoing applications. In: TAROS 2012, pp. 152–163 (2012)
11. Trifan, A.L., Neves, A.J.R., Lau, N., Cunha, B.: A modular real-time vision system for humanoid robots (2012)
12. Wilking, D., Röfer, T.: Realtime object recognition using decision tree learning. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 556–563. Springer, Heidelberg (2005)
13. Yánez, J.M., Leottau, L., Cano, P., Mattamala, M., Celedón, W., Silva, M., Silva, C., Saavedra, P., Miranda, P., Tsutsumi, Y., et al.: Uchile robotics team team description paper robocup 2013-standard platform league (2013)

A Review of Eight Years of CEABOT Contest: A National Wide Mini Humanoids Competition

Alberto Jardón, Félix Rodríguez, Juan G. Victores,
Santiago Martínez, and Carlos Balaguer

Systems Engineering and Automation Department
Carlos III University of Madrid
Leganés, Spain
{ajardon,ferodrig,jcgvicto,scasa,balaguer}@ing.uc3m.es

Abstract. This paper presents a review of CEABOT, the unique humanoids competition for university students at Spain wide level. This annual competition started in 2006 and it has been celebrated during the last eight editions. Aimed at initiate graduate and master's students in humanoid robotics, they have to program the sensor and motor skills of little humanoids built by themselves, or tune up a commercial kit. The challenge is to obtain the best platform able to complete a set of trials that are reviewed and renewed each year. In this paper, a review of similar contests, objectives and results will be discussed. A brief review of platforms and trials evolution will also allow presenting some key ideas for the promotion and future assurance of this initiative. Furthermore, a brief review of some didactic approaches experienced, and some insights about the future of this and similar contests that use robotic platforms to face similar goals will be presented.

Keywords: Humanoids contest, robot competitions, didactic approaches, teaching with robotics.

1 Introduction

During the last decades, many initiatives related with robotics' contests have appeared, and many of them still are alive. Mainly, there are three types of motivation for the use of robotic devices in contests and therefore three types of contest flavours. The first are "open to all" contests and fairs, e.g. any amateur and robotics hobbyist, like RoboGames ™ [1] or COMBOTS©[2] among many others, where the show and fun are the main goal. The second type of competitions is formed by these designed for students in any of their courses, from undergraduate to high school and University. Contest like RoboCup Junior [1] , VEX Robotics World Champioship [2] or FIRST LEGO LEAGE (FLL) [3] fit in this category, where didactic approaches and goals are

[1] RoboGames (formerly RoboOlympics) is ™ of non-profit The Robotics Society of America, Inc. http://robogames.net/

[2] COMBOTS is © of ComBots, LLC. http://combots.net/

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_4, © Springer International Publishing Switzerland 2014

clearly claimed. And the last type are scientific challenges, promoted by research institutions or research programs at international level, such as DARPA Robochallenge[3] or ICRA Robot Challenges[4], where fostering the current state of the art is the main motivation. In this paper, the focus will be placed in the second type of contests, although along the following review, many of these flavours can be found mixed in the same event with many trials.

A general problem is the increasing disinterests of young people, in particular girls in science and technology studies. To cope with this, some initiatives like ROBOCUP, the international scientific initiative [4] have grown in the last decades. RoboCup Junior (RCJ) focuses on educational aspects encouraging young students to deal with science and technology [1]. RCJ uses robots as technical tool to educate, motivate and inspire pupils and undergraduate students up to the age of 19. The work of Papert [5], with his pioneer LEGO/LOGO project, is considered a precursor of several works involving Robotics in Education. Various papers have been published on this topic, a good review can be found in [6] . The authors of [7] provide an evaluation of the FIRST Robotics Competition (FRC)[5]. FRC, which was founded in 1989, is a high school robotics initiative located in New Hampshire (United States). The program aims to get young people interested in science and technology. A detailed, but not exhaustive, list of robot competitions around the world can be found online[6] sorted by Date, instead of category.

The competitions of mini-humanoid robotics offer engineering students the chance to get some actively initiated a field of humanoid robotics. These contests are an easy way to get into the development of control systems for robot kinematics and recognition of the work environment, as well as the challenge of a small robot to provide the skills needed to complete certain tests. In addition, they are a way to compare the development of their own systems regarding to the other systems developed for the participants of these competitions. Furthermore, these competitions allow students from around the world and from different fields of work to be related to them so that they can start friendships or sharing knowledge.

To this end, this paper will review the Spanish annual championship of mini-humanoid robotics organized by the Spanish Committee of Automatics, named CEABOT, which has been celebrated for the last eight years. The aim of the competition is to show the skills of each humanoid robot by developing several tests to be performed autonomously. A review of existing robot competitions, their motivations and brief comparison among the robotics platform employed will be highlighted. A detailed description of CEABOT contest, trials and procedures will be reviewed. To conclude, some considerations about CEABOT experience of the last eight years and a future view and expectations will be presented.

---

[3] DRC is a prize competition funded by the US Defense Advanced Research Projects Agency. http://www.theroboticschallenge.org/

[4] ICRA conference starts to host Robot Challenge at 2008 and continues, each year with several challenges like Virtual manufacturing, Mobile Manipulation, etc. http://www.ieee-ras.org/conferences-workshops/icra

[5] FIRST, a not-for-profit public charity, was founded in 1989 to inspire young people's interest and participation in science and technology. http://www.usfirst.org/

[6] URL: http://robot.net/rcfaq.html

2 Humanoid Platform Based Competitions

As mentioned above, the well kwon international competitions as FIRST (FRC), LEGO®MINDSTORM®, VEX ROBOTICS or EUROBOT[7] [8] are not focused on humanoids. At Spanish level, the traditional ALCABOT-HISPABOT (based on wheeled robots)[8], AESSBot[9] in Barcelona, and many other younger initiatives like ARDERO (Abierto de Robótica ARDE[10]) promoted also by student associations, lack in the special interest for humanoids.

RobotChallenge, is maybe the biggest and may be most popular active competition in Europe. In March 2013 has celebrated its 10th anniversary. There were 470 robot designers from 23 different nations, which competed with their 399 self-made, autonomous robots in 15 different competitions. In addition to the traditional competitions - Robot Sumo, Line Follower, Puck Collect, Humanoid Sprint and Freestyle - flying robots took part in the "Air Race" for the second time. "LEGO Line Follower", a new category for robots built out of LEGO bricks, which is aimed primarily at beginners, was introduced.

The first RobotChallenge took place in March 2004 in Vienna, Austria. What started as a small event with only a few participants and three different competitions soon became one of the biggest robotic championships worldwide. Throughout the years, RobotChallenge adapted itself to the rapid changes in the international field of robotics: More and more competitions were added to the event, and each year participants were encouraged to experiment in the area of informatics, electronics, mechanics and artificial intelligence. Since 2006, the Robot Sumo competition was gradually divided into various weight categories. With the first Freestyle exhibition in 2007, RobotChallenge opened the competition for extra-ordinary robots with special abilities. They started the Humanoids sumo contest in the 2009 edition, and some of the authors were invited to participate and get involved in the rule definition for this novel contest. This year an UC3M team participated in the VI RobotChallenge 2009 European Robotics Championship, in Vienna, Austria, being the unique Spanish representative in the event. This event consisted in two independent contests: running and sumo fighting. The Spanish team obtained the first position among the five teams. After that experience, in 2009, the CEABOT organization opened the Free-style trial inspired by the RoboChallenge event, with the only limitation in the type of robot platform that must be humanoid like, but without weight and height limitations.

A special mention is required for the RoboCup Humanoid League. In the Humanoid League, autonomous robots with a human-like body and human-like senses plan and

---

[7] Created in 1998, EUROBOT is an international amateur robotics contest open to teams of young people, organized either in student projects or in independent clubs.

[8] ALCABOT web site: `http://asimov.depeca.uah.es/robotica/`

[9] Concursos de pruebas clásicas organizado por la UPC. `http://aess.upc.es/aessbot/`

[10] Concursos de pruebas clásicas organizado por la Asociación de Robótica y Domótica de España. `http://www.webdearde.com/`

play soccer against each other. Unlike humanoid robots outside the Humanoid League, the task of perception and world modeling is not simplified by using non-human like range sensors. In addition to soccer competitions, technical challenges take place. Dynamic walking, running, and kicking the ball while maintaining balance, visual perception of the ball, other players, and the field, self-localization, and team play are among the many research issues investigated in the Humanoid League. Also to mention, the Robot Soccer Competitions in Kid Size category (30-60 cm height), that uses DARwIn-OP[11], and the ICRA 2012 DARwIn-OP Humanoid Application Challenge[12]. DARwIn is medium size open platform humanoid robot, from the ROBOTIS company, with advanced computational power, sophisticated sensors, high payload capacity, and dynamic motion ability to enable many exciting research and education activities.

As presented before, there are many growing initiatives of educational courses in robotics, workshops, and other less formal initiatives [9], [10]. All of these activities have in common that they are frequently used to generate interest in engineering careers, and develop teamwork and communication skills. However, frequently they teach very little in the way of robotics specific pedagogy. Except for those mentioned above, the majority of courses tend to focus on designing and programming mobile robots, and rely less on advanced treatment of legged or serial chain articulated robots, such as humanoids. CEABOT was originally targeted at senior level undergraduates or first year master's students, to overcome this deficiency [11].

3 Description of the CEABOT Championship

In 2006 the CEABOT contest was born as a national competition of small humanoid's robots as an initiative of Prof. Carlos Balaguer. Although the first years there were few participants, CEABOT has been growing in importance, acquiring international status, and becoming one of the most important robotic contests in our country, with an average of 10 teams and 20 robots participating each year.

The tasks the participating robots must accomplish are stated in the CEABOT contest rules [12]. The robots must have an anthropomorphic constitution, and the mode of locomotion should be walking or running on two legs. Wheels, skids or similar devices are forbidden. Another specific characteristic of the CEABOT championship is that each robot must be completely autonomous. Autonomy must be achieved at locomotion, sensing, and processing level, and by being battery powered. Actuators, sensors, and processing power must be incorporated into the robot; it must make its own decisions. This issue makes CEABOT quiet different to other humanoid competition around the world like ROBO-ONE[13] [13] or KONDO BATTLE [14] and similar, really popular in Japan and USA, where the robots are tele-operated by a human controllers using a custom RC controller, and where the objective is pure entertainment.

[11] DARwIn's official webpage: `http://robotsource.org/`

[12] `http://www.icra2012.org/program/robotChallenge.php`

[13] ROBO-ONE is biped robot entertainment contest promoted by a robot Japanese retailer. `http://www.robots-dreams.com/`

3.1 Main Features of Mini-Humanoid Robots

The robot maximum weight is 3 kg and the maximum height is 50 cm. Furthermore, the robot foot length should be lower than 11 cm in any maximum measure. The most common platforms used in the CEABOT Spanish competition are shown in Fig 1. They are based on commercially available kits. These kit's prices are normally between 500 and 1000€, and must be completed with additional sensors and wiring. Sometimes some other teams have removed the original on-board processor and replaced it by PIC or Arduino based processors boards. These must rebuild the locomotion capabilities of the humanoid from low level, programming all the servo commands signal to achieve the desired motion. Others teams choose to exploit the original kit board's hardware resources and manufactures' programming tools at maximum level.

Fig. 1. Kondo KHR-2HV [15], ROBONOVA-1 (center) [16], BIOLOID (right) [17]

Other popular low size platforms like DARwIn-OP or NAO[14] are excluded intentionally by weight and size (note that other contests such as the RoboCup Standard Platform League, SPL, do allow the NAO platform), due non-affordable prices, in comparison with smaller and simplest commercially available humanoids robot kits.

3.2 Description of Tests

The contest has a set of test to determine the capabilities of the mini-humanoid robots participating. Each one of these test is associated to a different scenario. All the sensorial processing, algorithms execution and showed behaviour must be performed autonomously by the robot. Obviously to embed enough processing power on board is a challenge due the platforms limitations in power and payload autonomy. In the following, some details concerning the trials and scenarios are described.

- Obstacle race: This test is based on navigating in a partially known environment with obstacles, having to cross this area from one line to another, and returning. The measurements of the race scenario are shown in Fig 2. In this scenario, up to six rectangular blocks painted in white could be placed. The number of obstacles

[14] NAO is medium size humanoid platform.
http://www.aldebaran-robotics.com/en/

used can depend on configuration of the test during the competition, and its location is not previously known by teams. The score will depend on how long the robot lasts to finish the testing space and the imposed penalties. The penalties will be imposed if the robot needs help from the judges due to a fall or a clear mismatch in the obstacle avoidance behaviour that makes the robot to fall into a loop.

Fig. 2. Obstacle race measures. The ground will be painted in green (Pantone Code: 16C606 (R:22:G:198:B:6)).

At the beginning of the race, rectangular shaped obstacles (20x20 cm width and 50 cm height, Fig. 3) are placed inside the field. The robots may not move them.

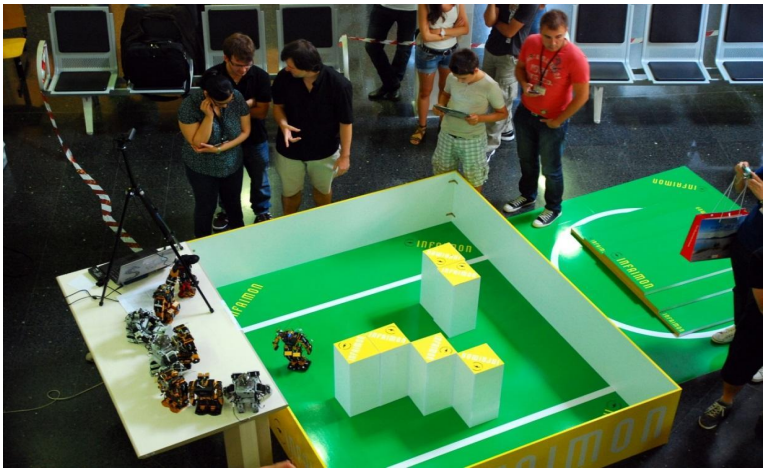


Fig. 3. The robot finds a gap between the obstacles and the wall. If the robot moves or throws an obstacle, the team will get a penalty. All the robots are place in the Committee's table until its turn arrives.

Finally, scores are given by formula (1).

$$P = \frac{(T_{MAX}(s) - T(s)) \cdot k_T}{T_{MAX}(s)} + \frac{d(cm)}{d_{MAX}(cm)} \cdot k_D - (2 \cdot pen)^{kp} \quad (1)$$

Where:

- T_{MAX} [s] is the maximum time (300 s).
- T [s] is the completion time in seconds. If the robot does not finish the race, this value will the maximum time (300s).

- d [cm] is the walked distance, measured from the white line to the feet.
- d_{MAX} [cm] is the maximum obstacle area distance (150 cm).
- pen [#] is number of penalties.
- k_T, k_D and k_P are time, distance and penalty constants, used to scalar the punctuations.

- Stairs: This test demonstrates the ability of the robot to climb stairs. All the steps are 3 cm high but there are three different lengths: 15, 25 and 50 cm, respectively (see Fig. 4).

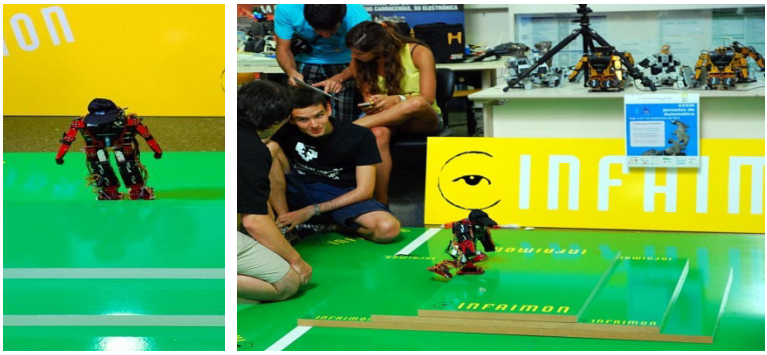


Fig. 4. A humanoid robot is climbing the stairs autonomously.

There is a time limit of 5 minutes. The team will get a penalty if the robot touches the ground with the arm/hand or if the robot falls down and cannot stand up by itself (only the referee will stand it up). For score calculation, teams will be organized by the number of penalties (less is better) and the time needed by the robot (less time is better). Depending on this ranking, the robots will get more points depending on the number of climbed steps.

- Sumo: In this trial, a battle by sumo is performed between two opponent's robots. The robots will try to locate the opponent robot and will try to knock him down. The sumo battle is performed in a circular ring, sized 1.5 m and painted in green, using the same Pantone colour as the others scenarios, as seen in Fig 5.

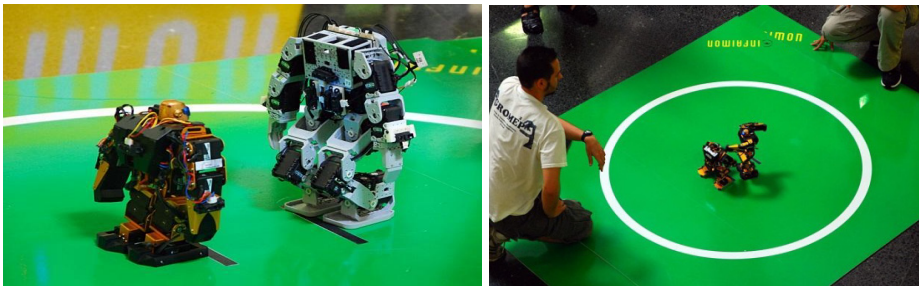


Fig. 5. Two robots fighting sumo in the circular ring

This trial is organized as a league, where all the robots should fight among all of them. Each combat has three rounds (2 minutes per round). The team (e.g. the robot) with highest score will win the round. The team's score will increase if the robot opponent: a) goes out the ring, b) touches the ground with the hand or knee, or c) is knocked down. Any non-competitive behaviour is sanctioned, e.g. erratic movements or a passive behaviour.

The Teams and Jury Panel

The members of a team should be students of any of the standard categories belonging to a University, Company or Research centre (e.g. undergraduate, graduate, MSc, or PhD student). Each team may consist of a maximum of five members.

The maximum number of robots per team is three. Obviously, it depends on each team to use only one robot per trial, or to use the same robot for all trials, etc. The registration is performed by mail through organizers. Each team should fill a form with all the relevant information: name, email, university, etc. about participants.

The judging panel performs the scheduling of the trials along the three days of the hosting event. The judging panel will be the responsible for scoring during the three aforementioned trials. With this aim, three members are designated: one referee, and two assistants. The referee will be the chair to take the main decisions during trials, with the help of the assistants.

4 Some Reflexions About CEABOT and Its Future

It is clear that designing this competition was not an easy task. Different types of goals were proposed, depending on the targeted audience. CEABOT, while attempting to be interesting and non-boring, was born in a scientific context, held by the CEA Robotics group, and focused on University researchers and students. Its aim is to foster the introduction to research, allow exchange of low cost approaches for exploring new scientific ideas and algorithms, but must maintain its appeal to students.

How to keep easy access to novel participants, while improving the difficulty of the tasks to perform in each test is not trivial. Adding or replacing trials or modifying rules in established tests implies changes that can bring failure (or success) to the event. How to measure this is also a relevant topic in the scientific community [18].

According to the initial CEABOT goal, "attract the student to the amazing world of the humanoid robotics", a complementary initiative has been adopted from the past five years by some of the co-authors (which are also co-founders and members), called ASROB[15]. They are currently combining efforts to create a new paradigm of education, where students, initially supervised by teachers, can develop new capabilities. This includes contributing with new ideas and experiences, working as a team with other students, and becoming more independent. The ultimate goal is to improve the student's abilities to face the challenges and difficulties in his/her professional life. The vision of our learning methodology was published in [19]. The

[15] ASROB: Robotics Society of the Universidad Carlos III de Madrid. http://asrob.uc3m.es

CEABOT initiative and ASROB has allowed creating the proper environment to set up an interesting experiment in peer-to-peer learning. Members of the class collectively set up a complex robotic system that is continuously evolving and that needs to be documented. Therefore, they have to maintain a community Wiki and a source code repository. They are also motivated to co-author and contribute to papers with their experiences (e.g., this one). From a curricular perspective, at most Universities, entering the first course is typically selective. However, Robotics Lab and ASROB are open to teach and introduce any student of any year and specialty (if any specialty). Another interesting approach has born from the ASROB incubator, like Robot Devastation: a new-generation shooter with augmented reality and real robots[16]. You can play online with other users with your PC or smartphone, moving robots in championships and campaigns.

4.1 Actual Difficulties of CEABOT Contest

Past editions have demonstrated that there are no negligible access barriers for students that are attracted to CEABOT contest: while humanoids robot kits have medium and low prices, the conference's accommodation and registration costs are expensive and obstruct their effective registration in the event. While the registration fee has been lower or also free for students participating at CEABOT, the travelling and accommodation costs are high due the itinerancy of CEABOT venue, that changes every year according to the University that hosts each year CEA's conference and the fact that the contest lasts the three days of the hosting conference.
It is important to also remark the personal effort of both students and teachers. Teams need many hours of tuning up, testing and programming in order to get a reliable platform minimally ready to achieve each one of the three trials. Although each year the difficulty of trials is modulated, each year the competition level trends to increase, and the help of past year veterans is a high value resource. Additionally, the dates for the conference are hosting the event every year in early September, forcing many participants to extend their work in labs in August (vacation in many Spanish institutions) to complete their preparations in time.

Tests and Contest Evolution
In section 3.2, the current set of tests has been described in detail. Originally, there were only three trials: robot sprint, stairs and sumo challenges. The sprint test was replaced after two editions, by the obstacle race, because stable gait skills at relative high speeds become a standard in almost teams. However, from 2010, the Free-style test was added with the intention of attracting more participants. As robot platforms become more capable, it is the intention of the organizers committee to evolve the current rules to present greater challenges. The current CEABOT scientific objectives are focused on postural path planning and control, and walking and climbing trials, where navigation in structured environments may only be successfully achieved by means of a proper interpretation of simple sensorial data, like IR proximity sensors,

[16] A demo video is available at: `http://www.youtube.com/watch?v=2RJayuBKR6Q`

inclinometers, and sometimes a gyro-compass. The quick evolution of robot platforms and decreasing prices of processors and complex sensors, like cameras, FRS etc., in conjunction with new open-source boards and source code, can drastically improve any off-the-shelf robot kit allowing achieve more sophisticated challenges. The issue of how newly available hardware or software can facilitate the inclusion of these topics, traditionally assumed to be too difficult, is also relevant. This is our own personal experience with the adoption of Arduino based Boards or x86 CPU boards.

With this in mind, the authors argue that the future of the competition is to maintain itself attractive enough to continue gaining popularity, especially among students, and increase the interest of these to participate and therefore initiate or strengthen their hobby in robotics. A possible critical point is the need of training programs for teachers who are often not confident with humanoids robotics, e.g. best practices by mentors, in order to reinforce their educational competences that help them improve their way to teach robotics.

4.2 Exporting the Contest: The iCOMIHU Proposal

The iCOMIHU competition is the CEA-GTRob initiative to become an international championship of mini-humanoid robots aimed at participation of European universities interested in this field. The first edition is scheduled for November of 2014, hosted the by IEEE-RAS HUMANOIDS 2014 conference. The underlying idea is to extend the CEABOT contest throughout all Europe. The first difficulty that appears is how to make the classification stage in order to get only a reduced group with the best of the teams. A deep review in the contest design, rules and scoring procedures is definitely needed.

5 Conclusions

The different successful robot competitions reviewed in Section 1, that have been organized in the last years, have shown a successfully way to let the student learners actively carry on experiments. Students are involved in physically assembling and programming a robot and using the on-board sensors, focusing on both the results and the abilities acquired of comparing the performance of the competing systems. This has been possible by means of very well defined rules, trials, procedures and metrics for scoring. As well as these scientific competitions have proved a quick way to attract substantial research efforts, rapidly produce high-quality working solutions, and additionally obtain a significant success in obtaining sponsors and popularity (e.g. media dissemination), CEABOT and other research initiation contests have proved to be a good tool for feeding students' curiosity and exploring new educational approaches [20], [21]. Along the past years, the authors' experiences show a very prolific collection of supervised works, ranging from Master Thesis to Final Degree Project works. Although a long-term evaluation is still not available to prove it formally, past years' results indicate a strong relation between robotic research and their educational and personal development among participants in this type of

educationally oriented contest. Due to this fact, it is mandatory to introduce some metrics for measuring the goals achievement obtained, as only score calculation in each trial could be no enough. Another question to keep in mind is related to the design of this Europe -wide competition of mini-humanoids. Does it be designed for easy measurement or adapt the competitions rules and procedures for easier benchmarking? How to manage the classification or scoring procedures, before the contest itself, for many teams of each nation subscribed is still an open question.

The experience gained in previous editions of CEABOT suggests that this kind of robotics competitions has a huge potential in exploring new educational approaches peer to peer and self-motivation of students. It is necessary take advantage of the learning experience that based on the effort of students, which learn to carry out technical projects with an initial supervision, investigate new resources and ideas, collaborate in teams, and develop new capabilities and personal attitudes.

The next editions of CEABOT must facilitate the student's hands-on exploration of legged locomotion, also at the undergraduate level. The setup of new hardware has been discovered as new challenge by itself and it could help in the expansion of CEABOT's scientific goals. The evolution of rules and contest procedures must be designed to facilitate the exchange of experiences, methodologies and materials to motivate students about science and technology, while keeping a balance of show, appeal and popularity that attract sponsors and assure the longevity of the contest.

References

1. Eguchi, A., Hughes, N., Stocker, M., Shen, J., Chikuma, N.: RoboCupJunior – A decade later. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 63–77. Springer, Heidelberg (2012)
2. VEX ROBOTICS Robotics Education & Competition Foundation (2011), http://www.roboticseducation.org/ (accessed Julio 2013)
3. Robótica Educativa NXT website (2009), http://us.mindstorms.lego.com/en-us/community/NXTLog/default.aspx
4. Visser, U., Burkhard, H.-D.: Robocup: 10 years of achievements and future challenges. AI Magazine 28(2), 115–132 (2007)
5. Papert, S.: The Children's Machine: Rethinking School in the Age of the Compute. Basic Books, New York (1993)
6. Demetriou, G.A.: Mobile Robotics in Education and Research. In: Mobile Robots - Current Trends. In: Gacovski, Z. (ed.) Intech (2011), http://www.intechopen.com/books/mobile-robots-current-trends/mobile-robotics-in-education-and-research
7. Melchior, A., Cohen, F., Cutter, T., Leavitt, T.: More than robots: An evaluation of the first robotics competition participant and institutional impacts. Technical report, Heller School for Social Policy and Management, Waltham, MA, An evaluation of the first robotics competition participant and institutional impacts (2005)
8. EUROBOT, Eurobot, International robotic contest (2011), http://www.eurobot.org

9. Ceccarelli, M.: Robotic Teachers' Assistants. IEEE Robotics & Automation Magazine 10(3), 37–45 (2003)
10. Greenwald, L., Kopena, J.: Mobile Robot Labs. IEEE Robotics & Automation Magazine 10(2), 35–32 (2003)
11. Jardón, A., Zafra, P., Martínez, S., Giménez, A.: CEABOT: Nationalwide Little humanoid robots competition; rules, experiences and new challenges. In: Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, Venice, Italy (2008)
12. Jardón, A., Sanz, P.J., Gómez, F., Felip, J., García, J.C.: CEABOT 2010 – Normativa. Comité Español de Automática (CEA). Universidad de Jaén, Spain (2010)
13. ROBO-ONE biped robot entertainment official web site, http://robo-one.com/ (accessed Julio 2013)
14. KONDO Ltd., Kondo Battle Oficial website, http://www.robospot.jp/html2/robospot_event_kobattle02.html (accessed Julio 2013)
15. K. K. C. LTD., http://www.ixs.co.jp (accessed Julio 2013)
16. ROBONOVA, http://robosavvy.com (accessed 2013)
17. Robotics, http://www.robotis.com/xe/bioloid_en (accessed Julio 2013)
18. Bonsignorio, F., Müller, V.C.: Benchmarking, Competitions and Replication of Experiments in Robotics and Cognition Research. In: Workshop on European Robotics Forum, Lyon, March 19-21 (2013)
19. González-Fierro, M., Jardón Huete, A., Martínez de la Casa, S., Stoelen, M. F., Victores, J.G., Balaguer, C.: Educational initiatives related with the CEABOT contest. In: Proceedings of SIMPAR 2010 Workshops Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, Darmstadt, Germany, November 15-16 (2010)
20. Buselli, E., Cecchi, F., Dario, P., Sebastiani, L.: Teaching Robotics through the Inquiry Based Science Education approach. In: 23rd International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum, Riva del Garda (TN), Italy (2012)
21. Kandlhofer, M., Steinbauer, G., Sundström, P., Weiss, A.: Educational Robotics - Evaluating long-term effects. In: International Workshop Teaching Robotics Teaching with Robotics, Integrating Robotics in School Curriculum, Riva del Garda (TN), Italy (2012)

Part XI

Perception for UAV

Visual Quadrotor Swarm for the IMAV 2013 Indoor Competition

Jose Luis Sanchez-Lopez, Jesús Pestana, Paloma de la Puente, Adrian Carrio, and Pascual Campoy

Centro de Automática y Robótica, CSIC-UPM (Spain)
{jesus.pestana,pascual.campoy}@upm.es
Web: www.vision4uav.com

Abstract. This paper presents a low-cost framework for visual quadrotor swarm prototyping which will be utilized to participate in the 2013 International Micro Air Vehicle Indoor Flight Competition. The testbed facilitates the swarm design problem by utilizing a cost-efficient quadrotor platform, the Parrot AR Drone 2.0; by using markers to simplify the visual localization problem, and by broadcoasting the estimated location of the swarm members to obviate the partner dectection problem. The development team can then focus their attention on the design of a succesful swarming behaviour for the problem at hand. ArUco Codes [2] are used to sense and map obstacles and to improve the pose estimation based on the IMU data and optical flow by means of an Extended Kalman Filter localization and mapping method. A free-collision trajectory for each drone is generated by using a combination of well-known trajectory planning algorithms: probabilistic road maps, the potential field map algorithm and the A-Star algorithm. The control loop of each drone of the swarm is closed by a robust mid-level controller. A very modular design for integration within the Robot Operating System (ROS) [13] is proposed.

1 Introduction

The motivation of the presented work is to design a low-cost framework for quadrotor swarm prototyping. The framework is designed to ease the main issues related to working with a multirobot system, which are obstacle avoidance and partner detection. The focus is to allow the team of designers to try various swarming behaviours on a real robotic swarm, so that the advantages of the proposed strategy can be experimentally demonstrated on an early stage of the development process. In order to obtain a cost-effective testbed the AR Drone 2.0 quadrotor was selected as the aerial swarm agent, as it is a low-cost but competitive platform. The second motivation is to participate in international robotics competitions using simple, but swarming, visual aerial robotic agents. More specifically, the presented work has been implemented to participate in the 2013 edition of the International Micro Air Vehicle (IMAV) Flight Competition.

There exists a large variety of applications which require a robotic system to densely navigate a wide area. Such applications can benefit from a swarming approach for the required data gathering of the problem at task, taking benefit from the robotic population. For instance, such an approach could be applied to security and surveillance tasks of middle sized areas.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_5,

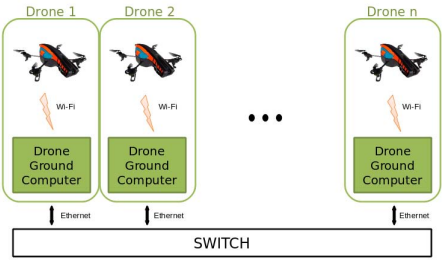


Fig. 1. The swarm is composed by identical robotic agents, which are composed of an AR Drone 2.0 which is commanded via Wifi from a ground station. The ground stations can communicate with each other under ROS via LAN, where one of them is running the roscore.

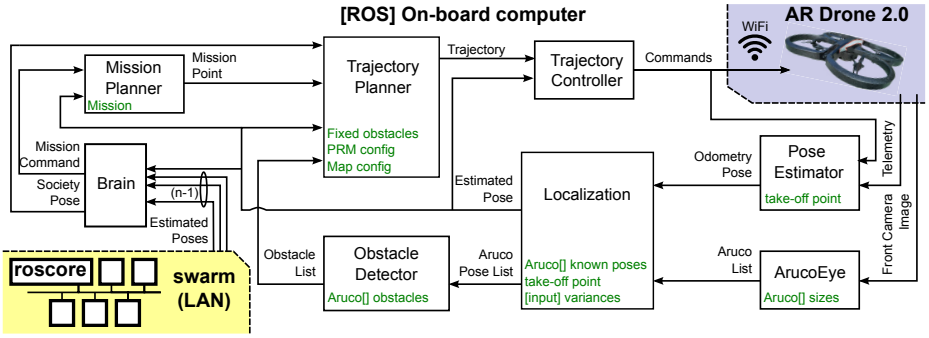


Fig. 2. Robotic agent software architecture. The ROS nodes of each agent are executed on a ground station which commands one AR Drone via WiFi. Each white box represents a module, and the green text inside it are configuration files. The localization module is implemented using an EKF which fuses the odometry based estimation with the ArUco visual feedback. This modules broadcasts the estimated pose to the mission and trajectory planning modules, to the controller modules and to the other robotic agents. The brain module receives the estimated position of the other robots and communicates it to the trajectory planner.

The IMAV Flight Competition is one of the most relevant competitions in Europe in the field of Autonomous Aerial Robotics and in the field of Small Remotely Piloted Air Systems (sRPASs). The Computer Vision Group (CVG) participated last year [20], in the 2012 edition, showing the potential and the research experience of the group. The learning experience obtained from the indoor dynamics competition encouraged us to keep working in the same line and also to try a swarming approach in the 2013 edition. Our motivation for participating in such competitions is to develop autonomous systems which can be later modified to perform civilian applications. This year's rules are significantly different with respect to 2012 edition. In the IMAV 2013 edition there is only one indoor competition (see [3]) which requires a high level of autonomy. The scenario has some fixed and previously known obstacles (a wall and four fixed poles) and several obstacles located at unknown positions (two windows and four obstacle poles). The indoor competition includes various challenges, among others: flying through a window,

flying through the obstacle zone, target detection and recognition, path following and precision landing.

After a deep analysis of the contest characteristics, a Visual Quadrotor Swarm was selected as the best option to join the 2013 IMAV indoor flight competition. A swarm composed by a significant number (more than 5) of relatively simple quadrotors (see section 2) is going to be used to achieve all missions except for the dropping one (mission 6). Additionally, as we decided to work with a visual swarm, a external visual aid described in section 3.2 is needed to solve the localization problem. Our swarm is going to be fully automated, and thus the level of autonomy is going to be "Autonomous Mission Control", requiring only one operator to start up and supervise/monitor the whole system (just in case something goes wrong, the operator could stop independent elements of the swarm, or the whole system).

2 System Description

The system is composed by a swarm of autonomous Unmanned Aerial Vehicles (UAVs). Each drone is autonomous enough to complete a previously defined mission avoiding obstacles and collisions with the other drones of the swarm. There is no high-level intelligence that controls or synchronizes the drones, the system then being a pure swarm. All the drones of the swarm share its pose in order to avoid the partner detection problem.

Each robotic agent is composed of the quadrotor platform and a ground station, see Fig. 1. The agent is composed of separate modules which work using the Robotics Operating System (ROS), see [13]. The Parrot AR Drone 2 [4] was selected as robotic platform, the characteristics of this platform are thoroughly explained in [5]. The ground computer communicates with the drone via Wi-Fi using the ardrone_autonomy ROS package [1]. On the other side, all the drone's ground computers are communicated via ethernet.

3 Drone's Modules

In this section the main modules, see figure 2, of each robotic agent of the swarm are described. The Pose Estimator and the Trajectory Controller modules are explained somewhere else [20,19].

3.1 Drone's Aruco Eye

The localization of the drone in the map is firstly estimated using the Pose Estimatmor module. Since this measure has drift, we need to correct it with an absolute measure when available using the module described in section 3.2. For this purpose, we use a visual external aid: ArUco Codes (figure 3, see [2]). This library allows to calculate the 3D pose of the camera with respect to each ArUco code.

There are two kind of ArUco codes defined in our system. The ones whose position is previously known and the ones whose position is unknown. Some of these ArUco codes are attached to all the obstacles (fixed known obstacles and fixed unknown obstacles).

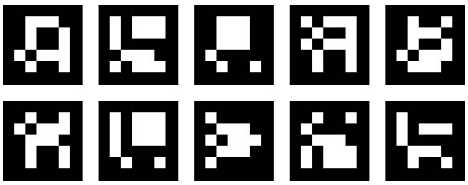


Fig. 3. Samples of ArUco Codes, which are markers used to simplify the obstacle detection and localization problems

With this distribution of codes, we are able to locate the drone with respect to the known ArUco codes, and to sense and locate the unknown obstacles with respect to the drone camera.

This visual aid helps us to solve two problems at the same in quite a straight forward manner: we solved the problem of sensing all the obstacles, that is a very hard task using only computer vision, and we also avoid the visual localization problem in a general environment, which is not solved precisely yet (even though there are some approaches like [11]).

3.2 Drone's Localization and Mapping

Localization in indoor environments is a challenging task for UAVs, especially if a low cost and very lightweight solution is required [17,21,14,12]. In the absence of GPS and laser sensors, visual approaches are very popular [21,14,12].

In the proposed system, the global localization of each drone is based on the IMU data and optical flow for the pose estimation, calculated by the Pose Estimator module. However, this measure has a drift which may be significant, so it should be corrected with more reliable information from the environment when available. For this purpose, we decided to use visual external aids, the ArUco codes [2] previously described. This library provides the 3D pose of the camera with respect to each ArUco code in a simple and convenient manner. The input of the localization node are hence the pose estimation result (similar to odometry) and the relative observations of the ArUco Codes, received by means of ROS messages.

Since the environment can be partially known a priori, some fixed landmarks are employed. ArUco codes with a priori known global poses are attached to the known poles. Other ArUco codes are placed on the wall and the unknown obstacles, and others are distributed over the floor, with an inclination angle of 45° and mounted on top of a platform 15 cm high. The latter are used so as to improve the visibility of ArUco codes when navigating among the poles; several tests were performed to qualitatively select an appropiate inclination working properly for the ArDrone's camera field of view. Simple and easy-to-use accessories for a fast arrangement of the ArUco codes in the environment were designed and built.

Localization with visual external aids for UAVs has been recently proposed in other works [21,14,12]. The method presented by Jayatilleke and Zhang [14] requires all the landmark poses to be known a priori and only works in limited areas, employing quite

a simple approach without filtering of any kind. The work by Faig *et al.* presents an interesting approach for local relative localization in swarms of micro UAVs, with the external marks always within the field of view. Our method was mainly inspired by the work by Rudol [21], but our models and formulation are quite different from those proposed by Conte [6].

We designed and implemented an Extended Kalman Filter (EKF) that allows the complete 6 DOF pose of the drone to be corrected by integrating the odometry data and the information from the external visual aids detection. The localization method benefits from the existence of known landmarks, but it also incorporates unknown detected features, using a Maximum Incremental Probability approach for building a map of 6 DOF poses corresponding to ArUco codes positioned in the environment. Similar methods for ground mobile robots were developed in previous work by de la Puente *et al.*, initially based on the observation of 2D point features with a laser scanner [8] and later based on the extraction of planar features from 3D point clouds generated by a tilting laser scanner [10,9].

In this work, the data association problem does not have to be addressed, since the ArUco readings provide unique ids for the observations and the landmarks. This way, loop closure is facilitated and enhanced robustness can be achieved with a not very cumbersome algorithm which showed nice empirical results in our initial tests.

Non linear state and observation models are used. At each iteration k, the prediction of the pose state x (6 DOF) is given by:

$$\tilde{\mathbf{x}}_\mathbf{k} = f(\mathbf{x}, \mathbf{u})_{\hat{\mathbf{x}}_{\mathbf{k-1}}, \mathbf{u}_\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k-1}} \oplus \mathbf{u}, \tag{1}$$

where the $\oplus$ operator corresponds to the composition of relative transformations in the 6D space. The noise in the odometry measurements is considered as gaussian white noise (as required to apply the EKF), and the odometry increment $\mathbf{u}$ is represented as $\mathbf{u} \sim N(\hat{\mathbf{u}}, Q)$.

The observation model is defined by the following innovation vector for an association of observation $\mathbf{o_i}$ and map landmark $\mathbf{l_j}$:

$$\mathbf{h_{i,i+5}} = \tilde{\mathbf{x}} \oplus \mathbf{o_i} - \mathbf{l_j} \tag{2}$$

The correction of the pose state is obtained by the update equation:

$$\hat{\mathbf{x}}_\mathbf{k} = \tilde{\mathbf{x}}_\mathbf{k} - W\mathbf{h_k} \tag{3}$$

where W is the Kalman gain matrix of the system. The covariance matrices are updated at each stage of the filter as required [22].

The environment is assumed to be static except for the presence of other drones. The accumulation of drift error if the drone is not able to detect ArUco codes all the time may require the incorporation of a forgetting mechanism so that the drone can navigate safely with local maps. In our tests thus far this has not been necessary due to the addition of extra ArUco codes over the floor, but this should be further investigated.

The input parameters of the algorithm (initial pose, covariance values, global poses and ids of the known landmarks) are read from an XML file, by means of the pugixml library [15]. The output is the corrected absolute pose of the drone and the list of global

poses of the landmarks belonging to the map. Other nodes subscribe to these topics, as shown in Fig. 2.

3.3 Drone's Obstacle Generator

Once the position of the unknown ArUco landmarks is obtained, they are processed in order to obtain higher level geometrical features in 2D to be used as obstacles by the trajectory planner. The map of obstacles is rebuilt at every iteration.

To do so, some prior information is required. Each of the obstacles is given a unique id and the ids of the ArUco codes belonging to it are provided. The radius of the poles is known. The poles are modeled with circles given by the coordinates of their center and the radius $c(x_c, y_c, r)$, while the walls are modeled with rectangles given by the coordinates of the center, the width and the length $R(x_c, y_c, w, l)$.

Given the observation of a landmark $\mathbf{l_j}$ belonging to pole i, an initial estimate of the circle i is very easily obtained:

$$(x_{c_i}, y_{c_i}) = \mathbf{l_j} + r\mathbf{dir} \tag{4}$$

with $\mathbf{dir} = (cos(yaw), sin(yaw))$. This initial estimate is further refined by the mean value of incorporating subsequent landmarks belonging to the same pole.

The distribution of the ArUcos of the windows has to be known more precisely. Currently, two different options are supported: the first solution is to place the ArUco codes at the corners of each window (with a predefined order) and the second solution is to place one ArUco code at each side of each window and two ArUco codes below each window (also with a predefined order). The second option seems to work best due to the fact that the ArDrone presents a horizontal field of view wider than the vertical field of view. Basic geometry is applied in order to obtain the rectangle models of the wall.

3.4 Drone's Trajectory Planner and Collision Avoidance

The objective of this module is the creation of a free collision 2D trajectory (horizontal coordinates x and y) to achieve a mission.

The module works as follows: a free-obstacles Probabilistic Road Map (PRM) [7] of the 2D map is generated off-line. The advantage of use a PRM instead of using a fixed-cell decomposition is that you can select the number of nodes in the graph and the neighbourhood of them. Also, if the robot is moving thought a zone with a lot of obstacles, new nodes can be added.

Once the free-obstacle graph is created, then an A-Star algorithm [18] searches the path using a potential field map function as the cost of the algorithm. This potential field map is built as a sum of a component that attracts the drone to the end of the obstacle zone and another component that repels the drone from any obstacle. The usage of a search algorithm (A-Star) instead of the potential field map alone [16], avoids the problem of the local minima.

There are three kinds of obstacle considered. The first type of obstacles are the fixed and previously known obstacles which are set during the start of the module and are

obstacles that never change its previously known position. The second type are the fixed and unknown obstacles that are received from the module described in 3.3 whose position could change over time, depending on how precisely are the ArUcos pose determined by the module 3.2. The last type are the unknown and moving obstacles that are other drones and are only considered in the path planning if they are near to the drone. The positions of other drones are received through the brain (see section 3.6).

Once the path is calculated using the A-Star algorithm, it is post-processed in order to obtain a shorter and more direct path, avoiding the noise produced by moving the robot from node to node of the PRM. The post-processing is done using the value of the potential field map. Fig. 4 shows a sample trajectory obtained by employing the exposed algorithm.

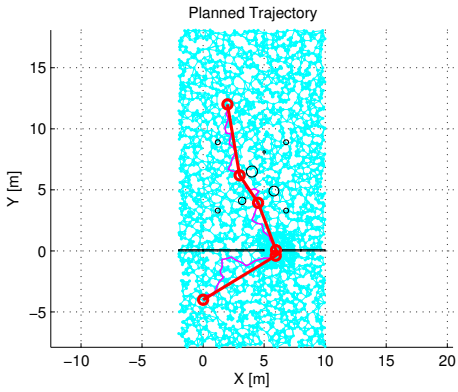


Fig. 4. Planned trajectory. In black, obstacles; in blue, the PRM; in magenta, the solution of the A-Star; in red, the post-processed trajectory

When a new pose of the drone or new positions of the obstacles are received, the planner checks if the new obstacles are outside the planned trajectory and if the drone is following the path. Otherwise, the trajectory is re-planned.

With this algorithm we solve the problem of the path planning and the collision avoidance, being able to navigate safely in the map using the Trajectory Controller module.

3.5 Drone's Mission Planner

The mission planner allows the operator to define a mission as a set of separate tasks; which are, in turn, fully described by a set of numeric parameters. The mission definition requires a xml file where the mission is described. It has available different tasks like take off, land, hover, sleep or move.

This module interacts with the trajectory planner (section 3.4) and with the localizer (section 3.2) when moving or with the brain (section 3.6) otherwise.

3.6 Drone's Brain

This is the very high-level module of each drone. It sends high-level commands to the drone like take-off, land or hover. It also communicates with other drones, receiving the pose of each other drone of the swarm.

The drone's brain communicates with all the modules, being in charge of monitoring each module state, activating or deactivating them.

4 Conclusions

This paper presented an overview of a whole swarm system designed to autonomously complete the indoor mission of the IMAV 2013 competition. The system is low-cost -employing Parrot ArDrone 2.0 quadrotors without any extra sensors- and deployment and setup are quite easy and straightforward due to the fact that only a limited number of known external ArUco codes is required.

The ArUco codes are used for localization and mapping, improving the pose estimation obtained from IMU data and optical flow by means of an EKF based method. The resulting map of ArUco codes is converted to higher level 2D geometrical obstacles used by a fast trajectory planner combining probabilistic roadmaps, the potential field map algorithm and the A-Star algorithm. All the drones have access to the global position of every other drone in the team. The corresponding obstacles are incorporated to obtain a safe trajectory. A robust mid-level controller employs the target global position given by the trajectory planner and the corrected pose of each drone in order to drive them to their respective goals, defined by a mission planner module. The system design and implementation is based on ROS, which makes code sharing, reutilization and monitoring easier.

This paper presents two additional contributions. First, a low-cost framework for visual quadrotor swarm prototyping was described. The framework allows to separate the complexity of some modules such as the localization and obstacle detection capabilities, from the swarm behaviour development and experimental testing. Second, the framework is used to prototype an aerial visual robotic swarm approach to security and surveillance applications, which is then used to participate on the IMAV2013 competition.

Some initial tests were carried out in simple environments, showing good performance results. We are currently building a real environment like the one shown in the competition rules in order to conduct more complex and realistic experiments and validations. Future work also includes designing and implementing a good graphical user interface for mission management. If the visibility of ArUco codes is not good in the whole area, the localization method may be modified so that old unknown obstacles get removed from the map. The trajectory planning algorithm may be updated with novel ideas.

References

1. ardrone autonomy ros stack,
`https://github.com/AutonomyLab/ardrone_autonomy/`

2. Aruco: a minimal library for augmented reality applications based on opencv, http://www.uco.es/investiga/grupos/ava/node/26
3. Imav, flight competition rules (2013), http://www.imav2013.org/index.php/information
4. Parrot ardrone 2.0 web, http://ardrone2.parrot.com/
5. The Navigation and Control Technology Inside the AR. Drone Micro UAV, Milano, Italy (2011)
6. Conte, G.: Vision-Based Localization and Guidance for Unmanned Aerial Vehicles. PhD thesis, Linkopings universitet (2009)
7. Motwani, R., Hsu, D., Latombe, J.C.: Path planning in expansive configuration spaces. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2719–2726 (1997)
8. de la Puente, P., Rodriguez-Losada, D., Pedraza, L., Matia, F.: Robot goes back home despite all the people. In: Proc. 5th. Conference on Informatics in Control, Automation and Robotics ICINCO 2008, Funchal, Portugal, pp. 208–213 (2008)
9. de la Puente, P., Rodriguez-Losada, D., Valero, A.: 3D Mapping: testing algorithms and discovering new ideas with USARSim. In: USARSim Workshop, IEEE Int. Conf. on Intelligent Robots and Systems, IROS (2009)
10. de la Puente, P., Rodriguez-Losada, D., Valero, A., Mata, F.: 3D feature based mapping towards mobile robots enhanced performance in rescue missions. In: Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, IROS (2009)
11. Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadrocopter. In: Proc. of the International Conference on Intelligent Robot Systems (IROS) (October 2012)
12. Faigl, J., Krajník, T., Chudoba, J., Saska, M., Přeučil, L.: Low-Cost Embedded System for Relative Localization in Robotic Swarms. In: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA). IEEE (2013)
13. Willow Garage. Ros: Robot operating system, www.ros.org/
14. Jayatilleke, L., Zhang, N.: Landmark-based localization for unmanned aerial vehicles. In: IEEE International Systems Conference (SysCon 2013), pp. 448–451 (2013)
15. Kapoulkine, A.: pugixml, http://pugixml.org/
16. Latombe, J.C.: Robot Motion Planning. Kluwer Academic (1991)
17. Mao, G., Drake, S., Anderson, B.D.O.: Design of an Extended Kalman Filter for UAV Localization. In: Information, Decision and Control (IDC 2007), pp. 224–229 (2007)
18. Raphael, B., Hart, P.E., Nilsson, N.J.: A formal basus for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4(2), 100–107 (1968)
19. Pestana, J.: On-board control algorithms for Quadrotors and indoors navigation. Master's thesis, Universidad Politécnica de Madrid, Spain (2012)
20. Pestana, J., Mellado-Bataller, I., Fu, C., Sanchez-Lopez, J.L., Mondragon, I.F., Campoy, P.: A general purpose configurable navigation controller for micro aerial multirotor vehicles. In: ICUAS 2013 (2013)
21. Rudol, P.: Increasing autonomy of unmanned aircraft systems through the use of imaging sensors. Master's thesis, Linkoping Institute of Technology (2011)
22. De Schutter, J., De Geeter, J., Lefebvre, T., Bruyninckx, H.: Kalman Filters: A Tutorial (1999)

Entropy-Based Search Combined with a Dual Feedforward-Feedback Controller for Landmark Search and Detection for the Navigation of a UAV Using Visual Topological Maps

Juan Pablo Fuentes, Darío Maravall, and Javier de Lope

Department of Artificial Intelligence, Faculty of Computer Science, Universidad Politécnica de Madrid, Madrid, Spain - Centro de Automática y Robótica (UPM-CSIC) Universidad Politécnica de Madrid, Madrid, Spain
juanpablo.fuentes.brea@alumnos.upm.es, dmaravall@fi.upm.es, javier.delope@upm.es

Abstract. We introduce a novel method for landmark search and detection for the autonomous indoor navigation of an Unmanned Aerial Vehicle (UAV) using visual topological maps. The main contribution of this paper is the combination of the entropy of an image, with a dual feedforward-feedback controller for the task of object/landmark search and detection. As the entropy of an image is directly related to the presence of a unique object or the presence of different objects inside the image (the lower the entropy of an image, the higher its probability of containing a single object inside it; and conversely, the higher the entropy, the higher its probability of containing several different objects inside it), we propose to implement landmark and object search and detection as a process of entropy maximization which corresponds to an image containing several target landmarks candidates. After converging to an image with maximum entropy containing several candidates for the target landmark, the UAV´s controller switches to the landmark´s homing mode based on a dual feed-forward/feedback controller aimed at driving the UAV towards the target landmark. After the presentation of the theoretical foundations of the entropy-based search. The paper ends with the experimental work performed for its validation.

Keywords: Unmanned Aerial Vehicles, Entropy search, Vision-based dual anticipatory reactive controllers, Nearest Neighbors Methods, Topological Maps.

1 Introduction

We have previously presented elsewhere[1] a dual anticipatory/reactive control architecture for the autonomous indoor navigation of an Unmanned Aerial Vehicle based on visual topological maps, in which the landmarks or relevant places are modeled as the vertices of a labeled graph and the graph´s edges correspond to specific UAV maneuvers. A critical requirement for a correct working of visual

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_6,

topological maps in robot navigation is obviously the search and recognition of the landmarks.

In this paper we approach this problem by proposing a novel method based on the use of the entropy[3] of an image as a means for object/landmark search and detection. The main idea behind the entropy-based search is the direct correlation between an image entropy and its probability of containing a single object (in the case of low entropy) or conversely of containing several different objects (in the case of high entropy). By using this simple idea, the critical task of landmark/object search and detection can be performed as a process of entropy maximization.

In Figure 1. we illustrate this empirical fact by displaying several images of different complexity and their corresponding entropy values.

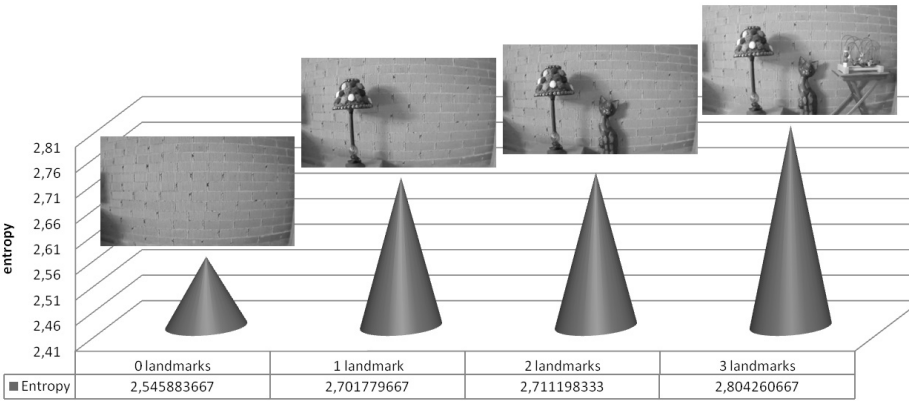


Fig. 1. In this figure we can observe the empirical fact that the higher the number of objects (landmarks) inside an image, the higher its entropy

This entropy-based landmark search could be substituded by a random search in which the robot also explores the environment to detect the target landmark although in a purely random strategy, which is less efficient than an entropy-based search strategy. After converging to an image with maximum entropy, that contains several candidates for the target landmark, the UAV´s controller switches to a homing mode in which a dual feed-forward/feedback controller is activated to guide the UAV towards the target landmark as described below. This process of switching between the UAV´s landmark Search Mode (S) and landmark Homing Mode (H) is controlled by the magnitude error ε or difference between the current image captured by the UAV´s onboard camera, and the target landmark´s image provided by the UAV´s navigation planner.

2 Entropy-Based Landmark Search

As previously stated, there is a direct correlation between the entropy of an image and the probability of the image containing several objects (in the case

of high entropy) or just a single object (in the case of low entropy). As visual landmarks in topological maps are usually chosen as outstanding single objects the task of visual landmark search, and detection can be formalized as a process of image entropy maximization aimed at converging to an image with maximum entropy that contains several candidates for the target landmark.

Therefore if we represent by u the UAV´s control variables (roll, pitch and yaw angles) this process of entropy maximization can be expressed as follows:

$$u = +\frac{\partial E(x,y)}{\partial u} \tag{1}$$

where E is the image entropy, as given by the standard definition of the entropy of the normalized histogram $H(I_k)$:

$$E[H(I_k)] = -\sum H(I_k)\, log_2\, H(I_k) \tag{2}$$

The expression (1) can be considered an entropy maximization process in the "control space". An equivalent entropy maximization process in the "physical space" is given by the following equations:

$$\dot{x} = +\frac{\partial E(x,y)}{\partial x} \tag{3}$$

$$\dot{y} = +\frac{\partial E(x,y)}{\partial y} \tag{4}$$

which can be expressed in terms of a control desired orientation for the UAV as follows:

$$\dot{\Phi} = tg - 1 \left[\frac{\frac{\partial E(x,y)}{\partial x}}{\frac{\partial E(x,y)}{\partial y}} \right] \tag{5}$$

where Φ stands for the UAV´s orientation.

3 The Feedforward/Feedback Controller for Landmark Homing

As previously mentioned, once the image containing the candidates for the target landmark has been obtained by the entropy maximization process in the Search Mode, the UAV´s controller switches to the Homing Mode (Figure 2.) to guide the UAV towards the target landmark.

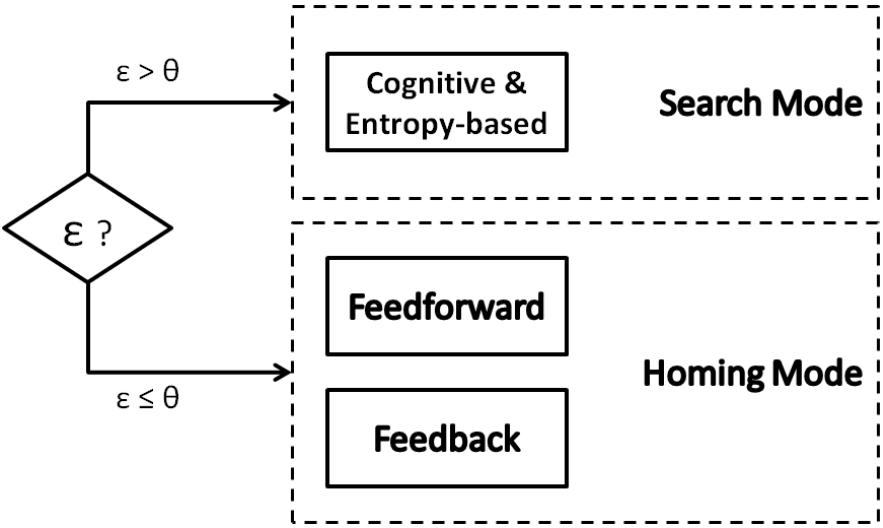


Fig. 2. Finite state automaton that models the UAV controller´s switching between the Search (S) and the Homing (H) modes depending on the error image magnitude (see the text for details) and according to the following heuristic rule: If { error is big} Then {Search Mode} Else{Homing Mode}

We have implemented this homing mode as a dual feed-forward/feedback control architecture (Figure 3.) which is constituted by the combination of a feedback module (either based on a conventional PD control or on a error gradient control) and a feed-forward module (based on either a Neurocontroller or a memory-based controller).

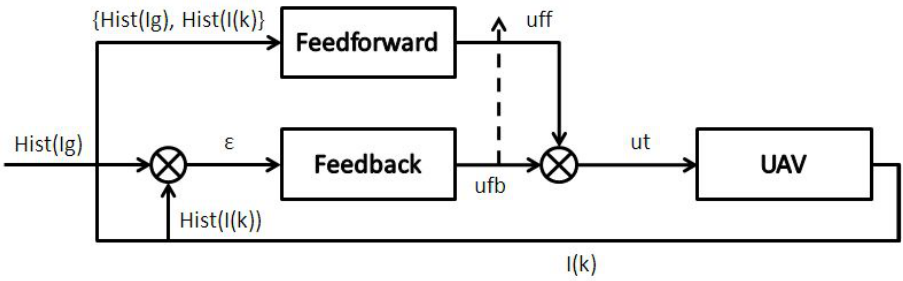


Fig. 3. The feedforward/feedback controller: Notice that the error signal ε is obtained as the difference between the histogram of the goal image $Hist[Ig]$ and the histogram of the current image $Hist[I(k)]$

This dual control architecture is shown in Figure 3., which shows the block-diagram of the dual feedforward / feedback controller[7,9]. Notice that both the feedforward or anticipatory controller and the feedback or reactive controller [4,6] receive as input the same image error, which is the difference between the target or desired image (i.e. the image corresponding to the next landmark in the topological map) and the current image captured by the UAV´s onboard camera.

More specifically, this vision-based error signal ε is obtained as the histogram of the identified landmark or histogram of the goal image $Hist[Ig]$ minus the histogram of the current UAV´s captured image $Hist[I(k)]$ during the k iteration of the controller. The feedback controller is implemented as a conventional PD control [8] and the feedforward controller is based on a inverse model [4,5] using a conventional neural network based algorithm.

4 A Preliminary Proposal of a Novel Method for the Detection and Recognition of Objects in Digital Images Based on Control Theory

The dual feedforward-feedback controller described in the preceding paragraph can be interpreted as a controller aimed at detecting and recognizing visual landmarks as the final convergence of the UAV´s controller to a zero error means that the current image captured by the UAV´s camera coincides with some of the prototype landmarks images stored in the UAV´s memory. Therefore, a zero error ε also means that the current image has been classified as one of the existing landmarks.

This control-based image recognition method can be straightforwardly formalized as a novel computer vision technique for the recognition of objects in digital images. As explained in the article, in which instead of controlling the UAV´s onboard camera by moving the UAV itself to converge to an image containing a specific object or landmark, we launch inside the image an automated search process aimed at detecting inside the image a specific object.

Thus, given the image of an object target I_{target} and given an image $I(x, y)$, the search of the target object is implemented as a dynamic search of the coordinates (x, y) of the center of a window sub-image I_w containing the target object according to an error gradient descent process:

$$\dot{x} = -\lambda \frac{\partial \varepsilon(x, y)}{\partial x} \tag{6}$$

$$\dot{y} = -\lambda \frac{\partial \varepsilon(x, y)}{\partial y} \tag{7}$$

where $\varepsilon(x, y)$ is an error signal defined as the difference of the histogram $H_{I_{target}}$ of the target object image I_{target} and the histogram $H_{I_w}$of the image

I_w inside the current explored window of $I(x, y)$. The search process can also be based on a conventional PD algorithm:

$$\dot{x} = k_p \varepsilon + k_d \frac{d\varepsilon}{dt} \tag{8}$$

$$\dot{y} = k_p \varepsilon + k_d \frac{d\varepsilon}{dt} \tag{9}$$

This dynamic search starts at an arbitrary random initial position $[x(0), y(0)]$ and hopefully converges to a final stable state of zero error, corresponding to a sub-image or window I_w of the image $I(x, y)$, exclusively containing a single pre-existing object or landmark, which has been finally detected and recognized once the dynamic search has converged to a state of zero error. Notice also, the intrinsic parallelism of this dynamic seach process as we can launch a multiple-class search inside the image $I(x, y)$.

The error signal ε can be obtained by either the L_1-norm (Manhattan distance) or by the L_2-norm (Euclidean distance) of both histograms:

$$\varepsilon_1 = \sum_{k=1}^{N} \mid (H_{I_{target}}[k]) - (H_{I_w}[k]) \mid \tag{10}$$

$$\varepsilon_2 = \sqrt{\sum_{k=1}^{N} \left((H_{I_{target}}[k]) - (H_{I_w}[k]) \right)^2} \tag{11}$$

where k is the position along of each image histogram.

5 Experimental Work

For the experimental work concerning the testing and validation of the proposed method for visual landmarks search and detection, we have used the quadrotor Parrot AR.Drone micro UAV as a well established and widely available robotics research platform [10,11].

In 2004 the Parrot company launched the project named AR.Drone with the final objective of producing a micro Unmanned Aerial Vehicle aimed at both the mass market of videos games and the home entertainment [11].

This ambitious project has involved from 5 to 12 engineers from Parrot with the support of external specialized companies like Sysnav and associated university partners like Mines Paris. Starting on August 2010, the A.R.Drone has been finally released on the market and it is widely available at a low price that make it possible to be used as a unique robotic platform for experimental work on UAVs .

Fig. 4. Parrot AR.Drone 2.0

All commands and images can be exchanged with a central controller via an ad hoc Wifi connection. The AR.Drone has an on-board HD camera and it has four motors to fly through the environment. This micro UAV supports four different control signals or degrees of freedom along the usual axes (roll, pitch, gaz and yaw).

This micro UAV has been extensively used for autonomous navigation at indoor environments [1,12].

5.1 Experiments on the Door Approaching and Crossing Maneuver

To test experimentally the proposed vision-based dual controller we have chosen the basic UAV´s navigation skill of "door approaching and crossing". The basic idea to test this UAV´s navigation skill is to get the UAV to fly towards a door as a target landmark in its visual topological map.

Once the UAV is approaching its target landmark and after its correct recognition it can activate its dual vision-based controller in order to safely transverse the door by monitoring and controlling the visual error.

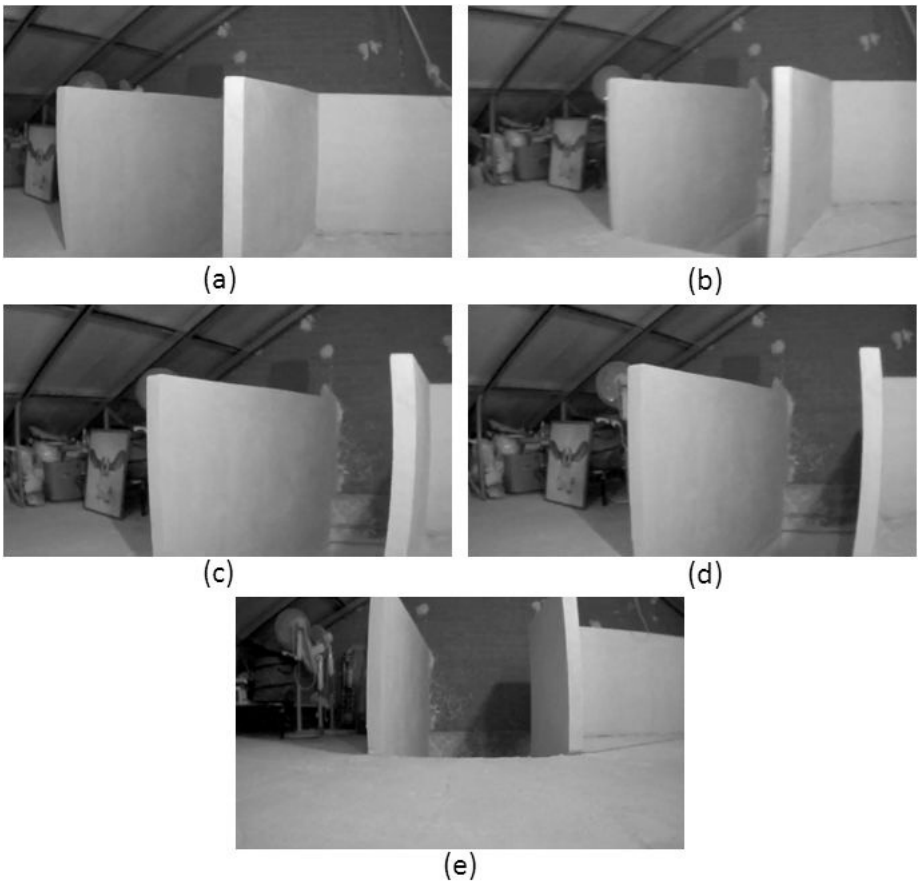


Fig. 5. Notice that the sequence (a)-(e) includes the successive images captured by the UAV while performing the maneuver: (a) is the initial state and (e) is the goal image

The Figure 5. shows a sequence of the images captured [2] by the onboard UAV´s camera while performing a door navigation maneuver. We have also displayed in Figure 6. the visual error signal that converges to zero as expected during this door navigation maneuver.

Notice also the control curves in Figure 7. giving an idea of the UAV´s control efforts applied during this maneuver. In both cases, k denotes the number of iterations of the controller during the maneuver (sequence (a)-(e)): (a) and (b) have been caught between $k = 1$ and $k = 9$, (c) and (d) between $k = 10$ and $k = 39$, and finally (e) corresponds from $k = 40$ to the end of the maneuver. At each iteration, the controller sends a control signal to the UAV; the time between two consecutive control signals is 30 ms.

Fig. 6. The visual error signal during the door-landmark approximation and crossing maneuver

During this approaching experiment, the UAV has used the pitch actuator (forward / back) and yaw actuator (rotation on its axis z), which were obtained by the controller based on the error signals received by the classifier K-NN.

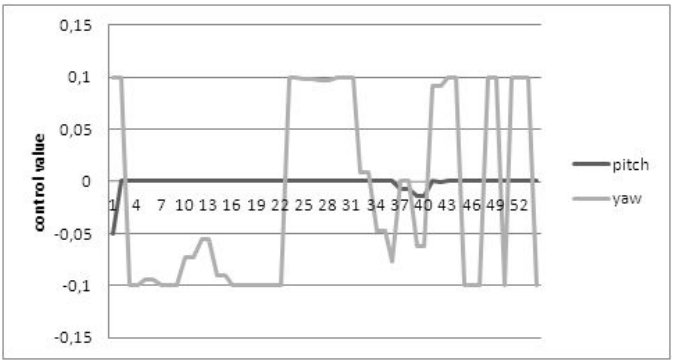


Fig. 7. The control signals during the door approximation and crossing maneuver

From the experimental results obtained in our laboratory and shown in Figures 6 and 7 we can conclude that the UAV is able to successfully perform in real time the fundamental skill of landmark door approximation and crossing by mean of the propossed vision-based dual feedforward/feedback controller.

5.2 Experiments on the Simultaneous Exploration and Obstacle Avoidance Maneuvers

The image entropy has also been applied for the implementation of both the exploration and the obstacle avoidance maneuvers. The first experiment was

performed when the UAV was inside an unknown state with a high error ε. In this case, the Search Mode was actived and the entropy of image captured was divided in three zones: right entropy H_r, center entropy H_c and left entropy H_l.

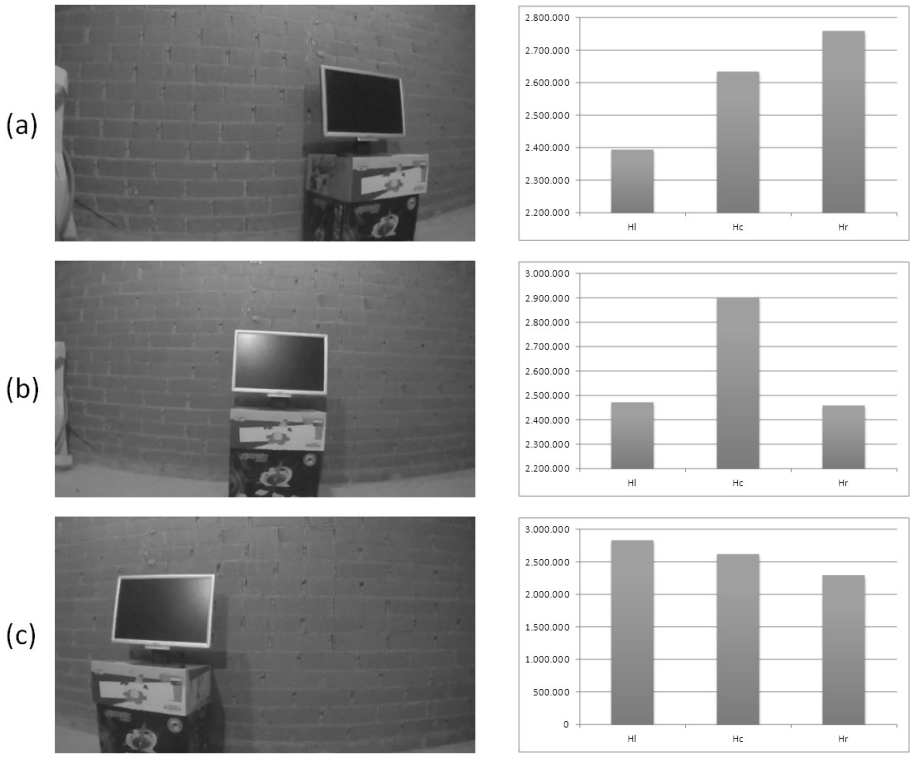


Fig. 8. The different image entropies H_r, H_c and H_l as computed during the simultaneous exploration and obstacle avoidance maneuvers: see the text for details

The Figure 8. shows that when H_r is the highest value (a) then the UAV must turn to the right zone of the environment. On the other hand, when the highest entropy is H_c (b), the UAV's controller must issue a forward maneuver to the landmark. Finally, the UAV turn to the left zone when the highest entropy is H_l (c).

The above image entropies variables H_r, H_c and H_l have also been applied when the UAV finds a obstacle in front of it.

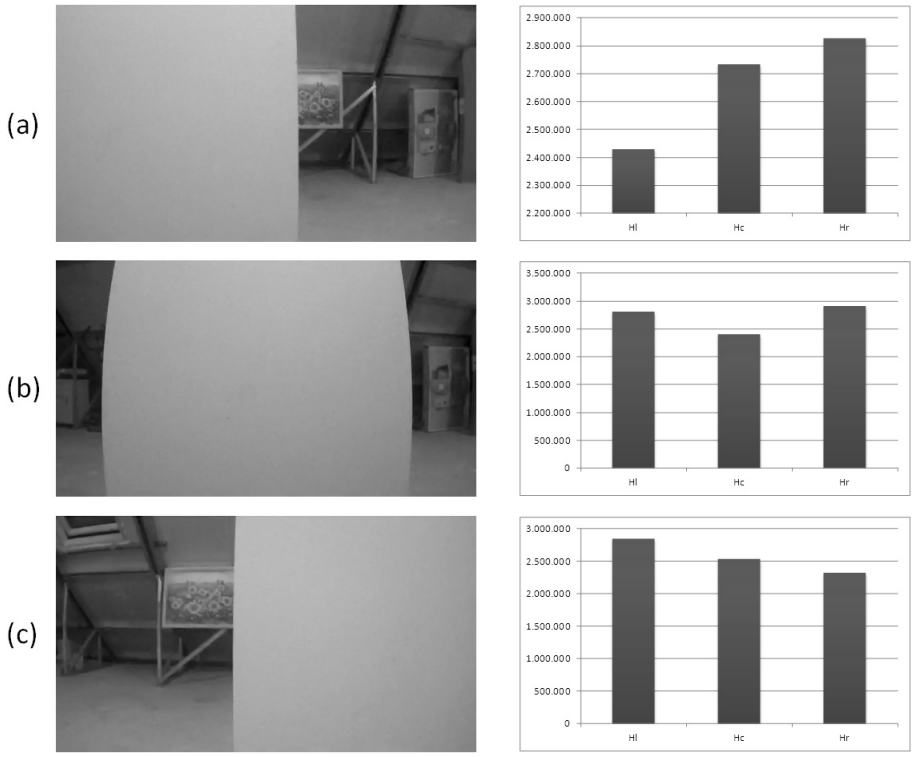


Fig. 9. The computed image entropies during the simultaneous performance of the exploration and obstacle avoidance maneuvers

The basic idea to implement the UAV's obstacle avoidance skill is to differenciate among several distinct relative positions of the obstacle and the UAV: right, left and in front of. Thus, in Figure 9. it is shown how the UAV is able to sucessfully avoid obstacles situated at different positions by conveniently using the entropy variables computed from the different relative positions: H_r, H_c and H_l.

6 Conclusions and Future Work

This paper has presented an entropy-based search combined with a dual feedforward-feedback controller for landmark search, and recognition for the autonomous indoor navigation of a UAV, that uses a visual topological map to autonomously navigate in the environment.

Apart from proposing the idea of using the image entropy for visual landmarks search, we have also proposed a solution to the problem of visual landmark recognition by means of a dual feedforward-feedback control architecture, in such a way that that the convergence of the UAV´s controller to a state of zero error

is equivalent to the final recognition of the corresponding visual landmark as the controller´s error signal, which is based on the difference between the current image captured by the UAV´s onboard camera and the target landmark´s image.

This idea of visual landmark recognition by controlling the UAV, has been formalized in this paper as a new method for object detection and recognition in digital images based on control theory. Thus, instead of controlling the UAV´s on board camera by moving the UAV itself, we propose to launch in the image an automated search process aimed at detecting inside the image a pre-specified object of interest by means of a control algorithm, in such a way that once the automatic search converges to a stable state of null error, it means that the object is finally detected and recognized. We plan to carry on future specific work on the experimental validation of this novel method, for object detection and recognition in digital images .

The paper finally presents the experimental work for the validation of the proposed topological navigation method centered on the basic skill for indoor navigation of door approaching and crossing, and also on the simultaneuos exploration and obstacle avoidance maneuvers based on entropy.

References

1. Maravall, D., de Lope, J., Fuentes Brea, J.P.: A Vision-Based Dual Anticipatory/Reactive Control Architecture for Indoor Navigation of an Unmanned Aerial Vehicle Using Visual Topological Maps. In: Ferrández Vicente, J.M., Álvarez Sánchez, J.R., de la Paz López, F., Toledo Moreo, F. J. (eds.) IWINAC 2013, Part II. LNCS, vol. 7931, pp. 66–72. Springer, Heidelberg (2013)
2. Maravall, D., de Lope, J., Fuentes, J.P.: Fusion of probabilistic knowledge-based classification rules and learning automata for automatic recognition of digital images. Pattern Recognition Letters (2013)
3. Oudeyer, P.-Y., Kaplan, F.: What is intrinsic motivation? A typology of computational approaches. Frontiers in Neurorobotics 1, Article.6 (2007)
4. Kawato, M.: Feedback-Error-Learning Neural Network for Supervised Motor Learning. Advanced Neural Computers (1990)
5. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Network 11(1998), 1317–1329 (1998)
6. Kawato, M.: Internal models for motor control and trajectory planning. Neurobiology 9, 718–727 (1999)
7. Imamizu, H., Kawato, M., et al.: Human cerebellar activity reflecting an acquired internal model of a new tool. Nature 403, 192–195 (2000)
8. Maravall, de, D., Lope, J.: Multi-objective dynamic optimization with genetic algorithms for automatic parking. Soft Computing 11(3), 249–257 (2007)
9. Barlow, J.S.: The Cerebellum and Adaptive Control. Cambridge University Press (2002)
10. Piskorski, S., Brulez, N., D'Haeyer, F.: A.R.Drone Developer Guide SDK 2.0, Parrot (2012)
11. Bristeau, P.-J., Callou, F., Vissiere, D., Petit, N.: The Navigation and Control Technology Inside the AR.Drone micro UAV. Preprints of the IFAC World Congress Milano (2011)
12. Jayatilleke, L., Zhang, N.: Landmark-based Localization for Unmanned Aerial Vehicles. IEEE (2013)

Topological Height Estimation Using Global Appearance of Images

Francisco Amorós, Luis Payá,
Oscar Reinoso, Luis Miguel Jiménez, and Miguel Juliá

Departamento de Ingeniería de Sistemas Industriales, Miguel Hernández University,
Avda. de la Universidad s/n, 03202, Elche (Alicante), Spain
{famoros,lpaya,o.reinoso,luis.jimenez,mjulia}@umh.es
http://arvc.umh.es

Abstract. In this work we present a comparison of different methods for topological height estimation applicable in UAVs navigation tasks using omnidirectional images. We take profit of the camera calibration information in oder to obtain different projections of the visual information from the omnidirectional images. The descriptors used to collect the visual information are based on the global appearance of the scenes. We test the algorithms using a real and dealing database.

Keywords: UAV, global appearance descriptors, zooming, omnidirectional image, topological navigation.

1 Introduction

Visual systems are commonly used in robotics navigation tasks. The richness of the information that a camera provides and the multiple possibilities of configurations and applications make them a popular sensing mechanism. We focus our work in omnidirectional vision and global appearance descriptors. In the literature, we can find numerous examples where omnidirectional visual systems are employed in navigation tasks, such as [19] and [9].

Classical research into mobile robots provided with vision systems has focused on local features descriptors, extracting natural or artificial landmarks from the image. Recent approaches propose processing the image as a whole, without local feature extraction. These global appearance techniques have demonstrate a good accuracy on the floor plane navigation in both location and orientation estimation [5], [3].

Nowadays, Unmanned Aerial Vehicles (UAVs) are becoming very popular as a platform in the field of robotic navigation research. In this sense, we can find in [12], [6], [18] different works that study the motion and attitude of UAVs using visual systems. Specifically, they are based on image feature extraction or image segmentation in order to extract valuable information of the scenes.

The aim of this paper is to extend the use of the global appearance descriptors to experiments where the height of the mobile robot changes. For that purpose,

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_7,

we suppose that the UAV is stabilized and the visual sensor has the same attitude, which corresponds with the perpendicular regarding the floor plane. In particular, we study the ability of height estimation using global appearance descriptors.

The experimental database is composed of omnidirectional images acquired using a catadioptric system composed of an hyperbolic mirror and a camera.

The remainder of the paper is structured as follows: Section 2 includes the global appearance descriptors we use in order to compress the visual information. Section 3 discusses the different methods used with the purpose of finding the relative height between images acquired in a same point in the floor plane. In the next section, the database used in the experiments is presented. Section 5 gathers the experimental results, and finally, the main conclusions are included in section 6.

2 Global Appearance Descriptors

In this section we summarize some techniques to extract the most relevant information from images. In particular, we describe descriptors based on the global appearance of scenes. These descriptors are computed working with the image as a whole, avoiding segmentation or landmarks extraction, trying to keep the amount of memory to a minimum.

2.1 Fourier Signature

In [11] the Fourier Signature is defined. It is possible to represent an image using the Discrete Fourier Transform of each row. So, we can expand each row of an image image $\{a_n\} = \{a_0, a_1, \ldots, a_{N-1}\}$ into the sequence of complex numbers $\{A_n\} = \{A_0, A_1, \ldots, A_{N-1}\}$:

$$\{A_n\} = \mathcal{F}[\{a_n\}] = \sum_{n=0}^{N-1} a_n e^{-j\frac{2\pi}{N}kn}, \quad k = 0, \ldots, N-1. \tag{1}$$

Taking profit of the Fourier Transform properties, we just keep the first coefficients to represent each row since the most relevant information concentrates in the low frequency components of the sequence. Moreover, when working with omnidirectional images, the modulus of the Fourier Transform of the image's rows is invariant against rotations in the perpendicular plane of the image. Representing each row of the original image as $\mathcal{F}[\{a_n\}]$ and $\mathcal{F}[\{a_{n-q}\}]$ the same row shifted q pixels, being q proportional to the relative rotation between images, the rotational invariance can be expressed with the shift theorem as:

$$\mathcal{F}[\{a_{n-q}\}] = A_k e^{-j\frac{2\pi qk}{N}}, \quad k = 0, \ldots N-1, \tag{2}$$

where $\mathcal{F}[\{a_{n-q}\}]$ is the Fourier Transform of the shifted sequence, and A_k are the components of the Fourier Transform of the non-shifted sequence.

2.2 2D Fourier Transform

When we have an image f(x,y) with Ny rows and Nx columns, the 2D discrete Fourier Transform is defined through:

$$\mathcal{F}[f(x,y)] = F(u,v) = \frac{1}{N_y}\sum_{x=0}^{N_x-1}\sum_{y=0}^{N_y-1} f(x,y)e^{-2\pi j\left(\frac{ux}{N_x}+\frac{vy}{N_y}\right)}, \\ u = 0,\ldots,N_x-1, v = 0,\ldots,N_y-1. \tag{3}$$

The components of the transformed image are complex numbers so it can be split in two matrices, one with the modules (power spectrum) and other with the angles. The most relevant information in the Fourier domain concentrates in the low frequency components. Furthermore, removing high frequency information can lead to an improvement in localization because these components are more affected by noise. Another interesting property when we work with panoramic images is the rotational invariance, which is reflected in the shift theorem:

$$\mathcal{F}[f(x-x_0,y-y_0)] = F(u,v)\cdot e^{-2\pi j\left(\frac{ux_0}{N_x}+\frac{vy_0}{N_y}\right)}, \\ u = 0,\ldots,N_x-1, v = 0,\ldots,N_y-1. \tag{4}$$

According to this property, the power spectrum of the rotated image remains the same of the original image and only a change in the phase of the components of the transformed image is produced, whose value depends on the shift on the x-axis (x_0) and the y-axis (y_0). Taking into account Eq. (4), the first row of the bidimiensional Fourier Transform, which corresponds with $v = 0$, is only affected by shifts on the x-axis, whereas the first column of the transform, which corresponds with $u = 0$, is only affected by shifts on the y-axis.

2.3 Spherical Fourier Transform

Omnidirectional images can be projected onto the unit sphere when the intrinsic parameters of the vision system are known. Being $\theta \in [0,\pi]$ the colatitude angle, and $\phi \in [0,2\pi)$ the azimuth angle, the projection of the omnidirectional image in the 2D sphere can be expressed as $f(\theta,\phi)$. In [4], it is shown that the spherical harmonic functions Y_{lm} form a complete orthonormal basis over the unit sphere. Any square integrable function defined on the sphere $f \in L^2(s^2)$ can be represented by its spherical harmonic expansion as:

$$f(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \widehat{f}_{lm}Y_{lm}(\theta,\phi), \tag{5}$$

with $l \in \mathbb{N}$ and $m \in \mathbb{Z}$, $|m| \leq l$. $\widehat{f}_{lm} \in \mathbb{C}$ denotes the spherical harmonic coefficients, and Y_{lm} the spherical harmonic function of degree l and order m defined in Eq. 6.

$$Y_{lm}(\theta,\phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta)e^{im\theta}, \tag{6}$$

where $P_l^m(x)$ are the associated Legendre functions.

It is possible to obtain a rotationally invariant representation from the Spherical Fourier Transform. Considering B the band limit of f, the coefficients of $e = (e_1, ..., e_B)$ are not affected by 3D rotations of the signal, where

$$e_l = \sqrt{\sum_{|m| \leq l} |\widehat{f}_{lm}|^2}. \tag{7}$$

We can find more information and examples of applications of the Spherical Fourier Transform in navigation tasks in other works such as [8], [10], [15], [7] and [16]

3 Height Estimation Methods

In this section we explain the different techniques used to obtain a measurement of the relative height of images captured in a same point. We make use of functions included in the Matlab Toolbox *OCamCalib* [14] to calibrate the camera and obtain different views of the visual information from the omnidirectional image.

3.1 Zooming of the Orthographic View

In [1], a method to obtain relative distance between images using zooming is presented. We propose to make use of the zooming concept with the purpose of measuring the vertical shift of a UAV.

However, we can not extract valuable information zooming the omnidirectional image directly. We need a representation of the image perpendicular to the robot movement. For that reason, we use the orthographic view [13] of the scene. In [9], [2], we find examples where orthographic view is used in robot navigation tasks.

We vary the distance of the plane where the omnidirectional image is projected to obtain different zooms of the bird-eye view by changing the focal distance.

The indicator of the vertical distance between two images using this method is the focal difference between both images.

After obtaining the orthographic view, we need to describe the scene. The descriptors we use are the Fourier Signature and the 2D Fourier Transform.

3.2 Camera Coordinate Reference System Movement

As shown in [17], given an image, it is possible to simulate the movement of the coordinate reference system (CRS) of the camera using the epipolar geometry, modifying the projection of the original image. The reprojected image, that uses the new CRS, reflects the movement of the camera.

Fist of all, we estimate the coordinates of the image in the real world in pixels. $m = [m_{x_{pix}}, m_{y_{pix}}]$ are the pixel coordinates regarding the omnidirectional image center. The camera calibration allows us to obtain the coordinates in the real world of the image. The image will be represented in the unit sphere $M \in \mathbb{R}^3$.

Then, we apply a change in the camera reference system:

$$M' = M + \rho \cdot T, \tag{8}$$

being T the unitary displacement vector in the z-axis, ($T = [0, 0, 1]^T$), and ρ a scale factor proportional to the displacement of the CRS.

Once we have the new coordinates of the image M', we can obtain the new pixel coordinates m'. Doing the association of the pixels of m with the new coordinates m', we obtain the new omnidirectional image that includes the camera CRS movement.

We have to take into account that, when we match the correspondences between m and m', some pixel coordinates of the new image might lay outside the image frame, and some other pixels might not have associated any value. We interpolate the values of the pixels that have not any association.

After obtaining the new coordinates of the image, we need to gather the visual information using a descriptor. Note that from M', we can obtain different representations of the visual information. Specifically, we use three different representations of the scene: the orthographic view of the omnidirectional image, the panoramic image, and the unit sphere. In Fig. 3, an example of each projection is shown.

We use the Fourier Signature and the 2D Fourier Transform to describe the orthographic and the panoramic views, whereas the Spherical Fourier Transform describes the unit sphere projection.

To obtain the height difference of two scenes captured in the same (x,y) position, we simulate different vertical CRS movements of the reference image, and compare them with the test image. Then, we look for the best image association, using the minimum Euclidean Distance of the image's descriptors.

The height difference using this technique is represented by the displacement scale factor ρ of the reference image.

4 Experimental Database

In order to carry out the experiments, we have acquired our own database of omnidirectional images in outdoor locations. We use a catadioptric system composed of a hyperbolic mirror and a camera with a resolution of 1280x960 pixels. The camera has been coupled to a tripod that allow us to have a range of 165cm in height.

The image acquisition has been done in 10 different locations. From every position, we capture 12 images in different heights. The minimum height is 125cm (h=1), and the maximum is 290cm (h=12), with a step of 15cm between consecutive images. Therefore, the database is composed of 120 images captured in real conditions. We do not vary the orientation of the images captured in a same location, although small rotations and short displacements have been unavoidable.

In the database, we include images near and far from buildings, garden areas and a parking. We also vary the time when the images are captured to change the illumination conditions and to have a more dealing database.

Fig. 1 and Fig. 2 include some examples of database images.

(a) (b) (c)

Fig. 1. Example of images captured in three different locations varying the relative position with the nearest building and the illumination conditions

(a) (b) (c)

Fig. 2. Example of images captured at three different heights in the same location. (a) is at a height of 125 cm, (b) is at a height of 200 cm and (c) is at a height of 290 cm.

In the experiments, we use different representations of the original visual information. Specifically, we compute the panoramic image, the orthographic view (or bird-eye view) and the projection onto the unit sphere. Fig. 3 includes an example of each representation.

5 Experiments and Results

Our goal is to check whether the different techniques provide a topological measurement in the image space proportional to the real change in height of the scenes. The topological indicator will depend on the height estimation method. For that purpose, we test all the methods included in Section 3 using three different experiments.

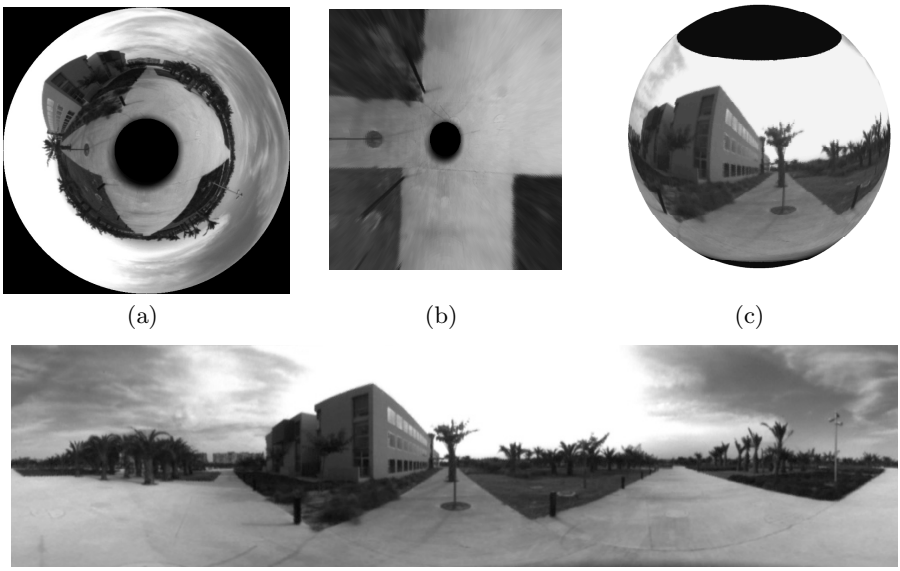


Fig. 3. Different projections of the same image. (a) Omnidirectional image, (b) Orthographic view, (c) Unit Sphere projection and (d) Panoramic view.

In the first experiment, we estimate the height of the images regarding lowest image in height ($h = 1$) for each location. We simulate several vertical shifts (with focal distance change or else the constant ρ variation depending on the technique). We compute the descriptor of the resulting images and create a comparison base with them. After that, we compare the other images captured in each localization, which are in different heights, with the base.

The match criteria is the minimum Euclidean Distance of the descriptors. In Fig. 4 we include the mean value and standard deviation of the height estimation the 10 different locations.

The second experiment is analogous to the first one, but we vary the reference image that forms the base. In this case, we choose the image corresponding to $h = 5$ (185 cm) as reference, having test images both below and above the comparison image. Fig. 5 includes the mean value and standard deviation of the results for the 10 different locations.

In the third experiment, we focus the analysis in the gradient of heights. For each location, we carry out as many comparisons as possible given a difference of heights, taking the reference images at different heights. Specifically, for $\Delta h = 2$ (i.e., 30 cm), we have 100 experiments; for $\Delta h = 4$, 80 experiments; for $\Delta h = 6$, 60; and for $\Delta h = 8$, 40 different comparisons. The results are included in Fig. 6.

Taking into account all the experimental results, we can confirm that all the methods present a monotonically increasing tendency as we increment the height lag between the compared images. Moreover, considering the results of the second

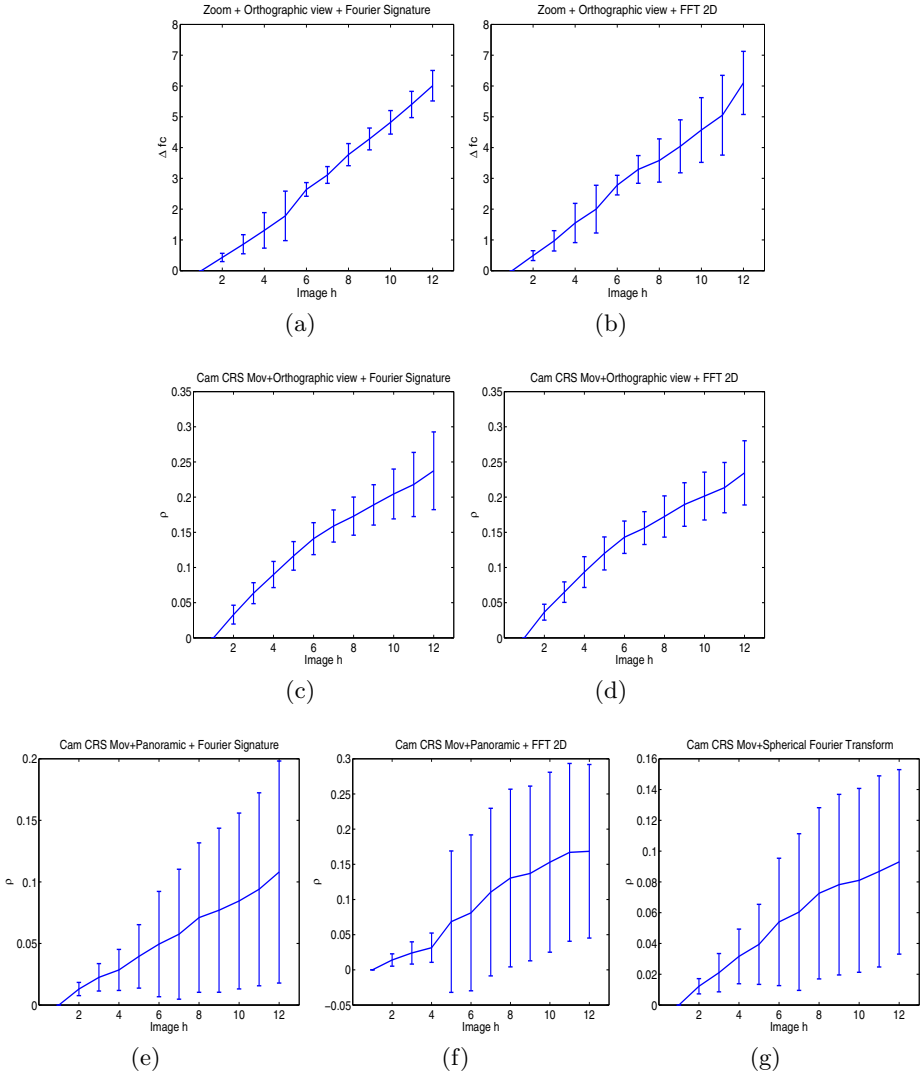


Fig. 4. Experimental results estimating the height regarding the image with h=0. Mean and standard deviation of all the different locations using the different methods: (a) Zooming over the the Orthographic view using the Fourier Signature descriptor, (b) Zooming over the the Orthographic view using the 2D Fourier Transform descriptor, (c) Camera CRS Movement with Orthographic view using the Fourier Signature descriptor, (d) Camera CRS Movement with Orthographic view using the 2D Fourier Transform descriptor, (e) Camera CRS Movement with Panoramic view using the Fourier Signature descriptor, (f) Camera CRS Movement with Panoramic view using the 2D Fourier Transform descriptor, and (g) Camera CRS Movement with Unit Sphere projection using Spherical Fourier Transform descriptor.

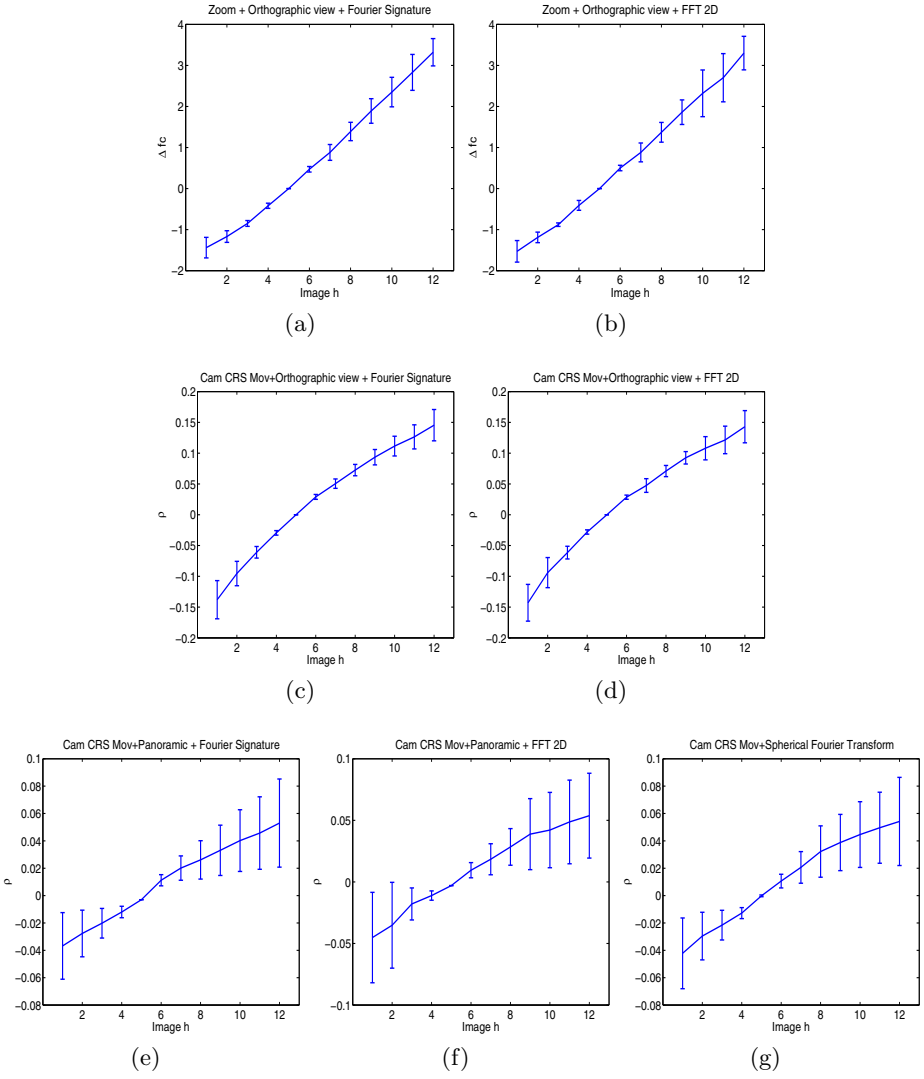


Fig. 5. Experimental results estimating the height regarding the image with h=5. Mean and standard deviation of all the different locations using the different methods: (a) Zooming over the the Orthographic view using the Fourier Signature descriptor, (b) Zooming over the the Orthographic view using the 2D Fourier Transform descriptor, (c) Camera CRS Movement with Orthographic view using the Fourier Signature descriptor, (d) Camera CRS Movement with Orthographic view using the 2D Fourier Transform descriptor, (e) Camera CRS Movement with Panoramic view using the Fourier Signature descriptor, (f) Camera CRS Movement with Panoramic view using the 2D Fourier Transform descriptor, and (g) Camera CRS Movement with Unit Sphere projection using Spherical Fourier Transform descriptor.

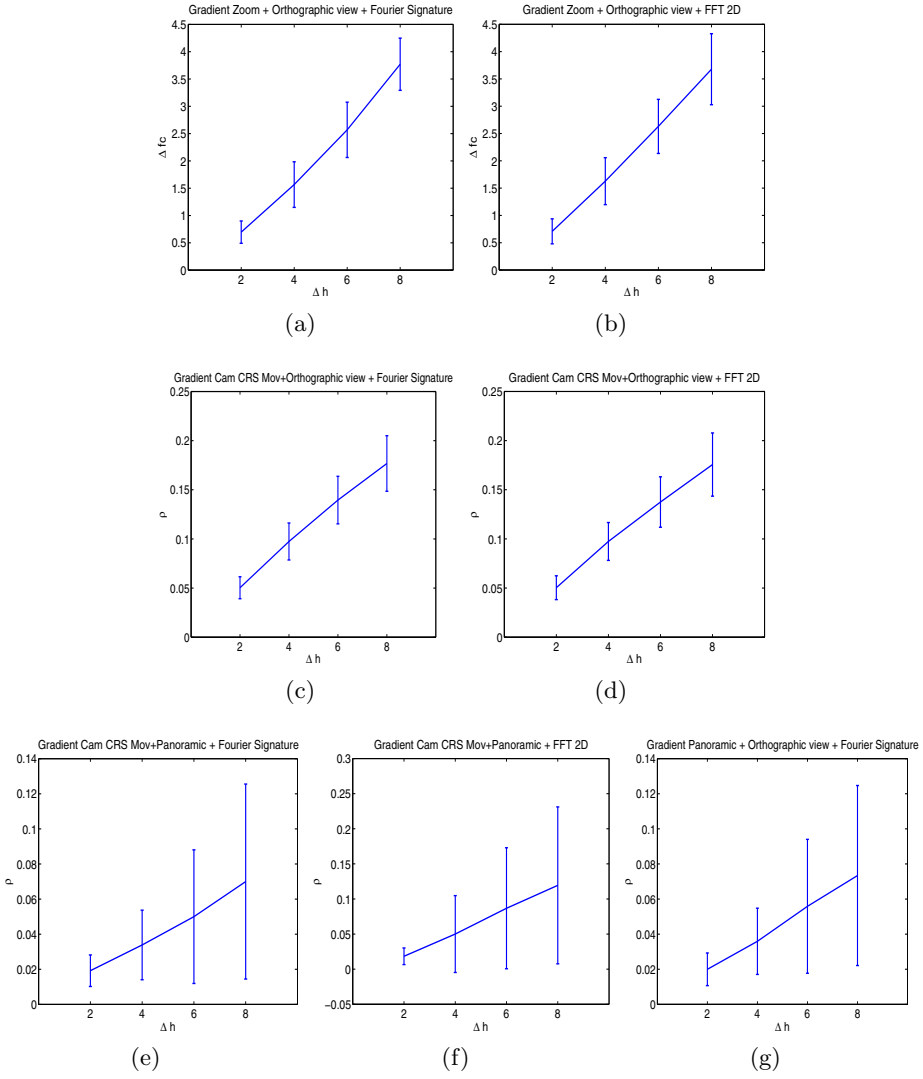


Fig. 6. Experimental results estimating four different gradients. Mean and standard deviation of all the possible experiments using the different methods: (a) Zooming over the the Orthographic view using the Fourier Signature descriptor, (b) Zooming over the the Orthographic view using the 2D Fourier Transform descriptor, (c) Camera CRS Movement with Orthographic view using the Fourier Signature descriptor, (d) Camera CRS Movement with Orthographic view using the 2D Fourier Transform descriptor, (e) Camera CRS Movement with Panoramic view using the Fourier Signature descriptor, (f) Camera CRS Movement with Panoramic view using the 2D Fourier Transform descriptor, and (g) Camera CRS Movement with Unit Sphere projection using Spherical Fourier Transform descriptor

experiment included in Fig. 5, when the test images are below the reference, the height indicator has negative sign. This allow us to determine the direction of the height difference.

The methods based on the orthographic view present better results than the techniques based on other image projections. As a rule, when we increase the height difference, the standard deviation increases. This is specially remarkable in the method based on the camera CRS movement that uses the panoramic view and the unit sphere projection.

When we simulate the CRS movement described in Eq.(8), we are applying the same displacement in all the pixels of the image, independently of the distance of the object depicted in the scene. However, when we change the height of the camera in the real world, the objects vary their position in the image depending on their relative position with the vision system. As an instance, the projection of objects that are far away from the camera changes less than the projection of closer objects when we vary the sensor location.

This is particularly notable when we work with the panoramic view or the unit sphere projection, as we use almost the whole image, that usually includes information of objects placed in different distances from the camera system. On the contrary, the orthographic view usually include elements that are at a similar distance (near the floor plane).

Despite this fact, the performance of all the algorithms are acceptable until a height lag of 45cm ($\Delta h = 3$).

Regarding to the descriptor used to represent the image, the Fourier Signature presents better accuracy than Fourier 2D, although there is no important difference in their performance.

In the experiments, we can also realize that the Spherical Fourier Transform over the unit sphere outperforms the Fourier Signature and the FFT 2D over the panoramic image. However, as stated above, the handicaps derived of the camera CRS movement technique affect the results.

6 Conclusions and Future Work

In this work we have presented a comparison of different topological height estimation techniques applicable in UAVs navigation tasks using omnidirectional images. The approaches we include in this work describe the visual information using global appearance descriptors. The experiments have been carried out using our own database captured in a real environment under challenging conditions.

The experimental results demonstrate that all methods proposed are able to estimate the relative height between two scenes captured in the same location for small height lags. However, the techniques based on the orthographic view of the scene present a better accuracy. Moreover, the Fourier Signature outperforms as a descriptor of the scenes.

The algorithm can deal with small rotations and short displacements, although it is not designed to work under bigger camera rotations or displacements. However, it would be to include the height estimation algorithm in a localization

system in order to locate the mobile and estimate the phase lag between the reference map and the current image. That way, we would be able to correct the phase lag between scenes and to use the height estimation algorithms proposed in this work.

The future work we should include the height estimation algorithm in a localization system in order to locate the mobile and estimate the phase lag between the map and the current image. It also should extend this research to include topological distance estimation taking into account 6D movements and topological mapping.

Acknowledgments. This work has been supported by the Spanish government through the project DPI2010-15308. Exploracin Integrada de Entornos Mediante Robots Cooperativos para la Creación de Mapas 3D Visuales y Topológicos que Puedan ser Usados en Navegación con 6 Grados de Libertad.

References

1. Amorós, F., Payá, L., Reinoso, Ó., Mayol-Cuevas, W., Calway, A.: Topological map building and path estimation using global-appearance image descriptors. In: ICINCO (2) (2013)
2. Bonev, B., Cazorla, M., Escolano, F.: Robot navigation behaviors based on omnidirectional vision and information theory. Journal of Physical Agents 1(1), 27–36 (2007)
3. Chang, C.K., Siagian, C., Itti, L.: Mobile robot vision navigation amp; localization using gist and saliency. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4147–4154 (2010)
4. Driscoll, J., Healy, D.: Computing fourier transforms and convolutions on the 2-sphere. Advances in Applied Mathematics 15(2), 202–250 (1994), http://www.sciencedirect.com/science/article/pii/S0196885884710086
5. Fernández, L., Payá, L., Valiente, D., Gil, A., Reinoso, Ó.: Monte carlo localization using the global appearance of omnidirectional images - algorithm optimization to large indoor environments. In: ICINCO (2), 439–442 (2012)
6. Han, K., Aeschliman, C., Park, J., Kak, A., Kwon, H., Pack, D.: Uav vision: Feature based accurate ground target localization through propagated initializations and interframe homographies. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 944–950 (2012)
7. Huhle, B., Schairer, T., Schilling, A., Strasser, W.: Learning to localize with gaussian process regression on omnidirectional image data. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5208–5213 (2010)
8. Makadia, A., Sorgi, L., Daniilidis, K.: Rotation estimation from spherical images. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 3, pp. 590–593 (2004)
9. Maohai, L., Han, W., Lining, S., Zesu, C.: Robust omnidirectional mobile robot topological navigation system using omnidirectional vision. Engineering Applications of Artificial Intelligence (2013), http://www.sciencedirect.com/science/article/pii/S0952197613000973

10. McEwen, J., Wiaux, Y.: A novel sampling theorem on the sphere. IEEE Transactions on Signal Processing 59(12), 5876–5887 (2011)
11. Menegatti, E., Maeda, T., Ishiguro, H.: Image-based memory for robot navigation using properties of omnidirectional images. Robotics and Autonomous Systems 47(4), 251–267 (2004)
12. Mondragón, I.F., Olivares-Méndez, M., Campoy, P., Martínez, C., Mejias, L.: Unmanned aerial vehicles uavs attitude, height, motion estimation and control using visual systems. Autonomous Robots 29(1), 17–34 (2010)
13. Roebert, S., Schmits, T., Visser, A.: Creating a bird-eye view map using an omnidirectional camera, pp. 233–240 (2008)
14. Scaramuzza, D., Martinelli, A., Siegwart, R.: A flexible technique for accurate omnidirectional camera calibration and structure from motion. In: IEEE International Conference on Computer Vision Systems, ICVS 2006, p. 45 (January 2006)
15. Schairer, T., Huhle, B., Strasser, W.: Increased accuracy orientation estimation from omnidirectional images using the spherical fourier transform. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, pp. 1–4 (2009)
16. Schairer, T., Huhle, B., Vorst, P., Schilling, A., Strasser, W.: Visual mapping with uncertainty for correspondence-free localization using gaussian process regression. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4229–4235 (2011)
17. Valiente, D., Gil, A., Fernández, L., Reinoso, Ó.: View-based slam using omnidirectional images. In: ICINCO (2), pp. 48–57 (2012)
18. Wang, C., Wang, T., Liang, J., Chen, Y., Zhang, Y., Wang, C.: Monocular visual slam for small uavs in gps-denied environments. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 896–901 (2012)
19. Winters, N., Gaspar, J., Lacey, G., Santos-Victor, J.: Omni-directional vision for robot navigation. In: Proceedings of the IEEE Workshop on Omnidirectional Vision, pp. 21–28 (2000)

Floor Optical Flow Based Navigation Controller for Multirotor Aerial Vehicles

Jesús Pestana[1], Ignacio Mellado-Bataller[1], Jose Luis Sanchez-Lopez[1], Changhong Fu[1], Iván F. Mondragón[2], and Pascual Campoy[1]

[1] Centro de Automática y Robótica, CSIC-UPM, Spain
jesus.pestana@upm.es, pascual.campoy@upm.es
Web: www.vision4uav.com

[2] Centro Tecnológico de Automatización Industrial Pontificia Universidad Javeriana, Colombia
imondragon@javeriana.edu.co
Web: http://www.javeriana.edu.co/blogs/imondragon/

Abstract. The presented work is focused on GPS-denied autonomous navigation for multirotor platforms. The problem of safe navigation based on noisy odometry measurements is discussed, and experimentally tested on the case of on-board optical flow measurements. A navigation controller is proposed which allows, through a group of configurable parameters, to ensure that the vehicle will fly on a speed specified flight envelope where the quality of the optical flow measurements is guaranteed. In order to attain safe navigation, the multirotor is modeled as a flying vehicle with specific kinematic constraints. The designed unperfect odometry based controller architecture has been experimentally tested on various multirotor vehicles, where the vehicles featured similar sensoring capabilities and the tolerance of our approach have been demonstrated. This work was implemented to compete in the International Micro Air Vehicle Conference and Flight Competition IMAV 2012, gaining two awards: the Special Award on "Best Automatic Performance - IMAV 2012" and the second overall prize in the category of "Indoor Flight Dynamics - Rotary Wing MAV". Most of the related code is available as two open-source projects hosted on GitHub.

1 Introduction

The motivation of this work is to experiment with unperfect odometry measurements as a means to stabilize the multirotor vehicle and to enable GPS-denied autonomous navigation. An important differente between Vertical Take-Off and Landing (VTOL) vehicles, such as multirotors, and ground robots is that they require to be stabilized using odometry or position measurements. If these measurements have a degraded quality for a long period of time (5-30 seconds), then due to small offset errors on the Inertial Measurement Unit (IMU) estimations the vehicle can end up crashing. Optical-flow based stabilization already exists and is implemented on a commercial multirotor, on the AR Drone 2.0, as explained in [1]. Optical-flow based navigation strategies are an on-going research topic on other groups, for example to enable fast calculations of the optical-flow [2], for collision and obstacle avoidance [3,4,5], and also to enable autonomous navigation [2,6].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_8,

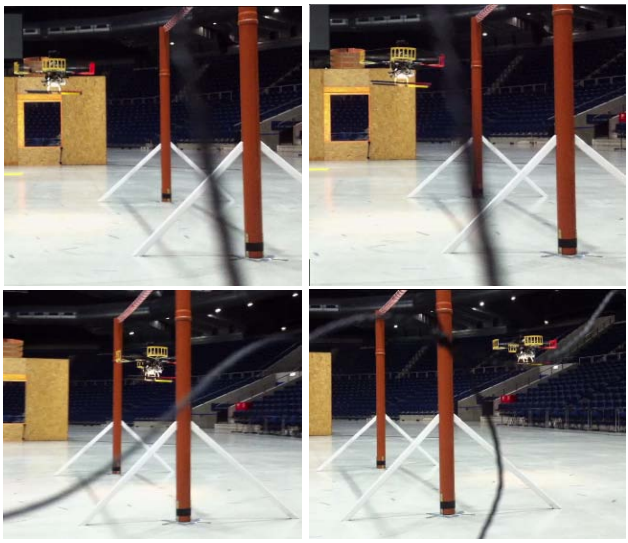


Fig. 1. The Asctec Pelican quadrotor, equipped as explained in Sec. 3, is shown flying in autonomous mode on the IMAV 2012 competition with an implementation of the controller presented in this paper. An EKF performs state estimation on the odometry measurements, and its estimation is fused with the position estimations from a Montecarlo Localization algorithm that is processing the Laser Range Finder readings. The quadrotor also had to be able to fly safely on areas where no obstacles were in range of the laser sensor. In this image sequence the quadrotor passes through a pair of poles using the laser sensor to estimate its position with respect to the poles. This flight gained the two IMAV 2012 awards stated in the abstract, see http://vision4uav.com/?q=IMAV12 . The full video of the flight is available online on http://vision4uav.com/?q=node/323 .

The presented research is focused on multirotor platforms, that are capable of flying in cluttered areas and hover if controlled properly. In order to obtain a semi-autonomous flying platform that can be commanded with high-level commands by an operator, the state estimation and the control problem must be addressed. On the presented and recent work by our group [7,8] a controller architecture to achieve stabilization and navigation objectives has been designed and experimentally tested on various multirotor vehicles: the AR Drone, the Asctec Pelican and the LinkQuad quadrotors. Among other related work, the presented architecture was implemented to compete, see Fig. 1, in the International Micro Air Vehicle Conference and Flight Competition IMAV 2012, gaining two awards: the Special Award on “Best Automatic Performance - IMAV 2012” and the second overall prize in the category of “Indoor Flight Dynamics - Rotary Wing MAV”. The code related to the presented work, which has been developed in our research group, is available in two open-source projects in GitHub[9,10], refer to section 3.2 for more information. Note that the work is still on-going and not all the control modes are currently available for the LinkQuad quadrotor.

2 Related Work

The following cited work has shown that the navigation control loops must be designed taking into account the non-linear dynamics of the multirotor, so that the control actions are coupled. If the control laws are well designed the multirotor can perform smooth trajectories in position coordinates, while orientating its yaw heading in any required direction.

Several labs have used a sub-millimeter accurate Vicon motion tracking system[11] to separate the control and the state estimation problems. Researchers at the GRASP lab of the University of Pennsylvania, and at the ETH - IDSC - Flying Machine Arena in the ETHZ, have been able to execute precise trajectory following[12,13], to perform aggresive maneuvers[13,14], and to perform collaborative tasks that require synchronization among the flying vehicles[15,16]. Relying on Vicon motion capture systems has simplified the research problem, which has shown that state estimation is the key to enabling many autonomous applications.

On the other hand, there are research groups that have shown successful autonomous multirotor navigation capabilities using only GPS positioning data. The STARMAC project[17], from Stanford University has shown several experimental tests on outdoors ranging from trajectory tracking tasks[18] to performing flips and stall-turn maneuvers[19,20].

Other groups have increased the situational awareness of the mUAVs using optical flow sensors and cooperative robots. For instance, a ground vehicle can estimate the position and attitude of the mUAV[21]. These two research works [3,5] are focused on navigation in corridors and obstacle collision avoidance strategies using multiple optical flow sensors. In outdoors navigation, optical flow sensors were used in a fixed-wing mUAV in[4] to avoid the collision with trees and other obstacles in a GPS waypoint trajectory.

3 System Overview

The presented research is a continuation on previous work by the same authors[22,23], that was implemented to participate on the IMAV 2012 indoors dynamics challenge[24]. Images from the flight that gained the awards of the IMAV 2012 competition are shown in Fig. 1, where the Asctec Pelican was controlled by the software architecture presented in this paper. In this case the localization problem was engaged using an Extended Kalman Filter (EKF) to fuse the odometry measurements, and a Particle Filter (PF) with a known map of the environment. The PF processed the Laser Range Finder readings providing estimated positions with respect to the pylons of the challenge[24]. The quadrotor also had to be able to fly safely on areas where no obstacles were in range of the laser sensor, where only the EKF provides the necessary feedback to the controller. The focus of this paper is to further researh on safe navigation based on unperfect odometry measurements, such as on-board optical flow measurements, and no position measurements are available.

This paper presents a navigation system architecture which controller can be configured depending on the limitations that are imposed by the kinematic capabilities of

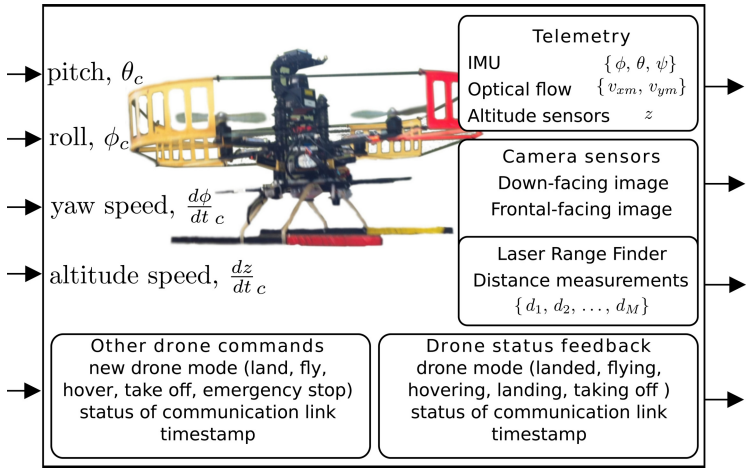


Fig. 2. Pelican on CVG Framework for interfacing with MAVs[9], user point of view. The Pelican offers several feedback channels: the telemetry package with IMU attitude, horizontal body speed and altitude data, and camera feedback channels where each camera is identified with an ID value. The drone modes are replicated from those available on the AR Drone.

the vehicle, by the measurement range of the speed estimators, or by precision requirements of the task at hand. The result is an architecture that was experimentally tested on GPS-denied environments, and that can be configured depending on the requirements of each phase of a task. This allows to have a setup for fast trajectory following, and another to soften the control laws and make the vehicle navigate more precisely and slowly whenever necessary.

3.1 Hardware Architecture

The implemented hardware architecture is inspired on the AR Drone quadrotor, which capabilities are described in[25]. An Asctec Pelican quadrotor was equipped with a camera and an on-board computer to obtain a similar setup. The advantages of using the Asctec Pelican are that all the algorithms can run on the on-board computer, and that the vehicle can be equipped with more sensors such as cameras and Laser Range Finders.

The Asctec Pelican shown in Figure 2 is equipped with an autopilot board that stabilizes the vehicle using information from GPS (only when outdoors), IMU, pressure altimeter and magnetometer fused using a Kalman Filter. This controller is embedded, closed, unmodifiable but gains are tunable. This quadrotor has also been equipped with a series of complementary sensors: a Hokuyo Scanning Laser Range Finder for horizontal depth mapping, a sonar altitude sensor for height estimation improvement and two cameras, one of them a downward looking sensor for velocity estimation based on optical flow. The optical flow is calculated using the OpenCV implementation of the pyramidal Lucas-Kanade algorithm for some image features selected by the Shi-Tomasi

method. The state estimation and control processing are exeuted onboard in an Atom Board, which has a dual core Atom 1.6 GHz processor with 1 GB of RAM.

3.2 Software Architecture

The AR Drone and the Asctec Pelican were both accomodated to be compatible with the CVG Framework for interfacing with MAVs. This framework, also called "MAVwork", was developed by our research group; it is an open-source project hosted on GitHub that can be accessed in [9], and its technical details can be found on [26,27]. One of the advantages of the framework is that it offers a common type of interface for every multirotor, allowing other software modules, such as the controller, to be compatible with multiple multirotors. The MAVwork interface has been designed to be similar to the one offered by the AR Drone, as this quadrotor has proven to be easy and reliable to setup for experimental tests. Among other advantages, MAVwork offers native take-off, landing and hovering flying modes; along with altitude hold capability. Fig. 2 summarizes the interface functions that the API and drone object offer to the developer.

The different components of the control and state estimation modules are shown in Fig. 3. An Extended Kalman Filter is used as data fusion algorithm to estimate the position and speed of the quadrotor. These estimates are then fed to the high-level (HL) controller that obtains the position, speed and feedforward commands that are given to the mid-level controller. The planning process that takes place in the HL controller uses a simple kinematic model of the multirotor that is specified by a little set of configurable parameters. The mid-level (ML) controller calculates and sends the commands to the CVG framework, which acts as interface between the controller and the quadrotor. Several saturation rules are used in order to perform the trajectory following task under certain safety conditions. The HL and ML controllers are further described in section 4. The state estimation and controller algorithms are available on an open-source project called "Multirotor Controller for MAVwork" hosted on GitHub that can be accessed in [10]. Its technical details are explained in the Msc. Thesis [28] and on the present paper.

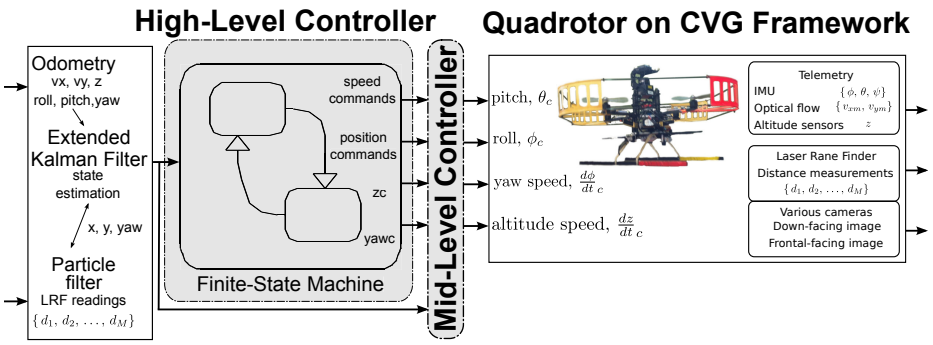


Fig. 3. General software architecture of the navigation controller. The State Estimation is used by the High-Level controller to determine optimized reachable speed commands to the mid-level controller, which sends commands to the drone through the CVG Framework[9,26,27]

The navigation of the multirotor is stabilized using odometry-based position estimates and for that reason the boundaries where the quality of these estimates is lost must be determined. For instance, the optical flow algorithm that is calculated on the down-facing images is affected by the texture and reflectiveness of the floor surface, and on the lighting conditions of the environment. The size of the floor texture details and the computing power of the on-board computer set a maximum speed boundary on the speed estimation. The maximum speed that could be reliably measured in the indoors playground where most of the tests where carried out was about 0.5-0.75 m/s, and so, this boundary was used in our indoors tests.

The presented software architecture has been implemented on the AR Drone and Astec Pelican quadrotors, and will be applied in the future to other multirotor platforms, more specifically to a LinkQuad[29] quadrotor and an OktoKopter from Mikrokopter [30]. The usage of a common software architecture increases code quality shareability and enables for different research lines to work in a cohesive way towards the objectives of our group.

4 Navigation Controller

This section introduces the navigation controller that uses a set of configurable parameters to adapt to different multirotors and sensor limitations. The parameters are inspired by the STARMAC project work described in[18]. First a simple multirotor point mass kinematic model is described. Then the high level controller state-machine is introduced with the description of the configurable parameters that are derived from the model. The final part of the section introduces the middle-level controller.

4.1 Multirotor Point Mass Kinematic Model

A multirotor is a flying vehicle that can exert force in any direction and thus can be simplified as a point mass moving in the 3D space. The vehicle tilts to exert force in the horizontal plane, and changes the average propeller speeds to control the force in the altitude coordinate. A maximum speed is also set by various reasons, for instance, due to the aerodynamic friction of the movement through the air. It is also important to take into account that the multirotor is commanded using feedback loop control laws that produce a response that can be characterized by a little set of parameters. The speed planifier can take advantage of a simple kinematic model built upon these facts in order to take decisions.

The body frame, $\{X_m, Y_m, Z_m\}$, of the vehicle, shown in Fig. 4, is described as follows. The $\{X_m, Y_m\}$ axis are horizontal and the Z_m axis points down to the floor. The X_m axis points to the front of the vehicle and the Y_m axis points to the right obtaining a orthonormal right-handed reference frame. The euler angles are denoted $\{\phi, \text{roll}\}$, $\{\theta, \text{pitch}\}$ and $\{\psi, \text{yaw}\}$, and they define the rotation between the body frame, $\{X_m, Y_m, Z_m\}$, and the world frame, $\{X, Y, Z\}$.

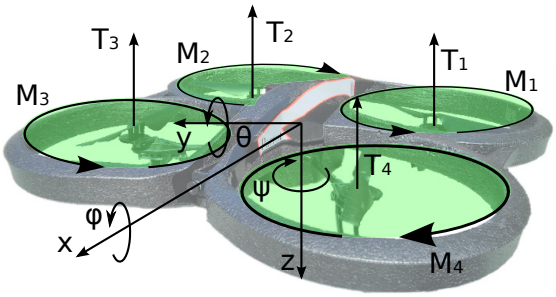


Fig. 4. Free body diagram of a quadrotor. The four propellers, and each of their performed thrust T_i and torque M_i are labeled 1-4. The euler angles are denoted $\{\phi, \text{roll}\}$, $\{\theta, \text{pitch}\}$ and $\{\psi, \text{yaw}\}$

The quadrotor rigid body dynamics model, see Eq. 1 and Fig. 4, is easy to infer from physical laws and is explained in multiple articles [18,13]:

$$\begin{aligned}
I_x\ddot{\phi} &= \dot{\psi}\,\dot{\theta}\,(I_y - I_z) + l\,(T_1 + T_2 - T_3 - T_4)\\
I_y\ddot{\theta} &= \dot{\phi}\,\dot{\psi}\,(I_z - I_x) + l\,(T_1 + T_4 - T_2 - T_3)\\
I_z\ddot{\psi} &= \dot{\theta}\,\dot{\phi}\,(I_x - I_y) + \textstyle\sum_{i=1}^{4} M_i\\
m\ddot{x} &= (-sin\phi\, sin\psi - cos\phi\, sin\theta\, cos\psi)\textstyle\sum_{i=1}^{4} T_i - K_{fr}\dot{x}^2\\
m\ddot{y} &= (-cos\phi\, sin\theta\, sin\psi + sin\phi\, cos\psi)\textstyle\sum_{i=1}^{4} T_i - K_{fr}\dot{y}^2\\
m\ddot{z} &= mg - cos\theta\, cos\phi\textstyle\sum_{i=1}^{4} T_i
\end{aligned} \tag{1}$$

Where the following the following variables have been introduced: the position of the quadrotor with respect to the world frame is denoted by the $\{x, y, z\}$ coordinates, the attitude of the vehicle is represented by the yaw ψ, pitch θ and roll ϕ euler angles; the quadrotor rigid body is characterized by its mass m and its three principal mass moments of inertia $\{I_x,\ I_y,\ I_z\}$, each propeller i generates a thrust T_i and a heading torque M_i, K_{fr} is an aerodynamic friction constant and the l constant is the arm between each pair of thrusts, in this case the distance between the propellers 1 and 2, as denoted in Fig. 4.

The following control loops are usually implemented inside the autopilot board of the multirotor: the attitude (roll, ϕ, pitch, θ, and yaw, ψ) control loops and altitude, z, control loop. The autopilot boards usually accept roll, pitch, yaw speed and altitude speed references. So that the saturation bounds for these variables is set by the autopilot board capabilities. Thus, it is only required to further develop the equations of mouvement in the horizontal plane.

If the flight is performed at constant altitude then, $\sum_{i=1}^{4} T_i \approx mg$, and taking the approximation to low roll and pitch angles, then the equations that define the mouvement in the horizontal plane are derived from Eqs. 1:

$$\begin{aligned}
m\ddot{x} &= (-\theta\, cos\psi - \phi\, sin\psi)\, mg - K_{fr}\dot{x}^2\\
m\ddot{y} &= (-\theta\, sin\psi + \phi\, cos\psi)\, mg - K_{fr}\dot{y}^2
\end{aligned} \tag{2}$$

Two additional attitude variables are defined $\{\phi_v, \text{virtual roll}\}$, $\{\theta_v, \text{virtual pitch}\}$, which control de position of the vehicle in the horizontal $\{X, Y\}$ plane. They are related to the actual $\{\phi, \theta\}$ through the yaw (ψ) angle as follows:

$$\begin{bmatrix}\theta_v \\ \phi_v\end{bmatrix} = \begin{bmatrix}\cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi)\end{bmatrix} \begin{bmatrix}\theta \\ \phi\end{bmatrix} \tag{3}$$

The attitude controller ($t_r \approx$ 200-300 ms) is significantly faster than the horizontal speed loop ($t_r \approx$ 1-2 s). This fact allows to correct the coupling introduced by the yaw commands in the horizontal speed loops using the decoupling law, defined by Eq. 3. The reference frame change defined by Eq. 3 is included in the controller, so that the $\{\phi_v, \theta_v\}$ can be used for the controller design.

The dynamic model of the multirotor is much simpler when expressed using the virtual angles and a maximum value for the quadrotor acceleration saturation can be infered from Eqs. 2 & 3 as follows:

$$\ddot{x} \approx -g\,\theta_v - (K_{fr}/m)\dot{x}^2 \rightarrow |\ddot{x}| \leq g\,|\theta_{v\,max}| - (K_{fr}/m)\dot{x}^2 \tag{4}$$

$$\ddot{y} \approx g\,\phi_v - (K_{fr}/m)\dot{y}^2 \quad \rightarrow |\ddot{y}| \leq g\,|\phi_{v\,max}| - (K_{fr}/m)\dot{y}^2 \tag{5}$$

The maximum horizontal speed of the vehicle, $v_{xy\,max}$, is physically limited by the aerodynamic friction. Thus, the maximum quadrotor speed is expressed by Eq. 7 and the following model:

$$\left(\dot{x}^2 + \dot{y}^2\right) \leq v_{xy\,max}^2 \tag{6}$$

The altitude acceleration can be modeled similarly, but the AR Drone only accepts altitude speed commands; which is the reason to constrain the altitude speed and not its acceleration:

$$|\dot{z}| \leq \frac{dz}{dt}_{c\,max} \tag{7}$$

The point mass model has to take into account that the vehicle is being commanded using feedback control laws, and that the controller will not be able to push the vehicle capabilities to the physical limits stated above. Thus the following set of parameters is selected based on this fact and on the previous considerations:

- The combination of Eqs. 7 & 6 set a maximum speed on any direction.
- The speed response of the vehicle can be characterized by the response time of the speed control loop $t_{rv\,max}$
- The maximum acceleration is physically limited by Eqs. 4 & 5, and can be further determined by experimental tests. But the actual accelerations obtained during trajectotry control depend on the performances of the controller and on the smoothness of the speed references set to the speed control loop. Thus, the maximum acceleration parameters in the horizontal plane are derived $\{a_{x\,max}, a_{y\,max}\}$.
- The trajectory around checkpoints can be approximately modeled by a sector of circumference thus requiring a centripetal acceleration that is limited by

$$a_{c\,xy} \;\leq\; g\{\;\theta_{v\,max},\;\phi_{v\,max}\} \;=\; \frac{v_t^2}{R_c} \tag{8}$$

where v_t and R_c are the average speed and curve radius during the turn.

4.2 Middle-Level Controller

The mid-level controller is shown in Fig.5 and it is inpired on previous research from other groups[12,13,18]. Its general architecture is a cascade controller consisting of an inner speed loop and an outer position loop. But it also includes control laws to take into account the overall system capabilities. Some characteristics of the control architecture decipted in Fig. 5 are:

- The position controller loop only commands achievable speeds to the speed controller loop, see left block in Fig. 5. The reference (feedforward) speed along the trajectory is saturated to $v_{at\,max}$ and $v_{at\,zmax}$; and the cross-track reference speed is saturated to $v_{ct\,max}$ and $v_{ct\,zmax}$. Both speed limits are set so that $v_{at\,max} + v_{ct\,max} \leq v_{xymax}$, $v_{at\,zmax} + v_{ct\,zmax} \leq v_{zmax}$.
- The planned along-track velocity $\{v_{at\,xc}, v_{at\,yc}, v_{at\,zc}\}$ is lower than the maximum velocity, thus, giving the relative position controller a speed margin to work on.
- The reference change block, which implements Eq. 3 ensures that the roll and pitch commands are decoupled from the yaw heading.
- The aerodynamic friction is partially linearized, in the NLF blocks, using the inverse of the identified aerodynamic friction.
- In the PID modules the derivate action is filtered to limit noise amplification and the integral action features anti-windup saturation.
- The variables $\{\theta_{ff}, \psi_{ff}\}$ are feed-forward commands that are calculated by the state machine using the planned acceleration and Eqs. 4 & 5 without considering the aerodynamic friction.

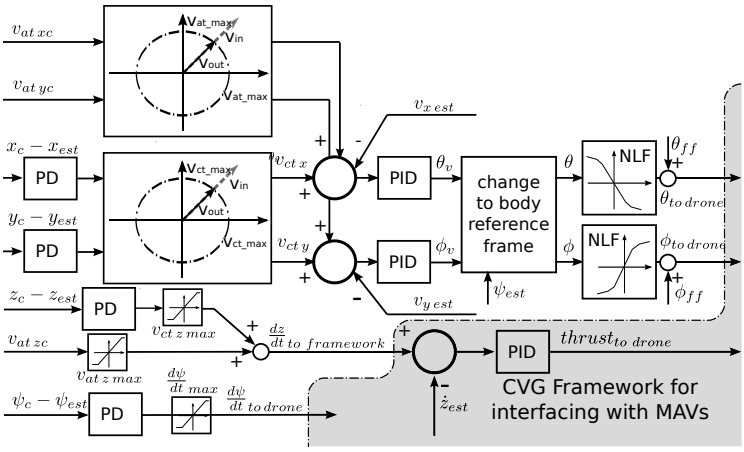


Fig. 5. The middle-level controller architecture is a cascade controller, which consists of an inner speed loop and an outer position loop. The reference change block, which implements Eq. 3 ensures that the roll and pitch commands are decoupled from the yaw heading. This allows for the yaw command to be set independently from the position and speed references. Thus, separating the left part of the diagram which is expressed in world coordinates, from the right part that is calculated in body frame coordinates.

4.3 High-Level Controller-State-Machine

The HL controller is implemented using a Finite State Machine (FSM), which uses the state estimation from the EKF to calculate the mid-level controller references. The FSM has three states corresponding to three control strategies: hover in a desired position, follow a straight segment of the trajectory and turn to head to the next trajectory segment. The work of the FSM is to navigate along a waypoint trajectory, and it can be summarized as the repetition of the following steps:

1. Follow the current straight segment, accelerating at first, and slowing down before reaching the next turn,
2. Perform the turn, controlling the speed direction to achieve a soft alignment with the next straight segment.

Every time the controller output is calculated the FSM calculates the planned speed at the current position $v_{plan}(s)$:

1. The speed, $\{V_{turn\,i}\}$, inner turn angle at checkpoint, $\{\alpha_i\}$, and radius, $\{R_{turn\,i}\}$, during each turn are precalculated each time the reference trajectory is changed, either because the whole trajectory is changed or because a waypoint was added. The radius is calculated so that the trajectory passes at a distance $R_{conf}/2$ from the checkpoint, where R_{conf} is the maximum distance at which the checkpoint is considered to be reached. The planned speed for each turn is selected either according to Eq. 8 either considering that the turn requires a stall-turn or turn-around maneuver, $v_{stall\,turn}$, which corresponds to a planned speed near zero.
2. The algorithm explained in Sec. 4.4 is used to calculate the planned speed, $v_{plan}(s)$, at the current position.

The FSM has also two additional operation modes where the quadrotor is controlled either only in position control, or in speed control. These can be used by the task scheduler to perform certain tasks.

The middle-level (ML) controller, see Fig. 3, receives position, $\{x_c, y_c, z_c\}$, yaw heading, ψ_c, speed, $\{v_{at\,xc}, v_{at\,yc}, v_{at\,zc}\}$, and tilt feed-forward, $\{\theta_{ff}, \psi_{ff}\}$, commands from the FSM, as shown in Fig. 5. The presented controller, which is inspired on those presented in [18,13], allows to use single mid-level architecture for the three control modes: trajectory, position and speed modes. These references to the ML controller are calculated by the FSM in such a way that the relative position and speed commands are orthogonal:

- The position references $[x_c, y_c, z_c]$ are caltulated projecting the estimated position $[x_{est}, y_{est}, z_{est}]$ onto the reference trajectory.
- The along-track speed commands, $[v_{at\,xc}, v_{at\,yc}, v_{at\,zc}]$, are parallel to the trajectory $v_{plan}(s)\,u_{tangent}$.
- The along-track speed commands, $[v_{at\,xc}, v_{at\,yc}, v_{at\,zc}]$, are derivated and smoothed using a low-pass filter to obtain an acceleration command based on Eqs. 4 & 5, so that the feed-forward attitude references, $\{\theta_{ff}, \phi_{ff}\}$, are calculated as follows:

$$\{\theta_{ff}, \phi_{ff}\} = \left\{-asin\left(\frac{\ddot{x}_m}{g}\right), asin\left(\frac{\ddot{y}_m}{g}\right)\right\}$$

4.4 Speed Planner Algorithm

The planned speed depends on the current speed and position of the multirotor, and on the previous and subsequent trajectory waypoints. The speed planner uses a uniform acceleration motion model to calculate both: the acceleration profile from the previous trajectory waypoints, and the deceleration profile to perform the following turns successfully. In addition to this model the speed loop response is modeled taking into account its response time.

The algorithm calculates the following distances for each of the neighbouring waypoints:

1. Δd_{chki}, the distance to waypoint i from the current estimated quadrotor position.
2. $\Delta d_{turni} = (R_{turni} + \frac{R_{conf}}{2}) sin(\alpha_i/2)$, the distance to waypoint i when the turn is to be started; where R_{conf} is the waypoint maximum clearance distance, $\{R_{turni}\}$ is the planned turn radius at waypoint i and $\{\alpha_i\}$ is the inner turn angle at waypoint i.
3. $\Delta d_{trvi} = t_{rvmax} \sqrt{v_{xest}^2 + v_{yest}^2}$, the distance required by the speed loop to slow down, where $\{v_{xest}, v_{yest}\}$ is the current estimated velocity and t_{rvmax} is the speed loop response time.

Then the optimal speed for the neighbouring checkpoints, v_{plan_i}, is calculated depending on the sign of $\Delta d = (\Delta d_{chki} - \Delta d_{turni} - \Delta d_{trvi})$. If $\Delta d \geq 0$ then $v_{plan_i} = \sqrt{V_{turni}^2 + 2a_{xymax}\Delta d}$, else if $\Delta d < 0$ then $v_{plan_i} = V_{turni}$. Where $\{V_{turni}\}$ is the planned speed at turn i. Finally the planned speed at the current position, $v_{plan}(s)$, is calculated as the minimum of the precaltulated optimal speeds $v_{plan}(s) = arg\,min_i\,(v_{plan_i})$. The coordinate s denotes the position of the quadrotor along the trajectory, and the expression $v_{plan}(s)$ highlights that the planned speed depends only on s and the following set of configurable parameters.

4.5 Configurable Parameters Set

From the prior explanation the set of configurable parameters is derived:

- $\{v_{atmax}, v_{ctmax}, v_{atzmax}, v_{ctzmax}\}$, are the saturation speed limits used inside the middle level controller that are imposed by either the kinematic capabilities of the vehicle, or the measurement range of the onboard sensors or the precision requirements of the task at hand.
- $\{R_{conf},\ v_{stallturn}, t_{rvmax}, a_{xymax}\}$, are the parameters that are used to determine the planned speed at each turn, $\{V_{turni}\}$, and to calculate the planned speed, $v_{plan}(s)$, at every controller step iteration.

5 Experimental Results

This section describes five separate experimental flights that show the navigation capabilities of the presented system architecture. All the flights are trajectory tracking

tests in autonomous navigation carried out in an indoors espacious room, simulating a GPS-denied situation. Thus, the GPS signal is not used in the estimation algorithms. All the flights were performed with the AR Drone 2.0 with the indoors hull, but similar flights with other drones can be found in [7]. The flight trajectories are shown in Fig. 6. The system has been tested following regular polygon trajectories on the XY plane, with varying altitude values of 1-3 m on each waypoint. A summary of the controller performances in these tests is shown in Table 1.

The AR Drone 2.0 was configured with the following parameters values. The maximum horizontal speed was set to 2 m/s, where $v_{at\,max} = 1.5$ m/s, $v_{ct\,max} = 0.5$ m/s. The FSM parameters were set to $R_{conf} = 0.25\ m$, $v_{stall\,turn} = 0.25\ m/s$, $t_{rv\,max} = 1.7$ s, and the maximum acceleration to $a_{xy\,max} = 0.5$ m/s^2. The configuration parameters were selected to work on trajectories made up of short segments, with the objective of optimizing speed while keeping low tracking error.

The first trajectory is a horizontal hexagon at 1 m of altitude, shown in the (a)&(b) subfigures of the Fig. 6. As shown the drone follows a trajectory made out of straight segments and circle segments around the corners. The (a) graph shows only the trajectory reference and the estimated position, and the (b) graph also shows the yaw references during a short period of time, as can be appreciated; the yaw control can be handled separately from the position control.

The rest of the trajectories, shown in the subfigures (c),(d),(e)&(f) have a varying number of waypoints and also a varying inside angle of the trajectory at their waypoints. Since they are all regular polygons on the XY plane, then the higher the number of waypoints the bigger the inner angles of the trajectory will be, allowing for faster passes through them. This is appreciated by the mean speed during trajectory tracking, see v_{mean} in Table 1. The other factor that affectes the maximum and mean speeds, v_{max} and v_{mean} respectively, is the distance between waypoints which is also shown in Table 1. Longer distances between waypoints allow the multirotor to accelerate to and deccelerate from higher speeds.

Table 1. Summary of results of the experimental tests: number of trajectory waypoints, and distance between waypoints in the XY plane, average, d_{mean}, and maximum, d_{max}, tracking error; average, v_{mean}, and maximum, v_{max}, speed during the test. Note that the trajectories are regular polygons when seen only on the XY plane. The tracking error is compared to the odometry based position estimate. The mean and maximum speed are calculated along the trajectory following part of the test only, were several laps were performed on each test. The table data correspond to the trajectories shown in Fig. 6

Subfig. in Fig. 6	a&b	c	d	e	f
number of trajectory waypoints	6	5	6	7	8
mean distance between waypoints	4.0	3.0	3.0	2.5	2.5
d_{mean} [m]	0.14	0.14	0.18	0.19	0.22
d_{max} [m]	0.56	0.26	0.34	0.42	0.39
v_{mean} [m/s]	0.81	0.59	0.73	0.74	0.89
v_{max} [m/s]	1.36	0.87	1.14	1.13	1.20

The error to the trajectory is measured against the odometry based position estimation. As expected this estimation is not accurate, and cummulates error over time. A ground truth to obtain an accurate tracking error is difficult to obtain without a motion capture system, which is the reason why it is not calculated. The drift depends on many circumstances that affect the quality of the optical flow speed estimation: texture of the floor, size of identifiable features in floor texture, illumination, altitude of the flight, yaw rotation speed, etc. To keep a good quality on the position measurements the yaw rotation speed commands are saturated at 50°/s, and are usually below 30°/s.

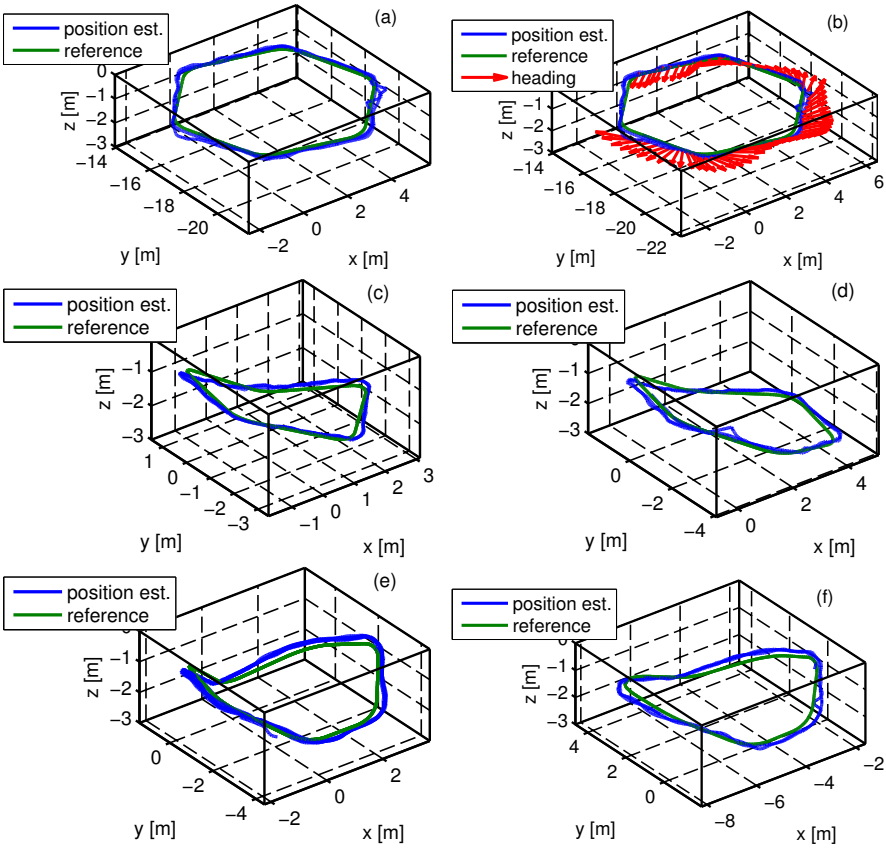


Fig. 6. Experimental tests performed using an AR Drone 2.0 for trajectory following. The graphs show the followed trajectory in the XYZ space. (a)&(b) correspond to the same flight, and the rest correspond to different flights. The trajectories are regular polygons on the XY plane, and (c),(d),(e)&(f) have waypoints with altitude references varying from 1 to 3 m. Several laps were finished on each test, and (b) shows how the yaw reference was varying during part of the laps. Some parameters that describe the controller architecture performance are shown in Table 1.

The flights were mostly anaffected during yaw heading rotation. In the flights exposed in this section the yaw rotation speed attained values of about 50 °/s. The experimental data clarify, however, that the vehicle tends to oscillate around the reference trajectory during the yaw rotation. High yaw rotation speeds do affect the position control, but 50 °/s is an acceptable value for many applications.

6 Conclusions

In this paper, we have extended the tests presented on a previous paper [7], with more 3D trajectories using an additional multirotor the AR Drone 2.0. The selected tests server to show how our architecture takes into account automatically the inner angle of the trajectory at each waypoint, and how the distance between waypoints is taken into account by the speed planner part of the architecture.

We presented our experimental work on the design of a multirotor navigation controller that allows for safer multirotor flights. The contribution of these works is two-fold: first, a discussion on various causes for multirotor navigation controllers malfunctioning is presented. Second, we derived a simple and robuts approach to address these causes natively in our middle-level controller architecture. As a further contribution to the research community, all the code related to this paper is available in two GitHub repositories.

This work is centered around the fact that the state estimation algorithms and the controller can interact in a very negative way when visual odometry estimations are faulty. This motivated our work to identify its causes and design an approach to overcome the problem. Our experimental work has led us to identify the safety boundaries under which our estimation algorithms work correctly. Then, our controller was designed to natively allow the saturation of the vehicle velocity. Although the speed can be saturated on the trajectory planning module, introducing this feature natively on the controller also allows to fly safely on speed and position control modes, or whenever the middle-level controller is utilized. The work has resulted in an overall increase of the safety and repeatibility of our experimental tests, which in turn allowed us to increase our test's efficiency and gain two awards on the IMAV 2012 competition. The experimental tests on our multirotor platforms have shown the robustness of our approach. The small trajectory tracking errors, measured against the EKF estimation, on these tests show that our efforts should be focused on improving our odometry and localization algorithms rather than the controller itself.

As an overall summary, the presented architecture was developed to allow a fast implementation on multiple multirotor vehicles. It permits to test the autonomous navigation software during real flight with a low cost vehicle, such as an AR Drone, and then easily make it portable to a more professional vehicle, such as an Asctec Pelican. The paper highlights the usage of a common drone model, and the issues that had to be solved to attain portability among multiple platforms. As future work, our group is interested on visual localization algorithms that can improve odometry and position measurements to the EKF; and the presented controller architecture will be tested in outdoors adding the GPS measurements for civilian applications research.

Acknowledgments. This work was supported by the Spanish Science and Technology Ministry under the grant CICYT DPI2010-20751-C02-01, and the following institutions by their scholarship grants: CSIC-JAE, CAM and the Chinese Scholarship Council. All the authors are with the Computer Vision Group, www.vision4uav.eu, which belongs to the Centre for Automation and Robotics, joint research center from the Spanish National Research Council (CSIC) and the Polytechnic University of Madrid (UPM).

References

1. The Navigation and Control Technology Inside the AR.Drone Micro UAV, Milano, Italy (2011)
2. Honegger, D., Meier, L., Tanskanen, P., Pollefeys, M.: An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications. In: International Conference on Robotics and Automation, ICRA 2013 (2013)
3. Zingg, S., Scaramuzza, D., Weiss, S., Siegwart, R.: MAV Navigation through Indoor Corridors Using Optical Flow. In: 2010 IEEE International Conference on Robotics and Automation, ICRA (2010)
4. Zufferey, J.-C., Beyeler, A., Floreano, D.: Autonomous flight at low altitude with vision-based collision avoidance and GPS-based path following. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). IEEE (2010), http://icra2010.grasp.upenn.edu/
5. Lippiello, V., Loianno, G., Siciliano, B.: MAV indoor navigation based on a closed-form solution for absolute scale velocity estimation using Optical Flow and inertial data. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 3566–3571 (2011)
6. Conroy, J., Gremillion, G., Ranganathan, B., Humbert, J.S.: Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. Autonomous Robots 27(3), 189–198 (2009)
7. Pestana, J., Mellado-Bataller, I., Fu, C., Sanchez-Lopez, J.L., Mondragon, I.F., Campoy, P.: A general purpose configurable navigation controller for micro aerial multirotor vehicles. In: 2013 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 557–564 (2013)
8. Pestana, J.: On-board control algorithms for Quadrotors and indoors navigation. Master's thesis, Universidad Politécnica de Madrid, Spain (2012)
9. Mellado Bataller, I.: A new framework for interfacing with MAVs (August 2012), https://github.com/uavster/mavwork
10. Pestana, J.: A general purpose multirotor controller compatible with multiple multirotor vehicles and with the mavwork open-source project (February 2013), https://github.com/jespestana/MultirotorController4mavwork
11. Motion capture systems from vicon, http://www.vicon.com/
12. Mellinger, D., Michael, N., Kumar, V.: Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) Experimental Robotics. STAR, vol. 79, pp. 361–373. Springer, Heidelberg (2012)
13. Michael, N., Mellinger, D., Lindsey, Q.: The GRASP Multiple Micro UAV Testbed. IEEE Robotics & Automation Magazine 17(3), 56–65 (2010)
14. Lupashin, S., Schollig, A., Sherback, M., D'Andrea, R.: A Simple Learning Strategy for High-Speed Quadrocopter Multi-Flips. In: 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), pp. 1642–1648 (2010)

15. Kushleyev, A., Kumar, V., Mellinger, D.: Towards a swarm of agile micro quadrotors. In: Proceedings of Robotics: Science and Systems, Sydney, Australia (July 2012)
16. Schölling, A., Augugliaro, F., Lupashin, S., D'Andrea, R.: Synchronizing the Motion of a Quadrocopter to Music. In: IEEE International Conference on Robotics and Automation ICRA, pp. 3355–3360 (2010), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509755
17. The Stanford/Berkeley Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) project, http://hybrid.eecs.berkeley.edu/starmac/
18. Hoffmann, G., Waslander, S.L., Tomlin, C.J.: Quadrotor Helicopter Trajectory Tracking Control. In: AIAA Guidance, Navigation and Control Conference and Exhibit 2008, pp. 1–14 (2008)
19. Hoffmann, G., Waslander, S., Tomlin, C.: Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3277–3282. IEEE (May 2009)
20. Gillula, J.H., Huang, H., Vitus, M.P., Tomlin, C.J.: Design of Guaranteed Safe Maneuvers Using Reachable Sets: Autonomous Quadrotor Aerobatics in Theory and Practice. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 1649–1654 (2010)
21. Rudol, P., Wzorek, M., Conte, G., Doherty, P.: Micro unmanned aerial vehicle visual servoing for cooperative indoor exploration. In: 2008 IEEE Conference on Aerospace (2008)
22. Pestana, J., Mellado-Bataller, I., Fu, C., Sanchez-Lopez, J.L., Mondragon, I.F., Campoy, P.: A Visual Guided Quadrotor for IMAV 2012 Indoor Autonomy Competition and Visual Control of a Quadrotor for the IMAV 2012 Indoor Dynamics Competition. In: 2012 Conference and Flight Competition International Micro Air Vehicle, IMAV (2012)
23. Fu, C., Pestana, J., I., Mellado-Bataller, Sanchez-Lopez, J.L., Campoy, P.: Visual identification and tracking for vertical and horizontal targets in gps-denied indoor environments. In: 2012 Conference and Flight Competition International Micro Air Vehicle, IMAV (2012)
24. International micro air vehicle conference and flight competition IMAV, program information for flight competition brochure (2012), http://www.dgon-imav.org/3.0.html#c214
25. Krajník, T., Vonásek, V., Fišer, D., Faigl, J.: AR-Drone as a Platform for Robotic Research and Education. In: Obdržálek, D., Gottscheber, A. (eds.) EUROBOT 2011. CCIS, vol. 161, pp. 172–186. Springer, Heidelberg (2011)
26. Mellado-Bataller, I., Mejias, L., Campoy, P., Olivares-Mendez, M.A.: Rapid prototyping framework for visual control of autonomous micro aerial vehicles. In: 12th International Conference on Intelligent Autonomous System (IAS 2012), Jeju Island, Korea (June 2012), http://eprints.qut.edu.au/50709/
27. Mellado-Bataller, I., Pestana, J., Olivares-Mendez, M.A., Campoy, P., Mejias, L.: MAVwork: A framework for unified interfacing between micro aerial vehicles and visual controllers. In: Lee, S., Yoon, K.-J., Lee, J. (eds.) Frontiers of Intelligent Auton. Syst. SCI, vol. 466, pp. 165–179. Springer, Heidelberg (2013)
28. Pestana Puerta, J.: On-board control algorithms for quadrotors and indoors navigation. Master's thesis, Universidad Politécnica de Madrid, Spain (October 2012)
29. UAS Technologies Sweden AB, LinkQuad quadrotor website, http://uastech.com/platforms.htm
30. MikroKopter, OktoKopter multirotor website, http://www.mikrokopter.de/ucwiki/en/MK-Okto

Air Drones for Explosive Landmines Detection*

C. Castiblanco, J. Rodriguez, I. Mondragon, C. Parra, and J. Colorado**

School of Engineering, Pontificia Universidad Javeriana
Cr 7 No 40-62, Bogota, Colombia
{jenny.castiblanco,j_rodriguezg,imondragon,carlos.parra,
coloradoj}@javeriana.edu.co},
http://www.javeriana.edu.co

Abstract. Explosive landmines have cost the lives of hundreds in several countries. This paper presents a field report of a low cost unmanned aerial vehicle -ARdrone 2.0- used as a complemented tool for landmine visual detection in rural scenarios. The main contribution of this work is the practical experimentation with an integrated tool. This tool is composed by the ARdrone quadrotor and a base station for monitoring the mission. Based on visual feedback from the onboard camera, algorithms for landmine detection are applied. Experiments have been carried out to quantify the performance of the platform by means of measuring the percentage of the detection at different set of flight speeds and altitudes. The goal is not only to detect fully visible landmines but also those partially buried. It has been observed an effective percentage of the detection over 80% when flying at low altitudes about $1m$ from the ground at speeds up to $2.2ms^{-1}$. The visual methods introduced herein might enable the ARdrone quadrotor to be used as a low-cost autonomous platform for safe area coverage applied to landmine detection in real scenarios.

Keywords: Aerial Sensing, Visual Recognition, Landmine Detection.

1 Introduction

Explosive landmines represent one of the most risky issues for people that live in conflict areas. The military has been the first to deploy machines as an attempt to overcome the risks involved when the landmine detection process is carried out by humans. Currently, there are fully autonomous systems which do not require a human operator for monitoring both detection and deactivation of explosive landmines, however these systems are highly expensive and also require qualified personnel [1,3,7,12,13]. Our goal in this paper is to integrate and evaluate a set of low cost technologies that allow the detection of explosive landmines autonomously and without compromising the mission.

* This work was supported by the SIRP group at the School of Engineering of Pontificia Universidad Javeriana in Bogota, Colombia. First and second authors have equally contributed in this research. First author is a researcher at SIRP with interest in control, artificial intelligence and vision applied to robotics. Second author is a researcher at SIRP with interest in aerial robotics, control and aeronautics.

** Corresponding author.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_9,

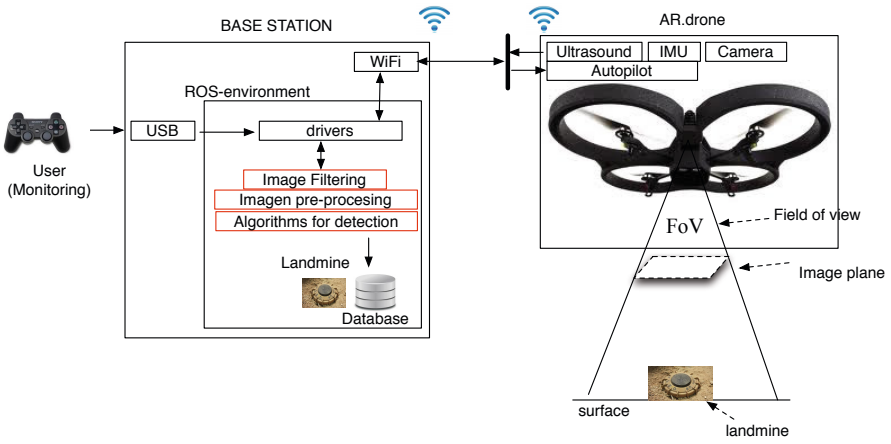


Fig. 1. Proposed architecture for landmine detection

In Colombia, between 1990 and 2013 were registered 10.413 victims caused by explosive landmines [10]. Recently, the local government has been developing humanitarian projects for demining tasks [4,6,9]. In this framework, the Ursula project [4] has been using a mobile wheeled robot to patrol and detect landmines mostly in flat fields. Nonetheless, the use of ground vehicles introduces several risky situations in cases where the system fails during the detection process. Likewise, the morphology of the robot constrains the area coverage task to uncluttered and flat ground scenarios. To overcome these drawbacks, the use of Unmanned Aerial Vehicles (UAV) enable several capabilities that enhance both navigation and security aspects of a landmine detection mission. Firstly, an UAV is able to autonomously cover a larger area of any type of terrain in less time and without compromising the mission. Secondly, it can be also used as a remote sensing platform for collecting key information about the terrain [14], [11,2].

Terrain information can be acquired using visual methods based on target localisation, tracking and image mosaicing techniques [8] which might provide a better understanding about the area to be covered. In this paper, we introduce a landmine detection system based on a low cost UAV equipped with cameras onboard that enable visual feedback of the terrain in real time. By capturing several sequences of images, visual algorithms for landmine detection are applied. The UAV is remotely controlled from a base station that uses the Robot Operating System (ROS) environment [5] for handling wireless communication, remote operation, flight control, image acquisition, filtering, pre-processing and the final landmine detection.

2 The Integrated Tool

The system is composed by a commercial UAV; The Parrot AR Drone 2.0 and a base station that handles both flight control and the landmine detection algorithms. The flight

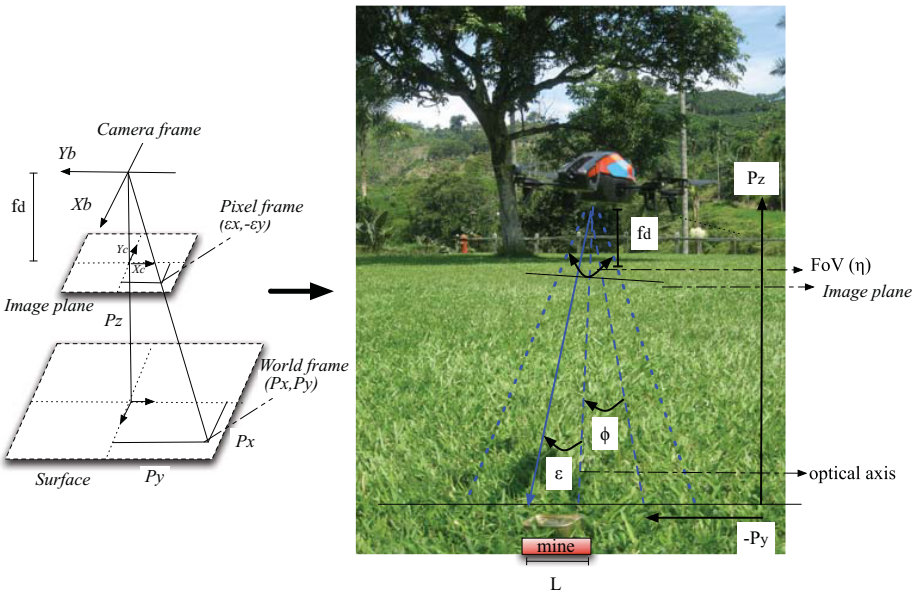


Fig. 2. left) camera model for the drone, (right) the geometry introduced by the vision system

control is integrated with ROS [5] whereas the landmine detection process is carried out by a set of algorithms that require an image captured by the bottom camera onboard the drone. Figure 1 shows the architecture for landmine detection. The visual algorithm is composed by three steps: i) the capture of a sequence of images, ii) image filtering, and iii) the landmine detection. These steps are also incorporated within ROS and require the OpenCV library to run.

3 Visual Sensing Algorithms

The detection process is performed using the still images captured by the bottom camera of the drone. To this purpose, Figure 2 shows the methodology used for the image capturing. The objective is to detect a target (landmine) using the vision sensor. It has to be ensured that the camera is mounted so that the optical axis of the camera is aligned with the body frame $z_b - axis$ and so that the $y_b - axis$ of the camera points out the right of the drone and the $x_b - axis$ of the camera points out the front of the drone. The camera model is depicted in Figure 2-left. The position of the target is described by the world frame (p_x, p_y, p_z) and the pixel location of the target in the image is with respect pixel frame $(\varepsilon_x, \varepsilon_y)$. The geometry for calculating p_x and p_y is shown in figure 2-right and correspond to the following equations:

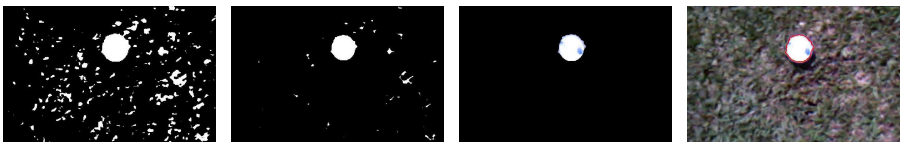
$$\begin{aligned} p_x &= -p_z \tan\left(\phi - \varepsilon_y \frac{\eta}{M_y}\right) \\ p_y &= p_z \tan\left(\phi - \varepsilon_x \frac{\eta}{M_x}\right) \end{aligned} \tag{1}$$

where η is the camera field-of-view, both M_x and M_y are the number of pixels along the camera $x_c - axis$ and $y_c - axis$ respectively and ϕ is the roll angle of the drone. By using the transformations from Eq. 1 is possible to have accurate images for detecting the target even in terrains with slope. In Colombia, most of the explosive landmines are fabricated and assembled using different kinds of cans, specially tuna's and sardine's cans. These artefacts require direct contact to be detonated, therefore they are located on the surface of the field (sometimes partially buried). Because of this, it is clear the use of visual sensing techniques as a complement approach for ensuring safety in the detection process.

The detection algorithm was based on two morphological processes applied to the stills: erode and dilate. Also, the algorithm removes areas of the images that do not contain relevant information for the landmine detection (area opening process). Equations 2 and 3 describe the mathematical process involved in the eroding and dilating process respectively. For both erosion and dilate processes of the image A by B, the set of pixel locations Z overlaps only with foreground pixels in A once the structuring element has been translated to location Z. In the case of the dilate process, $\hat{B}$ is the reflection of the structuring element B.

$$A \ominus B = \{z | (B_z \subseteq A)\} \tag{2}$$

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \varnothing\} \tag{3}$$



(a) (b) (c) (d)

Fig. 3. Experimental images captured by the drone during flight aimed for landmine detection using the proposed algorithm 1

To perform the image processing is necessary to convert the original image to the binary space aimed at improving the landmine detection. The eroding is basically used for eliminating the size of objects in the image and reducing noise. It requires a proper threshold for avoiding some parts of the target to be eliminated from the image. The setup of the threshold strictly depends on the size and shape of the morphological

Algorithm 1. Detection Process.

```
while flight process = 1 do
  1. imagebn ← binarization(imageInput);
  2. image1 ← erode(imagebn, morphological element(diamond), size element());
  3. image2 ← clear border(image1);
  4. image3 ← area opening(image2, tunedNumPixels());
  5. image4 ← dilate(image3);
  6. get whitePixelSize ← white elements(image4);
  if whitePixelSize < min2() && whitePixelSize > min1()+1 then
    7. image4 ← image2;
    8. image5 ← area opening(image4, tunedNumPixels());
    9. image5 ← dilate(image5);
  end if
  if whitePixelSize < min1 then
    10. image4 ← image2;
    11. image5 ← area opening(imageInput, tunedNumPixels());
    12. image5 ← dilate(imageInput);
  end if
  if detection landmine =1 then
    13. mineImage ← contour(image5);
    14. data base[counter] ← mineImage;
  else
    counter++;
    Go to step 1.
  end if
end while
```

element, which in this case has been based on a little diamond structure that removes the objects that are smaller than the landmine. Although the eroding process removes part of the noise in the image, this process does not eliminate larger objects within the image. Two approaches are used during the eroding process: cleaning the image borders and performing area opening. The former allows for removing objects that are in the contour of the image whereas the latter eliminates the objects with smaller pixel size than the threshold given by the user. This threshold must be smaller than the number of pixels of the landmine. Finally, the dilating is used to recover the parts of the landmine that were eliminated in the previous process. Additionally, a contour algorithm is used to mark the location of the landmine in the original picture. Algorithm 1 describes the landmine detection algorithm.

On the other hand, the landmine detection process is shown in figure 3a. It corresponds to the image captured by the drone after the binarization process. Figure 3b shows the resulting image after applying the eroding and clear border process and figure 3c depicts the obtained image after the area opening and dilating process. Finally, figure 3d shows the marked landmine in the original image once this one has been entirely detected. The experiment shown in figure 3 was performed with the drone flying at $2m$ of altitude over the field and a flight speed of $0.83ms^{-1}$.

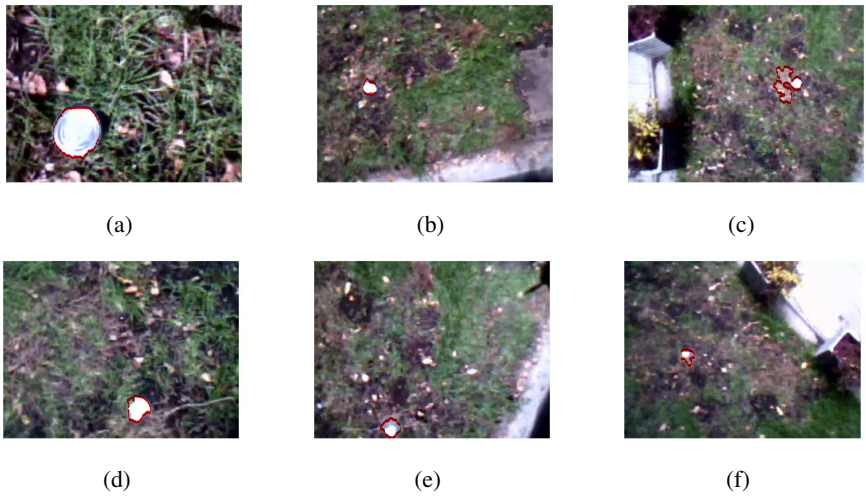


Fig. 4. Experimental detection response of Algorithm 1 for different altitudes: (top) fully visible landmine-like objects, (below) partially buried landmine-like objects: a,d) 1m above the surface, b,e) 2m above the surface, c,f) 3m above the surface

4 Experimental Testing and Results

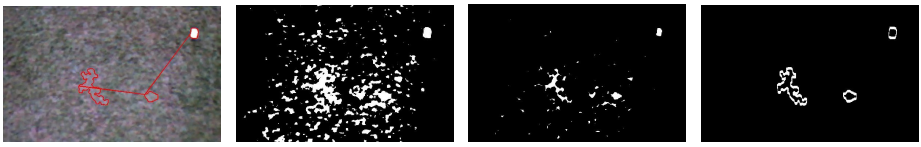
Outdoor tests were carried out using tuna cans as landmine-like objects and under different sunlight profiles and wind speeds. Some objects were randomly placed on the surface (fully visible) while others were placed partially buried. The test shown in figure 4 is twofold: firstly, the detection algorithm 1 is applied for different altitudes when the objects are fully visible (cf. figures 4a,b,c) and secondly when the objects are partially buried. Table 1 shows the average of the numerical values of the experiments carried out in figure 4. Note how the accuracy of the detection at $3m$ of altitude is compromised due to the low field of view range of the onboard 60 fps vertical QVGA bottom camera of the drone (resolution of 320x240). In general, the percentage of detection drops below 45% mainly because the pixel size of the noise is similar than the pixel size of the target. This behaviour is clearly shown in the experiments from figure 5. Finally, figure 6 shows how the accuracy of the detection (see algorithm 1) when the drone changes its position with respect to the object, which subsequently modifies the angle of the optical axis. In all the tests, the percentage of detection was higher than 80% when flying at $1m$ of altitude at a speed of $1ms^{-1}$.

5 Remarks

It has been experimentally demonstrated that a low cost UAV such as the AR drone can be used as an accurate complement tool for landmine detection tasks in roughed scenarios. Having an aerial platform equipped with visual sensors decreases the risks involved during the area coverage and enables to acquire key information of the terrain

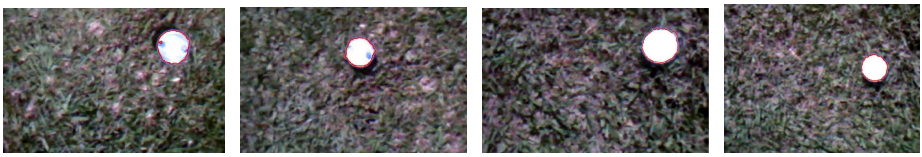
Table 1. Numerical values of the experiment from figure 4

Experiment type	Success (%) / Altitude (m)	Speed (Km/h)			
		1.5	3	5	8
Fully visible landmine-like objects	1	81.2	86.6	98.9	95.3
	2	97.2	80.5	63.8	83.3
	3	44.4	33.3	44.4	77.7
Partially buried landmine-like objects	1	60	83.3	80	73.3
	2	66.6	70	83.3	70
	3	6.6	13.3	23.3	33.3



(a) (b) (c) (d)

Fig. 5. Noisy images captured at $3m$ of flying altitude



(a) (b) (c) (d)

Fig. 6. Accurate landmine-like object detection for different roll (ϕ) values (cf. figure1)

thanks to the simple but effective visual algorithms applied. The landmine detection throughout images is a feasible approach to the Colombian scenario since most of the landmines do not have sophisticated mechanisms, which makes these artefacts somehow visible using a visual sensor. We have tested our system by measuring the average percentage of landmines detection in outdoor scenarios under three parameters: flight altitude, flight speed and visibility of the object. Based on the experimental results, we have determined that an altitude of $1m$ enables higher precision during the detection. Also, flying at higher speeds up to $8km/h$ ($2.2m/s$) enables better detection results due to the stability of the drone against external disturbances such as wind. Note in Table 1 that at $3m$ of altitude (partially buried object) the drone featured a 33.3% of detection flying at $8km/h$. This result is five times better compared to the profile described at $1.5km/h$. Current and upcoming work is oriented towards the geo-localization of the detected landmines into a map by means of image mosaicing techniques.

References

1. Abbas, S.M., Muhammad, A.: Outdoor rgb-d slam performance in slow mine detection. In: 7th German Conference on Robotics; Proceedings of ROBOTIK 2012, pp. 1–6 (2012)
2. Bharathi, T., Yuvaraj, S., Steffi, D., Perumal, S.: Vehicle detection in aerial surveillance using morphological shared-pixels neural (mspn) networks. In: 2012 Fourth International Conference on Advanced Computing (ICoAC), pp. 1–8 (2012)
3. Celebi, A., Gullu, M., Erturk, S.: Mine detection in side scan sonar images using markov random fields with brightness compensation. In: 2011 IEEE 19th Conference on Signal Processing and Communications Applications (SIU), pp. 916–919 (2011)
4. Coronado, J., Rizo, J., Campo, C.: Sistema móvil para la detección y localización de minas antipersonales. Pontificia Universidad Javeriana. Artículo Proyecto de grado Ingeniería Electrónica (2002)
5. Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadrocopter. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2815–2821 (2012)
6. Garcia-Caceres, R.G., Araoz-Durand, J.A., Palacios-Gomez, F.: Planning of a supply chain for anti-personal landmine disposal by means of robots. Innovar, Scieloco 22, 51–68 (2012)
7. Kang, S.P., Choi, J., Suh, S.B., Kang, S.: Design of mine detection robot for korean mine field. In: 2010 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pp. 53–56 (2010)
8. Martínez, C., Mondragón, I., Campoy, P., Sánchez-López, J., Olivares-Méndez, M.: A hierarchical tracking strategy for vision-based applications on-board uavs. Journal of Intelligent and Robotic Systems, 1–23 (2013)
9. Melo, K., Paez, L., Hernandez, M., Velasco, A., Calderon, F., Parra, C.: Preliminary studies on modular snake robots applied on de-mining tasks. In: Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC), pp. 1–6 (2011)
10. Programa de Acción Integral conta Minas Antipersonal, I.: Situación nacional víctimas de minas antipersonales en colombia, programa presidencial para la acción integral conta minas antipersonal (2013), `http://www.accioncontraminas.gov.co/Situacion/Paginas/SituacionVictimasMinasAntipersonal.aspx`
11. Pahsa, A., Kaya, P., Alat, G., Baykal, B.: Integrating navigation amp; surveillance of unmanned air vehicles into the civilian national airspaces by using ads-b applications. In: Integrated Communications, Navigation and Surveillance Conference (ICNS). pp. J7-1–J7-7 (2011)
12. Shimoi, N., Takita, Y.: Remote mine sensing technology using a mobile wheeled robot rat-1. In: 2010 International Conference on Control Automation and Systems (ICCAS), pp. 622–626 (2010)
13. Thành, N., Hào, D., Sahli, H.: Infrared thermography for land mine detection. In: Hammoud, R. (ed.) Augmented Vision Perception in Infrared, pp. 3–36. Springer, London (2009)
14. Valente, J., Barrientos, A., del Cerro, J., Rossi, C., Colorado, J., Sanz, D., Garzón, M.: Multirobot visual coverage path planning: Geometrical metamorphosis of the workspace through raster graphics based approaches. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2011, Part III. LNCS, vol. 6784, pp. 58–73. Springer, Heidelberg (2011)

Part XII

Flexible Robots

Passivity-Based Control Improvement of Single-Link Flexible Manipulators by a Two-Degree-of-Freedom PID Motor Controller

Andrés San-Millán Rodríguez[1,⋆] and Emiliano Pereira González[2]

[1] Escuela Técnica Superior de Ingenieros Industriales, Universidad de Castilla-La Mancha, Av. Camilo José Cela, s/n 13071, Ciudad Real, España
[2] Escuela Politécnica Superior, Universidad de Alcalá, Ctra.Madrid-Barcelona, Km. 33,600, 28805 Alcalá de Henares, Madrid, España

Abstract. A new approach to a previous passivity-based control scheme of single-link flexible manipulators is presented herein. Such previous scheme achieves precise positioning of the link tip by combining a position angular control of the motor (inner loop) with a link vibration damping (outer loop), which can be designed independently by decoupling joint and link dynamics with a linear strain feedback. Although, precise positioning can be achieved under large tip payload changes, the used inner loop cannot eliminate the steady-state position error due to the nonlinearities present in the motor. The contribution presented in this work consists of using a two-degree of freedom PID motor controller to solve this problem, eliminating thus the steady-state error while also improving the settling time of the angular position. Simulation and experimental results are carried out to illustrate these improvements.

1 Introducción

"Flexible robots" or "flexible manipulators" exhibit many advantages over their rigid counterparts such as they are lighter robots which can be driven using smaller amounts of energy, being more suitable for aerospace industry. In addition, these robots are safer to operate due to their reduced inertia and their inherent property to transform kinetic energy into potential energy (i.e., link strain). However, the vibrations make accurate positioning or trajectory tracking a challenging task, motivating a huge research in this topic [1].

In order to address control objectives, such as tip-position accuracy or suppression of residual vibration, many techniques derived from the control theory have been applied to flexible robots. Most of these control techniques can be classified in two groups: adaptive control [2,3,4], and robust control [5,6,7]. Both

⋆ This paper was sponsored by the Spanish FPU Program (Ministerio de Educación, Cultura y Deporte). This paper was sponsored by the Spanish Government Research Program with the Project DPI2012-37062-CO2-01 (Ministerio de Economia y Competitividad) and by the European Social Fund.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_10,

robust control and adaptive control usually require complex design methodologies and may show unstability issues due to high frequency unmodeled dynamics of the system due to spillover effects.

Control designs based on the partial differential equations can solve the aforementioned stability problems, while also simplifying the control design. Some examples are the direct strain feedback control (DSFB) [8], and the passivity-based control [9]. However, these techniques present some limitations owing to the coupling between the dynamics of the motor and the link of the flexible manipulator, which may lead to slow time response and position error.

The previous work [10] proposes a passivity-based control scheme consisting of two nested loops which are designed independently by decoupling joint and link dynamics with a linear strain feedback. This scheme damps the vibrations of the tip which appear after the movement of the robot, being also robust to large changes in the payload and to spillover effects. However, the loop used to control the motor angular position cannot remove the steady-state error due to the frictions present in the reduction gear and the motor.

In the present work, this loop is substituted by a two-degree of freedom (2DOF) PID motor controller. Thus, the steady-state error is eliminated while also improving the settling time of the angular position. In addition, a step by step design methodology is proposed to keep the design simplicity of [10]. A design example is simulated and implemented in practice to illustrate these improvements.

2 Dynamic Modelling

The scheme of the flexible robot to be controlled is shown in Fig. 1. The system is divided in two subsystems: the subsystem comprising the motor and the gear box, and the subsystem comprising the flexible link with a payload of mass M_p and inertia J_P. The equations of dynamic equilibrium in the joint are:

$$\Gamma_m(t) = n_r K_m V = J_0 \ddot{\theta}_m(t) + \nu \dot{\theta}_m(t) + \Gamma_f(t) + \Gamma_{coup}(t), \tag{1}$$

where Γ_m is the applied torque by the motor, K_m is a constant relating the control voltage of the motor V with the torque in the motor side, J_0 is the hub inertia, n_r is the reduction ratio, θ_m is the angular position of the motor, ν is the viscous friction coefficient, Γ_f is the torque owing to Coulomb friction and Γ_{coup} is the coupling torque between the link and the joint.

The dynamic behavior of an Euler-Bernoulli beam is governed by the following PDE (see, for example [11]),

$$EIw^{IV}(x,t) + \rho w(\ddot{x},t) + c\dot{w}(x,t) = f(x,t). \tag{2}$$

where L is the length of the link, ρ is the volumetric density, EI is the stiffness, w is the deflection, $f(x,t)$ is a distributed external force, c is a damping constant and overdots and primes indicate time and spatial derivatives, respectively.

Solving this equation (see [12]) the transfer functions of the system can be obtained. The assumptions made are that movement of the link is constrained to

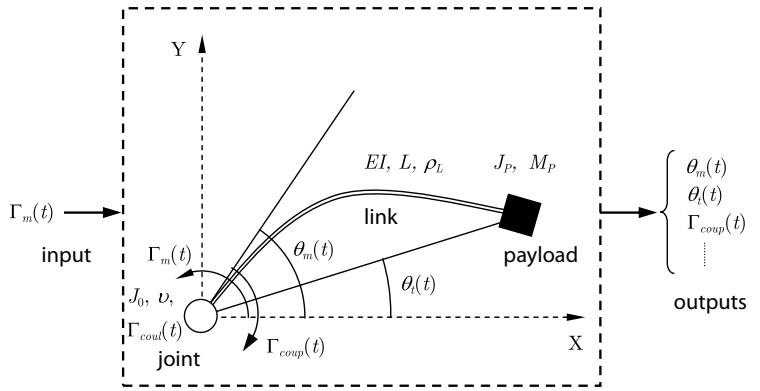


Fig. 1. Robotic system scheme

the horizontal plane (so the gravity effects are negligible), and that the deflections of the link are much smaller than the axial deformation.

The relationship between the coupling torque and the strain measured at the base of the link, which is used to decouple the link from the motor+reduction Gear, is as follows:

$$\Gamma_{coup}(t) = -EIw_0^{''}, \tag{3}$$

where $w_0^{''}$ is the strain at the base of the flexible link.

The model dynamics is derived by considering the link as an Euler-Bernoulli beam (see [12]). In addition, the experimental platform built in the laboratory of the E.T.S.I.I. in Ciudad Real, whose identified system parameters are shown in Table 1, is used to derive the dynamics model needed in the design of the controllers.

Table 1. Experimental platform parameters

Stiffness (Nm^2)	EI	2.4
Thickness (m)	h	0.002
Width (m)	b	0.05
Length (m)	L	1.26
Mass density (Kg/m^3)	ρ	2680
Inertia of the rotor and hub (Kg/m^2)	J_0	0.79
Motor constant (Nm/V)	K_m	0.474
Reduction gear ratio	n_r	50
Viscous friction coefficient (Kgm^2/s)	ν	3.65
Torque owing Coulomb friction (Nm)	Γ_f	17.06

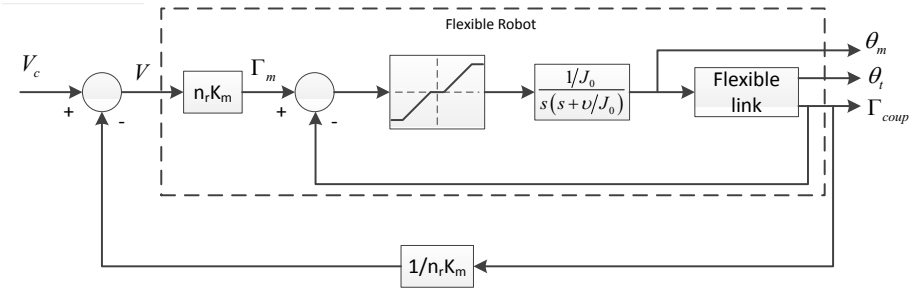


Fig. 2. Decoupling the flexible link and the motor+reduction gear dynamics compensating the coupling torque

3 Control Design Methodology

The objective of this section is to present the control design methodology based on the passivity property existing between the integral of the coupling torque Γ_{coup} (output) and the motor angle θ_m demonstrated in [10]. The control strategy consists of three sequential steps: First the link and motor dynamics are decoupled, then the controller of the angular position is designed, and finally an outer control loop is designed in order to damp the vibrations of the tip of the link.

The contribution presented in this work consists of substituting the PD motor controller of the inner loop, which was proposed in the design methodology presented in [10], by a 2DOF PID. The tuning of this 2DOF PID motor controller is based on a minimization process, which consists of minimizing the error between the output of the system and a step shaped reference control signal. Once the 2DOF PID is thus correctly tuned, the steady state error is removed and the settling time of the motor angular position is minimised, thus improving the results obtained in [10] without a significant increase of the complexity of design.

3.1 Decoupling the Link and the Motor Dynamics

In order to decouple the link and the motor dynamics is necessary to compensate the coupling torque as showed in Fig. 2. Compensating the coupling torque leads from the original model of the plant to the equivalent one showed in Fig. 3. The coupling torque is obtained in this work by using a strain-gauge bridge placed at the base of the beam (see Eq. (3)). The strain signal is amplified by the dynamic strain amplifier (Kyowa DPM600) and filtered by a second-order Butterworth filter with its cutoff frequency set to 300 Hz.

3.2 Design of the Inner Loop Controller Robust to Nonlinearities

After decoupling the motor and link dynamics, the modified motor position control scheme is shown in Fig. 4. Note that the whole model of the motor,

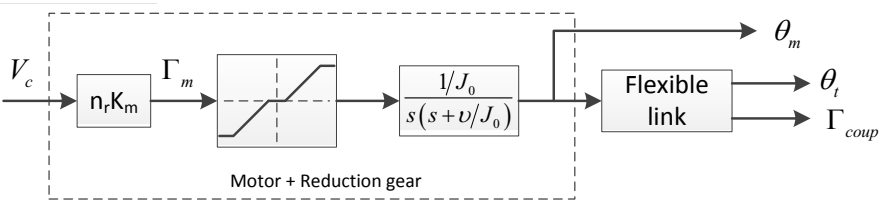


Fig. 3. Equivalent decoupled system

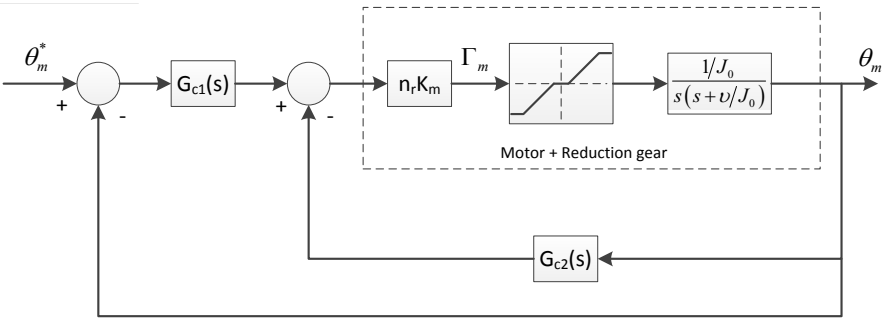


Fig. 4. 2DOF PID applied to the motor+reduction gear model

including the nonlinearities caused by the friction and the saturation of the motor, is considered in order to design the 2DOF PID controller [13].

The relative order of the inner loop dynamics must be less or equal than two in order to guarantee the stability and the passivity properties of the overall control system [10]. Considering this restriction, the 2DOF PID controller design can be divided into the following steps: i) choose the transfer functions $G_{c1}(s)$ and $G_{c2}(s)$ so that the closed loop dynamics (considering only the linear part the motor+reduction gear dynamics) has a relative degree less or equal than two, ii) establish restrictions in the parameters of $G_{c1}(s)$ y $G_{c2}(s)$ in order to simplify the resulting transfer function so that such transfer function has real and coincident zeroes and poles, leading to a second order critically damped system, and iii) adjust the zeroes and poles of the resulting closed loop system in order to minimize the following functional:

$$J(p_1, p_2) = \int_0^\infty (u(t) - y(t))^2 \, tdt, \tag{4}$$

where y(t) is the response of the controlled system (considering the nonlinearities of the motor and the reduction gear), and $u(t)$ is a step reference signal of amplitude equal to 0.5 radians.

If the linear part is only considered in Fig. 4, the transfer function between θ_m and θ_m^* is as follows:

$$G_{BC}(s) = \frac{G_{c1}(s)A}{s^2 + sB + A(G_{c1}(s) + G_{c2}(s))} \tag{5}$$

where A and B are equal to $n_r K_m/J_o$ and ν/J_0, respectively. If it is imposed $G_{BC}(s)$ to being a critically damped second order system, the following functions $G_{c1}(s)$ and $G_{c2}(s)$ can guarantee this restriction:

$$G_{c1}(s) = \frac{a_2 s^2 + a_1 s + a_0}{s^2 + gs}, \quad G_{c2}(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + gs}. \tag{6}$$

Thus, substituting the equalities (6) into (5) the following expression is obtained:

$$G_{BC}(s) = \frac{A(a_2 s^2 + a_1 s + a_0)}{s^4 + (g+B)s^3 + (Bg + A(b_2 + a_2))\, s^2 + A(b_1 + a_1)s + A(b_0 + a_0)}. \tag{7}$$

Therefore, if it is considered that the transfer function (7) has the following form:

$$G_{BC}(s) = \frac{p_2^2 (s + p_1)^2}{(s + p_1)^2 (s + p_2)^2}, \tag{8}$$

the equations for designing the parameters of (6) from the values of p_1 and p_2 are the following:

$$\begin{aligned}
a_2 &= \frac{p_2^2}{A}, \quad a_1 = \frac{2p_2^2 p_1}{A}, \quad a_0 = \frac{p_2^2 p_1^2}{A} \\
b_2 &= \frac{B(B - 2p_2 - 2p_1)}{A} + \frac{p_2^2 + p_1^2 + 4p_2 p_1}{A} - \frac{p_2^2}{A} \\
b_1 &= \frac{2p_2^2 p_1 + 2p_1^2 p_2}{A} - \frac{2p_2^2 p_1}{A} \\
b_0 &= 0, \quad g = -B + 2p_2 + 2p_1.
\end{aligned} \tag{9}$$

Note that Eq. (8) is an approximation because the Coulomb friction is not considered. This approximation can be used for a range of values for p_1 and p_2. Thus, the optimal values of p_1 and p_2 are obtained by simulation the control scheme of Fig. 4 when the overshot is less than 10%. Fig. 5 shows the evaluation of Eq. (4) after considering the above restrictions, where a minimum value of J can be seen.

3.3 Design of the Outer Loop

The whole control scheme is shown in Fig. 6, where the inner loop is approximated by the following transfer function:

$$G_{BC}(s) \simeq \frac{p_2^2}{(s + p_2)^2} \tag{10}$$

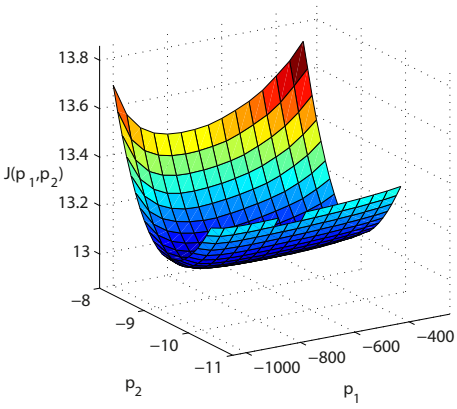


Fig. 5. Evolution of (4) with respect p_1 and p_2

Note that the nonlinearities of the motor and the coupling between dynamics have been compensated in the previous design steps. This approximation was validated via experimental results with different reference inputs. Thus, if the approximation (10) is considered adequate and if the outer control is defined as follows:

$$C(s) = (1/s)C'(s) = (1/s)K_c(s + \lambda), \tag{11}$$

the passivity property existing between the integral of the coupling torque (measures with the strain gauges) and the angular position of the motor (measured in the encoder of the motor) can be considered to deduce the following necessary and sufficient condition to guarantee the stability of the overall control system(see [10] for more details):

$$Re\left\{\frac{K_c p_2^2(j\omega + \lambda)}{(j\omega + p_2)^2}\right\} > 0, \forall\omega > 0, \tag{12}$$

which can be summarized in:

$$\lambda < 2p_2 \tag{13}$$

Bearing in mind this restriction on the value of λ , the remaining parameter of $C'(s)$ (K_c) is chosen so that the complex conjugate poles corresponding to the first mode of vibration of the link become into a double pole placed in the real axis (critically damped dynamic for the closed loop system).

Note that Eq. (13) is a simple condition of inequality (the same as for the one showed in [10]). However, if a PID is used for the inner control loop instead the 2DOF PID proposed herein, this stability condition is more complex because $G_{BC}(s)$ cannot be approximated by a critically damped second order system.

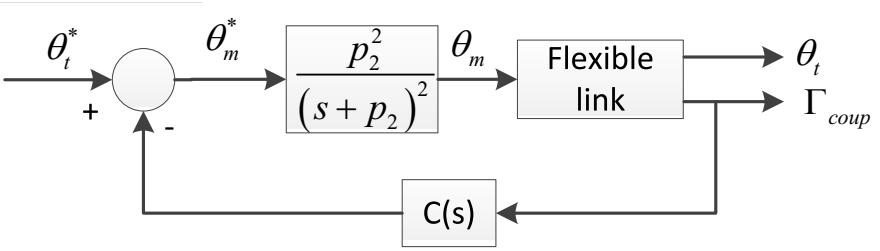


Fig. 6. General control scheme

4 Experimental Results and Simulations

The identification of the joint dynamics is firstly carried out. This dynamics is as follows:

$$\frac{\theta_m}{V_c} = \frac{39.26}{s(s+4.24)} \tag{14}$$

Based on this model, the functional (4) is minimized to obtain the parameters of Eqs. (6), which determine the optimal regulator for the angular position control. The parameters obtained after the minimisation are shown in the Table 2.

Table 2. Optimal values for the 2DOF PID control

Parameter	Value	Parameter	Value
a_2	2.06	b_2	12218
a_1	2801.7	b_1	$2.1 \cdot 10^5$
a_0	$9.5 \cdot 10^5$	b_0	0
g	1371.8		

Next the transfer function of the flexible link is identified from the experimental platform defined in Table 1:

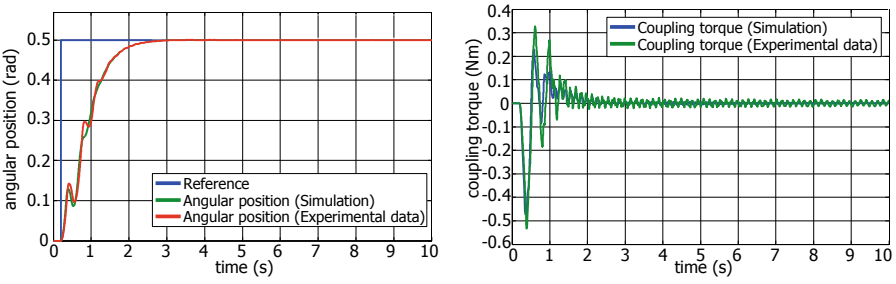
$$\frac{\Gamma_{coup}}{\theta_m} = s^2 \left(\frac{5}{s^2+0.075s+43.419} + \frac{3.5}{s^2+0.131s+1720.2} + \frac{3}{s^2+0.417s+13439} + \frac{3}{s^2+0.821s+52020} \right) \tag{15}$$

Once the system dynamics are known and the inner loop is designed, the values of λ and K_c can be obtained. Thus, the outer controller, which places the

poles of the first vibration mode of (15) into a double pole placed in the real axis and guarantees the condition defined into Eq. (13), is tuned as follows:

$$C(s) = \frac{0.84(s + 4.76)}{s} \tag{16}$$

The results obtained applying a step input of amplitude equal to 0.5 radians to the controlled system are shown in Figs. 7(a) and 7(b). It can be seen that the steady state error of the system is null and the residual vibration is approximately zero after 2s.



(a) Evolution of the angular position (b) Evolution of the coupling torque

Fig. 7. Step input response (Simulation and experimental results)

Finally, a small oscillation can be appreciated in Fig. 7(b). This oscillation, which corresponds to a high vibration mode, is not significant to obtain a precise tip positioning. It should be remarked that the passivity property (13) does not consider the dynamics of the strain-gauge bridge, the signal amplifier and the low pass filter. These dynamics, which could make the system unstable [14], will be considered in future works.

5 Conclusions

This work has proposed a modification of a previous passivity based control scheme for single link flexible manipulators. This modification is based on substituting the PD motor controller proposed in [10] by a 2DOF PID motor controller. In addition, this modification has been taken into account in a step by step design methodology, which simplifies the controller tuning parameters. The resulting control scheme has been simulated and implemented in a laboratory structure, showing its effectiveness to remove the steady state error position and the residual vibration.

References

1. Feliu, V.: Robots flexibles: Hacia una generación de robots con nuevas prestaciones. Revista Iberoamericana de Automática e Informática Industrial
2. Yang, T.C., Yang, J.C.S., Kudva, P.: Load-adaptive control of a single-link flexible manipulator. IEEE Transactions on Systems, Man and Cybernetics 22(1), 85–91 (1992)
3. Feliu, J.J., Feliu, V., Cerrada, C.: Load adaptive control of single-link flexible arms based on a new modeling technique. IEEE Transactions on Robotics and Automation 15(5), 793–804 (1999)
4. Dharne, A.G., Jayasuriya, S.: Robust adaptive control of residual vibration in point-to-point motion of flexible bodies. Journal of Vibration and Control 13(7), 951–968 (2007)
5. Chen, Y.P., Hsu, H.T.: Regulation and vibration control of an fem-based single-link flexible arm using sliding-mode theory. Journal of Vibration and Control 7(5), 741–752 (2001)
6. Amiri, M., Menhaj, M.B., Yazdanpanh, M.J.: A neural-network-based controller for a single-link flexible manipulator: Comparison of ffnn and drnn controllers. In: IEEE International Joint Conference on Neural Networks, IJCNN 2008 (IEEE World Congress on Computational Intelligence), pp. 1686–1691 (June 2008)
7. Moudgal, V.G., Kwong, W.A., Passino, K.M., Yurkovich, S.: Fuzzy learning control for a flexible-link robot. IEEE Transactions on Fuzzy Systems 3(2), 199–210 (1995)
8. Luo, Z.H.: Direct strain feedback control of flexible robot arms: new theoretical and experimental results. IEEE Transactions on Automatic Control 38(11), 1610–1622 (1993)
9. Liu, L.-Y., Yuan, K.: Noncollocated passivity-based pd control of a single-link flexible manipulator. Robotica 21(2), 117–135 (2003)
10. Pereira, E., Diaz, I.M., Cela, J.J.L., Feliu, V.: A new design methodology for passivity-based control of single- link flexible manipulators. In: 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1–6 (September 2007)
11. Meirovitch, L.: Principles and Techniques of Vibrations. Prentice Hall International (1997)
12. Bellezza, F., Lanari, L., Ulivi, G.: Exact modeling of the flexible slewing link. In: Proceedings of the 1990 IEEE International Conference on Robotics and Automation, vol. 1, pp. 734–739 (1990)
13. Ogata, K.: Ingeniería de control moderna 4ED. Pearson educación. Pearson-Prentice-Hall (2003)
14. Forbes, J.R., Damaren, C.J.: Single-link flexible manipulator control accommodating passivity violations: Theory and experiments. IEEE Transactions on Technology 20(3), 652–262 (2012)

Vibration Suppression Controller for a Flexible Beam on a Cart Using SMC*

S. Hassan HosseinNia, Inés Tejado, Daniel Torres, and Blas M. Vinagre

Department of Electrical, Electronics and Automation Engineering,
University of Extremadura, Spain
{hoseinnia,itejbal,datglez,bvinagre}@unex.es

Abstract. In this paper, vibration suppression of a flexible beam on a moving cart is investigated based on sliding mode control (SMC). A linear model which describes the motion of the beam-cart system, as well as the vibrational motion of the beam, are firstly derived. In the proposed control strategy, a PD controller is used to control the position of the cart, and a corrector switching law is added to attenuate the beam vibration. Indeed, the linear controller plays the role of the equivalent control law to make the states of the system reach the surface. Simulation and experimental results are given to demonstrate de effectiveness of the proposed control strategy.

Keywords: Beam-Cart System, Vibration Control, Sliding Mode Control.

1 Introduction

Flexible beams are found in many practical engineering applications, such as robotic arms, space structures, etc. The challenge of controlling vibrations has been approached with very different methods (refer to [1] for a survey on the topic). However, in spite of all the research devoted to modelling and controlling these kind of robots, there is no a universal satisfactory solution for the control, which is clearly demonstrated by the number of recent articles presenting new improved solutions for vibration control. Among them, sliding mode control (SMC) have been successfully applied to suppress vibrations in flexible links [2,3,4,5].

On the other hand, a considerable number of studies have analyzed vibrations of beams fixed to moving carts (e.g. see [6,7,8,9,10]. Stabilizing a flexible beam on a cart using a distributed Port-Hamiltonian approach is studied in [6]. Similarly in [8], the authors proposed a modified pulse sequence method with a robust internal-loop compensator (RIC) to suppress single-mode residual vibration and obtain accurate positioning of the beam-mass-cart system. In [9], vibration suppression is studied to stabilize beam-cart system using piezoelectric transducers by decomposed parallel adaptive neuro-fuzzy control. A SMC design methodology is applied to solve the position tracking problem of an uncertain very lightweight single-link flexible robot arm in [10].

* This work has been supported by the Spanish Ministry of Economy and Competitiveness under the project DPI2012-37062-C02-02.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_11,

Throughout this work, the authors sought a control scheme robust with regard to payload and actuator friction changes, without the establishment of any bounding limits on these parameter variations.

Given this context, the objective of this paper is to present preliminary results of the suppression control of single-mode vibration of an elastic beam fixed on a moving cart by means of SMC, in which the position of the cart is controlled and, at the same time, the cart itself is considered as the actuator for beam vibration suppression. It should be worth mentioning that this effort represents a part of an ongoing investigation to develop control strategies for teleoperated robotic arms with flexible links.

The rest of the paper is organized as follows. Section 2 describes the beam-cart system and gives its dynamic model. In Section 3, the design of the controller to suppress the vibration of the beam is addressed, as well as the stability analysis of the controlled system. Section 4 presents simulation and experimental results of the controlled system. Finally, Section 5 draws the main conclusions of this paper.

2 The Beam-Cart System

This section describes the system under our interest, the flexible beam on a cart shown in Fig. 1, and gives its model.

2.1 Description

Figure 1 shows the connection scheme of the experimental set-up, which consists of:

- The beam-cart system, which is based on the Digital Pendulum (model 33-935) by *Feedback*. As observe, it consists of an aluminum beam of length L mounted on a cart –fixed to it at the bottom. The cart is actuated by a DC motor and can be moved along a rail of limited length. At the rail ends there are two limit switches which cut off the DC motor power when the cart overruns the limit points.
- A gyroscope *Gyrosense* CRS 03-03S, which allows to determine the vibration of the beam from the measured angular velocity.
- A digital box, which serves as interface for the system and the computer. Mainly, it contains the power amplifier and is equipped with two switches: the main power switch and the switch for cutting off the DC motor power.
- A computer, where the Real Time Workshop Toolbox of MATLAB runs for the validation of the proposed control.
- A PCI 1711S data acquisition board by *Advantech* mounted on the computer, which implements a high-performance data acquisition hardware for real-time applications. Its D/A output is wired to the power amplifier input of the digital box which drives the DC motor.

2.2 Dynamic Model

Consider the linear motion of the beam-cart system along the rail. The elastic beam is assumed to follow the Bernoulli-Euler beam model and to be clamped tightly on the

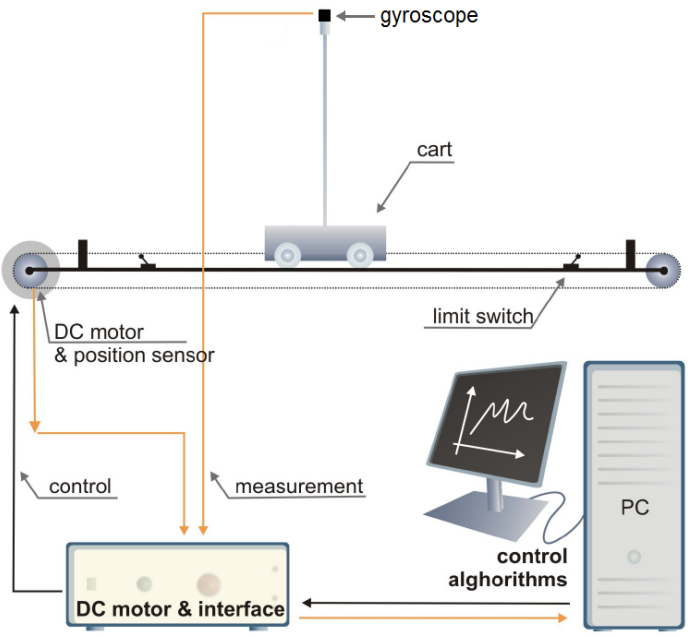


Fig. 1. Experimental set-up of the beam-cart system

moving cart. In deriving the equations of motion, it is assumed that the beam can only vibrate in vertical plane.

The dynamics of the cart is given by:

$$m\frac{d^2x_c}{dt} = F - F_C - \Gamma_c - \Gamma_{coup}, \tag{1}$$

where x_c is the position of the cart, m is the mass of the system, F_C is the friction force, F is the force applied to the cart, Γ_c is the (unknown) Coulumb friction torque and Γ_{coup} is the coupling torque between the flexible beam and the cart. Replacing the friction by $F_C = f_c\frac{dx_c}{dt}$, where f_c is the friction coefficient, (1) can be expressed as:

$$m\frac{d^2x_c}{dt} = F - f_c\frac{dx_c}{dt} - \Gamma_c - \Gamma_{coup}. \tag{2}$$

Considering

$$x_{c_1} = x_c$$
$$x_{c_2} = \dot{x}_c,$$

the state space representation of the cart is:

$$\begin{bmatrix} \dot{x}_{c_1} \\ \dot{x}_{c_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{fc}{m} \end{bmatrix} \begin{bmatrix} x_{c_1} \\ x_{c_2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F - \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} (\Gamma_c + \Gamma_{coup})$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_{c_1} \\ x_{c_2} \end{bmatrix} \tag{3}$$

Concerning flexible beams, it is well-known that they present infinite vibration modes depending on the specific dynamical conditions. Among them, we consider the single-mode vibration model for flexible joints, i.e.,

$$\omega_n{}^2(x_c - x_b) = \ddot{x}_b + 2\zeta\omega_n\dot{x}_b \tag{4}$$
$$\omega_n{}^2(x_c - x_b) = \Gamma_{coup}, \tag{5}$$

where x_b is the position of the end of the beam, ω_n is the natural frequency of the vibration and ζ, the damping coefficient.

Therefore, the dynamics of the whole system is given by the cart model, (3), the flexible beam model, (4), and the coupling torque produced by translation of the flexible beam, (5). Let us define the following vector:

$$X = \begin{bmatrix} x_{c_1} & x_{c_2} & x_{b_1} & x_{b_2} \end{bmatrix}^T, \tag{6}$$

where $x_{b_1} = x_b$ and $x_{b_2} = \dot{x}_b$. So, the model of the system is given by:

$$\begin{aligned} \dot{X} &= AX + B(u - (\Gamma_c + \Gamma_{coup})), \\ y &= \begin{bmatrix} y_b \\ y_c \end{bmatrix} = CX, \end{aligned} \tag{7}$$

where $u = F$, $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_n{}^2 & -2\zeta\omega_n & \omega_n{}^2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{f_c}{m} \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m} \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

To determine both ω_n and ζ parameters of the single-mode vibration of the beam, several experiments were carried out as follows: the beam was displaced an angle of the equilibrium position with the car stopped and released to vibrate –response with no input, just an initial condition. The angle of the beam was recorded during 100 seconds for six different initial angles (see an example in Fig. 2). It can be seen that the vibration of the beam can be modelled by a second order system. Figure 3 shows the spectra estimation of each experiment using FFT with 2^{17} points. As observe, ω_n is 21.52 rad/s. Next step was to identify the damping coefficient, ζ. To do so, it was considered that the covering curve of the response has the form

$$y(t) = x_0 e^{-\zeta\omega_n t}.$$

The coefficient ζ was determined considering different points for a given response and for different experiments, obtaining $\zeta = 0.003$. Hence, the vibration of the beam can be approximated by the following second order system:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{463.11}{s^2 + 0.13s + 463.11}.$$

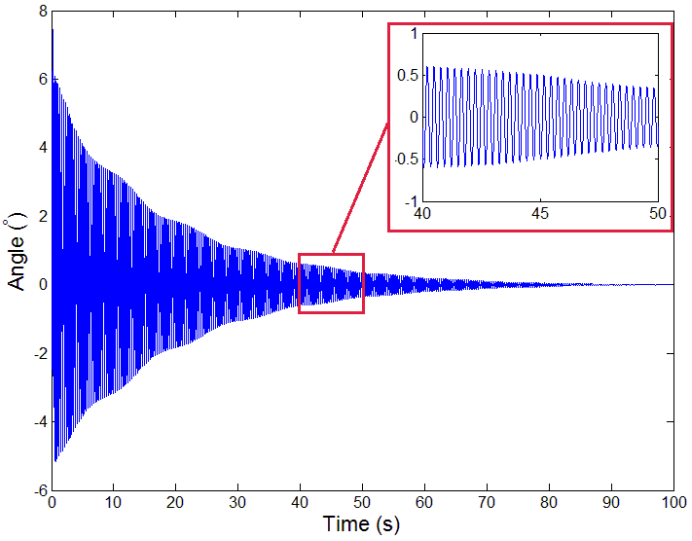


Fig. 2. Example of evolution of the angle of the beam (experiment 1)

Figure 4 compare the identified second order system with the experimental responses.

3 Vibration Suppression Control

This section addresses the design of the vibration suppression controller and the stability analysis of the controlled system.

3.1 Controller Description

The controller design is divided into two steps: firstly, design of a linear PD controller u_{PD} for the cart, neglecting the nonlinear components due to friction and coupling in (1), and, secondly, design a switching control law u_s to suppress the vibration and the effects of nonlinear terms. Figure 5 shows the control block diagram of the beam-cart system.

Therefore, the cart system to be considered is:

$$\begin{bmatrix} \dot{x}_{c_1} \\ \dot{x}_{c_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{fc}{m} \end{bmatrix} \begin{bmatrix} x_{c_1} \\ x_{c_2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u_{PD}.$$

In this case, gain crossover frequency of $\omega_{gc} = 3$ rad/s and phase margin of $\phi_m = 60°$ are specified to tune the PD controller,

$$u_{PD} = k_p(y_d - y_c) + k_d(\dot{y}_d - \dot{y}_c), \tag{8}$$

where y_d is the desired reference to the cart. Substituting the system parameters f_c and m by 0.36 N and 1.55 kg, respectively, the controller parameters which fulfill the specifications are $k_p = 8$ and $k_d = 4$, as shown in Fig. 6.

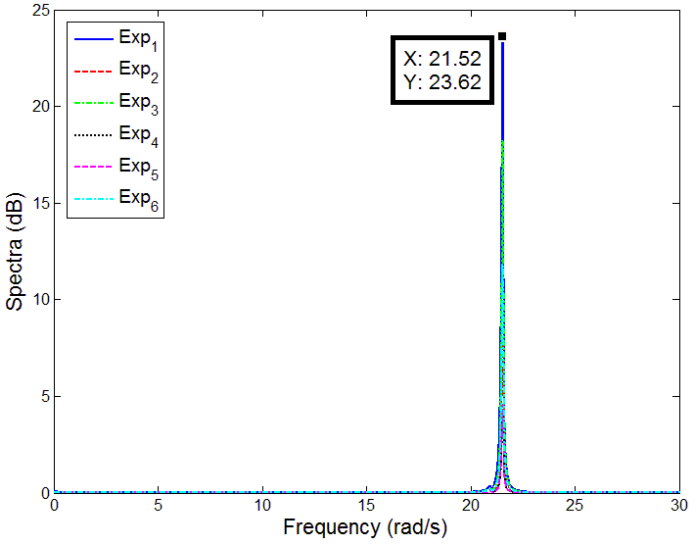


Fig. 3. Spectra estimation by FFT

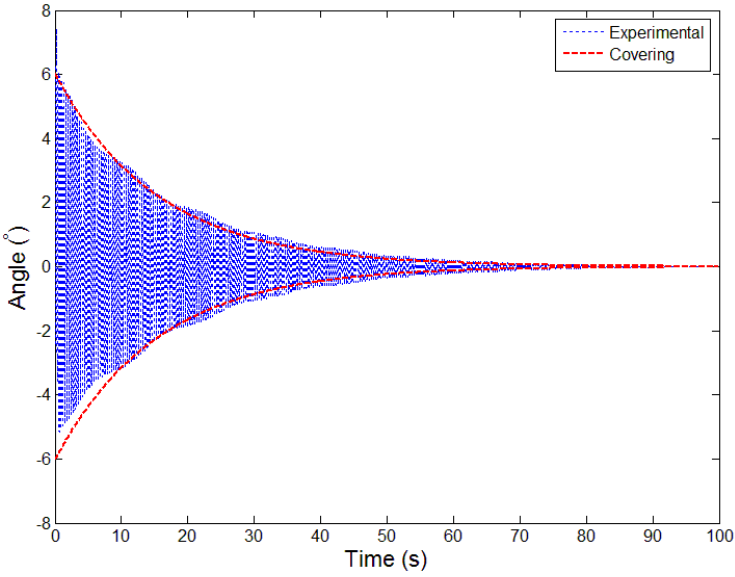


Fig. 4. Identification results for the vibration of the bar: comparison with the covering curve (experiment 1)

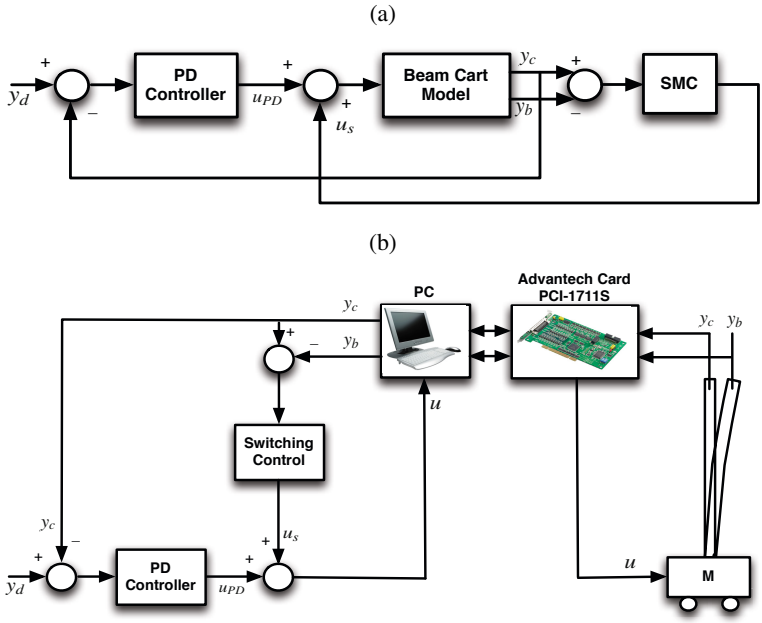


Fig. 5. Block diagram of the controlled beam-cart system: (a) Simplified control loop (b) Implemented control loop

Now, the whole beam-cart system (7) will be considered. To this end, a switching control input is added to the PD controller in order to suppress the vibration as follows:

$$u = u_{PD} + u_s. \tag{9}$$

And the switching controller u_s is

$$u_s = K_s sign(S), \tag{10}$$

where K_s is the switching parameter which will be tuned in order to eliminate the vibration and S is the sliding mode surface:

$$S = k_1(y_c - y_b) + k_2(\dot{y}_c - \dot{y}_b), \tag{11}$$

being k_1 and k_2 positive real values.

It should be remarked that, in order to tune k_1, k_2 and K_s, the stability of the controlled system is going to be analized in the following subsection.

3.2 Stability Analysis

Let us choose a positive Lyapunov function as

$$V = \frac{1}{2}S^2. \tag{12}$$

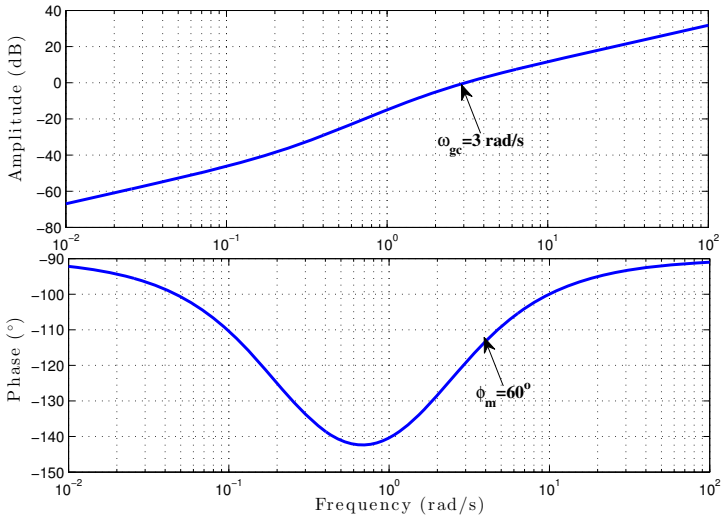


Fig. 6. Bode diagram of the beam-cart system controlled with the designed PD

¿From Lyapunov's stability theory we know that the system reaches $S = 0$ in finite time if the above Lyapunov function satisfies:

$$S\dot{S} \leq -\eta\,|S| \tag{13}$$

for some strictly positive constant $\eta \in \mathbb{R}$. By differentiating V, the above stability condition reduces to:

$$S\dot{S} = S[k_1(\dot{y}_c - \dot{y}_b) + k_2(\ddot{y}_c - \ddot{y}_b)] = S\left[k_1(\dot{x}_{c_2} - \dot{x}_{b_2}) + k_2(\dot{x}_{c_2} - \dot{x}_{b_2})\right] =$$
$$S\left[k_1(x_{c_2} - x_{b_2}) + k_2(-\tfrac{f_c}{m}x_{c_2} - {\omega_n}^2 x_{b_1} - 2\zeta\omega_n x_{b_2} + u_{PD} - \Gamma_c - \Gamma coup + K_s sign(S))\right] \leq$$
$$\left|k_1(x_{c_2} - x_{b_2}) + k_2\left(-\tfrac{f_c}{m}x_{c_2} - {\omega_n}^2 x_{b_1} - 2\zeta\omega_n x_{b_2} + u_{PD} - \Gamma_c - \Gamma coup\right)\right||S| + K_s\,|S| \leq -\eta\,|S|\,.$$

Assuming

$$|g(X)| = \left|k_1(x_{c_2} - x_{b_2}) + k_2\left(-\frac{f_c}{m}x_{c_2} - {\omega_n}^2 x_{b_1} - 2\zeta\omega_n x_{b_2} + u_{PD} - \Gamma_c - \Gamma coup\right)\right| \leq \gamma, \tag{14}$$

the system reaches the surface in finite time when $K_s \leq -(\gamma + \eta)$.

3.3 Controller Design

In order to tune k_1, k_2 and K_s to fulfil condition (14), the estimation of γ is required. The vibration of the beam, in comparison with the movement of the cart, is very small and $\frac{f_c}{m}x_{c_2} + 2\zeta\omega_n x_{b_2} << {\omega_n}^2 x_{b_1}$. As a result, $k_1(x_{c_2} - x_{b_2})$ –assuming a small value for k_1; it was set to 0.01– and $\frac{f_c}{m}x_{c_2} + 2\zeta\omega_n x_{b_2}$ are negligible. Therefore, (14) can be rewritten as:

$$\left|k_2\left(-{\omega_n}^2 x_{b_1} + u_{PD} - \Gamma_c - \Gamma_{coup}\right)\right| \leq \gamma. \tag{15}$$

On the other hand, the control effort is in the range $[-2.5, 2.5]$ and the rail is limited to 0.3 m, which means that $x_{c_1} < 0.3$, so $\left|u_{PD} - \Gamma_c - \Gamma_{coup}\right| \leq 2.5$. Then, due to practical limitations, $x_{b_1} \approx x_{c_1} \leq 0.3$ and ${\omega_n}^2 = 463.11$, and therefore, we have

$$141.43k_2 \leq \gamma. \tag{16}$$

As mentioned, the maximum value for the control signal is 2.5, thus $|K_s|$ should be also limited to this value. Indeed,

$$(141.32k_2 + \eta) \leq |K_s| \leq 2.5 \tag{17}$$

Therefore, choosing $k_2 = 0.01$ and $K_s = -1.5$, condition (17) is fulfilled with $\eta \leq 0.0857$.

Figure 7 depicts the numerical simulation of condition (14) when applying the designed controller. It can be seen than $|g(X)| < 1.047$, which verifies the accuracy of the estimation of γ.

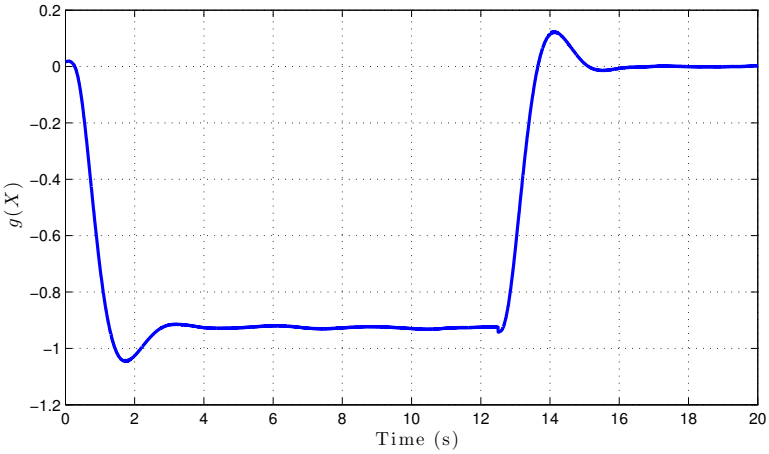


Fig. 7. Numerical simulation of (14) when applying the designed controller

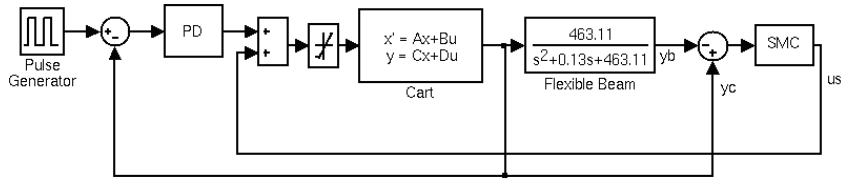
4 Simulation and Experimental Results

This section presents simulation and experimental results when controlling the beam-cart system in the MATLAB/Simulink environment. The vibration is sensed by a gyroscope installed on the top of the beam. The aim is to force the cart to follow a pulse reference with an amplitude of 0.2 m using the previously tuned PD controller. The sharp changes in pulse reference will cause vibration in the beam, which will be attenuated by a switching control law. Parameters of the system and the controllers are summarized in Table 1. The simulation block diagrams of the controlled system and the sliding mode controller are depicted in Fig. 8.

Table 1. Parameters of the system and the controllers

	Parameter	Value
PD controller	k_p	8
	k_d	4
SMC	k_1	0.01
	k_2	0.01
	K_s	1.5
System	f_c	0.36 kg/s
	m	1.55 kg
	ω_n	21.52 rad/s
	ζ	0.003

(a)



(b)

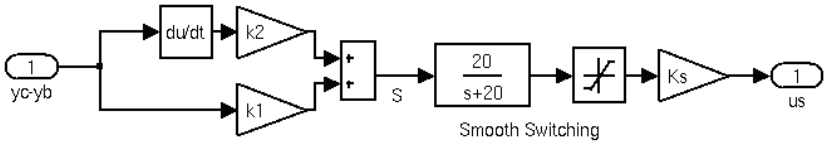


Fig. 8. Simulink block diagrams: (a) Controlled system (b) Details of the sliding mode controller for smooth switching

Simulation and experimental results corresponding to the controlled system are shown in Fig. 9. It is important to remark that only linear elements were considered in the simulation model –this fact may explain the differences observed between simulation and experimental results. Fig. 9(a) shows the results using *sign* function as switching function. As observe, a considerable chattering effect appears produced by the high switching action of the control signal, which becomes the main implementation problem of this controller due to actuator practical limitations. In the literature, numerous techniques have been proposed to mitigate this phenomenon in SMC (see e.g. [11,12,13,14] and references therein). One common and feasible solution to reduce the chattering is to replace *sign* by the following function:

$$\begin{cases} -(1+e^{-nt}) & t < -1 \\ t(1+e^{-nt}) & -1 \le t \le -1 \\ 1+e^{-nt} & t > 1 \end{cases}. \tag{18}$$

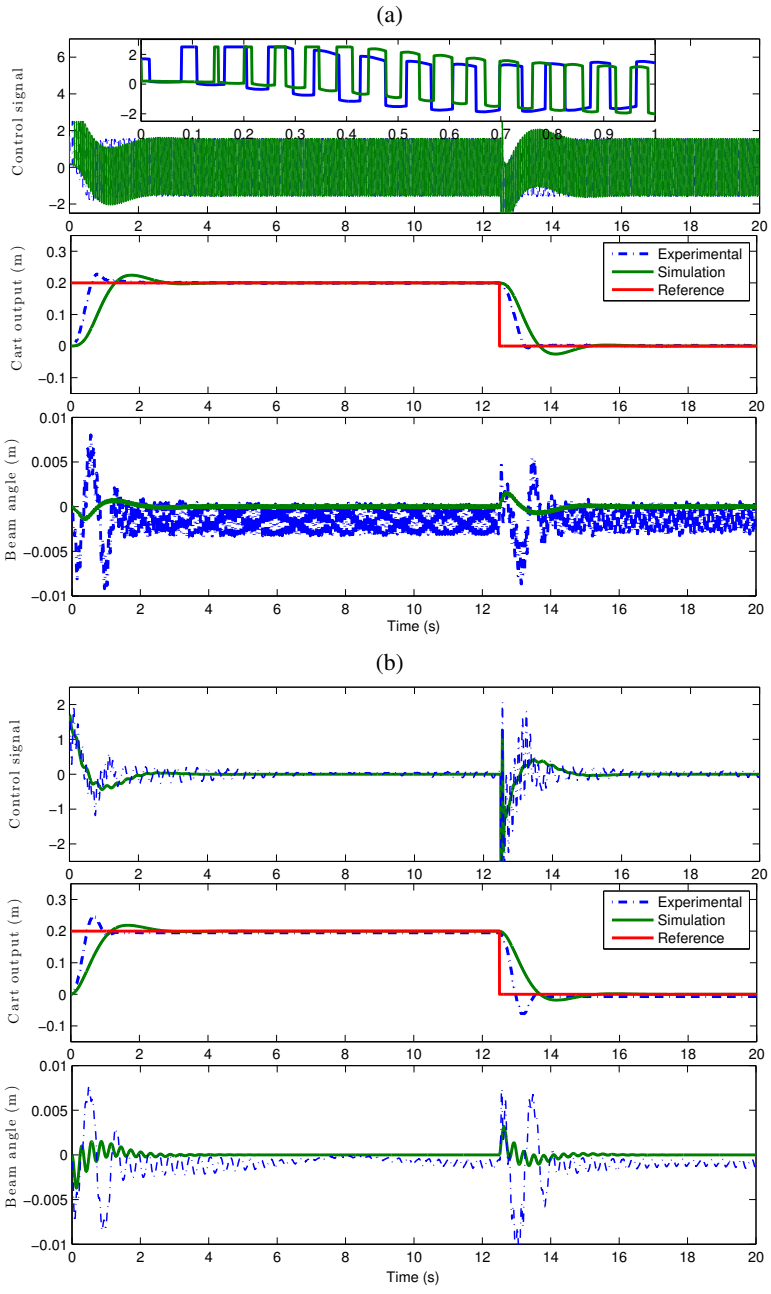


Fig. 9. Experimental and simulated results: (a) Using SMC (b) Using smooth SMC

where n can be chosen taking into account the application, more precisely, the sensitivity of the actuator. To this respect, the lower value of n, the smoother the signal but, on the other hand, the sliding mode would be lost, so that there should be a compromise between having smoother signal and sliding mode. In this case, n was set to 20. The results related to smooth switching are shown in Fig. 9(b). As observe, the chattering phenomena is considerably reduced, the control signal is smoother and the vibration is significantly attenuated. The vibration suppressed in less than 2 s for both experiments and the experiment results verifies the simulation results.

5 Conclusions

In this paper, a SMC controller is designed to suppress the vibration of a flexible beam on a moving cart. The stability is analyzed to tune the parameters of the switching control. Experimental and simulated results are given to show that the proposed control can be effective to attenuate the vibration in flexible links. Likewise, it is also demonstrated that nonlinearities and coupling effect of beam and cart dynamics in the system model can be compensated by adequate tuning of the switching parameter, K_s.

Our future work will focus on developing a more complete simulation model of the system with nonlinear elements and designing other control strategies.

References

1. Benosman, A., Vey, G.L.: Control of flexible manipulators: A survey. Robotica 22(5), 533–545 (2004)
2. Hu, J., Zhu, D.: Vibration control of smart structure using sliding mode control with observer. Journal of Computers 7(2), 411–418 (2012)
3. Gu, H., Song, G., Malki, H.: Chattering-free fuzzy adaptive robust sliding-mode vibration control of a smart flexible beam. Smart Materials and Structures 17(3), 035007 (2008)
4. Hu, Q., Wang, Z., Gao, H.: Sliding mode and shaped input vibration control of flexible systems. IEEE Transactions on Aerospace and Electronic Systems 44(2), 503–519 (2008)
5. Pai, M.C., Sinha, A.: Sliding mode output feedback control of vibration in a flexible structure. Journal of Dynamic Systems, Measurement, and Control 129(6), 851–855 (2007)
6. Banavar, R., Dey, B.: Stabilizing a flexible beam on a cart: A distributed port-hamiltonian approach. Journal of Nonlinear Science 20, 131–151 (2010)
7. Ghaith, F.A., Hamdan, M.N.: Dynamic modeling and control of elastic beam fixed on a moving cart and carrying lumped tip. Jordan Journal of Mechanical and Industrial Engineering 5(1), 61–70 (2011)
8. Park, S., Kim, B.K., Youm, Y.: Single-mode vibration suppression for a beam-mass-cart system using input preshaping with a robust internal-loop compensator. Journal of Sound and Vibration 241(4), 693–716 (2001)
9. Lin, J., Chao, W.-S.: Vibration suppression control of beam-cart system with piezoelectric transducers by decomposed parallel adaptive neuro-fuzzy control. Journal of Vibration and Control 15(12), 1885–1906 (2009)

10. Mamani, G., Becedas, J., Feliu, V.: Sliding mode tracking control of a very lightweight single-link flexible robot robust to payload changes and motor friction. Journal of Vibration and Control 18(8), 1141–1155 (2012)
11. Chang, J.-L.: On chattering-free dynamic sliding mode controller design. Journal of Control Science and Engineering 2012, 564906 (2012)
12. O'Toole, M.D., Bouazza-Marouf, K., Kerr, D.: Chatter Suppression in Sliding Mode Control: Strategies and Tuning Methods. In: ROMANSY 18 Robot Design, Dynamics and Control, vol. 524, pp. 109–116. Springer (2010)
13. Chen, M.S., Chen, C.H., Yang, F.Y.: An LTR-observer-based dynamic sliding mode control for chattering reduction. Automatica 43(6), 1111–1116 (2007)
14. Lee, H., Utkin, V.I.: Chattering suppression methods in sliding mode control systems. Annual Reviews in Control 31(2), 179–188 (2007)

Modal Filtering Properties of Piezoelectric Laminates with Directional Actuation

João C.P. Reis* and José Sá da Costa

IDMEC/LAETA, Instituto Superior Técnico, Universidade de Lisboa
{reis,sadacosta}@dem.ist.utl.pt

Abstract. A study of the modal filtering properties of piezoelectric laminates based on directional characteristics of actuation is presented. Piezoelectric laminates are often used to perform collocated vibration control of structural elements like flexible robot links. The relation between the directional properties of the piezoelectric actuation, and their influence on the excitation of vibration modes is specifically addressed. An example application shows that it is possible to design a multi-active-layer laminate to generate a specified in-plane force state with piezoelectric actuators of fixed directional characteristics. Several of these designs are used in the example to perform selective excitation of vibration modes, based only on the specified angles of actuation and without changing the actuator position on the structure.

Keywords: Vibration control, Piezoelectric, Spatial filtering, Induced strain actuation.

1 Introduction

Vibration control using piezoelectric actuation is one of the methods for dealing with the structural vibrations of flexible link robots that as gained some attention from the research community in recent years [1–7]. It is common that planar flexible robots are considered, and so only the bending modes of the links are controlled by the piezoelectric actuators. To achieve this, monolithic piezoelectric plates with in-plane isotropic actuation properties are often used, properly bonded to the flexible links, forming a laminated active structure. However, for a robot with a more general serial configuration, the links may also have flexible behavior for other displacement degrees of freedom, resulting in more complex strain distributions on the surface of the link during oscillations. For these cases, the in-plane isotropic actuation properties of monolithic plates may represent a severe limitation. Recently, a number of piezoelectric actuators has been developed with different characteristics and actuation possibilities. In [8] an electrode design named *InterDigitated Electrodes* (IDE) is used, that allows for the poling

* This work was supported by FCT, through IDMEC, under LAETA. Projects: PEst - OE/EME/LA0022/2011-LAETA/IDMEC/CSI and SurgRob - PTDC/EME-CRO/114571/2009.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_12,

of the piezoceramic material to be made along a direction belonging to the plane of the lamina. This design results in directional (non-isotropic) in-plane actuation for a monolithic plate. In [9] a composite actuator with piezoceramic fibers is presented. This actuator is made of cylindrical fibers of piezoceramic material in epoxy matrix, packaged in insulating film with an etched IDE pattern, and has both non-isotropic actuation and elastic characteristics. These composite actuators with cylindrical fibers are known as *Active Fiber Composites* (AFC). Another version of this concept, named *Macro Fiber Composite* (MFC), is presented in [10] where rectangular cross section fibers are used. In [11] an MFC actuator with special fibers made of a single piezoelectric crystal is presented. Several recent studies have also addressed the application of directional piezoelectric actuators to structural control [12–15], including the control of both bending and torsion modes in tubular structures.

This paper presents a study of the application of directional properties of laminates with piezoelectric laminas to vibration problems. The role of the directionality of actuation in the excitation of vibration modes is addressed at the laminate level, and an example application to the selective excitation of vibrations modes in a plate is presented.

2 Active Laminates

2.1 Work on Laminates in Free Vibration

The work per unit area performed by in-plane forces and moments in a laminate, in conditions of plane stress and considering the Kirchoff assumptions, is given by

$$\overline{u}\, h = \frac{1}{2} \begin{bmatrix} \mathbf{N}^T & \mathbf{M}^T \end{bmatrix} \begin{bmatrix} \varepsilon^0 \\ \kappa \end{bmatrix} \tag{1}$$

where $\overline{u}$ is the mean energy density through the total thickness h of the laminate, $\mathbf{N}$ is the vector of in-plane forces per unit length, $\mathbf{M}$ is the vector of moments per unit length, ε^0 is the vector of the local mid-surface strains, and κ is the vector of local curvatures. For a laminate in free vibration, the vector of local strains and curvatures may be decomposed into an infinite series of modal components with modal amplitude ν_j resulting in

$$\overline{u}\, h = \frac{1}{2} \begin{bmatrix} \mathbf{N}^T & \mathbf{M}^T \end{bmatrix} \left(\sum_{j=1}^{\infty} \nu_j \begin{bmatrix} \varepsilon^0 \\ \kappa \end{bmatrix}_j \right) \tag{2}$$

Assuming that the only acting forces and moments are induced in the laminate by piezoelectric laminas, the work performed by these actuators on mode j is given by

$$\overline{u}_j\, h = \frac{\nu_j}{2} \begin{bmatrix} \mathbf{N}_p^T & \mathbf{M}_p^T \end{bmatrix} \begin{bmatrix} \varepsilon^0 \\ \kappa \end{bmatrix}_j \tag{3}$$

According to [9], considering the above stated assumptions, the piezoelectrically induced forces $\mathbf{N}_p$ and moments $\mathbf{M}_p$ for the case of active laminas with constant properties through the thickness are given by

$$\begin{bmatrix} \mathbf{N}_p \\ \mathbf{M}_p \end{bmatrix} = \sum_{k=1}^{n_p} h_{p\_k} \begin{bmatrix} \mathbf{I} \\ \bar{z}_k \mathbf{I} \end{bmatrix} \mathbf{e}_k^* E_k \tag{4}$$

where n_p is the number of active laminas, $\mathbf{I}$ is the 3×3 identity matrix and h_{p_k} and $\bar{z}_k$ represent the thickness and the out-of-plane mean coordinate of active lamina k respectively. Also E_k is the scalar value of applied electric field in the poling direction, and $\mathbf{e}_k^* = [e_{31}\ e_{32}\ 0]_k^T$ is the vector of piezoelectrically induced stress constants of lamina k after considering the plane stress assumptions. Subscripts 1 and 2 refer to the in plane coordinates of the lamina, while subscript 3 refers to the out-of-plane coordinate perpendicular to that plane. Poling direction of the actuator is assumed to be the standard direction 3 also according to [9]. While this is not the case for actuators with IDE, it carries no loss of generality as long as it is mathematically in accordance with the applied electric field.

If we consider the case where each lamina is allowed to have its own orientation θ_k relative to the referential of the generated forces, a rotation matrix $\mathbf{T}_k$ must be considered in Eq. (4) such that

$$\begin{bmatrix} \mathbf{N}_p \\ \mathbf{M}_p \end{bmatrix} = \sum_{k=1}^{n_p} h_{p\_k} \begin{bmatrix} \mathbf{I} \\ \bar{z}_k \mathbf{I} \end{bmatrix} \mathbf{T}_k^{-1} \mathbf{e}_k^* E_k \tag{5}$$

where

$$\mathbf{T}_k = \begin{bmatrix} \cos^2\theta_k & \sin^2\theta_k & 2\sin\theta_k\cos\theta_k \\ \sin^2\theta_k & \cos^2\theta_k & -2\sin\theta_k\cos\theta_k \\ -\sin\theta_k\cos\theta_k & \sin\theta_k\cos\theta_k & \cos^2\theta_k - \sin^2\theta_k \end{bmatrix}$$

Eq. (3) can be used in the maximization or minimization of the influence of the active laminate on a specific vibration mode, or set of vibration modes. It is clear from this expression that the relation between the vector of forces and moments generated by the piezoelectric laminas, and the vector of local strains and curvatures that result from the mode shape solutions is crucial to this problem.

Two special cases are common in flexible link robots: the case of a link with tubular thin-walled cross section, for which $||\varepsilon_j^0|| >> ||\kappa_j||$; and the case of a link with thin rectangular cross section, common in the planar robot geometry, for which the inverse relation occurs. For each case the work per unit area can be approximated by Eqs. (6) and (7) respectively.

$$\overline{u}_j\, h \approx \frac{\nu_j}{2} \mathbf{N}_p^T \varepsilon_j^0 = \frac{\nu_j}{2} \sum_{k=1}^{n_p} h_{p\_k} E_k \mathbf{T}_k^{-1} \mathbf{e}_k^*\, \varepsilon_j^0 \tag{6}$$

$$\overline{u}_j\, h \approx \frac{\nu_j}{2}\mathbf{M}_p^T \kappa_j = \frac{\nu_j}{2}\sum_{k=1}^{n_p} \bar{z}_k h_{p\_k} E_k \mathbf{T}_k^{-1}\mathbf{e}_k^*\, \kappa_j \tag{7}$$

Given vectors ε_j^0 and κ_j for a specific mode of vibration, the only difference between the analysis of these two expressions are the variables $\bar{z}_k$ in Eq.(7), representing the moment arm of each active lamina relative to the mid-surface. The problem of maximization or minimization of the influence of the active laminate on vibration mode j amounts to maximize or minimize, respectively, the value of $|\overline{u}_j|$ in each point of the active area of the laminate.

From Eqs.(6) and (7) it is clear that, if all terms of the sum have the same sign, the maximization of $|\overline{u}_j|$ can be achieved by maximizing the absolute value of each term, meaning that if each individual active lamina has maximum influence over the work per unit area of mode j, then the total value for all laminas will also be maximized. The roles of $\bar{z}_k$, h_{p_k} and E_k to the maximization of each term are also clear, and it is possible to show that if axis 1 and 2 of each active lamina are defined such that $|e_{31}| \geq |e_{32}|$, then the alignment of each lamina k with the principal direction of mid-surface strain yields the maximum absolute value of $\mathbf{T}_k^{-1}\mathbf{e}_k^*\, \varepsilon_j^0$ in Eq. (6), while the alignment of each active lamina with the principal direction of curvature yields the maximum absolute value of $\mathbf{T}_k^{-1}\mathbf{e}_k^*\, \kappa_j$ in Eq. (7).

The case of minimization of the actuation influence over mode j requires, for each case, that $\mathbf{N}_p^T \varepsilon_j^0 = 0$ or that $\mathbf{M}_p^T \kappa_j = 0$. This means that the active laminate must be able to generate a local force vector that is orthogonal to the local strain or curvature vector of mode j. If it were possible to tailor the piezoelectric and elastic characteristics of the active laminas, such that vector $\mathbf{e}_k^*$ would fulfill the needs of each design case, then the problem would be trivial. However, since currently available actuators have fixed piezoelectrically induced stress properties, the need arises to design a laminate, with two or more active laminas, that is able to perform the task of generating a specific in-plane force state.

2.2 Design of an Active Laminate for a Specific Force State

From Eq.(5), considering a laminate with a pair of active laminas with the same properties, the generated in-plane force can be written as

$$\mathbf{N}_p = \begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = h_p \left(E_1 \mathbf{T}_1^{-1} + E_2 \mathbf{T}_2^{-1}\right) \begin{bmatrix} e_{31} \\ e_{32} \\ 0 \end{bmatrix} \tag{8}$$

For a given type of active lamina, the design of the laminate has four unknown values θ_1, θ_2, E_1 and E_2, and only three equations are available to determine these values. However, using an analogy between the in-plane force vector and Mohr circle for plane stress, it is possible to obtain closed form solutions to the design problem by considering additional constraints.

Let the force state $\mathbf{N}_p$ be described in a cartesian plane by points (N_x, N_{xy}) and $(N_y, -N_{xy})$. These points lie on a circle with center c and radius r, and are uniquely defined by the principal direction 2ψ, relative to the center of the circle, given by

$$\begin{aligned} c &= (N_x + N_y)/2 \\ r &= \sqrt{N_{xy}^2 + ((N_x - N_y)/2)^2} \\ \tan 2\psi &= 2N_{xy}/(N_x - N_y) \end{aligned} \tag{9}$$

Following the same analogy, an active lamina has a characteristic circle with center position c_p and radius r_p that is defined by its piezoelectrically induced stress constants

$$\begin{aligned} c_p &= (e_{31} + e_{32})/2 \\ r_p &= |e_{31} - e_{32}|/2 \end{aligned} \tag{10}$$

It is possible to show that if the relation $r/|c| > r_p/|c_p|$ holds, then choosing $\theta_2 = \theta_1 + \pi/2$ leads to the minimization of $|E_1| + |E_2|$. Using this relation as the fourth constraint the design problem solution is

$$\begin{aligned} \theta_1 &= \psi & E_1 &= \tfrac{1}{2h_p}\left(\tfrac{c}{c_p} + \tfrac{r}{r_p}\right) \\ \theta_2 &= \psi + \tfrac{\pi}{2} & E_2 &= \tfrac{1}{2h_p}\left(\tfrac{c}{c_p} - \tfrac{r}{r_p}\right) \end{aligned} \tag{11}$$

When the case is such that $r/|c| < r_p/|c_p|$ one finds the value of $|E_1| + |E_2|$ to be constrained by the value of c. Then by considering $E_1 = E_2$ as the fourth constraint the solution is

$$\begin{aligned} \theta_1 &= \psi + \arccos\left(\tfrac{r\,c_p}{c\,r_p}\right)/2 & E_1 = E_2 &= \tfrac{1}{2h_p}\tfrac{c}{c_p} \\ \theta_2 &= \psi - \arccos\left(\tfrac{r\,c_p}{c\,r_p}\right)/2 \end{aligned} \tag{12}$$

The case where $r/|c| = r_p/|c_p|$ is trivial, since only a single lamina is required. The solution in this case can also be obtained from Eq. (12). If a single lamina is used then $\theta_1 = \psi$ and $E_1 = c/h_p c_p$.

3 Planar Flexible Robot Link Example

3.1 Plate Model

For the case of a planar robot link subjected to no significant mid-surface strains Eq.(7) applies. Some care must be taken in the design of the laminate sequence, in order to ensure that no resulting in-plane forces are generated by the actuators.

The chosen laminated structure is built out of four active laminas and a passive isotropic substrate made of aluminium with the following stacking sequence:

$$[\theta_1^{PZT}/\theta_2^{PZT}/iso^{Al}/\theta_1^{PZT}/\theta_2^{PZT}]$$

This choice of layer sequence allows for the generation of pure bending moments $\mathbf{M}_p$. This is achieved by applying the following electric fields to the layers, according to each layer position in the sequence

$$[\,E_1/E_2/\,0\,/-E_1/-E_2]$$

The result is $\mathbf{N}_p = 0$. The active layers chosen for this application have properties similar to the actuators in [8], meaning that they are elastically isotropic, but generate in-plane directional actuation due to the IDE design. This allows for the angle of actuation to be selected in the design stage without changing the elastic characteristics of the laminate, that otherwise could also modify the mode shape solutions of the vibration modes, and therefore require an iterative design procedure.

Although the piezoelectric layers cover the entire plate, meaning that there are no elastic discontinuities in the model, the applied electric fields are limited to a small area portion near the clamped edge. Also, a small tip mass was added the lower free vertex of the plate, so that the mode shape solution will show more interesting strain angles according to the vibration mode considered. In Fig. 1 the mode shapes of the first four modes of vibration resulting from a finite element modal analysis are presented.

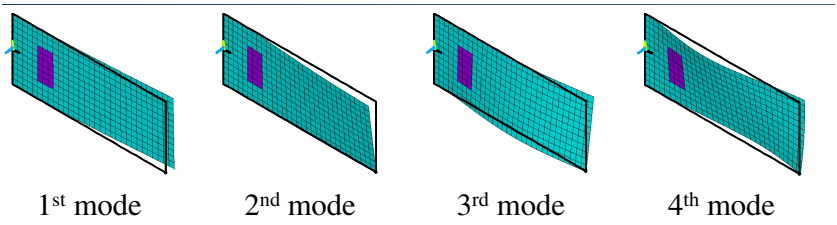


Fig. 1. Mode shapes for the first four modes of vibration with natural frequencies: $0.95Hz$, $13.6Hz$, $28.6Hz$ and $56.2Hz$

Assuming small out-of-plane displacements and $\varepsilon^0 \approx 0$ the strains are related to local curvature by $\varepsilon = z\,\kappa$. The top surface strains ε^S can therefore be used to calculate the local curvature through $\kappa = 2\,\varepsilon^S/h$. This relation allows the active laminate to be designed based on the principal directions of the top surface strain for each mode of vibration. Accordingly, Eq.(7) can be written as

$$\overline{u}_j\,h \approx \frac{\nu_j}{h}\mathbf{M}_p^T\varepsilon_j^S \tag{13}$$

The top surface strains of each finite element of the active area of the plate are represented in Fig. 2 for each mode of vibration. All elements have the same area. The mean value of principal strains for the actuated area of the plate, and the corresponding principal strain angle, are presented in Table 1.

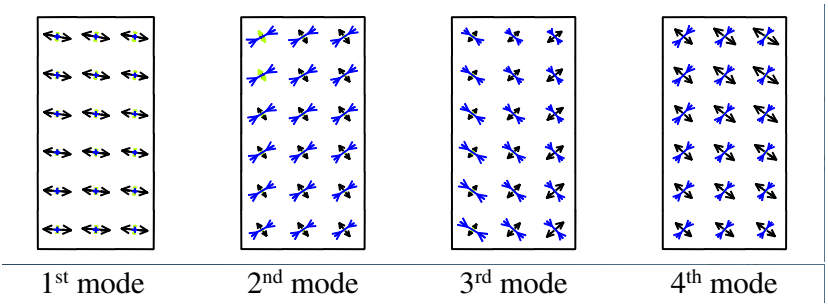


Fig. 2. Representation of surface strains of the active portion of the plate for the first four modes of vibration

Table 1. Top surface strain values for the principal directions

	Mode 1	Mode 2	Mode 3	Mode 4
$\bar{\varepsilon}_I$	3.90×10^{-3}	-4.32×10^{-2}	-5.92×10^{-2}	1.46×10^{-1}
$\bar{\varepsilon}_{II}/\bar{\varepsilon}_I$	-0.282	-0.579	-0.709	-0.865
$\bar{\theta}_I(^\circ)$	-7.37	33.9	-43.4	-39.6

3.2 Actuation Design

The maximization of the actuator influence is achieved by aligning both actuator directions with the principal direction of the mean strain over the area of the top surface. Assuming that, for the separate study of each mode of vibration, the main reference frame is aligned with the principal direction of the top surface strain, then $\theta_1 = \theta_2 = 0$ in all the active area portion of the plate.

The minimization of the actuator influence requires that $\mathbf{M}_p^T \varepsilon_j^S = 0$. Considering the same assumption as before, one finds that in order to fulfill this condition

$$\mathbf{M}_p \propto \begin{bmatrix} -\bar{\varepsilon}_{II}/\bar{\varepsilon}_I \\ 1 \\ a \end{bmatrix}_j \tag{14}$$

where a may take any value, since the third element of ε_j^S written in relation to the principal direction is zero. The chosen laminate stacking sequence and applied electric fields imply that $\mathbf{M}_p = z_{eq}\mathbf{N}_p$, where $z_{eq} \neq 0$ is the equivalent moment arm of the active layers that share the same orientation. This means

that the design procedure for $\mathbf{N}_p$ described in section 2.2 can be directly applied to this problem. Let $a = 0$, the resulting design parameters are

$$c_j = \left(- \left(\frac{\bar{\varepsilon}_{II}}{\bar{\varepsilon}_I} \right)_j + 1 \right) / 2$$

$$r_j = \sqrt{\left(\left(- \left(\frac{\bar{\varepsilon}_{II}}{\bar{\varepsilon}_I} \right)_j - 1 \right) / 2 \right)^2} \tag{15}$$

$$\psi_j = \frac{\pi}{2} \; (rad)$$

The relevant properties of the active laminas are the same as those of Midé QP10ni transducers. For these actuators and for all vibration modes, the relation $r/|c| < r_p/|c_p|$ holds. The corresponding solutions for the active lamina angles are presented in Table 2.

Table 2. Values of active lamina angles relative to the principal direction of each vibration mode, and relative to the design angle ψ_j

	Mode 1	Mode 2	Mode 3	Mode 4
$\theta_1(^\circ)$	120.73	128.43	130.82	133.23
$\theta_2(^\circ)$	59.27	51.57	49.18	46.77
$\theta_1 - \psi_j(^\circ)$	30.73	38.43	40.82	43.23
$\theta_2 - \psi_j(^\circ)$	-30.73	-38.43	-40.82	-43.23

Note that the specific electric field values resulting from Eq. (12) are not relevant for this example as long as $E_1 = E_2$.

3.3 Results

Results from several finite element simulations are presented in Figs. 3-5. In all simulations the maximum possible voltage amplitude allowed by the QP10ni transducers was applied to the actuators, and the out-of-plane displacement of the upper vertex of the plate was used as output. Also, a damping coefficient of 0.01 was assumed for all four vibration modes to prevent numerical problems. In Fig. 3 the magnitude of frequency response is shown for all the eight different plate designs. For each case, the dotted line represents the FRF magnitude for the design that maximizes the influence over the respective vibration mode, while the continuous line represents the design that is supposed to minimize this influence. In all four cases the minimization curves show no significant resonant peak at the natural frequency of the vibration mode.

In Figs. 4 and 5, the design solutions are put into context against other possible choices of angles for the active laminas. Each point of the series was obtained by feeding the actuators with a sinusoidal voltage of maximum amplitude at the natural frequency of interest. This was done for a set of plate models spanning the complete design range of θ_k with equally spaced values. The points represent the magnitude of the tip displacement obtained for each plate model, multiplied by -1 when the phase is inverted. In Fig. 4 it is possible to see that the average principal direction of strain at the top surface, that was chosen as the direction of actuation for each mode of vibration, is closely aligned with the maximum magnitude values obtained in simulation in all four cases, which was expected from theory.

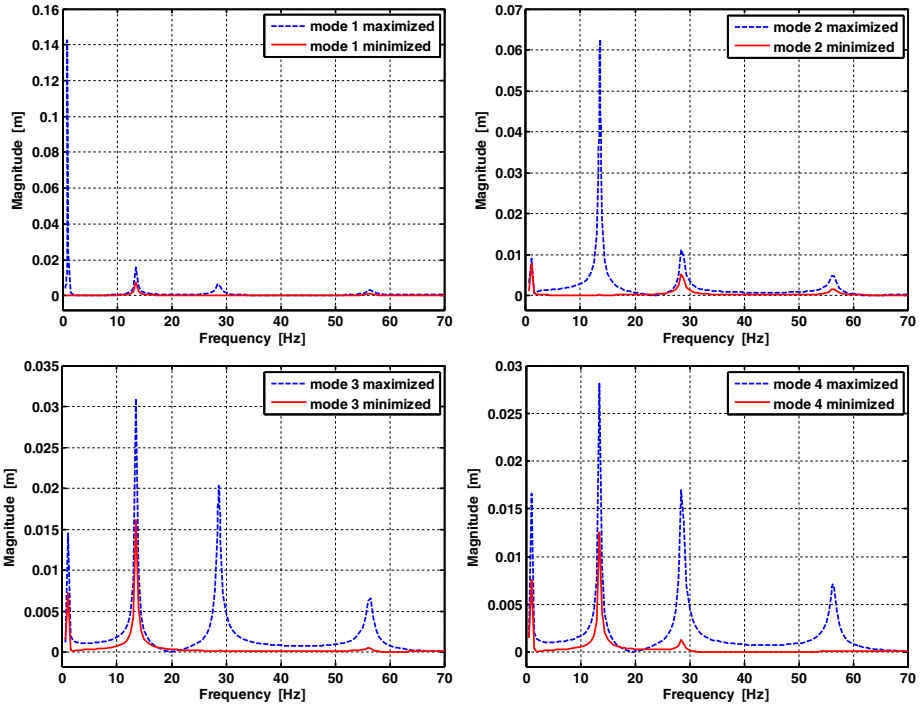


Fig. 3. Magnitude of frequency response for each mode

In Fig. 5 the values concern the influence minimization case. An evaluation of the behavior of magnitude as a function of the distance between the active lamina angles and the design angle ψ_j is presented. The resulting distances span the interval $[0°, 90°]$. It is possible to see in this case, that the angles resulting from the design procedure, and that are given in Table 2, are also in close alignment with the zero-crossing points of the signed magnitude representation.

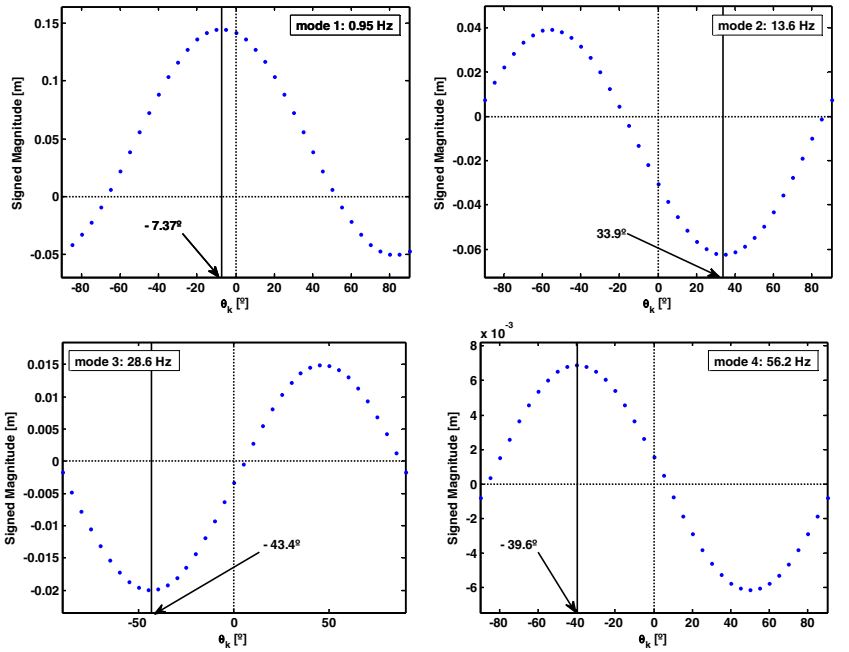


Fig. 4. Evaluation of the designs for influence maximization

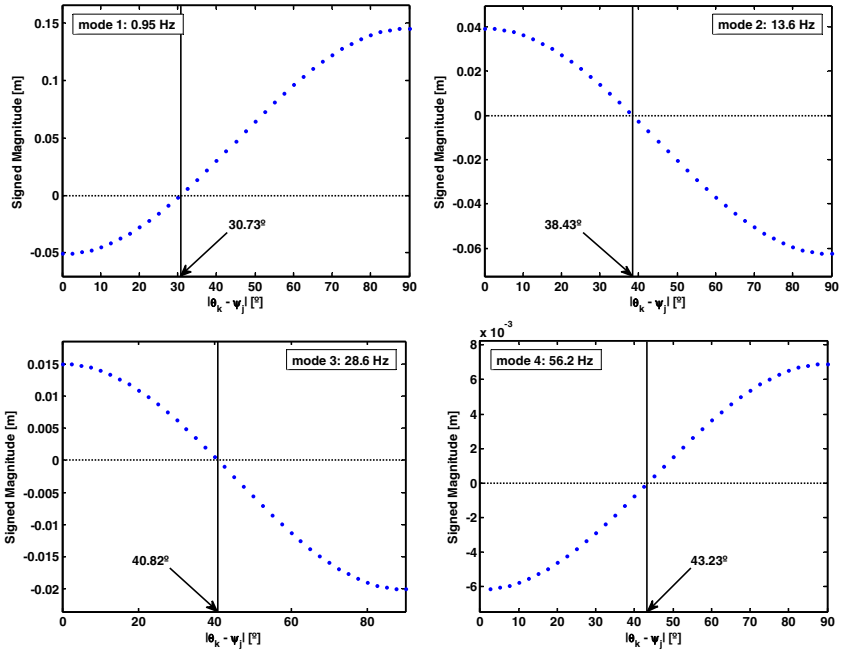


Fig. 5. Evaluation of the designs for influence minimization

4 Conclusions

A study of the modal filtering properties of piezoelectric laminates with directional actuation characteristics was presented. The study established the relation between piezoelectrically induced forces in the laminate, and the work performed by these forces on vibration modes. A design methodology, based on an analogy between Mohr circle for plane stress and the piezoelectrically induced forces, was used to design a set of multi-active-layer laminates with maximized or minimized actuation influence over selected modes of vibration. Simulation results for the FRFs of the laminates designed for minimization of influence showed no significant magnitude peaks at the frequencies of the targeted vibration modes. Both the maximization and the minimization results were put into context against other possible choices of actuation angles for the multi-active-layer laminate. These results confirmed the design assumptions, and the angles obtained from the design solutions.

References

1. Khorrami, F., Zeinoun, I., Tome, E.: Experimental results on active control of flexible-link manipulators with embedded piezoceramics. In: Proceedings of 1993 IEEE International Conference on Robotics and Automation, Atalanta, U.S.A, vol. 3, pp. 222–227 (May 1993)
2. Denoyer, K.K., Kwak, M.K.: Dynamic modeling and vibration suppression of a slewing structure utilizing piezoelectric sensors and actuators. Journal of Sound and Vibration (189), 13–31 (1996)
3. Choi, S.B., Cho, S.S., Shin, H.C., Kim, H.K.: Quantitative feedback theory control of a single-link flexible manipulator featuring piezoelectric actuator and sensor. Smart Materials and Structures (8), 338–349 (1999)
4. Moallem, M., Kermani, M.R., Patel, R.V., Ostojic, M.: Flexure Control of a Positioning System Using Piezoelectric Transducers. IEEE Transactions on Control Systems Technology 12(5), 757–762 (2004)
5. Shan, J., Liu, H.T., Sun, D.: Slewing and vibration control of a single-link flexible manipulator by positive position feedback (PPF). Mechatronics 15(4), 487–503 (2005)
6. Gurses, K., Buckham, B.J., Park, E.J.: Vibration control of a single-link flexible manipulator using an array of fiber optic curvature sensors and PZT actuators. Mechatronics 19(2), 167–177 (2009)
7. Reis, J.C.P., Sá da Costa, J.: Motion Planning and Actuator Specialization in the Control of Active-Flexible Link Robots. Journal of Sound and Vibration 331(14), 3255–3270 (2012)
8. Hagood, N.W., Kindel, R., Ghandi, K., Gaudenzi, P.: Improving transverse actuation of piezoceramics using interdigitated surface electrodes. In: Proc. SPIE Smart Structures and Materials 1993: Smart Structures and Intelligent Systems, Albuquerque, NM, USA (1993)
9. Hagood, N.W., Bent, A.A.: Development of Piezoelectric Fiber Composites for Structural Actuation. In: 34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, LaJolla, California, USA, pp. 3625–3638 (1993)

10. Wilkie, W.K., Bryant, G.R., High, J.W., Fox, R.L., Hellbaum, R.F., Jalink, A., Little, B.D., Mirick, P.H.: Low-Cost Piezocomposite Actuator for Structural Control Applications. In: Proceedings, SPIE 7th Annual International Symposium on Smart Structures and Materials, Newport Beach, CA, pp. 323–334 (2000)
11. Wilkie, W.: Anisotropic Piezocomposite Actuator Incorporating Machined PMN-PT Single Crystal Fibers. In: 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Palm Springs, CA, U.S.A (2004)
12. Neugebauer, R., Drossel, W.G., Kranz, B., Kunze, H.: Multimode vibration reduction concept for machine tools and automotive applications. In: Proceedings of SPIE, vol. 5762, pp. 252–260 (2005)
13. Deraemaeker, A., Benelechi, S., Benjeddou, A., Preumont, A.: Analytical and Numerical Computation of homogenized properties of MFCs: Application to a composite boom with MFC actuators and sensors. In: III ECCOMAS Thematic Conference on Smart Structures and Materials, Gdansk, Poland (2007)
14. Bhadbhade, V., Jalili, N., Mahmoodi, S.N.: A novel piezoelectrically actuated flexural/torsional vibrating beam gyroscope. Journal of Sound and Vibration 311, 1305–1324 (2008)
15. Reis, J.C.P., Sá da Costa, J.: Design and experimental control of an active-flexible tubular manipulator link. Journal of Intelligent Material Systems and Structures 23(15), 1741–1757 (2012)

Slip Detection in Robotic Hands with Flexible Parts

Raul Fernandez*, Ismael Payo, Andres S. Vazquez, and Jonathan Becedas

Universidad de Castilla La Mancha,
E.T.S. Ingenieros Industriales, Ciudad Real, Spain
{raul.fernandez,ismael.payo,andress.vazquez,jonathan.becedas}@uclm.es

Abstract. It is well known that tactile sensing is of major importance in robots that interact with the environment. It provides relevant information for the robot to detect contacts with surfaces and manipulate objects. Flexible fingers have demonstrated its usefulness in the field. Due to their high flexibility they can detect contacts and can be safely used to control forces with high accuracy over any object. Among their advantages are low weight, low kinetic energy, low inertia, high flexibility, durability and resistance. In this paper, we demostrate how to take advantage of flexible links in order to detect slipping. We show how the same principle can be applied in force sensors based on flexible parts. In particular, we also show the slip detection with a simple and low cost sensor based on flexible beam deformation, with the finger torque sensor of the Barrett Hand, and with a 6 axis force sensor mounted in the wrist of an industrial robot.

Keywords: Slip Detection, Flexible Robots, Flexible Finger Grippers, Force Tactile Sensors, Grasping, Manipulation.

1 Introduction

Tactile sensing has increasing interest in the robotics field. It allows the robots to physically interact with the environment and modify it by improving basic capabilities such as touching, detecting surfaces and collisions, and advanced capabilities such as grasping and manipulation of objects [19], [27]. These characteristics are of special relevance if the environment, in which the robot is interacting with, is unknown [10].

To carry out these tasks, scientists have designed specific tools to be installed in the robots. The most popular tools are anthropomorphic hands and grippers [5]. They require to have installed accurate and sophisticated force and torque sensors to detect the contact and to control the exerted forces over a surface. However those sensors usually have limited working range and high costs [6], [9].

* This work was supported by the Consejería de Ciencia y Tecnología of the Junta de Comunidades de Castilla-La Mancha, Spain, under Contract PPIC10-0202-8377, and by the European Social Fund.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_13,

Nevertheless other approaches, based on the use of flexible structural elements, have appeared. We usually know them as flexible robots [11]. They changed the concept of tactile sensing of robotic hands and grippers [3], [7], [14], [23], [29].

The use of flexible robotic fingers improves the performance of tasks where there is contact with the environment [12], [26]. In flexible elastic fingers, the structure can act as a sensor in a contact by measuring the forces and torques. The structure of flexible robots is very lightweight. This characteristic presents several advantages such as low energy consumption, low kinetic energy and low inertia. They also absorb the energy of the impact with a surface. As a consequence, the system can move faster and safer. On the other hand, robotic systems that use flexible elements require more complex modeling and control than robots with rigid parts only. More information about modeling and control of flexible manipulators can be found in [4] and [25]. Specifically, in this work we used the flexible fingers gripper designed in [3] based on previous works (see Fig. 1) of the authors [2], [24].

Many aspects have to be taken into account when rigid or flexible grippers are used for manipulation. In particular, slip detection is crucial to successfully grasp an object. In the scientific literature we found different techniques to detect slip [13], [16], [17], [18], [20], [21], [22], [28]. In summary the slipping can be detected in three manners: i) estimating the friction coefficient between the grasping surfaces, which is not an accurate indirect measurement. ii) Analyzing the changes in the contact footprint over a tactile array sensor: changes in the shape of the contact with the object and changes in the pressure-force distribution. This strategy is not useful if the object is not rigid. iii) Detecting microvibrations. This is a direct and reliable measure of the slipping, but of dynamic nature, so the dynamic sensors traditionally used, as accelerometres, are noisy and usually detect the slipping too late to respond to it and prevent it with a closed loop control law.

In this work we have used a method based on the analysis of microvibrations in order to detect slipping in our flexible grippers during grasping tasks. We have also tried this method on three force sensors based on structural deflection, showing its applicability to these sensors.

The paper is structured into 6 sections: Section 2 deals with the description of the flexible gripper; In Section 3 we present our slip detection algorithm; In Section 4 some experimental tests are shown; In Section 5 we present the slip detection with other strain-gauge based force sensors; And finally, in Section 6 the main conclusions of this work are summarized.

2 Flexible Link Gripper

The type of grippers analyzed in this work is an underactuated mechanism constituted by rigid parts (palm) and flexible parts (flexible fingers) as shown in Fig. 1.a. The two symmetric flexible beams (fingers) of the gripper are actuated by a *dc motor* by means of a worm gearbox. The flexible finger of the gripper shown in Fig. 1.b and Fig. 2 can be used as force sensors [3].

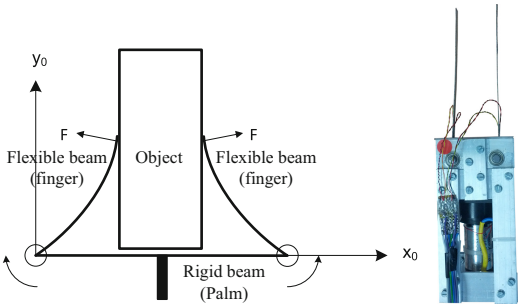


Fig. 1. a) Gripper scheme b) Flexible gripper

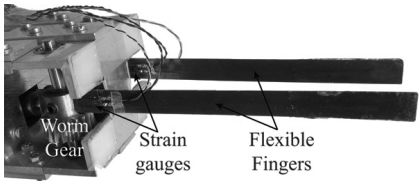


Fig. 2. Gripper parts

We assume that the flexible fingers are modelled according to the Euler - Bernoulli equation for beams subjected to small deflections and the neutral axis is non-extensible. The Euler-Bernoulli static equation for a beam deflected in one dimmension is

$$\frac{d^2y}{dx^2} = \frac{M}{EI} \tag{1}$$

where x, y are the cartesian coordinates of the elastic curve of the beam, M is the bending moment, E is the elastic modulus of the beam material and I is the second moment of area. If we assume that the bending moment is caused by a contact force F perpendicular to the x axis, the deflection of the flexible finger can be obtained by integrating twice the previous equation.

$$y(x) = \frac{Fx^2(3\lambda - x)}{6EI} \tag{2}$$

where λ is the distance from the contact point, between the finger and the object, to the base of the finger.

On the other hand, the dynamic model of the gripper that relates the contact force F with the voltage input to the motor can be defined as [3].

$$Au = \ddot{F} + B\dot{F} + \xi \tag{3}$$

where A and B are constant of the system and ξ is an input disturbance in the system.

A force feedback control system was designed in [3] based on the previous dynamic model. We have used the same control law to regulate the contact force applied by the gripper. In addition we have caused a slipping contact between the flexible finger and the grasp object to analyze in detail this type of contacts with our system, as it will be explained in Section 4.

3 Slip Detection Algorithm

When the surfaces of two objects come into slipping contact some structural vibrations of high frequencies arise in the objects. It has been tested in previous works (e.g. [16]) and it will also be demonstrated in the present study.

In order to detect a slipping contact in real time we have used an algorithm based on the Discrete Fourier Transform (DFT) of the strain gauges signals attached to the flexible structure. This method allows us to detect structural micro-vibrations of high frequencies in real time and therefore slipping contacts.

It is well known that the DFT of a signal φ is defined by the following equation.

$$\Phi(k) = \sum_{n=0}^{N-1} \varphi(n) e^{-i2\pi k \frac{n}{N}}; \quad k = 0, ..., N-1. \tag{4}$$

where N is the number of samples used. Instead of evaluating Eq. (4) directly, which requires $O(N^2)$ operations (O denotes an upper bound), we have used the Colley-Turkey algorithm [8]. This algorithm is a Fast Fourier Transform (FFT) that computes the same results in $O(N \log N)$ operations (lower computational cost).

The power spectrum of Φ can be calculated as

$$Pw(k) = \frac{\Phi(k)\,\Phi^*(k)}{N} \tag{5}$$

where $*$ denote the conjugate.

The detection method is summarized as follows: 1) the FFT of the signal Φ is computed each N samples eliminating the DC component; 2) the power spectrum of Φ is calculated in every iteration, obtaining the peak value and its frequency; 3) the peak value of the power spectrum is multiplied by its frequency; 4) finally, a slipping contact is detected when this later value exceeds a threshold defined empirically.

In order to detect slipping contact (associated to microvibration of high frequencies) in real time, the sampling frequency was chosen to be $f_s = 1\ kHz$ and the FFT was computed with the last $N = 64$ samples of the strain gauges signals. This means that the time needed to detect slipping is $N/f_s = 64\ ms$.

4 Experimental Results

The objective of this section is to analyze the behavior of the flexible gripper in order to detect slipping. First of all a brief description of the setup used for performing all experiments is introduced.

4.1 Experimental Setup

The flexible gripper was attached to a 6DOF manipulator (model Stäubli RX-90). This manipulator was used to move the gripper toward an object and to apply a certain force. The motion control of this manipulator is not explained here because it is not the objective of this study. The strain gauges used (model Kyowa KFG) have a resistance of $120 \pm 0.2\ \Omega$ and a gauge factor of 2.24 ± 1. Commercial strain gauge amplifiers (model Vishay BA660) were used for conditioning the gauges signals. All gauges signals were read with a computer by means of a commercial data acquisition system (model NI cRIO-9024) and were processed in real time with the commercial data logging application LabView. The sampling frequency for real time data acquisition tasks was chosen to be 1 kHz.

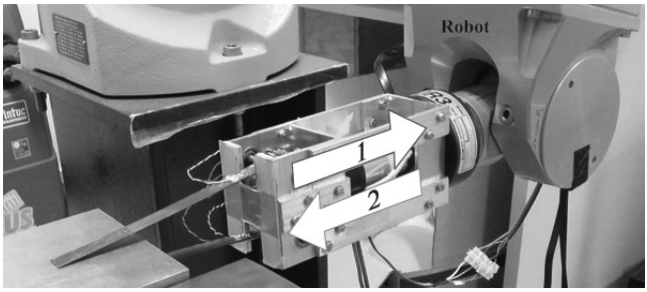


Fig. 3. Experimental setup with gripper

The contacted object was chosen to be a steel plate (flat surface) as shown in Fig. 3. The test was performed according to the following sequence: 1) the fingers were slipped on the surface of the plate; 2) the fingers were slipped on the opposite direction. Fig. 4.a shows the gauge signal of one of the fingers, measured by the strain gauges during the experiment.

4.2 Results

As seen in Fig. 4.a, the fingers slip in the plate from $t = 3.24\ s$ to $t = 4.65\ s$. From $t = 4.65\ s$ to $t = 5.01\ s$ the gripper kept still. From $t = 5.01\ s$ to $t = 6.06\ s$ the fingers slip in opposite direction. Finally, the gripper was stopped at $t = 6.06\ s$. Fig. 4.b (top) and (middle) show, respectively, the peak value of the power spectrum and its frequency at each iteration ($N = 64$ samples, i.e. each 64 ms) and Fig. 4.b (bottom) shows the product of both signals. It is observed that the slipping contact can be detected when this product exceeds a certain threshold (after performing many experiments this threshold was chosen to be 0.5). Note that high frequencies caused by noise in the signal are ignored by this method, detecting high frequencies associated to high spectrum peaks, two conditions which only appear with slipping contacts. Note also that the slip vibration frequency is about $125Hz$.

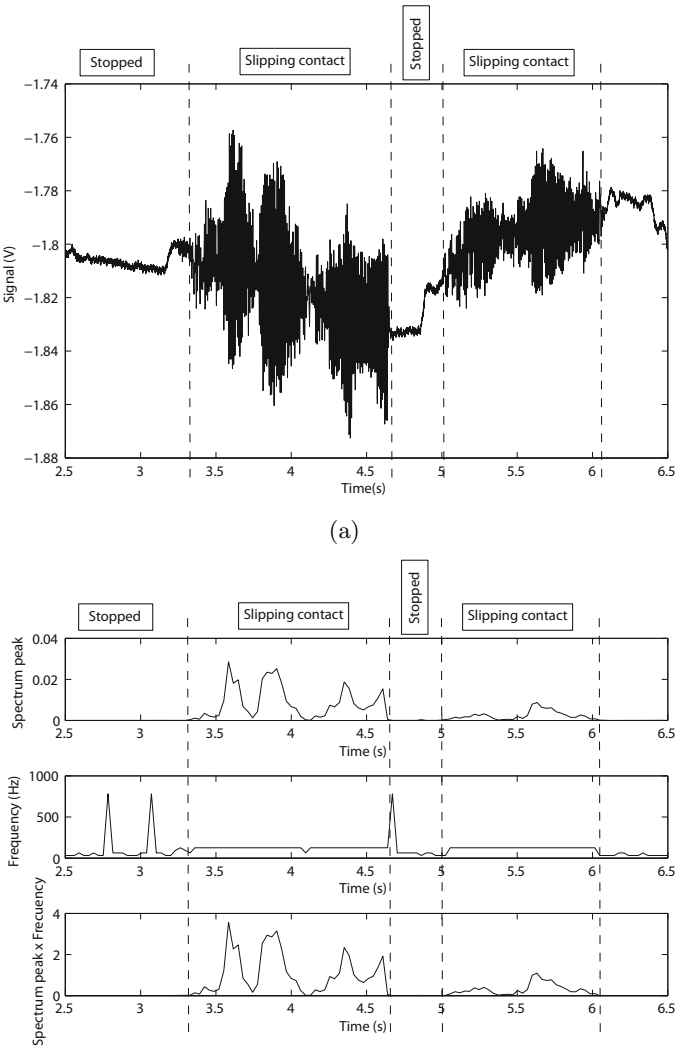


Fig. 4. a) Gauge voltage Finger 1 as a function of the time. b) Spectrum peak and its frequency as a function of the time (gripper).

5 Slipping Detection with Other Strain-Gauge Based Force Sensors

In addition to previous analysis, in this section we analyze if the same slip detection procedure can be applied to other strain-gauge based force sensors. In particular we have tested 3 different sensor systems based on structural deformation.

5.1 Tactile Force Sensor

This sensor has been developed by the authors with the aim of taking advantage of flexible robotics technology. The sensor is constituted by two structural parts as shown in Fig. 5: 1) passive part (semi-cylindrical metallic cover of aluminium) which comes into contact with the object and 2) sensed part (elastic cylindrical beam of methacrylate) where some strain gauges are strategically attached. The elastic beam is clamped at both ends to the robotic fingertip. Fig. 5.b shows the sensor mounted in the fingertip of a Barret Hand and Fig. 5.c shows a cross section of the sensor and their dimensions.

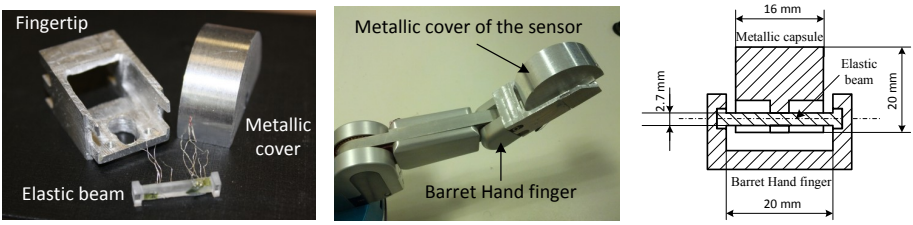


Fig. 5. a) Metallic cover of aluminium. b) Elastic beam. c) Cross section of the sensor

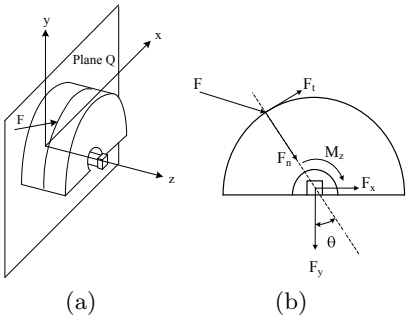


Fig. 6. a) Applied force. b) Decomposition of force F.

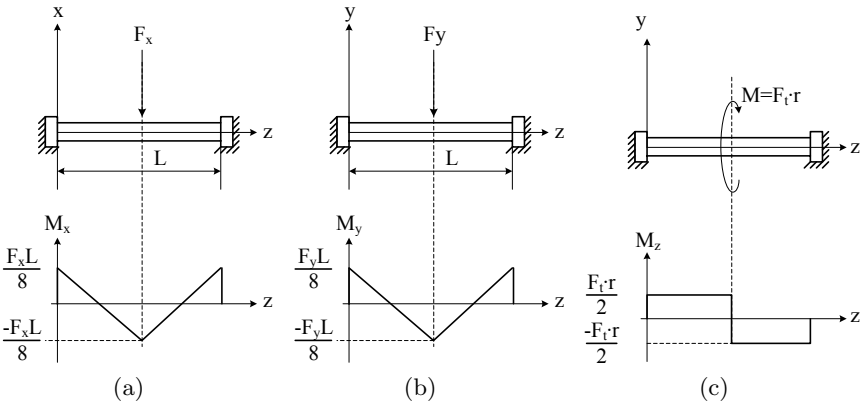


Fig. 7. a) Bending moment M_x. b) Bending moment M_y. c) Torsional moment M_z.

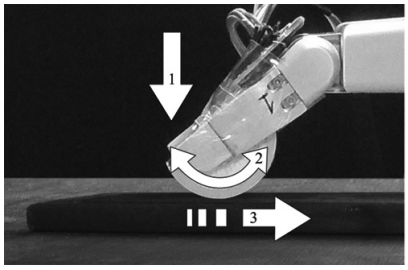


Fig. 8. Experimental setup with tactile force sensor

In order to model the sensor it is assumed that the force F is applied in the plane Q (see Fig. 6.a). The cover is clamped to the elastic beam at its midpoint as shown in Fig. 5.c and therefore the applied force is transmitted to the elastic beam through this point. The applied force F can be decomposed into normal and tangential forces (F_n and F_t) relatives to the surface of the metallic cover as shown in Fig. 6.b. In addition, the normal force can be decomposed into Cartesian coordinates relatives to the axis (X, Y) associated to the elastic beam as shown in Fig 6.b.

Fig. 7 shows the bending moments (M_x, M_y) and the torsional moment M_z transmitted to the elastic beam, caused respectively by F_x, F_y and F_t. These transmitted moments can be measured by using strain gauges on the elastic beam. Bending moments M_x and M_y are measured by two pairs of gauges placed in opposition at the ends of the beam and the torsional moment M_z is measured by two gauges placed at 45° with respect to the axial axis of the beam. With these configurations and an appropriate electrical connection to a Wheatstone bridge (2–gauge system), superimposed effects can be cancelled, measuring only the desired variable [15].

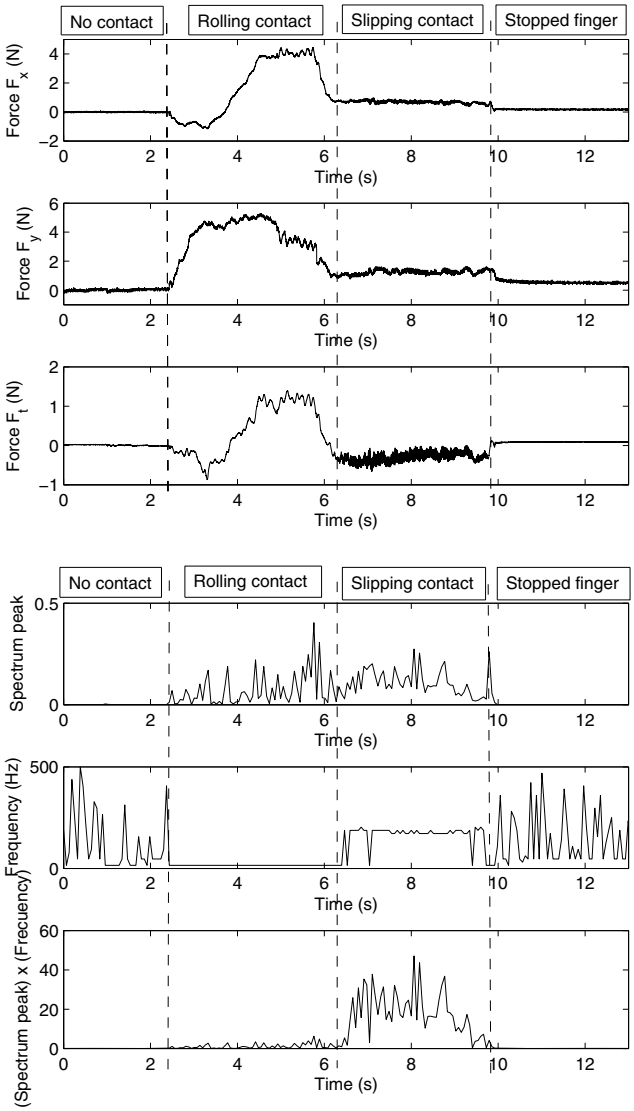


Fig. 9. a) Forces F_x, F_y and F_t in tactile sensor as a function of the time. b) Spectrum peak and its frequency as a function of the time.

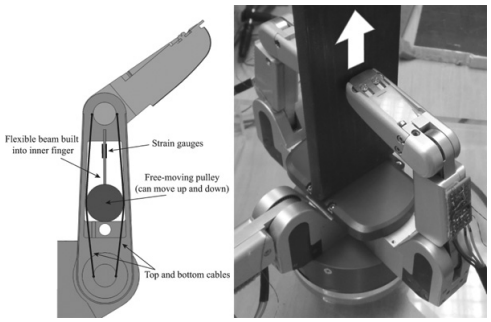


Fig. 10. a) Cutaway diagram of finger reveals internal strain gages on BarrettHand. b) Experimental setup with BarrettHand.

Experimentation and Discussion The objective of the test was to demonstrate the effectiveness of the tactile force sensor to detect slipping. A steel table (flat surface) was chosen to be the contacted object. The test was performed according to the following sequence: 1) the fingertip was moved toward the table until the sensor came into contact 2) the finger was rotated, allowing the semi-cylindrical surface of the sensor to roll on the table (rolling contact; 3) The finger was linearly moved producing the tactile sensor to slip on the surface of the table (slipping contact); 4) the finger was stopped. Fig. 8 illustrates the aforementioned sequence.

Fig. 9.a shows the forces F_x, F_y and F_t, measured by the strain gauges during the experiment. It can be seen that the sensor came into contact with the table at $t = 2.43\ s$. From $t = 2.43\ s$ to $t = 6.20\ s$ the cylindrical surface of the sensor rolled on the table. From $t = 6.20\ s$ to $t = 9.8\ s$ a slipping contact occurred. Finally, the finger stopped its movement at $t = 10\ s$. Fig. 9.b (top) and (middle) show, respectively, the peak value of the power spectrum and its frequency at each iteration ($N = 64$ samples, i.e. each 64 ms) and Fig. 9.b (bottom) shows the product of both signals. Threshold for the slipping contact was chosen to be 10 after performing many experiments. Note that high frequencies caused by noise in the signal (from $t = 0\ s$ to $t = 2\ s$ and from $t = 10\ s$ to $t = 13\ s$ as shown in Fig. 9.b (middle)) and high spectrum peaks caused during the rolling contact as shown in Fig. 9.b (top), are ignored by this method, detecting high frequencies associated to high spectrum peaks, two conditions which only appear with slipping contacts. Note also that the slip vibration frequency is about $180Hz$.

5.2 Barrett Hand Torque Finger Sensor

Each finger of the BarrettHand was mounted with a force sensing mechanism consisting of a flexible beam, a free-moving pulley, a pair of cables, and two strain gauges as is shown in Figure 10.a [1]. Basically, when a force is applied to the last phalange of the finger, the cables get tight which moves the pulley,

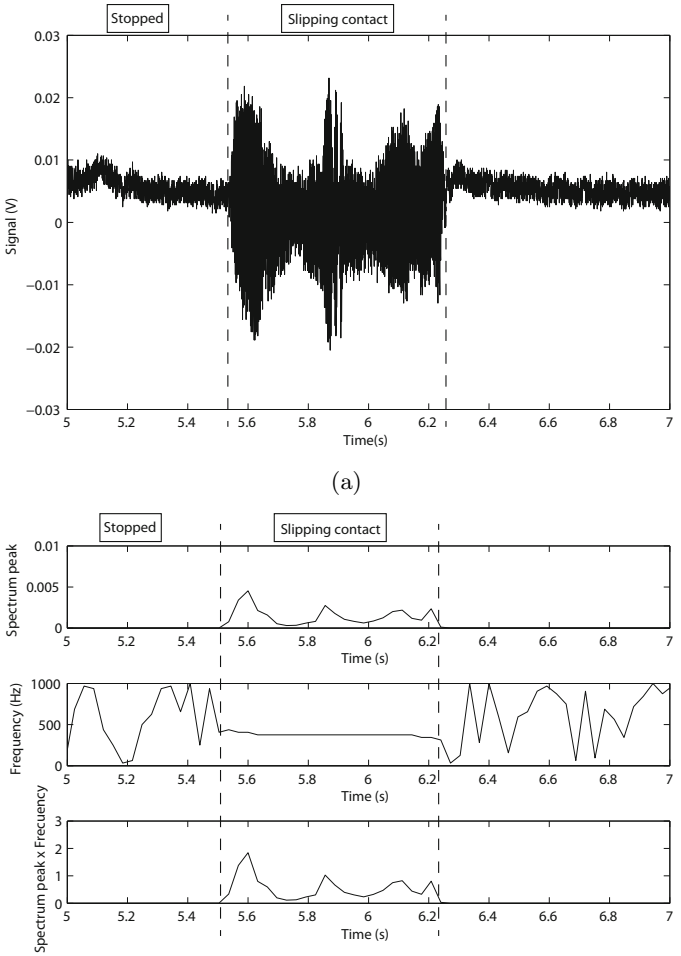


Fig. 11. a) Gauge signal Finger 1 as a function of the time. b) Spectrum peak and its frequency as a function of the time (BarrettHand).

bending the flexible beam built into inner finger. This deformation is measured by the strain gauges.

Experimentation and Discussion. The BarrettHand was used as an end effector of a Stäubli RX90 robot for these experiments. In the same way that in the previous experiment, the contacted object was chosen to be a steel plate as shown in Fig. 10.b. The test was performed according to the following sequence: 1) the BarrettHand grasped the steel plate with two fingers; 2) the fingers were

slipped on the surface of the plate until they lost contact with the object. Fig. 11.a shows the gauge signal of one finger during the experiment. It is seen that the fingers slipped on the plate from $t = 5.50\ s$ to $t = 6.25\ s$.

Structural microvibrations of high frequencies caused by the slipping contact were detected with this sensor as we can see in Fig. 11.b. In the Barrett system the frequency of slipping vibration was detected at $375Hz$. The frequency graph of Fig. 11.b shows that this system has a lot of high frequency noise, however the detection method still works properly.

5.3 JR3 6-axis Force/Torque Sensor

Manipulator robots usually mount multi-axis force/torque sensors on the wrist that are very useful in manipulation tasks. Usually, these sensors have installed, like in previously described sensors, strain gauges to measure deformations in order to calculate the loads applied. We have used in this work the model 67M25A3 of JR3 (Fig. 12.) which is able to measure in Fx and Fy with a standard measurement range of $100N$ and a digital resolution $0,025N$ and in Fz with a standard measurement range of $200N$ and a digital resolution of $0.05N$.

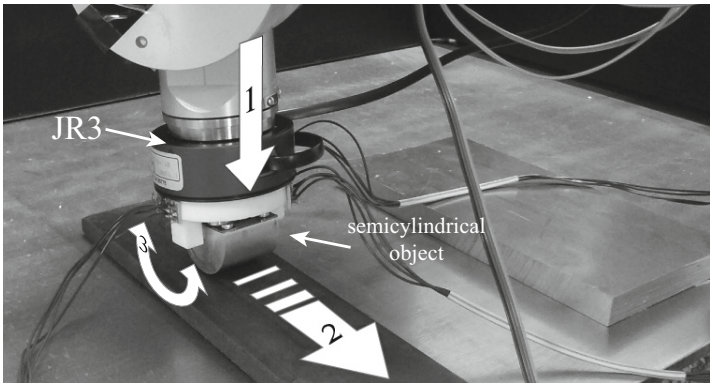
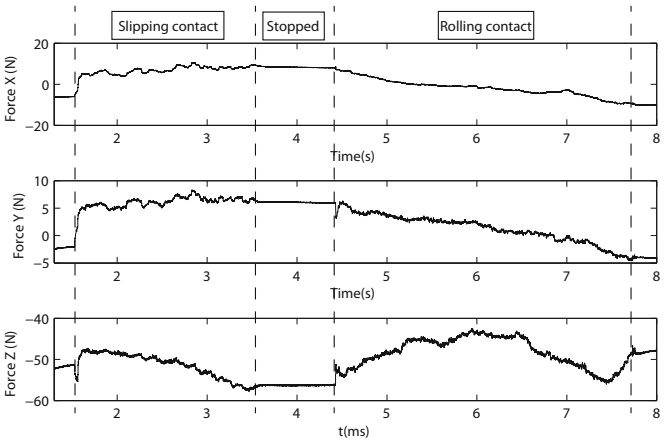
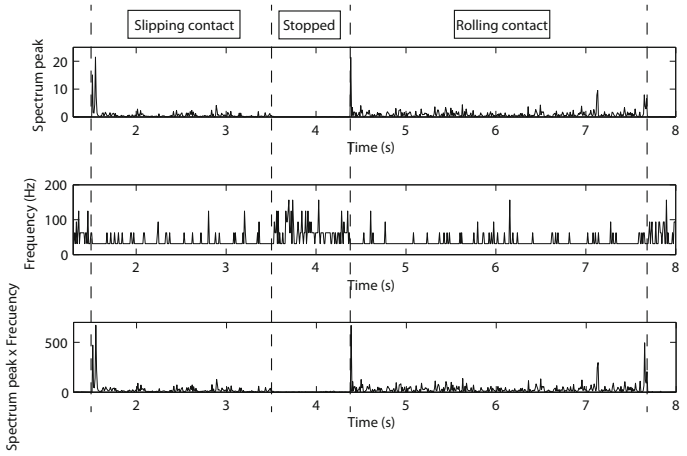


Fig. 12. Experimental setup with JR3

Experimentation and Discussion. This sensor was mounted on the wrist of a Stäubli RX90 robot. As shown in Fig. 12, it was necessary to add a semicylindrical object as an end-effector to perform the following experiment: 1) the semicylindrical object was moved toward the table until it came into contact with it; 2) the object was slipped on the surface of the table (slipping contact); 3) the robot rolled the semi-cylindrical surface of the sensor on the table (rolling contact). Fig. 13 a. shows the forces F_x, F_y and F_z, measured during the experiment. It is seen that the object came into contact with the table at $t = 0.82\ s$. From $t = 1.53\ s$ to $t = 3.55\ s$ a slipping contact occurred. From $t = 4.43\ s$ to



(a)



(b)

Fig. 13. a) Forces in sensor JR3 as a function of the time. b) Spectrum peak and its frequency on F_z as a function of the time.

$t = 7.75\ s$ the cylindrical surface of the object rolled on the table. Finally, the robot stopped its movement at $t = 7.75\ s$.

Fig. 13.b shows the spectrum analysis of signal F_z, which, based on its magnitude, was considered as the best one to detect slipping. However, structural microvibrations of high frequencies were not detected. Basically, this is because of the relative amplitude and frequencies of the vibration modes when a slip occurs, which is not different in the rolling contact case. We have concluded that this is due to the high rigidity of this type of sensors.

6 Conclusions

We tested the flexible fingers gripper to detect slipping between the contact surfaces. Slip detection highly contributes to the increase in the knowledge of what is occurring during contact, and highly improves manipulation [20].

We have presented a simple algorithm, based on DFT, that allow us to detect slipping with our flexible gripper in 64ms as presented in Section 4

In addition, we have analyzed the usefulness of our method in other sensors based on strain gauges. We have concluded that the algorithm works properly in two of the three sensors analyzed. As a consequence of this study we can say that the slip detection algorithm does not work properly for sensors in which rigidity is high, because the amplitude and frequencies of their vibration modes cannot be distinguishable in slipping or rolling contacts.

References

1. Allen, P.K., Miller, A.T., Oh, P.Y., Leibowitz, B.S.: Integration of vision, force and tactile sensing for grasping. In: Int. J. Intelligent Machines, Citeseer (1999)
2. Becedas, J., Payo, I., Feliu, V.: Generalized proportional integral torque control for single-link flexible manipulators. IET Control Theory and Applications 4(5), 773–783 (2010)
3. Becedas, J., Payo, I., Feliu, V.: Two-flexible-fingers gripper force feedback control system for its application as endeffector on a 6 dof manipulator. IEEE Trans. on Robotics 27(3), 599–615 (2011)
4. Benosman, M., Le Vey, G.: Control of flexible manipulators: a survey. Robotica 22(5), 533–545 (2004)
5. Bicchi, A.: Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity. IEEE Transactions on Robotics and Automation 16(6), 652–662 (2000)
6. Castellanos-Ramos, J., et al.: Tactile sensors based on conductive polymers. Microsystem Technologies 16(5), 765–776 (2010)
7. Choi, S., Lee, C.: Force tracking control of a flexible gripper driven by piezoceramic actuators. Journal of Dynamic Systems, Measurement, and Control 119(3), 439–446 (1997)
8. Cooley, J., Turkey, J.: An algorithm for the machine calculation of complex fourier series. Math. Comput. 19, 297–301 (1965)
9. da Silva, J., de Carvallo, A., da Silva, D.: A strain gauge tactile sensor for finger-mounted applications. IEEE Transactions on Instrumentation and Measurement 51(1), 18–22 (2002)
10. Dollar, A., Jentoft, L., Gao, J., Howe, R.: Contact sensing and grasping performance of compliant hands. Automous Robots 28, 65–75 (2010)
11. Dwivedy, S.K., Eberhard, P.: Dynamic analysis of flexible manipulators, a literature review. Mechanism and Machine Theory 41(7), 749–777 (2006)
12. Feliu, V.: Robots flexibles: hacia una nueva generación de robots con nuevas prestaciones. Revista Iberoamericana de Automática e Informática Industrial 3(3), 24–41 (2006)
13. Gunji et al., D.: Grasping force control of multi-fingered robot hand based on slip detection using tactile sensor. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2605–2610 (2008)

14. Hanafusa, H., Asada, H.: Stable prehension by a robot hand with elastic fingers. In: Proc. Seventh Int. Symp. Industrial Robots, pp. 361–368 (1977)
15. Hoffmann, K.: Applying the wheatstone bridge circuit. Hottinger Baldwin Messtechnik GmbH (2001)
16. Holweg, E., et al.: Slip detection by tactile sensors: algorithms and experimental results. In: Proceedings of the IEEE Conference on Robotics and Automation, pp. 3234–3239 (1996)
17. Hosoda, K., Tada, Y., Asada, M.: Internal representation of slip for a soft finger with vision and tactile sensors. In: Proceedings of the Int. Conference on Intelligent Robots and Systems, pp. 111–115 (2002)
18. Kawamura, T., et al.: Measurement of slip, force and deformation using hybrid tactile sensor system for robot hand gripping an object. International Journal of Advanced Robotic Systems 10(83), 1–8 (2012)
19. Kawasaki, H., Komatsu, T., Uchiyama, K.: Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii. IEEE Transactions on Mechatronics 7(3), 296–303 (2002)
20. Kawasaki, H., Komatsu, T., Uchiyama, K.: Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii. IEEE Transactions on Mechatronics 7(3), 296–303 (2002)
21. Lazzarini, R., Magni, R., Dario, P.: A tactile array sensor layered in an artificial skin. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, vol. 3, pp. 114–119 (1995)
22. Melchiorri, C.: Slip detection and control using tactile and force sensors. IEEE Trans. on Mechatronics 5(3), 235–243 (2000)
23. Odhner, L., Dollar, A.: Dexterous manipulation with underactuated elastic hands. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 5254–5260 (2011)
24. Payo, I., Feliu, V., Cortazar, O.D.: Force control of a very lightweight single-link flexible arm based on coupling torque feedback. Mechatronics 19, 6334–6347 (2009)
25. Pereira, E., Becedas, J., Payo, I., Ramos, F., Feliu, V.: Control of flexible manipulators. theory and practice. In: Jimenez, A., Al Hadithi, B.M. (eds.) Robot Manipulators Trends and Development, pp. 267–296. InTech (2010)
26. Shimoga, K.: Robot grasp synthesis algorithms: a survey. International Journal of Robotics Research 15, 230–266 (1996)
27. Tegin, J., Wikander, J.: Tactile sensing in intelligent robotic manipulation - a review. Industrial Robots 32, 64–70 (2005)
28. Tremblay, M.R., Cutkosky, M.R.: Estimating friction using incipient slip sensing during a manipulation task. In: Proceedings of the IEEE Conference on Robotics and Automation, pp. 429–434 (1993)
29. Yang, K., Gu, C.: A novel robot hand with embedded shape memory alloy actuators. In: Proc. Instn. Mech. Engrs. Part C: J. Mechanical Engineering Science (2002)

Feedback Linearizing Controller for a Flexible Single-Link Arm under Gravity and Joint Friction

Juan C. Cambera, José A. Chocoteco, and Vicente Feliu

Department of Electrical, Electronic and Automatic Engineering, University of Castilla-La Mancha, Ciudad Real, Spain
{JuanCarlos.Cambera,Vicente.Feliu}@uclm.es,
JoseAbel.Chocoteco1@alu.uclm.es

Abstract. This paper presents a control scheme for a flexible single-link arm moving under gravity. The purpose of this controller is the precise positioning of the arm tip in presence of link vibrations and joint friction. In order to achieve this, a two nested loop control is used: The inner loop is responsible for the motor positioning, and the outer loop suppresses the vibrations in the arm structure. On the inner loop design, a linear algebraic controller is considered. On the outer loop, an input/state linearization control is chosen. Simulation results are presented to demonstrate the performance of the controller.

Keywords: flexible link robot, non-linear control, feedback linearization.

1 Introduction

In the last three decades a considerable amount of literature has been published on flexible link robotics. This topic has drawn great deal of attention because of its promising advantages over the conventional rigid robotics manipulators. By using lighter materials and slender designs, a flexible link arm has potential advantages in building cost, operational speed, power consumption, maneuverability, operation security, among others [1]. Unfortunately, the same building considerations that bring all these benefits, introduce structural flexibility on robot arms. This property produces vibrations that reduce substantially the accuracy of robot movements. In order to deal with this drawback, important efforts have been carried out on sensor development, and on dynamic models and control strategies. But because of the complex nature of the problem a few important issues still remain open [2].

The first studies on flexible links robots were funded by a NASA research program in the seventies. The main goal was to build very large lightweight manipulators in order to reduce its launching cost [3]. Nowadays there is big interest in taking full advantage of flexible arms for earth applications. But, because of the initial motivation, most of the research done until now has been focused on

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_14,

the domain of flexible link robots moving in a gravity-free environment. Gravitational force effect is a major concern to take into account for a big part of the on-earth oriented applications.

In the earth scenario, there are some relevant progress in the use of flexible mechanical structures for manipulation. The presence of flexibility is associated with an inherent passive compliance. This feature, together with the lower inertias typical from this kind of manipulators, minimizes the potential risk of injury in task where humans and robots share the same workspace [4]. The KUKA LWR4 prototype is a relevant example of a robot designed based on this premise [5][1]. Another field of interest is the search and rescue robotic applications, where mobile robots with long manipulators are useful to interact with the environment. Here, a flexible manipulator is advisable as long as it reduces the power consumption and the overall weight of the system, two major constraints in mobile robotics (Gryphon robot is a good example of this, see [7]). In other areas, the same principles used in the flexible robotics have been effectively applied to suppress the vibration. This is particularly the case of the development of control algorithm for vibration reduction and improvement of the accuracy in bridge and gantry cranes(see [8]).

In the Industrial Engineering School of the University of Castilla-La Mancha we are developing a robotic system for inspection tasks. The system is composed of a very large and lightweight arm of 3 DOF placed on a relatively small mobile platform. This paper deals with the control of the first DOF of the system, which is a flexible arm moving vertically, and supporting a mass in its tip. The earliest efforts found towards solving this problem, used the classical rigid manipulator controllers to control an elastic manipulator [9]. Nowadays there are two main streams: the feedback and the feedforward approaches. The feedback approach still considers the classical controllers for rigid manipulators but additionally include an oscillation feedback to intentionally suppress the vibrations [10] [11]. In the feedforward strategy the command is shaped, so the excitation of natural frequencies is reduced [12]. In recent studies [13], both approaches- feedback and feedforward - are combined in order to take advantage of their respective strengths and minimize its weaknesses.

To solve this problem, in this paper we propose a two nested loop control structure, which is an approach that falls into the first category mentioned in the paragraph above. In this structure, the inner loop is responsible for the motor positioning, and the outer loop is in charge of canceling the arm vibration and tracking a tip point trajectory[2]. The inner loop control implements an algebraic controller capable of rejecting the frictional effects in the motor, and making

[1] Because the large inertias of the conventional industrial manipulator, collaborative work with humans is only allowed under restrictive conditions of velocity. Experiments presented in [6] explain the influence of the robot mass and velocity during blunt unconstrained impact

[2] The two nested loop scheme has been widely used in our research group as a general strategy for controlling flexible link arms(see for instance [14]). However, we have been constantly looking for new ways to implement the inner and the outer loop in accordance to the flexible link problem.

a good tracking of the reference. For the outer loop we suggest an input-state feedback linearization control. Some previous work using the two nested loop control structure in flexible robots under gravity can be found in [4] [15] [16].

The feedback linearization technique applied to flexible link robots has been extensively studied [17] [18] [19]. The idea behind this technique is to transform, by exact state transformation and feedback, the nonlinear system dynamics into a fully linear one, so that the linear control principles can be applied [20]. As long as flexible link arm dynamics of the end-effector has in the most general case a non-minimum phase nature, the feedback linearization principles required special considerations, and its contributions to the position tracking control problem are limited [17] [18]. In most of flexible link studies the tip position control have been solved under conditions of no Coulomb friction. Even though its effects lower the accuracy of the end-effector positioning in practice. In this paper, we combine the two nested loop control strategy with the feedback linearization techniques, in order to exploit its mutual strengths. The feedback linearization limitations do not affect our solution since we consider a minimum phase dynamics system (and not the most general case). By designing an inner loop we cancel the Coulomb friction effects in the motor positioning. While in the outer loop, we transform a non-linear problem in a linear one, by applying a feedback linearization.

This paper is organized as follows. Section 2 introduces briefly the dynamic model of the flexible arm. In Section 3, an overview of the control design is presented. Section 4 and Section 5 are devoted to the design of the inner and the outer loop controller. In Section 6, simulations of the control system are shown. Finally, Section 7 presents conclusions and future work.

2 Dynamic Model

For the purpose of the controller synthesis, the dynamic model of a flexible link robot is presented in two coupled sub-systems. The first subsystem represents the motor dynamics; the second one describes, strictly speaking, the flexible link dynamics.

2.1 DC Motor Model

In a DC motor the differential equation relating the armature current $i(t)$ to the motor position $\theta_m(t)$ is given by

$$k_{\mathrm{m}}\, n\, i(t) = K_{\mathrm{m}} i(t) = J\ddot{\theta}_{\mathrm{m}} + \nu\dot{\theta}_{\mathrm{m}}(t) + \Gamma_{\mathrm{st-coul}}(t) + \Gamma_{\mathrm{coup}}(t) \ . \tag{1}$$

where k_{m} represents the torque constant of the motor, n is the gear reduction, K_{m} represents the torque constant equivalent (motor+gear), J is the moment of inertia due to the motor rotor and the gearbox inertias, ν is the viscous friction coefficient, $\Gamma_{\mathrm{st-coul}}$ describes the static and Coulomb friction, Γ_{coup} is the coupling torque between the motor and the flexible link.

In relation to the static and Coulomb friction, a very basic model is considered. It is defined in terms of a constant (Γ_c) as follows

$$\Gamma_{st-coul} = \begin{cases} \Gamma_c \text{sgn}[\dot{\theta}_m(t)] & \text{if } \dot{\theta}_m \neq 0 \ . \\ K_m i(t) - \Gamma_{coup}(t) & \text{if } \dot{\theta}_m = 0 \text{ and } \|K_m i(t) - \Gamma_{coup}(t)\| < \Gamma_c \ . \\ \Gamma_c \text{sgn}[K_m i(t) - \Gamma_{coup}(t)] & \text{if } \dot{\theta}_m = 0 \text{ and } \|K_m i(t) - \Gamma_{coup}(t)\| > \Gamma_c \ . \end{cases} \quad (2)$$

2.2 Flexible Link Model

The flexible link dynamic model relates the motor position (θ_m) and the coupling torque (Γ_{coup}) to the tip position (θ_t). The following model considers a massless link[3] with a point mass situated in its tip. The movement of the arm is affected by the gravitational force. In this model, the sources of potential energy are the bending of the link and the gravitational force acting on the tip mass. The torsion and the compression effects of the link are not taken into account. Using the Euler-Bernoulli beam theory, as explained in [21], a dynamic model is found which is described by

$$c(\theta_m - \theta_t) = \Gamma_{coup} \ . \quad (3)$$

$$c(\theta_m - \theta_t) = ml^2 \frac{d^2\theta_t}{dt^2} + mgl\cos(\theta_t) \ . \quad (4)$$

where c is the stiffness of the arm constant, m is the tip mass, l is the link length and g is the gravitational constant. The rotational stiffness of the arm could be expressed in terms its length, its Young module (E) and its cross-sectional moment of inertia(I). This relation is given by

$$c = \frac{3EI}{l} \ . \quad (5)$$

As long as the flexible link dynamics is non-linear, the natural frequency varies based on the arm orientation. It can be approximately calculated by linearizing(4) for an arbitrary orientation, where the motor position and the tip position are θ_{m_x} and θ_{t_x}, respectively. The linearized equation can be written as

$$c(\bar{\theta}_m - \bar{\theta}_t) = ml^2 \frac{d^2\bar{\theta}_t}{dt^2} - mgl\sin(\theta_{t_x})\bar{\theta}_t \ . \quad (6)$$

where $\bar{\theta}_t = \theta_t - \theta_{t_x}$ and $\bar{\theta}_m = \theta_m - \theta_{m_x}$. This equation has the same structure of an oscillatory motion differential equation. Thus, the natural frequency is

$$\omega_n(\theta_{t_x}) = \sqrt{\frac{c - mgl\sin(\theta_{t_x})}{ml^2}} \ . \quad (7)$$

The schematic model of the flexible link is depicted in Fig.1.

[3] This assumption is valid as long as the tip mass is many times greater than link's mass. By doing this, the dynamic model only includes one vibrational mode.

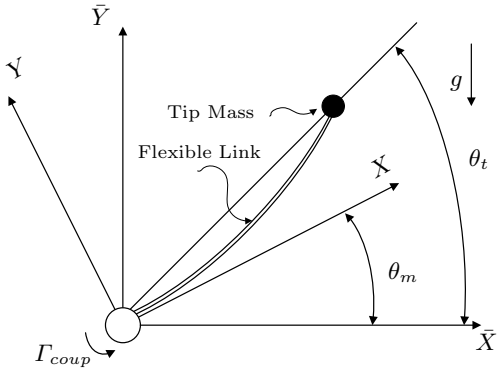


Fig. 1. Flexible Link Schematic Model

3 Control Design

Previous studies have shown that the Coulomb friction in the joints has important effects on the control of very lightweight flexible arms especially when they move at low speeds and accelerations [22]. The same studies indicate that the simplest method to reduce these effects, is to use high gain linear feedback to improve the robustness to Coulomb friction disturbances. However, as long as most of the flexible arms are non-minimum phase systems, this gain cannot be arbitrarily increased without making the system unstable [4]. In order to cope these drawbacks, in a wide sense, Feliu et al. proposed in 1989 to control a flexible link arm using a two nested loop control scheme [23]. The inner loop controls the motor position by using the error between the present and the commanded motor position; the outer loop controls the tip position by means of the error between the present and the desired tip position. The latter generates a position reference for the inner loop. By doing this, the control is divided in two simpler problems that can be solved independently, with their own constraints. The proposed control scheme is shown in Fig.2. The following sections show the design of the inner and the outer loop.

4 Inner Loop Design

On the inner loop, two relevant issues have to be taken into account in order to design a reliable control. The first issue is that the motor dynamics is not independent from the flexible link dynamics. A coupling torque compensation term solves this problem (see Sect.6 for more details). The second issue is the presence of nonlinearities and non modeled components of friction. For the sake

[4] The dynamic model presented in this paper corresponds to a minimum phase system. However, the same control scheme is still valid to effectively deal with the Coulomb friction problem.

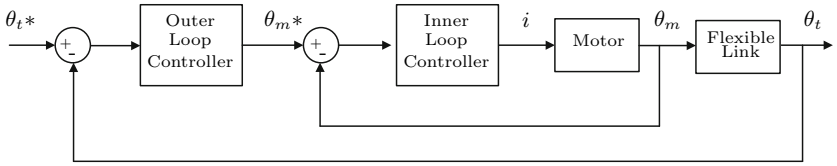


Fig. 2. General Control Scheme

of simplicity, and considering that in our model, static and Coulomb friction, takes the form of a step (see (2)), these terms are considered as a disturbance in the designing process. Then, the design of the inner loop controller can be done by using the classical linear control theory but considering an effective step disturbances rejection.

One of the most common controllers used to solve this problem is a PD with its derivative action in the feedback path. This controller solves the problem when no static and Coulomb disturbance is present. But if this were not the case, a steady state error would occur. In order to cancel it, there are two straightforward solutions. First, to introduce a feedforward compensation to cancel the static friction when the motion starts. And second, to consider an integrative action to eliminate the steady state error. Regarding the first solution, an effective feedforward compensation requires a exact modeling of the friction. This is very complex considering that static term is affected even by temperature changes. With respect to the second solution, it cancels the steady state error, but in return an overshoot shows up when the command signal changes abruptly (an step input, for instance). We have considered the algebraic controller shown in Fig.3. It has shown to be capable of tracking step inputs without any of the mentioned shortcomings.

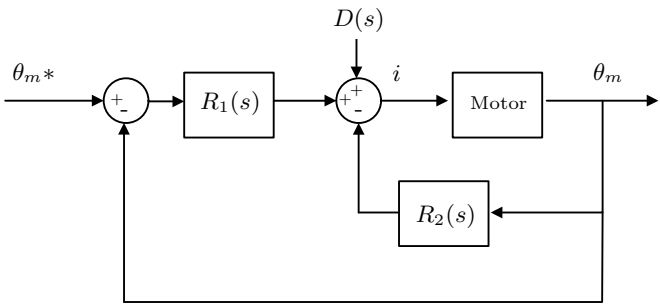


Fig. 3. Inner Loop Scheme

From the scheme above, the relationship between the motor position (θ_m) and the desired trajectory(θ_m^*) and the disturbance signal(D) is given by

$$\theta_m(s) = \frac{AR_1(s)}{s^2 + Bs + A(R_1(s) + R_2(s))}\theta_m^*(s) + \frac{A}{s^2 + Bs + A(R_1(s) + R_2(s))}D(s) \ . \tag{8}$$

where A, B are the motor parameters $A = K_m/J$ and $B = v/J$, and $R_1(s)$ and $R_2(s)$ can be expressed as

$$R_1(s) = \frac{\lambda_2 s^2 + \lambda_1 s + \lambda_0}{s(s+\mu)} \ . \tag{9}$$

$$R_2(s) = \frac{(k_2 - \lambda_2)s^2 + (k_1 - \lambda_1)s + (k_0 - \lambda_0)}{s(s+\mu)} \ . \tag{10}$$

The parameters k_1, k_2, k_3, λ_1, λ_2, λ_3 should be selected to cancel two zeros with two poles in $\frac{\theta_m(s)}{\theta_m^*(s)}$. So that the resulting transfer function has a structure similar to (11). The poles canceled still appear in the relation $\frac{\theta_m(s)}{D(s)}$, so they should be allocated in the same position that the others, in order to cancel the disturbance with the same speed as motor position tracking is performed.

$$\frac{\theta_m(s)}{\theta_m^*(s)} = \frac{1}{(\alpha s + 1)^2} \ . \tag{11}$$

These zero-pole cancellations reduce the dynamics between the motor position reference (θ_m*)and the tip position (θ_t) from 6th to 4th order. Furthermore, the absence of zeros in the inner loop makes possible the use of the feedback linearization technique in the outer loop, without considering any internal dynamics. This will be demonstrated in the next section. In short, this controller does not only guarantee a feasible tracking of motor position commands, but it also paves the way for the outer loop design.

5 Outer Loop Design

To control the tip position the feedback linearization technique is used. The design process is as follows. Let $\mathbf{x} \in \mathbb{R}^4$ be the state, defined by $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$, where $x_1 = \theta_t$, $x_2 = \dot{\theta}_t$, $x_3 = \theta_m$, $x_4 = \dot{\theta}_m$. And let u be the system input $u = \theta_m^*$. Then, the state equation of the system including the flexible link and the inner loop dynamics is

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u = \begin{bmatrix} x_2 \\ \frac{c}{ml^2}(x_3 - x_1) - \frac{g}{l}\cos(x_1) \\ x_4 \\ \frac{1}{\alpha^2}x_3 - \frac{2}{\alpha}x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\alpha^2} \end{bmatrix} u \ . \tag{12}$$

Let us assume that it is possible to design an input-state linearization control. It means, a feedback linearization controller with no internal dynamics associated. Then, it is necessary to check the controllability and the involutivity of the system [20]. The controllability matrix is defined in terms of *the Lie bracket of* f *and* g ($ad_f g$) and its repeated Lie brackets ($ad_f^i g$) as follows

$$\left[g, ad_f g, ad_f^2 g, ad_f^3 g\right] = \begin{bmatrix} 0 & 0 & 0 & -\frac{c}{\alpha^2 l^2 m} \\ 0 & 0 & \frac{c}{\alpha^2 l^2 m} & \frac{2c}{\alpha^3 l^2 m} \\ 0 & -\frac{1}{\alpha^2} & -\frac{2}{\alpha^3} & -\frac{3}{\alpha^4} \\ \frac{1}{\alpha^2} & \frac{2}{\alpha^3} & \frac{3}{\alpha^4} & \frac{4}{\alpha^5} \end{bmatrix} . \tag{13}$$

As long as it has full rank for $c > 0$ and m, l, $\alpha < \infty$, the system is controllable. The second condition is satisfied if the vector fields $\left[g, ad_f g, ad_f^2 g\right]$ form an involutive set. Because these vectors are constant, this condition is trivially satisfied. Therefore, the system showed in (12) is input-state linearizable.

The next step is finding out the state transformation $\mathbf{z}(\mathbf{x}) = [z_1, z_2, z_3, z_4]^T$ and the input transformation so that input/state linearization is achieved. In order to guarantee these properties the state transformation should satisfy the next set of constraint equations on z_1.

$$\nabla z_1 ad_{\mathbf{f}}^k \mathbf{g} = 0 \quad k = 0, 1, 2 \ . \tag{14}$$

$$\nabla z_1 ad_{\mathbf{f}}^3 \mathbf{g} \neq 0 \ . \tag{15}$$

These conditions are satisfied when $z_1 = x_1 = \theta_t$. The complete state transformation can be computed as

$$\mathbf{z} = \begin{bmatrix} z_1 \\ L_{\mathbf{f}} z_1 \\ L_{\mathbf{f}}^2 z_1 \\ L_{\mathbf{f}}^3 z_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \frac{c}{ml^2}(x_3 - x_1) - \frac{g}{l}\cos(x_1) \\ (-\frac{c}{ml^2} + \frac{g}{l}\sin(x_1))x_2 + (\frac{c}{ml^2})x_4 \end{bmatrix} . \tag{16}$$

This transformation leads to the next system

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ z_4 \\ (\frac{g}{l}{z_2}^2 - \frac{g}{l\alpha^2})\cos(z_1) + (\frac{2g}{l\alpha}{z_2}^2 + \frac{g}{l}z_3)\sin(z_1) + \\ -(\frac{c}{ml^2\alpha^2})z_1 - (\frac{2c}{ml^2\alpha})z_2 - (\frac{c}{ml^2} + \frac{1}{\alpha^2})z_3 - (\frac{2}{\alpha})z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{c}{ml^2\alpha^2} \end{bmatrix} u \ . \tag{17}$$

The nonlinear feedback control law is $u = \alpha(\mathbf{z}) + \beta(\mathbf{z})v$, where v is the new input of the linearized system, and $\alpha(\mathbf{z})$ and $\beta(\mathbf{z})$ are

$$\alpha(\mathbf{z}) = -\frac{{L_{\mathbf{f}}}^4 z_1}{L_{\mathbf{g}} {L_{\mathbf{f}}}^3 z_1} \ . \tag{18}$$

$$\beta(\mathbf{z}) = \frac{1}{L_{\mathbf{g}} {L_{\mathbf{f}}}^3 z_1} \ . \tag{19}$$

Thus, the input u can be written explicitly as

$$u = z_1 + (\frac{mgl}{c} - \frac{mgl\alpha^2}{c}{z_2}^2)\cos(z_1) - (\frac{2mgl\alpha}{c}z_2 + \frac{mgl\alpha^2}{c}z_3)\sin(z_1) + 2\alpha z_2 + (\alpha^2 + \frac{ml^2}{c})z_3 + (\frac{2ml^2\alpha}{c}z_4) + (\frac{ml^2\alpha^2}{c})v \ . \quad (20)$$

By substituting (20) into (17) the following linear system is obtained

$$\dot{\mathbf{z}} = \begin{bmatrix} 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v \ . \quad (21)$$

Considering that the link tip position $z_1 = x_1 = \theta_t$ should track a predefined trajectory z_1^*, the next linear control law is proposed

$$v = {z_1^*}^{(4)} - k_3({z_1}^{(3)} - {z_1^*}^{(3)}) - k_2({z_1}^{(2)} - {z_1^*}^{(2)}) - k_1(\dot{z_1} - \dot{z_1^*}) - k_0(z_1 - z_1^*) \ . \quad (22)$$

Denoting $e_1 = z_1 - z_1^*$, the tracking error dynamics can be expressed as

$${e_1}^{(4)} + k_3{e_1}^{(3)} + k_2{e_1}^{(2)} + k_1\dot{e_1} + k_0e_1 = 0 \ . \quad (23)$$

The above dynamics is exponentially stable if the positive constants k_i are chosen properly. A general diagram of the outer loop control scheme is shown in Fig.4. All the equations considered in the control algorithm are summarized on it. The states x_2 and x_4 are obtained from derivation of the signals x_1 and x_3, respectively.

In our experimental platform, this control law cannot be directly implemented as long as the full state $\mathbf{x}$ is not measured. However, it can be replaced with an estimation $\mathbf{x^e}$. The only measurements available are the motor position (θ_m) and the coupling torque (Γ_{coup}). Considering these signals and (3), an state estimator $\mathbf{x^e}$ can be computed as

$$\mathbf{x^e} = \begin{bmatrix} {x_1}^e \\ {x_2}^e \\ {x_3}^e \\ {x_4}^e \end{bmatrix} = \begin{bmatrix} \theta_{\mathrm{m}} - \frac{\Gamma_{\mathrm{coup}}}{c} \\ \dot{\theta}_{\mathrm{m}} - \frac{\dot{\Gamma}_{\mathrm{coup}}}{c} \\ \theta_{\mathrm{m}} \\ \dot{\theta}_{\mathrm{m}} \end{bmatrix} \ . \quad (24)$$

6 Identification

The control algorithm exposed in this paper was conceived to control a flexible arm platform built in our laboratory. Its mechanical structure consists mainly of a Maxon DC motor with gear reduction, and a duraluminium slender link attached to the gear shaft. The motor will be driven by a Maxon EPOS controller

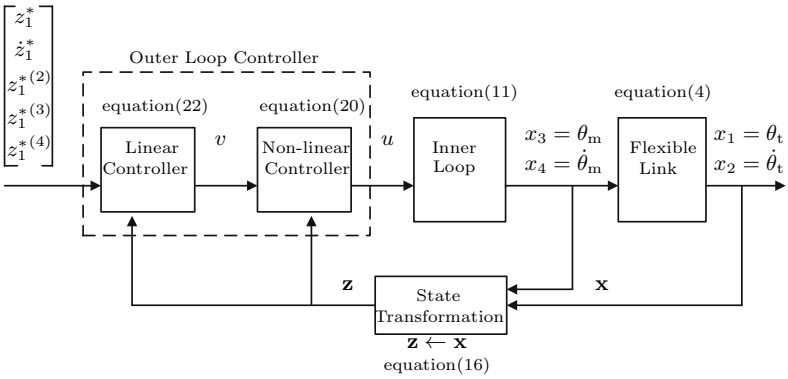


Fig. 4. Outer Loop Scheme

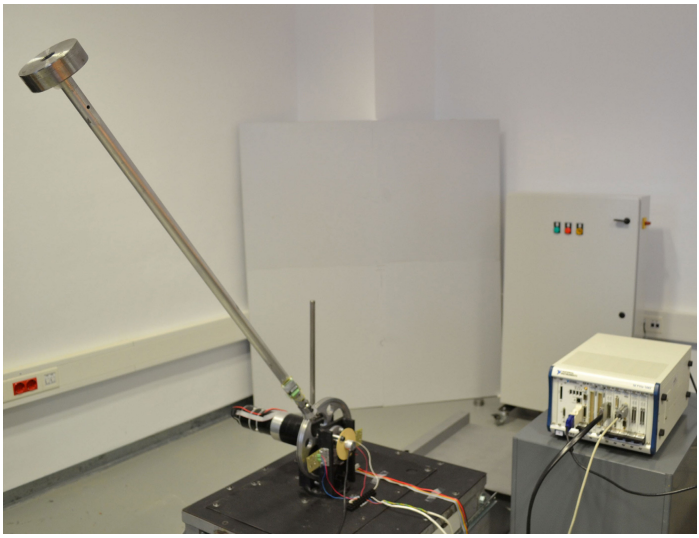


Fig. 5. Experimental platform

configured to work as a current amplifier, and the control law will be implemented in a National Instrument PXI system. The sensory system is composed by an encoder mounted on the motor shaft, and a strain-gauge structure placed at the root of the arm to measure the coupling torque. The platform is shown in Fig.5.

In addition to the obvious intentions behind a identification process, it was also useful to test some statement assumed to be true in the dynamic model. In the next paragraphs we briefly mention the procedure followed in the identification of the motor and flexible link parameters.

In the motor datasheets, the motor torque constant was the only parameter information useful for our control algorithm. The other important parameters, the viscous and the Coulomb friction constants and the rotor inertia, do not appear, or its values do not include all the desirable effects. The rotor inertia and the viscous friction of the motor were identified by adjusting the velocity response of the motor when a step current commands was applied. To identify the Coulomb friction constant, we measured the current from which the motor starts to move and then we multiplied this quantity by the motor torque constant equivalent. These experiments considered the motor with no load (no flexible arm attached). Figure 6 shows the steady state velocity at the gear shaft for a set of armature currents. In this figure, a dead zone-band can be noticed due to the static friction.

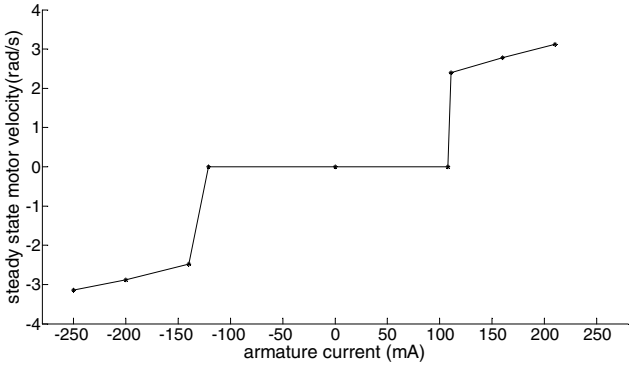


Fig. 6. System response(θ_t) with and without outer loop.

In relation to the flexible link, its length and its rotational stiffness were identified from the coupling torque estimation. It seems a little illogical to identify the arm´s length when it can be directly measured. But taking into account that the arm's root is attached to a mechanical part with no negligible dimensions, an equivalent length is necessary in order to improve the accuracy. The experiments were done as follows. The arm's root was fixed for a set of angles (θ_m), and then it was kindly hit with a hammer to stimulate its vibrational modes. From the resulting torque measurements, and applying the Fast Fourier Transform (FFT), the natural frequencies for every position were obtained. These values were fitted to the theoretical values given by (7). As an example, in Fig.7 we present the FFT of the torque measurement for $\theta_m = 40$ degrees . In this figure, only one vibrational mode stands out. This observation is consistent with the assumption done in the flexible link dynamic model, where only one vibrational mode was modeled by considering the whole mass to be concentrated in the arm's tip. Another parameter identified, was the coupling torque noise variance. It is a consequence of the noise signals on the strain-gauge measurements, and affects

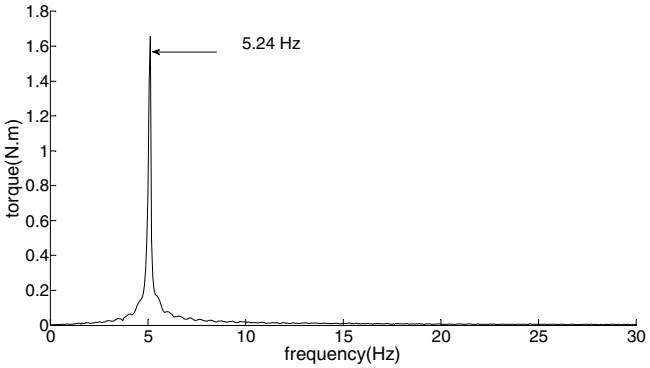


Fig. 7. FFT of the torque measurement for $\theta_m = 40$ degrees

negatively the full state estimation $\mathbf{x^e}$. Especially, the state ${x_2}^e$ that requires the first derivative of the coupling torque (see (24)).

The most relevant parameters of the platform are shown in Table.1.

Table 1. System parameters

Parameter	**Symbol**	**Value**
Motor torque constant equivalent	K_m	18.4140 $\frac{\text{N.m}}{\text{A}}$
Motor rotor inertia	J	1.2656 kg.m^2
Motor viscous friction	ν	2.320 $\frac{\text{N.m.s}}{\text{rad}}$
Gear reduction	n	93
Coulomb friction constant	Γ_c	2.136 N.m
Tip mass	m	2.27 kg
Arm mass	-	0.61 kg
Arm's length	l	0.96 m
Arm's rotational stiffness	c	2276.3 $\frac{\text{N.m}}{\text{rad}}$
Maximum arm's natural frequency	$\max[\omega_n]$	5.27 Hz
Minimum arm's natural frequency	$\min[\omega_n]$	5.23 Hz
Coupling torque noise variance	-	0.0165 N^{2}m^2

7 Simulations

Simulations were performed using *Matlab/Simulink* with a sampling time of T=0.002s. The inner loop poles were allocated in $-36.52\frac{\text{rad}}{\text{s}}$. These poles were placed there in order to achieve a motor response as fast as possible without reaching its saturation limit. The outer loop poles, for its part, were allocated in $-18\frac{\text{rad}}{\text{s}}$.

For the purpose of these simulations, the aim of the control algorithm was to make the system to track a fourth order trajectory from 0 to 75 degrees (approx. 1.3 rad) in 1 second. This fourth order trajectory was considered not only because its bounded fourth order derivative guarantees a bounded signal control (see (22) and (20)), but also because it can be easily designed to avoid surpassing the elastic limit constraints of the bar. Details on the generation of these fourth order trajectories applied to flexible link robots can be found in [24]. The coupling torque noise variance was considered in the simulations. The simulation results are presented below.

Figure 8 shows a comparative study of the behavior of the system with and without outer loop. With outer loop, the trajectory tracking was performed with little overshoot in the transient response, perfect vibration suppression and no steady state error in the tip response. On the other hand, when the inner loop is closed and the outer loop is open, the tracking performance of the commanded trajectory deteriorates. The residual vibration represents a positioning error of 1.6 cm approx.

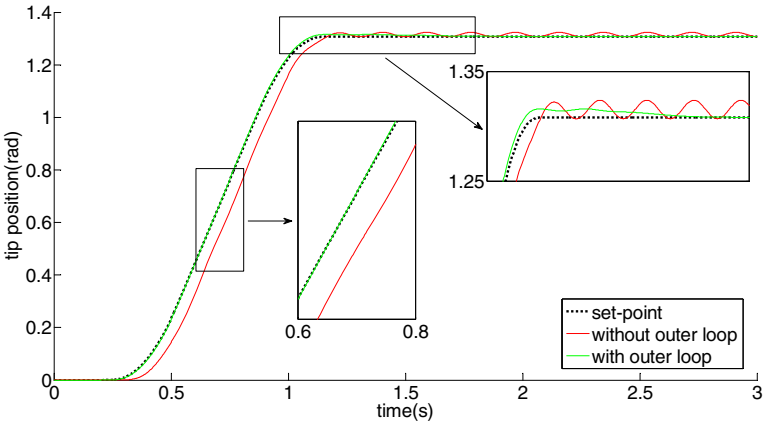


Fig. 8. System response(θ_t) with and without outer loop

Figure 9 shows the trajectory tracking of the tip point in presence of payload uncertainties. Here, the transient response presents a little overshoot when the payload mass is incremented. In steady state, an effective vibration suppression has been achieved. However, it presents a steady state error as result of a control law based on a nominal payload.

On a final note about the simulations, it is important to mention that the noise effect was completely negligible for the noise levels identified.

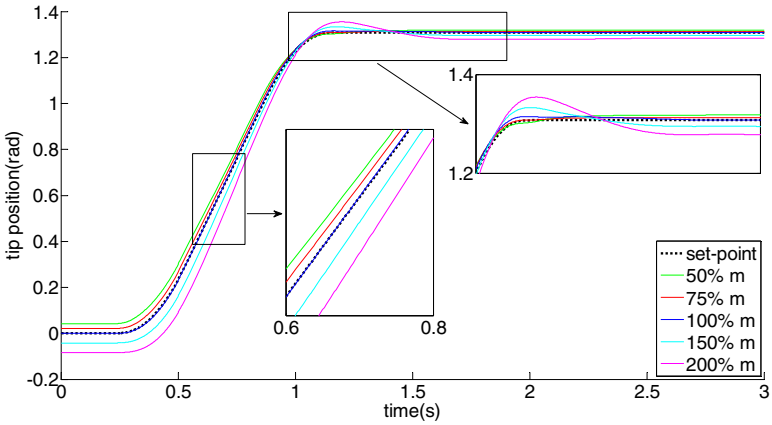


Fig. 9. System response(θ_t) to mass changes

8 Conclusions

A two nested controller has been designed for tracking tip trajectories of one-link flexible arm moving under gravity. In the inner loop, an algebraic controller was used in order to perform a reliable tracking of the commanded signal even in the presence of Coulomb friction disturbances. In the outer loop, the feedback linearization technique allowed us to transform the nonlinear system dynamics into a fully linear one. With this linearized dynamics, we easily set a linear control law to effectively suppress the arm vibration and track a tip point trajectory.

The resulting inner loop, has been intentionally designed to meet two conditions: it has to behave as a second order system and it must have no zeros. The former of these conditions reduces the order dynamics considered in the outer loop from 6th to 4th. The latter was a necessary condition for designing a controller base on input-state linearization. In other word, it prevents us from dealing with the problems associated with the internal dynamics. Those zeros and poles canceled were allocated in the exact position we designed for, and do not change as result of the outer loop design. To complete the control algorithm, we designed a full-order estimator from the encoder and the strain-gauge structure measurements. The estimator does not depend on the payload mass uncertainties, and is a simpler option than a nonlinear observer, for this specific case.

Simulation results have shown a good performance of the control under nominal conditions: perfect vibration cancellation with no steady state error and low overshoot. In the other hand, in presence of payload uncertainties, noticeable overshoot and steady state errors showed up, but vibrations were still effectively suppressed. Other simulations, not presented in this paper, have shown similar

results in presence of arm's stiffness constant uncertainties. Further work will focus on correcting these drawbacks by using a mass estimator and an adaptative control law. Experimental validation will be also carried out.

Acknowledgements. This study was financially supported by the Spanish Ministry of Economy and Competitiveness and by the European Social Fund with projects DPI2009-09956 and DPI2012-37062-CO2-01.

References

1. Dwivedy, S.K., Eberhard, P.: Dynamic analysis of flexible manipulators, a literature review. Journal of Mechanism and Machine Theory 41(7), 749–777 (2006)
2. Benosman, M., Le Vey, G.: Control of flexible manipulators: A survey. Robotica 22, 533–545 (2004)
3. Book, W.J.: Modeling, desing and control of flexible manipulators arms. PhD thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering (1974)
4. Malzahn, J., Phung, A.S., Hoffmann, F., Bertram, T.: Vibration control of a multi-flexible-link robot arm under gravity. In: 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1249–1254. IEEE (2011)
5. Bischoff, R.: From research to products: The development of the kuka light-weight robot. In: 40th International Symposium on Robotics, Barcelona, Spain (2009)
6. Haddadin, S., Albu-Schäffer, A., Hirzinger, G.: Approaching Asimov's 1st law: The impact of the robot's weight class. In: Robotics: Science and Systems Conference Workshop: Robot Manipulation: Sensing and Adapting the Real World (RSS 2007), Atlanta, USA (2007)
7. Freese, M., Matsuzawa, T., Oishi, Y., Debenest, P., Takita, K., Fukushima, E.F., Hirose, S.: Robotics-assisted demining with Gryphon. Advanced Robotics 21(15), 1763–1786 (2007)
8. Sorensen, K.L., Singhose, W., Dickerson, S.: A controller enabling precise positioning and sway reduction in bridge and gantry cranes. Control Engineering Practice 15(7), 825–837 (2007)
9. De Luca, A., Siciliano, B.: Regulation of flexible arms under gravity. IEEE Transactions on Robotics and Automation 9(4), 463–467 (1993)
10. Mansour, T., Jiang, X., Konno, A., Uchiyama, M.: Experimental verification on vibration suppression of a flexible manipulator using mpid controller. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 2896–2901. IEEE (2008)
11. Franke, R., Malzahn, J., Nierobisch, T., Hoffmann, F., Bertram, T.: Vibration control of a multi-link flexible robot arm with fiber-bragg-grating sensors. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 3365–3370. IEEE (2009)
12. Sarkar, P.K., Yamamoto, M., Mohri, A.: On the trajectory planning of a planar elastic manipulator under gravity. IEEE Transactions on Robotics and Automation 15(2), 357–362 (1999)
13. Malzahn, J., Ruderman, M., Phung, A.S., Hoffmann, F., Bertram, T.: Input shaping and strain gauge feedback vibration control of an elastic robotic arm. In: 2010 Conference on Control and Fault-Tolerant Systems (SysTol), pp. 672–677. IEEE (2010)

14. Feliu, V., Rattan, K.S., Brown Jr., H.B.: Adaptive control of a single-link flexible manipulator in the presence of joint friction and load changes. In: Proceedings of the 1989 IEEE International Conference on Robotics and Automation, pp. 1036–1041. IEEE (1989)
15. Feliu, V., García, A., Somolinos, J.: Gauge-based tip position control of a new three-degree-of-freedom flexible robot. The International Journal of Robotics Research 20(8), 660–675 (2001)
16. Berrio, C.F.C., García, F.J.C., Battle, V.F.: V. feedback linearization based control for trajectory tracking of a 2 degree of freedom whisker (5), 324–331 (2010)
17. De Luca, A., Siciliano, B.: Joint-based control of a nonlinear model of a flexible arm. In: American Control Conference, pp. 935–940. IEEE (1988)
18. Vandegrift, M., Lewis, F.L., Zhu, S.: Flexible-link robot arm control by a feedback linearization/singular perturbation approach. Journal of Robotic Systems 11(7), 591–603 (1994)
19. Wang, D., Vidyasagar, M.: Control of a class of manipulators with a single flexible link: Part i-feedback linearization. Journal of Dynamic Systems, Measurement, and Control 113(4), 655–661 (1991)
20. Slotine, J., Li, W.: Applied Nonlinear Control. Pearson Education (1991)
21. Feliu, V., Rattan, K.S., Brown, H.B.: Modeling and control of single-link flexible arms with lumped masses. Journal of Dynamic Systems, Measurement, and Control 114, 59–69 (1992)
22. Feliu, V., Somolinos, J.A., Cerrada, C., Cerrada, J.A.: A new control scheme of single-link flexible manipulators robust to payload changes. Journal of Intelligent and Robotic Systems 20(2-4), 349–373 (1997)
23. Feliu, V., Rattan, K., Brown, H.B.: Adaptive control of a single-link flexible manipulator in the presence of joint friction and load changes. In: Proceedings of the 1989 IEEE International Conference on Robotics and Automation, vol. 2, pp. 1036–1041 (1989)
24. Ramos, F., Feliu, V., Payo, I.: Design of Trajectories with Physical Constraints for very Lightweight Single Link Flexible Arms. Journal of Vibration and Control 14(8), 1091–1110 (2008)

Design of Modular Robot System for Maintenance Tasks in Hazardous Facilities and Environments

Prithvi Pagala, Manuel Ferre, and Manuel Armada

Centre for Automation and Robotics (UPM-CSIC)
Universidad Politécnica de Madrid, 28006 Madrid, Spain
{ps.pagala,m.ferre}@upm.es, manuel.armada@car.upm-csic.es
http://www.car.upm-csic.es/

Abstract. In the hazardous environments present in particle accelerators and nuclear facilities require maintenance and specific type of interventions. The maintenance task by the intervening personnel are planned to ensure personal safety but they are sometimes exposed to residual radiation. Hence inclusion of robotic systems into these environmental conditions are being encouraged to increase the availability of the facility and reduce personal radiation doses. However, the environment and tasks required present challenging conditions for robotic systems in terms of design, architecture, task planning and execution. This paper presents and addresses the hardware and software design choice of a modular robotic system as solution to overcome the challenges and presents a prototype heterogeneous modular robot system (SMART) . The simulation results of robot configurations overcoming the challenges is also presented.

Keywords: Modular robots, large science facilities, ionizing, radiation, fault tolerance, safety, hostile environments.

1 Introduction

Different conventional robots have established their use in various environments like automation and remote explorations. Focusing on the industrial and scientific facilities, industrial arms [11] and mobile teleoperated platforms [8] are among other conventional robots that have successfully performed predetermined tasks. Custom designed conventional robots [20] have been designed and deployed in hazardous environments [3] to perform missions. Some of the hazardous environments being deep sea missions, clean up of toxic waste, nuclear plant decommissioning and others. Despite the advantages shown by the conventional robots as single specialized robot system, they are limited by their fixed robot configuration and leads to limitation in extending capabilities of the robot. Conventional robots development time and cost for new platform are long and fixed [5] respectively in comparison to an adaptable system. Milti robot systems provide flexibility in performing a missions but there is an overhead on the operators to adapt and maintain the different robots.

Living organisms exhibit adaptation for the survival and when robot systems are able to adapt to the environment in performing tasks then larger possibilities for applications can be seen. If a robot system is able to evolve depending on the needs, then the necessity to develop new platforms is eliminated. Modularity in robot design is the ability

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_15,

of robot to separate and recombine into different system components [14] [7]. Modular robots explore in the direction of adaptability and flexibility in robot design [21], as they are composed of individual modules that can connect in different ways with each other to manifest different morphologies or robot configurations [15].

Modular robots are broadly classified into single basic modules (Homogeneous - e.g., M-TRAN [13]) and different basic modules (Heterogeneous systems - e.g., SMART [2]). The use of a few elements and identical modules provides the opportunity to replacing modules on failure and include redundancy for safety to perform a task increases. Fault tolerant design similar to the practice in space missions [1] can be applied by duplicating modules. Developing a modular robots has lower fabrication costs than developing different conventional robots that perform different tasks due to reuse of the modules to achieve different morphologies and functionality. The complexity of the system increases with the number of modules and therefore needs to be considered to ensure robust control and manipulation of the modular robot system. Different modules of the modular robot system allow the robot configuration to dynamically adapt to continue performing operations such as locomotion, navigating and executing the tasks in the environment, even after being impeded by obstacles.

Ionising radiation hazardous environments and workspace are present in large facilities like nuclear fission, nuclear fusion reactors, nuclear fuel processing and large scientific facilities which conduct experiments in particle physics using accelerators. At the large scientific facilities CERN (European Organization for Nuclear Research, Geneva) particle physics experiments are conducted and accelerator facilities are used. The accelerator beams interact with targets and other equipment in the path and activation occurs on prolonged exposure. The strength of activation depends on the length of exposure, strength of the beam, material properties and others. During the shutdown of the accelarator facility for maintenance or other interventions personnel entering the restricted sections are exposed to the residual radiation. Different measures to ensure the crew safety are implemented like, As Low As Reasonable Achievable (ALARA) are followed at CERN [19]. At CERN multiple accelerators are active to increase the beam energy through the tunnel network. As new experiment and processes are setup the residual radiation will increase with time. Hence deploying robots to perform the tasks in accelerator tunnels is preferred, provided that the platform achieves the safety and robustness required for deployment in such facilities.

In this paper we discuss the design of the hardware and software of a prototype of modular robot SMART. The challenges of the environment and tasks performed are presented in section 2. In Section 3 the design of the second iteration modular robot (SMART), which attempts to overcome these challenges both from the hardware and software architecture perspective is presented. Results are presented in section 3 and the applications of the modular robot and its simulator are presented in section 4. Future work and conclusions are presented in section 5.

2 Constraints and Task Requirement

For this paper the requirements and constraints are taken from the ISOLDE facility at CERN. ISOLDE is an isotope separator on-line (ISOL) radioactive ion beam facility

and has two industrial manipulators to exchange sources. At ISOLDE facility different radioactive species are produced at the target by the bombardment of high-energy proton beam. Figure 1 shows the 3D layout of the facility. The section highlighted in red is the location where two industrial manipulators are employed to exchange and store targets [12]. The need to find a larger and longer storage of the radiation emitting used targets is present. As the industrial robots are limited to moving along the rail on which they are mounted new robot solutions are being explored to transport the used targets to a different storage cell for longer storage. Like, mobile robots collaborating with the industrial manipulators. To solve the same problem we present modular robots as an option.

The modular robot solution needs to also consider being used for other facilities and equipment like collimator, beam splitters, etc. Apart from the industrial robot deployment, an overhead robotic system like TIM (Train inspection monorail) [9] and Mantis remote manipulators [18] are also actively working in the facility.

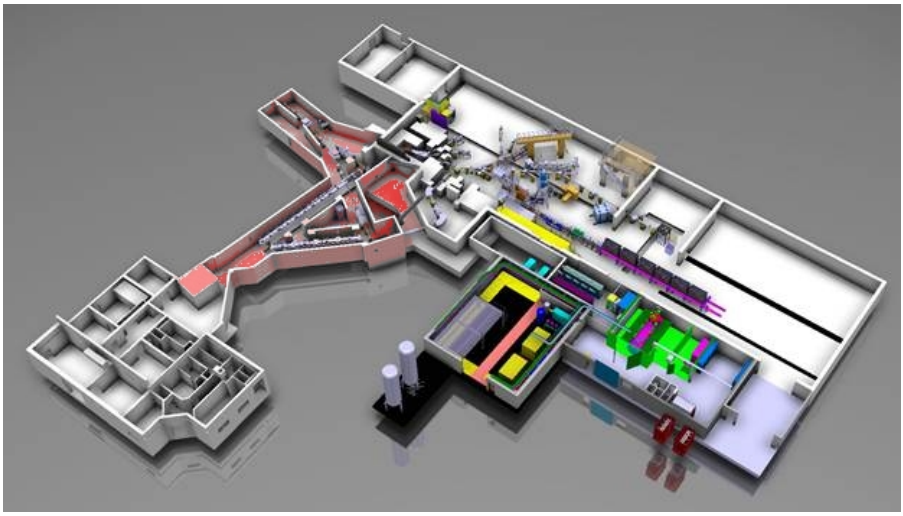


Fig. 1. Layout of the ISOLDE Facility [4]

2.1 Constraints

Generalizing the requirements, the new robot deployment employed should be flexible to collaborate with the existing remote handling strategies and industrial robots deployed in the facility by augmenting to their capabilities. Also, the new robot deployment should be capable of dynamically adjusting to the needs of task execution and have short development time which over time saves cost. During the course of operation unforeseen situations can be encountered and special tools and remote handling strategy may be needed, hence the robot architecture should support the same. Lastly increased use of the new robot platform to reduce human exposure to the hazardous environment. Hence, the new robot platform should be capable of performing basic preventive, corrective maintenance and measurements.

The previously mentioned requirements also have physical constraints which are as follows-

Hazardous environment due to the ionizing radiation which is fatal on prolonged exposure and causes single event upsets [6]. Therefore special need to protect the electronics need to be taken. *Safety* is a critical constraint which applies to both the equipment in the facility and deployed robots. The deployed robot should not cause harm to the facility or itself and the facility can not have major changes to accommodate robot deployment.

Hence, considering the above constraints it would be beneficial for the facility if robot deployment is a modular robot system. As, modular robots are more flexible than conventional robots and cut the development time of creating new conventional robots as new tasks arise. A heterogeneous modular robot system is suited more than homogeneous modular robot as requirements and the environment constraints force the robot to protect its electronics due to ionization dose and different types of actuation are required depending on the tasks. The advantages of such a deployment being, reduced down time of the facility, effective execution of the tasks and collaboration with existing robot solutions in the facility.

Tasks. The tasks are directed towards reducing the interaction of human beings with this hazardous environment and hence performing as many tasks as possible using the robot system would be ideal. The tasks have been simplified and split as follows:

The tasks are *remote inspection and manipulation* in the facilities to maintain the optimum functioning of the facility. The *remote inspection* can be radiation survey, visual inspection and others. This ensures reduction in human intervention during the planning phase of the shutdown. The *remote manipulations* during the intervention for planned tasks with specialised tools at different sections of the tunnel for long duration is also required. The locations can be at varying height and orientation.

Failure to comply with safety and robustness would require manual intervention and recovery which needs to be avoided at all costs and defeats the purpose of robot deployment.

3 Modular Robot System

SMART heterogeneous modular robot system composed of three types of modules that provide processing/control, movement, and specialised functionalities is presented. To justify the use of a heterogeneous modular system and to validate its compliance with the requirements the prototype and simulations of the system are presented.

3.1 Prototype

A heterogeneous modular robot system is chosen over a homogeneous system to satisfy the requirement to have different types of actuation and sensors with additional protection from ionising radiation to the electronics in the power and control module. The SMART robot's architecture is classified into three different type of modules based on functionality: *power and control modules* (P&C), *joint modules* (J) and *specialized*

modules (S). The modules are designed considering the need for partial or complete module upgrade in the future and therefore providing more flexibility. The heterogeneity in the design allows addition of new modules depending on new needs.

Power and Control Module is a 150 mm x 55 mm x 90 mm cuboid as shown in Figure 2. It is designed to decouple the electronics, communications and power source from the rest of the modules. The electronics is spread out rather than using high density chips to reduce the chances of single event upsets (SEUs) and accumulated dose due to radiation and also provides the opportunity for better shielding. It allows the use of additional computational units with the module and upgrade in the future as the technology improves. More P&C modules can be used in parallel to achieve fault tolerant processing similar to the practice in critical modules of space mission.

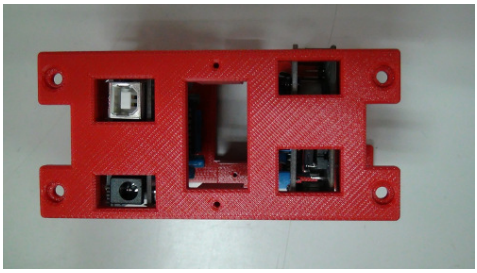
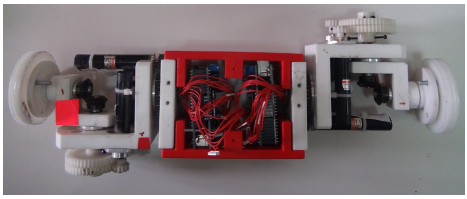


Fig. 2. Power and control (P&C) Module of SMART Robot

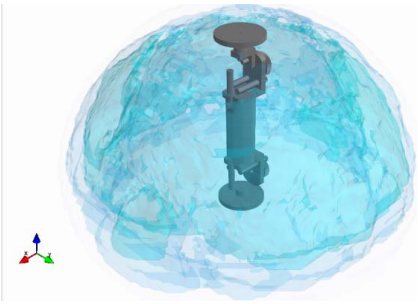
Joint Module is the actuation provider for the robot configuration depending on the type of actuation required it can be linear, rotary, spherical, etc with multiple degrees of freedom (DoF) and with the necessary actuation requirement of torque and velocity. In the prototype , three rotational DoF module is used. This module has been reused from the previous SMART [2] modular robot system which is a 120 mmx 55mm x90 mm cuboid. The design of the J module allows S or P&C modules to be attached and detached depending on the task performed. It is mandatory to have a P&C module attached to a J module for the functioning of the modules.

Figure 3(a) shows the prototype which uses two J modules modules along with a P&C module and two wheel S modules to form a wheeled robot configuration. Figure 3 shows the kinematic reachability of the robot when fixed vertically like an arm manipulator. Various S modules can be attached to the joint module like gripper, radiation probe, camera etc.

Specialized Module are described as the different tools, end-effector and platforms (base, power) of the robot. It may be a an attachment to help locomotion like wheel (W), a sensor device like radiation measurement, a tool (e.g gripper), an accessory (e.g power platform). This type of module can be attached to a P-C module, J module or another S module. The S module provides more flexibility and diversity to locomotion and manipulation tasks. They can be developed to be task specific.



(a) Prototype wheeled robot



(b) Kinematic reachability

Fig. 3. Modular Robot Configuration

The design approach is towards increased reusability and cost effectiveness in various robot configuration and individual module usage. The ability to reconfigure the functions of the modules helps the robot in planning and adjusting to tasks, environments and positions. Examples being remote inspection of a magnet console or radiation sampling at different locations and beam heights with varying restrictions of angle or position of interaction.

3.2 Hardware and Software Design

Modularity provided with the implementation of modules is extended and exploited by the software design to improve fault detection and tolerance. The fault tolerance is improved in software design by using concurrent processes. Hardware failure of modules on occurrence need to managed by reconfiguration of the robot configuration. Example when master P&C module fails then the next P&C module in priority is able to take over the functionality. It is capable of reconfiguring the robot configuration and the failed module for possible recovery. As mentioned above, to ensure easy upgrade of the P&C module and ionising tolerance it uses controllers with spread circuit and not system on chip solution, wired communication was choose over wireless for the same reason. The module transmits and receives data over a two wire serial communication. The P&C module is designed to work with two wire system, among the solutions of CAN bus, two wire and serial communication protocol, the two wire communication is used between the modules in the prototype. It is designed so that it can be easily changed to wireless communication if the need arises.

The figure 4 shows the hardware and software design adopted for the prototype of modular robot system. It consists of a backbone platform of ROS (robot operating system) [17] which provides middle level interface between robot actuators, sensors and others with the processing nodes through publishing and subscribing to topics. It makes it possible to easily implement high level controllers and planners for the different modules of the robot. Many ROS tools that help in optimising (command line tools- CTL)

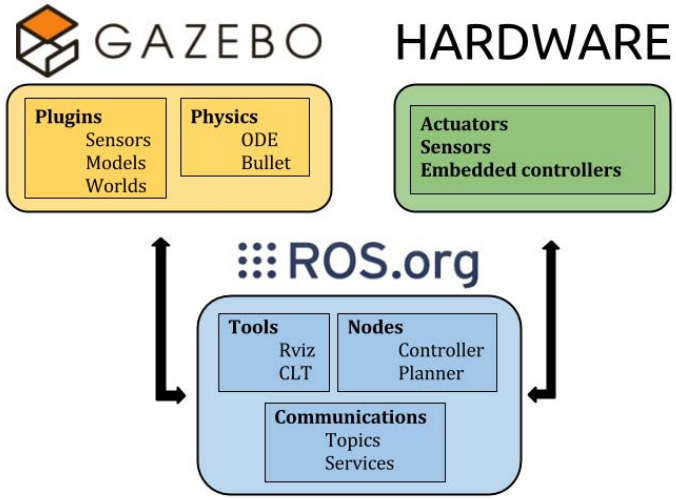


Fig. 4. Hardware and Software Architecture

the robot function and robot visualisation (Rviz) are used. This architecture extends the modularity in robot configuration with modularity in the software architecture (figure 4). Figure also depicts the capability of connecting the three dimension Gazebo simulator [10], where sensors plugins , robot dynamics and physics can be tested.

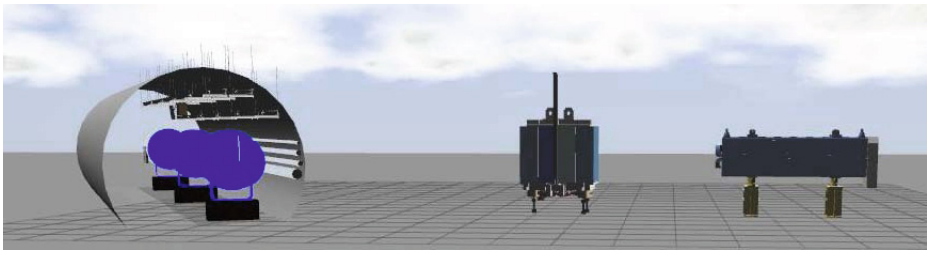


Fig. 5. CAD Models Imported into the Simulation Environment for Testing

Gazebo simulator is capable of generating realistic sensor feedback and rigid-body physics. The modelling of a modular robot system is complex due to its capability of changing and forming different robot configurations with a set of modules. The simulation models for SMART are obtained through the characterization of the geometric and kinematic model of each module of the system. The resulting simulated configurations exhibit the same kinematic parameters as the prototype of the SMART robot. It supports importing computer aided designs (CAD) of the remote facility workspace into the simulator, figure 5 shows imports of three different CAD models from the accelerator facility into the simulator. This provides realistic environment to test and perform the tasks.

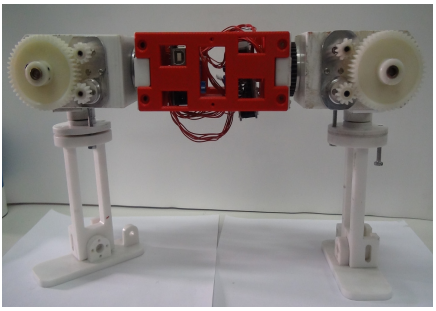
The prototype presents the proof of working of the system and the actual deployment would use radiation hardened equipment using the same architecture.

3.3 Robot Configurations and Simulation

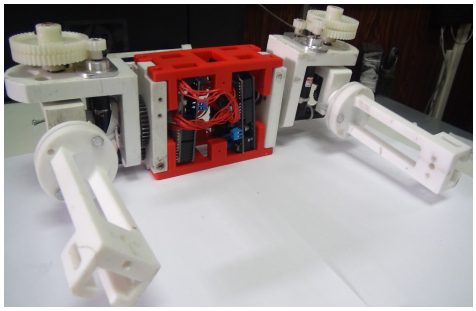
Various robot configurations can be achieved using the basic modules. Three different robot configurations are presented using a P&C module, two J modules and different S modules. They are wheeled robot configuration using wheel S module (figure 3(a)), legged robot configuration using leg S module (figure 6(a)) and probe robot configuration using probe S module (figure 6(b)).

In the robot configuration two J modules determine the different degrees of freedom for actuation which can be between one and three for the prototype J modules. Hence, the robot configurations can have upto 6 DOF in total to achieve locomotion and orientation. Having multiple joint modules it would be essential to manage the different actuation.

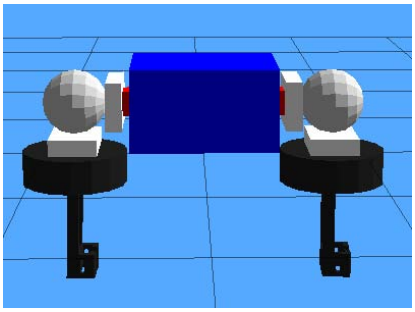
Model of wheeled robot configuration featuring different functionalities that can be achieved by this robot configuration are shown in figure 7. Figure 7(a) shows the wheel rotation which if using one DOF would be able to perform differential wheeled loco-



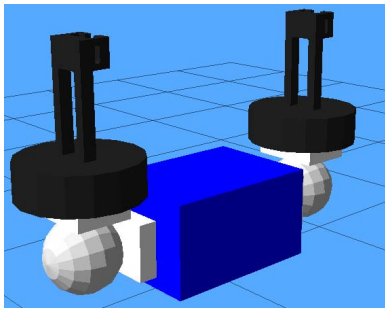
(a) Legged robot configurations



(b) Probe robot configurations

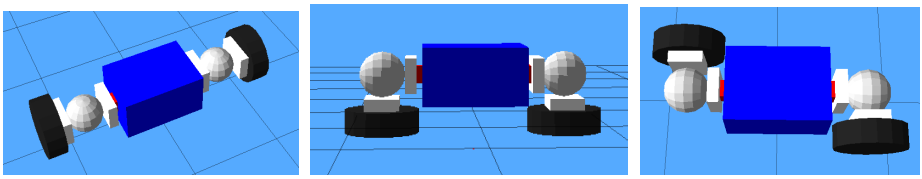


(c) Simulation - Legged robot



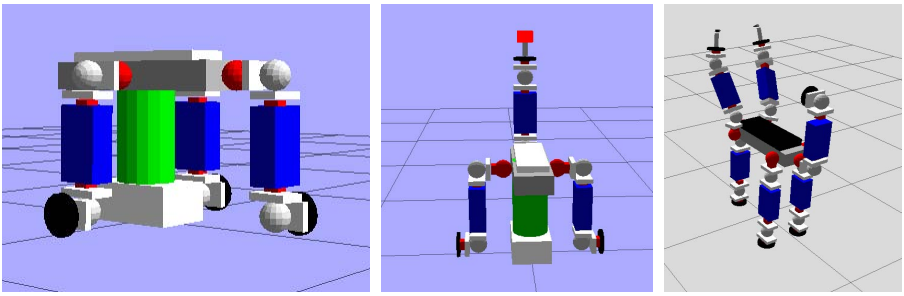
(d) Simulation - Probe robot

Fig. 6. Prototype and Simulation SMART Robot Configurations



(a) Wheel rotation (b) Joint rotation (c) Tangential movement

Fig. 7. Simulation model - Wheeled configuration using 3 DOF J module



(a) Locomotion with power base (b) Radiation inspection with power base (c) Locomotion and manipulation configuration

Fig. 8. Robot Configuration with and without power base

motion. It can perform differential drive and tangential motion when using two DOF J modules as seen in figure 7(c). Figure 7(b) shows the third joint rotation of the robot to be able to lift the robot configuration onto the horizontal wheel S modules. Considering the requirement of reaching *target location* which may require both legged and wheeled locomotion to reach destination figure 8 shows some possibilities of the same. Wheeled locomotion is advantageous locomotion on flat surfaces and legged locomotion for negotiating steps. LMRC (locomotion or manipulation robot configuration) is the robot configuration consisting of two joint modules, a power and control module and various specialised modules. Figure 8(c) shows a robot configuration which is able to exhibit wheeled and legged locomotion along with manipulation by using six LMRC connected with an specialised base plate module. Three of the six LMRC are seen to act as supports, while the two of the other three are used for manipulation task and last one is used to provide better understating of the environment using specialised modules like camera, radiation probe etc. Figure 8(a) and 8(b) show the use of base plate connected with three LMRC and mounted onto a power base S module. The power base s module provides the power for execution of remote maintenance tasks for long duration (minimum of four working hours) and also acts as a support for the robot configuration when performing manipulation with the LMRC. The power base S module doesn't provide actuation and is equipped with spring loaded ball transfer unit on the floor surface to provide least friction during locomotion and its also used to ensure no tipping of the robot occurs during the manipulation tasks. In figure 8(a) it can be seen that all the

LMRC are used for locomotion while in figure 8(b) one of the LMRC is used with radiation probe to acquire the radiation information at beamline. All the the LHRC are capable of manipulation and locomotion. After performing the task the power bases are stored along with other modules in safe zones away from the radiation behind concrete shield in each section near the access gates. Hence the used power base can be docked in the charging station after use.

The requirement of *custom tools* necessary to perform a task is addressed with the use of tool case in base plate S module and the power base S module. The base module has a restriction to the number of tools it can carry depending on the size, while the power base module holds section specific tools. The custom tools are individual S modules or a combination of many S modules e.g. camera, lighting, gripper and special section dependent tools. Long *endurance* for situations where manipulation and inspection needs to be performed for long hours is made possible with the help of power base S module that provides minimum of 4 hours of working time and also functions as a support for the Robot configuration.

The SMART robot prototype along with simulations validate the hardware and software architecture and the robot design. Although the platform presents many advantages, intuitive human machine interface is required to be provided to the operator. The execution of the planned tasks in simulation and mock-ups provides information to optimise the task execution through reduction of redundant movements and different control strategies.

4 Applications

The applications of the heterogeneous modular robot system SMART Hardware and the simulator are in remote manipulation, inspection, training and emergency response.

The modular robot is able to adapt to different accelerators and their constraints to perform remote *manipulation* and *inspection*. As various accelerators beam lines are at different heights and locations the modular robot system is capable to performing inspection and survey immediately after shutdown to provide information for task planning and reduce downtime of the facility. *Collaboration* with the other robot deployments and between different modular robot configuration would increase the flexibility of task execution.

The simulator is useful to develop new possibilities and configurations of the robot and test them. It also provides a platform to *train* operators during the deployment of the modular robot. Data logs from the robot deployment task execution and mock-up task execution can be used to provide more realistic training to new operators. The simulator also provides the opportunity to add computer aided design models of the remote environments to provide realistic environment.

4.1 Advantages

The traditional advantages of a modular robot system over conventional robot like adaptability, lower cost, etc are provided. The SMART modular robot design also

provides easier upgrade and replacement of modules, like the power and control module upgrade with advancement in fault tolerant electronics and better power source. Also additional higher computational modules can be added to the P&C module to provide higher level control and execution of advanced algorithms for autonomy.

Optimisation and Planning. The task execution of the different robot configurations can be planned and optimised using the simulator and during the execution of tasks in the mockup of the workspace. This approach would not only increase the optimisation and planning of the modular robot movement [16] by reducing redundant movement but also in reducing and distributing the overall radiation dose taken by different modules. The different approaches can be tests in the simulator and data form mock-up trials can be used to optimise the task execution.

Fault Tolerance and Detection. It can be achieved by combining both the hardware and software approach. Duplicating (having redundant modules) the hardware, similar to what is done in space crafts [22] to ensure high degree of safety by having multiple units performing the same task and cross validating each other to detect faults is the hardware approach. While implementing concurrent processes in software, along with checking with task execution model is the software approach. Fault detection can also be achieved by comparing the data logged during the task execution in the mockup and simulation (task execution model) versus the execution in the remote location. Extension of this process would be to create respective models for the different tasks depending on robot configuration, with which it is possible to detect and predict faults. In the case of failure in a module, example if an actuator in the joint module fails then the other modules compensate for the lost degrees of freedom by reconfiguring their functions.

5 Conclusions and Future Work

The prototype modular robot system presented overcome the requirements constraints considered for the design. The design choices used for the hardware and software extend towards robust robot system through fault detection and recovery. Also, the software and hardware architecture is validated using the prototype. The scale of the modular system and materials used for a deployment would depend on the requirements. Various robot configurations have been evaluated in simulation and results presented showing the possibility of deployment of a modular robot in an ionizing radiation facility to do basic maintenance and reduce the downtime of the facility. The SMART robot prototype is able become a platform to test and validate the safety and robustness of the modular robots use in facilities. Future work would be focussed on testing the robustness and safety of the modular robot system along with strategies to manage complexity due to addition of modules, faults, multiple degrees of freedom and human machine interface. In the future, there could be fault tolerant, robust and safe modular robots that can avoid human intervention in hazardous environments along with offering adaptability and cost effectiveness during task execution .

References

1. Aviziens, A.: Fault-tolerant systems. IEEE Transactions on Computers 100(12), 1304–1312 (1976)
2. Baca, J., Ferre, M., Aracil, R., Campos, A.: A Modular Robot System Design and Control Motion Modes for Locomotion and Manipulation Tasks. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2010)
3. Briones, L., Bustamante, P., Serna, M.: Wall-climbing robot for inspection in nuclear power plants. In: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pp. 1409–1414 (1994), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=351292
4. CERN: HIE-ISOLDE Project, http://hie-isolde.web.cern.ch/hie-isolde/
5. Cohen, R., Lipton, M.G., Dai, M.Q., Benhabib, B.: Conceptual Design of a Modular Robot. ASME Journal of Mechanical Design 114(1), 117–125 (1992), http://link.aip.org/link/?JMD/114/117/1
6. Dodd, P.E., Massengill, L.W.: Basic mechanisms and modeling of single-event upset in digital microelectronics. IEEE Transactions on Nuclear Science 50(3), 583–602 (2003)
7. Fukuda, T., Buss, M., Hosokai, H., Kawauchi, Y.: Cell Structured Robotic System CEBOT - Control, Planning and Communication Methods. In: An International Conference on Intelligent Autonomous Systems 2, pp. 661–671. IOS Press, Amsterdam (1989)
8. Horne, R.A., Lohmann, K.D., Coull, L., Coin, A.Y., Therville, A., Lips, R., Desrozier, M.: H.K.: MANTIS: a compact mobile remote handling system for accelerator halls and tunnels. ANS Meeting on Remote Systems and Robotics in Hostile Environments 30(CONF-7811109-) (1978)
9. Kershaw, K., Chapron, F., Coin, A., Delsaux, F., Feniet, T., Grenard, J.L., Valbuena, R.: Remote inspection, measurement and handling for LHC. In: 2007 IEEE Particle Accelerator Conference (PAC) pp. 332–334 (2007), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4440202
10. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), vol. 3, pp. 2149–2154. IEEE (2149), http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1389727
11. Kugler, E., Fiander, D., Johnson, B., Haas, H., Przewloka, A., Ravn, H.L., Simon, D.J., Zimmer, K.: The new CERN-ISOLDE on-line mass-separator facility at the PS-Booster. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms 70(1), 41–49 (1992)
12. Kugler, E.: The ISOLDE facility. Hyperfine Interactions 129(1-4), 23–42 (2000)
13. Kurokawa, H., Yoshida, E., Tomita, K.: Self-reconfigurable M-TRAN structures and walker generation. Robotics and Autonomous Systems 54(2), 142–149 (2006), http://linkinghub.elsevier.com/retrieve/pii/S0921889005001533
14. Kusiak, A., Huang, C.C.: Development of modular products. IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part A 19(4), 523–538 (1996)
15. Matsumaru, T.: Design and control of the modular robot system: TOMMS. In: Proceedings of the 1995 IEEE International Conference on Robotics and Automation, vol. 2, pp. 2125–2131. IEEE (1995)
16. Pamecha, A., Ebert-Uphoff, I., Chirikjian, G.: Useful metrics for modular robot motion planning. IEEE Transactions on Robotics and Automation 13(4), 531–545 (1997), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=611311

17. Quigley, M., Gerkey, B.: ROS: an open-source Robot Operating System. ....on open source ...(Figure 1) (2009), http://pub1.willowgarage.com/ konolige/cs225B/docs/quigley-icra2009-ros.pdf
18. Therville, A., Hunt, K., Horne, R.A., Lips, R., Desrozier, M., Lohmann, K.D., Coin, A., Coull, L.: MANTIS 2 a new long range remote vehicle and servo-master-slave manipulator for the CERN accelerator complex. Tech. rep
19. Vignes-Magno, A.: Radiation Protection Safety Code, Geneva (2006), https://edms.cern.ch/file/335729/LAST_RELEASED/F_E.pdf
20. Yamamoto, S., Electric, T., Company, P.: Development of inspection robot for nuclear power plant. In: Proceedings of the 1992 IEEE International Conference on Robotics and Automation (1992), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=220030
21. Yim, M., Shen, W.M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S.: Modular Self-Reconfigurable Robot Systems [Grand Challenges of Robotics]. IEEE Robotics & Automation Magazine 14(1), 43–52 (2007)
22. Zhuo-hua, D., Zi-xing, C., Jin-xia, Y.: Fault diagnosis and fault tolerant control for wheeled mobile robots under unknown environments: A survey. In: IEEE International Conference on Robotics and Automation. pp. 3428–3433 (April 2005), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1570640

Improving the Motion of a Sensing Antenna by Using an Input Shaping Technique

Daniel Feliu Talegon, Claudia F. Castillo, and Vicente Feliu Batlle

Department of Electrical, Electronic and Automatic
Engineering, University of Castilla-La Mancha, Ciudad Real, Spain
www.uclm.es

Abstract. This work proposes an open loop control for driving a flexible-link based sensor. The control strategy is based on an (IS) Input Shaping technique in order to reduce link vibrations and includes two algebraic controllers to deal with precise motor positioning, as well as minimize the high non-linear motor friction. The antenna performs free azimuthal and vertical movements as part of a first driving stage when moving the antenna fast and accurately, before collision and while searching for precise areas of objects. The vertical movement is clearly non-linear due to the gravity effect. Some experiments illustrate that the controllers provided significant results driving the motors accurately and reducing large part of the flexible-link vibration.

Keywords: Input shaper, flexible link, algebraic controller, antenna sensor.

1 Introduction

The study on tactile systems offers a promise in using artificial flexible-links for a range of different tasks, including object sorting navigation in occluded environments, as well as obtaining the three-dimensional tactile information, such as the shape and texture of nearby surfaces, as it has been explained by [1]. There are several optical sensing methods available but none of those methods can replicate the sense of touch that is lost when performing a specific task. A tactile system is independent of illumination and could work in special conditions including fog, darkness, glare and environments with reflections. The sensor as a mechanism is divided into two main parts: the sensing element, such as a load-cell or a (F-T) force and torque sensor and the flexible-link placed on the top of the sensing element.

Some flexible links for active touch have been built, for instance, [4], [5], [6], [7], [8], among others. In the works of [4], [5], and [7] the contact point was obtained from processing the vibrations that appeared in the mechanical structure as consequence of the impact. Other works, e.g. [6] and [8], ignored the dynamics completely and made use of the elasticity equation of the link static deflection. Subsequently, the contact point was obtained from angle and force measurements by kinematic considerations. All these previous studies model the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_16,

system dynamics for obtaining the contacted point when the link touches an object, but not for designing or planning the antenna free motion and its active vibration damping. The antenna control without vibration damping could have difficulties in approaching special points accurate and fast enough. The link could also collides with the object permanently which constrains the sensor acuity.

In the work by [8], a whisker is controlled via a (PID) Proportional Integral Derivative feedback law that follows a sinusoidal trajectory despite the disturbance torques from the contacted object. The torque and shear force alternate with clockwise and counter clockwise sweeps and the control system minimizes transients while ensuring reasonably accurate tracking. In the research done by [7], the experimental platform uses a (RL) Repetitive Learning controller that moves a whisker across an object while regulating the vertical contact force using feedback of hub moment and encoder measurements. The work by [9] presents a bio-mimetic tactile sensor and a movement strategy based on free single movements of the link. Some control methods tried to control the active movement of a robotic antenna by modelling the antenna assuming a lumped masses model [10] and cancelling the flexible-link vibrations by means of the inversion of the system dynamics [11].

Previous research activity on sensing antennae describes how low control latency of the link movement after a contact constrains the sensory acuity [9]. An antenna active motion strategy should include the free motion for placing the link fast and precisely while in turn reducing vibrations due to every movement or after a contact. It minimises the time for every manoeuvre when searching an object. Subsequently, a contact algorithm for obtaining the collision instant should switch from using the antenna position control to force control. Finally, the contact point and the force direction are obtained when performing the force control mode which keeps the contact in order to get the right information from the object. Other ways, the link vibration of every movement before and after a contact would make the sensing task high time consuming and the collected information somewhat vague. This work is devoted to the cope with the first stage of the antenna free motion control. In this first stage, it has to perform two control tasks: motor positioning and tip positioning trying to get rid of the permanent oscillation. The flexible-link of our platform is very flexible and the dynamics existing between the motor torques and the tip position is non-minimum phase.

Two high gain closed loops are needed for controlling the each joint position and compensating the friction. Therefore, two algebraic controllers were designed following the methodology described in [17]. The antenna link vibration modes cannot easily be removed because: 1) the link vibration frequencies are high, 2) the limited motor bandwidth, and 3) The use of a relatively cheap real-time computer system with low sampling rate. This last issue led us to instead apply open loop control techniques.

This work proposes an open loop control strategy which is based on an (IS) Input Shaping technique to reduce the vibrations of the flexible link, as it has been explained in the works of [12] and [13]. The IS technique consists in

generating a command reference that does not excite the vibration modes of the link. It generates command references for the motors by passing the desired tip trajectory through a finite impulse filter. The main problem for the IS technique is dealing with system uncertainties produced by model errors or changes in the system parameters. In order to solve it, robust, learning and adaptive approaches have been proposed, such as in [14] and [15]. IS techniques are only suited for linear time-invariant systems while our antenna is highly non-linear. Then, this research aims to study the robust IS for reducing the antenna tip vibrations. This paper is organized as follows. Section 2 presents our antenna prototype and its dynamic model. Section 3 develops the proposed motion control system. Experimentation of the control system is described in Section 4, and Section 5 presents some conclusions.

2 The Experimental Antenna

The proposed sensing antenna consists of a flexible-link, a sensor holder (load-cell) where the antenna is attached to two DC mini servo actuators PMA-5A motor sets (from Harmonic drive) which includes reduction gears and incremental encoders. The setup is shown in Figure 1. The system runs under Microsoft Windows XP 2002, INTEL Core (TM) 2 Quad CPU, Q8200 2.33 GHz, 3 GB of RAM. The data acquisition and control algorithms are programmed using Labview 7.1. System simulations, identification and comparisons are performed using Simulink/MATLAB R2010a. The system data acquisition (measurements, the control signals and written data) sampling time is Ts=0.004 s.

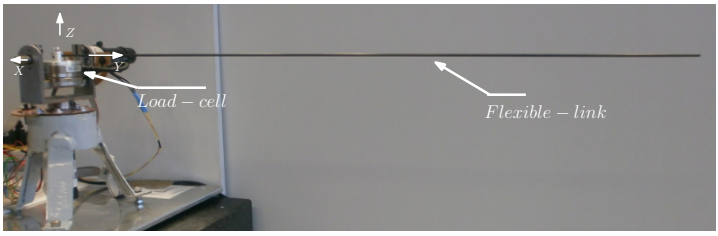


Fig. 1. Experimental Platform

2.1 Antenna Features

The workspace of the (2DOF) two Degrees of Freedom antenna is spherical and constituted with azimuthal and attitude motions. The link length has been determined by the desired workspace dimension, see Figure 2. The tip position is expressed in spherical coordinates θ_t and ϕ_t, motor angles are θ_m and ϕ_m, being θ and ϕ azimuthal and attitude angles, respectively. The antenna features are listed in Table 1, where E is the Young module, I is the inertial moment due to the flexible link cross section, g is the gravity constant, r is the link radius, m is the link mass and l is the length of the flexible link.

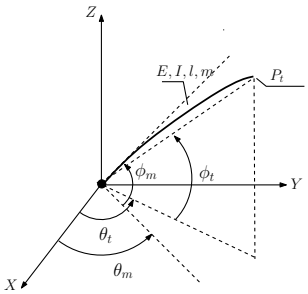


Fig. 2. Flexible-link schematic Model

Table 1. Flexible-link Characteristics

Feature	Quantity	Unit
l	1	m
r	0.001	m
m	4.7×10^{-3}	kg
E	1.15×10^{11}	$\frac{\mathrm{N}}{\mathrm{m}^2}$

2.2 DC Motor Dynamics

The dynamics of the two motors are given by the following differential equations:

$$\hat{\Gamma}_\theta = \hat{K}_\theta V_\theta = \hat{J}_\theta \ddot{\hat{\theta}}_m + \hat{\nu}_\theta \dot{\hat{\theta}}_m + \hat{\Gamma}_\theta^{nlc} + \hat{\Gamma}_\theta^{coup} \tag{1}$$

$$\hat{\Gamma}_\phi = \hat{K}_\phi V_\phi = \hat{J}_\phi \ddot{\hat{\phi}}_m + \hat{\nu}_\phi \dot{\hat{\phi}}_m + \hat{\Gamma}_\phi^{nlc} + \hat{\Gamma}_\phi^{coup} \ . \tag{2}$$

In these equations, the voltages V_θ and V_ϕ are the control signals. As it is assumed that both motors are endowed with servo amplifiers of very fast dynamics, the currents of the motors, and then the motor torques $\hat{\Gamma}_\theta$ and $\hat{\Gamma}_\phi$ are assumed to be proportional to the previous control signals. $\hat{K}_\theta$ and $\hat{K}_\phi$ are the motor constants that define such proportionality. $\hat{\nu}_\theta$ and $\hat{\nu}_\phi$ are the corresponding viscous friction coefficients, and $\hat{J}_\theta$ and $\hat{J}_\phi$ are the motor rotational inertias. $\hat{\Gamma}_\theta^{coup}$ and $\hat{\Gamma}_\phi^{coup}$ are the coupling torques between the motors and the link. $\hat{\Gamma}_\theta^{nlc}$ and $\hat{\Gamma}_\phi^{nlc}$ are the Coulomb friction terms which are non-linear. Both motors have gears with a reduction ratio n. Variables and parameters with upper hat are referred to the motor side of the gears. The same variables and parameters without the upper hat are referred to the link side of the gears. For example, the conversion between the angles of the azimuthal motor is given by $\hat{\theta}_m = n\theta_m$, and the conversion between the torques of this same motor is $\hat{\Gamma}_\theta = \Gamma_\theta / n$. The motor parameters are given in Table 2.

Table 2. Actuator Identification

Motor	$\hat{K}$	$\hat{J}$	$\hat{\nu}$	$\frac{\hat{T}_{Coul}}{\hat{K}}$	Saturation
Parameters	$\frac{\text{Nm}}{\text{V}}$	kgm^2	Nms	V	V
ϕ_m	0.003	1.85×10^{-7}	2.85×10^{-6}	0.42	1.2
θ_m	0.003	6.18×10^{-7}	3.04×10^{-6}	0.48	1.2

2.3 Link Vibration Frequencies

The flexible-link dynamics has been modelled assuming distributed masses where the resulting infinite high-order dynamics is truncated yielding reduced order models. Other models are based on spatial discretization by means of finite elements or lumped masses. Some of these works can be found in [3], [2] and [16]. An interesting feature of the IS technique proposed here to reduce motion vibrations is that it does not need the complete dynamic model. Only the knowledge of the vibration frequencies of the link are required in order to properly design an IS.

The vibration frequencies of a flexible-link with distributed mass and without tip payload, which rotates in an horizontal plane (without gravity effects), can be obtained as the solutions of the equation (e.g. [2]):

$$c \cdot sh - s \cdot ch + \frac{J}{\rho} \cdot \beta_i^3 \cdot (1 + c \cdot ch) = 0 \tag{3}$$

where J is the rotational motor inertia referred to the link side ($\hat{J}n^2$), ρ is the linear mass density, $c = \cos(\beta \cdot l)$, $s = \sin(\beta \cdot l)$, $ch = \cosh(\beta \cdot l)$ and $sh = \sinh(\beta \cdot l)$. Every solution β_i, $1 \leq i < \infty$ of the previous characteristic equation yields an associated vibration frequency:

$$\omega_i = \beta_i^2 \sqrt{\frac{E \cdot I}{\rho}} \tag{4}$$

which corresponds to a link with a pinned end. Vibration frequencies of a link with a clamped end can also be obtained from (3) by taking its limit if $J \to \infty$. In this case, the equation becomes $c \cdot ch = -1$ and combined with (4) yields the required angular frequencies.

Simulation of the movements considering the flexible-link parameters of Table 1 showed that:

- Only the vibration modes associated to the two lowest frequencies (ω_1 and ω_2) were noticeable.
- The mode of the lowest frequency (ω_1) was much more significant in the time response of the tip than the mode of the second frequency (ω_2).
- These two above items are valid for the flexible-link with one pinned end and the flexible link with one clamped end.
- In the case of attitude angular motion, the vibration frequencies slightly change in function of the angle ϕ_t, because of the non-linear effect of the

gravity (it induces a distributed moment of the link with respect to the base which can be approximated by $\int_0^l \rho \cdot g \cdot \cos(\phi_t) \cdot x \cdot dx$.

3 System Control

As it was mentioned in the Introduction, the control system has two components. The first one consists of the closed loop control of the two motor positions (angles), and the second one is the open loop control of the tip position. Both components are developed in the following subsections.

3.1 Motor Control

PID controllers with a low pass filter term ensure good trajectory tracking, compensate disturbances such as unmodelled components of the friction, and are robust to parameter uncertainties, providing precise and fast motor positioning responses. An algebraic methodology is used to tune the parameters of these controllers. Throughout this section we denote a particular joint by the index $'i'$ according to the first column of Table 3.

Using (1) and (2), considering that Coulomb friction is a step like disturbance that can be compensated by using the closed-loop control, and considering that the coupling torques are insignificant (due to the high gear-box ratio n), the following transfer function from the input voltage to the motor angular position is found as

$$\hat{G}_{m,i}(s) = \frac{\hat{K}_{\theta,i}}{s \cdot (\hat{J}_i \cdot s + \hat{\nu}_{n,i})} = \frac{\hat{g}_{n,i}(s)}{\hat{g}_{d,i}(s)} \ , \tag{5}$$

where $\hat{g}_{n,i}$ and $\hat{g}_{d,i}$ are the numerator and denominator polynomials of $\hat{G}_{m,i}(s)$.

A two degrees of freedom PID controller is used, of the form shown in Figure 3. This controller can be easily implemented and it also provides a robust closed loop system behaviour for tracking the desired trajectory. The closed loop poles can be arbitrarily placed following the algebraic methodology presented in [17]. We decided to place all the closed loop poles in the same location p_i, as shown in Table 3.

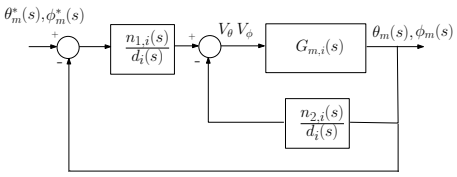


Fig. 3. Algebraic Controller

Table 3. Control Parameters

Joints i	*Poles* p_i	$a_{0,i}$	$a_{1,i}$	$a_{2,i}$	$b_{0,i}$	$b_{1,i}$	$b_{2,i}$	$c_{a,i}$
ϕ	-70	148210	4234	30.2	0	4234	126	264
θ	-60	267110	8903	74.9	0	8903	347	235

The closed-loop transfer function $M_i(s)$ from the actual output $\theta_m(s)$, $\phi_m(s)$ to the input reference $\theta_m^*(s)$, $\phi_m^*(s)$ (angles seen from the link side of the gear) is written as follows:

$$M_i(s) = \frac{G_{m,i}(s)n_{1,i}(s)}{d_i(s) + G_{m,i}(s)(n_{1,i}(s) + n_{2,i}(s))} \quad , \tag{6}$$

where $G_{m,i}(s)$ is the motor transfer function (5) and $n_{1,i}(s)$, $n_{2,i}(s)$ and $d_i(s)$ are polynomial expressions of the controller transfer functions. The tuning equation of the controller can be expressed as:

$$d_i(s) \cdot g_{d,i}(s) + g_{n,i}(s) \cdot (n_{1,i}(s) + n_{2,i}(s)) = (s - p_i)^4. \tag{7}$$

being $g_{n,i}$ and $g_{d,i}$ the numerator and denominator polynomials of $G_{m,i}(s)$. In order to guarantee a zero steady state error to step commands and disturbances at the motor input (Coulomb friction), and an appropriate pole placement, the algebraic controller is written as:

$$n_{1,i}(s) = a_{2,i}s^2 + a_{1,i}s + a_{0,i}, \tag{8}$$

$$n_{2,i}(s) = b_{2,i}s^2 + b_{1,i}s + b_{0,i} \tag{9}$$

and

$$d_i(s) = s(s + c_{a,i}). \tag{10}$$

These terms yield a closed loop transfer function with a denominator of four order. Then four poles have to be placed in the location p_i, justifying the order of the term expressed in the right hand side of equation (7).

Combining equations (5)-(10), yields 4 conditions to tune 7 parameters. Other 3 conditions are obtained from designing the numerator of $M_i(s)$, i.e., the polynomial $n_{1,i}(s)$. We choose to design a $n_{1,i}$ such that the gain of $M_i(s)$ becomes the unity, and its two zeros are cancelled with two of the poles of $M_i(s)$. Then 7 linear equations are obtained from which controller parameters $a_{2,i}$, $a_{1,i}$, $a_{0,i}$, $b_{2,i}$, $b_{1,i}$, $b_{0,i}$ and $c_{a,i}$ can be easily calculated. The parameters of the controllers of the two motors after such algebraic tuning are given in Table 3, and the resulting closed loop transfer functions of the motors are:

$$M_i(s) = \frac{1}{(1 + \alpha_i \cdot s)^2}, \quad \alpha_i = -p_i^{-1}. \tag{11}$$

3.2 IS Technique

The IS is a well known feed-forward technique used to reduce vibrations caused by movements of flexible structures. This technique basically removes the frequency components that could excite the vibration modes from the reference signal: it acts as a bandstop filter at the vibration frequencies of the link. Details about the foundations and refinements of this technique can be found, e.g. in [12], [13], [14], [15]. The IS control scheme that we propose for our antenna is represented in Figure 4, where the motor controlled dynamics has been included. The scheme shows that the input to our IS is the desired trajectory for the tip and it provides an output that is used as reference to the motor. A multi-variable IS controller is used. It is formed by two mono-variable terms IS^{θ} and IS^{ϕ} which are responsible of removing the vibration in the θ and ϕ movements.

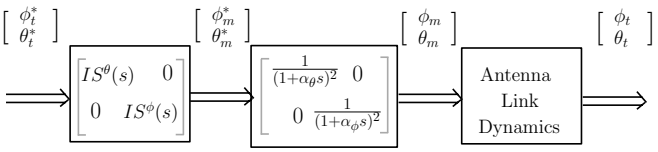


Fig. 4. Control Diagram

Taking into account the results provided at the end of Sec.2.3, we consider three configurations for the IS:

$$IS_1(s) = \frac{1 + e^{-\frac{\pi}{\omega_1} \cdot s}}{2}. \tag{12}$$

It removes only the first vibration angular frequency ω_1 from the reference,

$$IS_2(s) = \frac{1 + e^{-\frac{\pi}{\omega_1} \cdot s}}{2} \cdot \frac{1 + e^{-\frac{\pi}{\omega_2} \cdot s}}{2} \tag{13}$$

which simultaneously removes the first and second vibration angular frequencies ω_1 and ω_2 from the reference. Finally, the following IS

$$IS_3(s) = \left(\frac{1 + e^{-\frac{\pi}{\omega_1} \cdot s}}{2} \right)^2 \tag{14}$$

removes the first vibration angular frequency ω_1, and is robust to slight variations of this frequency around its nominal value. The use of the last IS is basically motivated by the fact stated in Sec.2.3, since the vibration frequencies slightly change with the attitude angle in the vertical movement.

Figure 5 shows the frequency responses of these three filters, which expresses the sensitivity function of each IS. The vertical axis is the non-dimensional vibration amplitude, calculated from the ratio of the shaped amplitude of vibration to the unshaped amplitude vibration. It expresses the residual vibration

as a percentage. The horizontal axis is the normalized frequency obtained by dividing the actual frequency of the system ω by the first angular frequency of the model ω_1. The level of vibration tolerance was defined for our application 5% which is also shown in Figure 5.

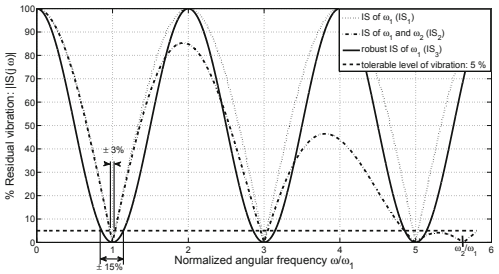


Fig. 5. Sensitivity functions of the IS

Note that for the sensitivity curves of IS_1 and IS_2 shown in Figure 5, as the system frequency deviates from the design frequency (ω_1 in our case), the amount of residual vibration increases rapidly. This figure also shows that IS_3 is a shaper more robust to modelling errors. IS_3 is synthesized by forcing the derivative of the IS frequency response at the modelled frequency ω_1 to be zero. This constraint of the derivative tends to keep the vibration at a low level, even when there are modelling errors. It can be expressed as $IS_3(s) = (IS_1(s))^q$, being q an integer larger than 1. The simplest robust IS_3 has been chosen for this application, which is obtained making $q = 2$. This shaper is usually denoted as Zero Vibration and Derivative (ZVD).

Figure 5 shows that the allowed range by the IS_1 is $\pm 3\%$ considering the error in the modelled first vibration frequency - range in which the residual vibration is under the tolerance level -. Then IS_1 is not appropriate for our device, because the first vibration frequency may experience variations larger than this range in the vertical movements due to the non-linear effect of the gravity. This figure also shows that the allowed range by the proposed IS_3 is $\pm 15\%$ considering the error in the modelled first vibration frequency.

4 Experimental Results

The parameters of the motor transfer functions $G_{m,i}(s)$ were identified by performing experiments and using the Parameter Estimation toolbox of Matlab / Simulink. The identification results are shown in Table 2. The closed loop poles and the calculated coefficients of the controller transfer functions are presented in Table 3. The poles have been chosen considering the servo actuators saturation limit (maximum allowable control signal before saturation Cs=1.2 V). In order to obtain the critically damped response, four coincident poles have been

Table 4. Trajectories (Parabolic Curves)

Case	rad
ϕ_t^*	$0 \rightarrow 0.7854$
θ_t^*	$0 \rightarrow 0.7854$
Trajectory time	1.2 s
Experiment time	4 s

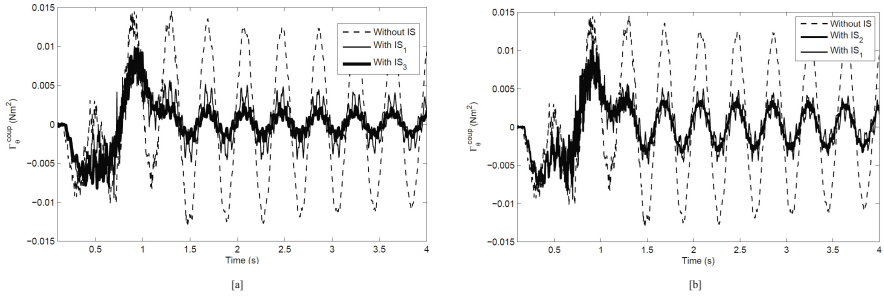


Fig. 6. Torque of azimuthal movement, system without IS compared to the system with: (a) IS_1 and IS_3; (b) IS_1 and IS_2

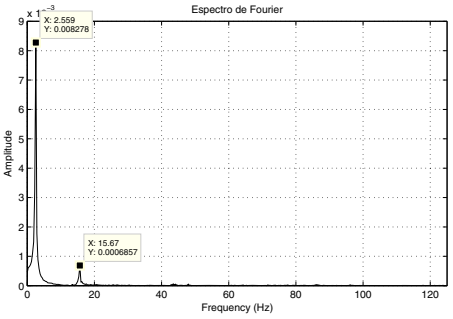


Fig. 7. Frequency spectrum, azimuthal case

placed as far as possible of the imaginary axis of the complex S-plane, taking into account the sampling time constraints $T_s = 0.004s$. The reference trajectories are parabolic curves of a second order which are designed to meet the needs for driving flexible links. Every manoeuvre has been designed to neither produce large link deflections, or reach the motor saturation or the flexible link mechanical constraints (see Table 4).

Figure 6 shows the moment at the base of the link (coupling torque) when an azimuthal manoeuvre is carried out (θ_t^* in Table 4). It illustrates the following cases using only motor control and motor control combined with one of the IS_1, IS_2 and IS_3 techniques. Figure 6a compares the system performance without IS to the system with IS_1 or IS_3. It shows that IS_3 reduces the vibration amplitude

about one fourth of its value when it is compared to the system without IS. Figure 6b compare the system performance without IS to the system with IS_1 or IS_2. This figure shows that both IS reduce the vibration amplitude about one third of its value when it is compared to the system without IS. The IS_2 reduces the vibration associated to the second frequency, effectively. However, the IS_1 does not reduce the vibration of the second frequency.

Figure 7 shows the Fourier spectrum of the coupling torque signal presented in Figure 6 (motor control, system without IS). It illustrates that only two vibration modes are relevant which is in accordance with the conclusions from the modelling and simulation in Sec.2.3. It also shows that the first mode is much more significant that the second one which means that the second mode has little influence in the tip response and that cancelling this mode yields little improvement in the antenna movements. In Figure 6b, the cancellation of the second mode by the IS_2 does not significantly reduce the amplitude of the vibration. Therefore, the most representative performances are going to be IS_1 and IS_3.

Figure 8a and Figure 8b illustrate the motor responses when using the IS_1 and IS_3, respectively. They show a good tracking of the motor references that were achieved using the designed motor controllers: zero steady state error with a delay of about 0.04 s in following the trajectory. Under the assumption that only one vibration mode is relevant, it is possible to design the following tip position estimator

$$\theta_t(t) = \theta_m(t) - \frac{\Gamma_\theta^{coup}(t)}{C} \quad , \phi_t(t) = \phi_m(t) - \frac{\Gamma_\phi^{coup}(t)}{C}, \tag{15}$$

where C is the rigidity coefficient of the link (obtained from static deflections). The tip position estimators are approximated and they are valid only under the assumption of a single vibrational mode.

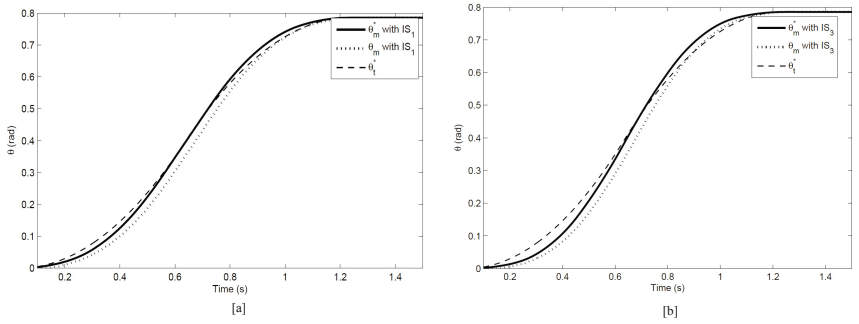


Fig. 8. Azimuthal case, motor response to: (a) IS_1 and (b) IS_3

Figure 9a and Figure 9b depict the tip position estimated from (15) when the IS_1 and IS_3 were used respectively. These figures show that the vibrations were effectively removed from the tip during antenna manoeuvres. Vibration

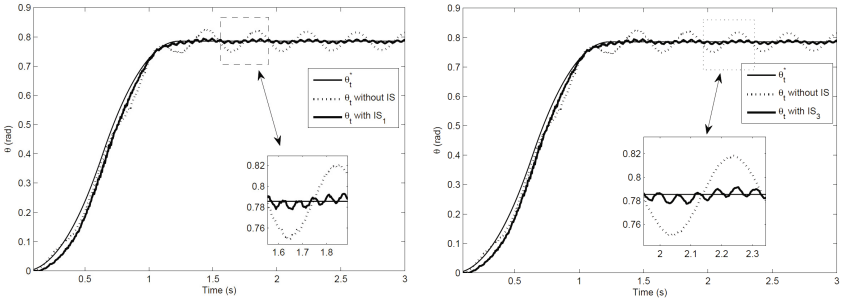


Fig. 9. Antenna tip positioning to the inputs without IS and with: [a] IS_1 and [b] IS_3

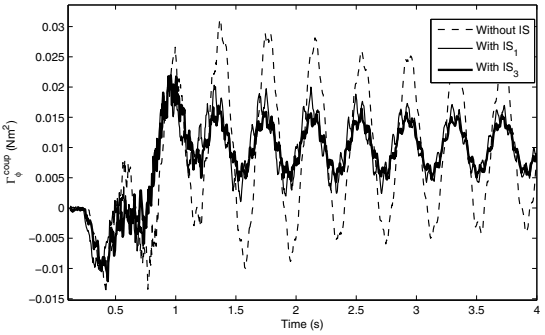


Fig. 10. Torque of vertical movement, IS_1 and IS_3 comparison

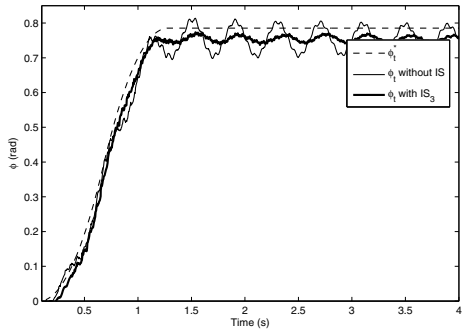


Fig. 11. Antenna tip positioning to the inputs without IS and with IS_3

Table 5. Antenna Frequencies

Azimuthal	1^{st} frequency (Hz)	2^{nd} frequency (Hz)
Model	2.74	15.42
Experimental	2.62	15.63
Attitude	1^{st} frequency (Hz)	2^{nd} frequency (Hz)
Experimental 0^o	2.61	16.08
Experimental 30^o	2.45	15.01
Experimental 45^o	2.42	15.15
Experimental 60^o	2.32	15.08

frequencies of the link were determined experimentally (for the one end clamped case) for azimuthal and attitude movements. In attitude movements, they were calculated for several attitude angles. These results are shown in Table 5. In particular, a variation of about $\pm 15\%$ of the nominal frequencies in the attitude movements is shown. It suggests that, according to Figure 5 the IS_1 is not appropriate for removing the vibrations in this movement, but the IS_3 is well suited for that.

Figure 10 shows the moment at the base of the link (coupling torque) when an attitude manoeuvre is carried out (ϕ_t^* in Table 4). It considers the cases of using: a) only motor control, b) motor control combined with IS_1 and c) motor control combined with IS_3. It shows that both IS reduce the vibration amplitude when it is compared to that of the system without IS. The IS_1 reduces about one half of the vibration value and the IS_3 about one fourth of this value. Figure 11 depicts the estimated tip angle obtained in this experiment. This figure shows that vibrations are reduced significantly, but they are not as much attenuated as in the horizontal motion experiment because of the non-linear effects of gravity.

Finally, a combined azimuthal-attitude manoeuvre was carried out (according again to Table 4). Figure 12a and Figure 12b show the experimental 3D trajectories of the antenna tip (estimated from applying (15) to both angles and after the direct kinematic transformation). It uses only motor control and motor control combined with the IS_3, respectively. The figures illustrate the projection of these 3D trajectories on a plane which is tangent to the antenna workspace sphere at a point which is the desired target of the antenna manoeuvre. These figures show that the IS_3 reduces to one third the amplitude of the residual vibrations produced at the end of the manoeuvre when using only the motor control (without IS).

Table 6. Vibration Reduction

Case	Av_{mc} (mm)	Av_{IS} (mm)	Red %
$\Delta\theta_t$	41.4	8.76	78.84
$\Delta\phi_t$	57.8	17.45	69.81
$\Delta\theta_t$, $\Delta\phi_t$	59.36	18.75	68.41

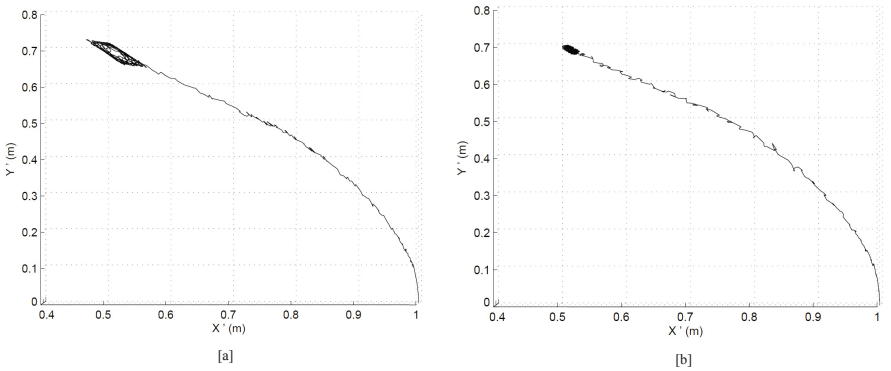


Fig. 12. 3D movement: [a] without IS and [b] with IS_3

In Table 6, quantitative data of the most relevant system results are shown. It allows to illustrate and compared the final system performance. The vibration amplitude is measured considering three antenna movements (θ_t^*, ϕ_t^* and both cases simultaneously). It considers the IS_3 vibration amplitude since it proved to be the best IS performance, and in this case it was compared to the system without IS. The table also depicts the percentage of vibration reduction Red, Av_{mc} and Av_{IS} are the amplitude vibration of the system without IS and with the IS_3, respectively. The following equation is used to obtain the percentage of vibration reduction

$$Red = \frac{Av_{mc} - Av_{IS}}{Av_{mc}} \cdot 100\%. \tag{16}$$

5 Conclusion

A 2DOF flexible sensing antenna has been designed and built. The precision and the amount of data collected by antenna sensor, obtained when the link searches for an object or collides, could be improved by the use of the proposed controllers which are designed to achieve fast and vibration-free motion.

The antenna system drives a flexible-link using DC servo actuators, therefore, the control strategy has to deal with non-linear motor friction, non-linear link dynamics and link vibrations at high frequencies. The control strategy combines a closed loop control of the motors and an open loop control for vibration damping. The motor control consisted of two PID controllers with a low pass filter and demonstrated to be robust to the non-linear friction and parameter uncertainties.

Several experiments testing IS schemes proved that the use of a robust input shaping (IS_3) for vibration damping was the best option considering control complexity and efficiency. Experiments illustrated that fast movements were achieved while reducing the tip vibration to about 30% of its value if only motor control had been implemented, and nearly 70% when the system included

motor control and the IS technique. Therefore, the tip positioning precision was increased as consequence of using the proposed controllers.

Future research includes improving the vibration attenuation by combining IS techniques with feedback linearization, so that the non-linear nature of the link dynamics is taken into account.

Acknowledgments. This study was financially supported by the Spanish Ministry of Economy and Competitiveness and by the European Social Fund with project DPI2012-37062-CO2-01.

References

1. Prescott, T.J., Pearson, M.J.: Whisking with robots. IEEE Robotics and Automation Magazine 16(3), 42–50 (2009), doi:10.1109/MRA.2009.933624
2. Bellezza, F., Lanari, L., Ulivi, G.: Exact modelling of the flexible slewing link. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, pp. 734–804 (1990)
3. Dwivedy, S.K., Eberhard, P.: Dynamic analysis of flexible manipulators, a literature review. Mechanism and Machine Theory 41(7), 749–777 (2006)
4. Kaneko, M., Kanayama, N., Tsuji, T.: Active antenna for contact sensing. IEEE Transactions on Robotics and Automation 14(2), 278–291 (1998)
5. Ueno, N., Kaneko, M., Svinin, M.: Theoretical and experimental Investigation on dynamic Active Antenna. In: Proc. of the International Conference of Robotics and Automation, vol. 4, pp. 3557–3563 (1996)
6. Clements, T.N., Rahn, C.D.: Three dimensional contact imaging with an actuated whisker. IEEE Transactions on Robotics 22(4), 844–848 (2006)
7. Zhao, H., Rahn, C.D.: Repetitive Learning Control of a Flexible Whisker in Tip Contact with an Unknown Surface. Journal of Vibration and Control 17(2), 197–203 (2010)
8. Scholz, G.R., Rahn, C.D.: Profile sensing with an actuated whisker. IEEE Transactions Robotics and Automation 20(1), 124–127 (2004), doi:10.1109/TRA.2003.820864
9. Pearson, M.J., Mitchinson, B., Charles, S., Anthony, G., Prescott, P., Tony, J.: Biomimetic vibrissal sensing for robots. Trans. R. Soc. B (2011), doi:10.1098/rstb.2011.0164 366 2011
10. Castillo, C.F., Castillo, F.J., Feliu, V.: Experimental Validation of 2 degrees of freedom whisker sensor dynamic model. In: IFAC World Congress, vol. 18(Part 1), pp. 3148–3152 (2011), doi:10.3182/20110828-6-IT-1002.02360
11. Castillo, C.F., Castillo, F.J., Feliu, V.: Inverse dynamics feed forward based control of two degrees of freedom whisker sensor. In: IEEE International Conference on Mechatronics (ICM), Istanbul, Turkey, pp. 684–689., doi:10.1109/ICMECH.2011.5971202
12. Singer, N.C., Seering, W.C.: Preshaping command inputs to reduce system vibration. ASME Journal of Dynamics System, Measurement and Control 112(1), 76–82 (1990)

13. Huey, J.R., Sorensen, K.L., Singhose, W.E.: Useful applications of closed loop signal shaping controllers. Control Engineering Practice 16(7), 836–846 (2008)
14. Park, J., Chang, P.H.: Learning input shaping technique for non-LTI systems. ASME Journal of Dynamic Systems, Measurement, and Control 123(2), 288–293 (2001)
15. Pereira, E., Trapero, J.R., Diaz, I.M., Feliu, V.: Adaptive input shaping for manoeuvring flexible structures using an algebraic identification technique. Automatica 45(4), 1046–1051 (2009)
16. Sutton, R.P., Halikias, G.D., Plummer, A.R., Wilson, D.A.: Modelling and H^{∞} control of a single-link flexible manipulator. Proc. IMechE 213(I), 85–104 (1999)
17. Sira-Ramirez, H., Marquez, R., Fliess, M.: On the generalized PID control of linear dynamic systems. In: 6th Eur. Control Conf., Porto, Portugal, pp. 166–171 (2001)

Part XIII

Robust Techniques for Planning, Navigation and SLAM

Towards Exploiting the Advantages of Colour in Scan Matching

Fernando Martín[1], Jaime Valls Miró[2], and Luis Moreno[1]

[1] Carlos III University, Madrid, Spain
{fmmonar,moreno}@ing.uc3m.es
http://roboticslab.uc3m.es
[2] University of Technology, Sydney, Australia
Jaime.VallsMiro@uts.edu.es
http://ims.uts.edu.au/

Abstract. Colour plays an important role in the perception systems of the human beings. In robotics, the development of new sensors has made it possible to obtain colour information together with depth information about the environment. The exploitation of this type of information has become more and more important in numerous tasks. In our recent work, we have developed an evolutionary-based scan matching method. The aim of this work is to modify this method by the introduction of colour properties, taking the first steps in studying how to use colour to improve the scan matching. In particular, we have applied a colour transition detection method based on the *delta E* divergence between neighbours in a scan. Our algorithm has been tested in a real environment and significant conclusions have been reached.

Keywords: Differential Evolution, Scan Matching, RGB-D, Colour Properties, Delta E.

1 Introduction

The information provided by colours is a crucial factor in human perception. If we are approaching a dark place, it is not easy to distinguish between objects and free space. If we see fire in a house, it is a signal of danger. If we are walking in the beach, the shore can be distinguished by the blue of the sea. One traditional task in robotics consists of building maps using depth sensors such as laser range finders. This task is commonly referred to as mapping. If the sensor works in three dimensions (3D), it is possible to obtain a volumetric representation. In the last years, the development of cheap new sensors that provide depth and colour information, such as the Microsoft Kinect, has made it possible to utilize colour properties when building volumetric maps. Although colour properties have been widely used in fields such as computer vision, it is not common to use them when building volumetric maps with depth information. The objective of this work is to make an initial study about how to exploit the advantages of colour in a specific application (scan matching).

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_17,

Mapping is closely related to the well-known Simultaneous Localization and Mapping (SLAM) problem, originally introduced by Leonard and Durrant-White [1] basing on an earlier work by Smith *et al.* [2]. The SLAM problem for mobile robots consists of building a map of an unknown environment while exploring the environment at the same time, using this map.

The work reported in this paper is focused on the scan matching (or registration) problem, which is a particular aspect of the mapping task. The scan matching can be defined as the estimation of the metric relation (position and orientation) between a pair of scans, using the information provided by these scans. This is a very common problem in mobile robotics because the information about the environment that is going to be represented is often given by an exteroceptive sensor with depth information.

A scan matching algorithm for 3D environments has been created in our recent work [3]. The core of this technique is the Differential Evolution (DE) method, developed by Storn and Price [4], which is a particle-based evolutionary algorithm that evolves in time to the solution that minimizes a fitness function. If the cost function is chosen in an adequate way, the scan matching problem can be solved.

The scan matching methods traditionally have failures when there is an important change in orientation between scans. In these cases, an initialization algorithm like RANSAC has been used to solve this problem [5]. There are some colour properties that are more invariant to rotation than the raw 3D data. An initial idea is to try to improve our DE-based scan matching algorithm in order to increase the robustness under sharp turns.

In this work, we incorporate colour properties into the cost function of the scan matching method. This can be done because the sensors obtain RGB information together with depth information. In particular, we propose to extract the colour transitions, which are represented by neighbours with a significant change in colour, and to use these points of interest to perform the scan matching. Due to its interesting properties, the *delta E* divergence (unit of measurement proposed by the International Commission on Illumination, CIE in french) has been chosen as a suitable measurement to obtain these points of interest.

The objective of this work is to take the first steps in studying how to use colour to improve our method. The new technique has been tested and compared to the original version in a real environment, and significant conclusions have been reached.

1.1 Related Work

SLAM is one of the most popular topics in mobile robotics. The interest about 3D maps has increased within the recent past due to the availability of efficient 3D sensors. For example, Nuchter *et al.* [6] have proposed a technique based on the alignment of 3D scans using the Iterative Closest Points [7] (ICP) scan matching method combined with a heuristic for loop detection and a global relaxation method. Hähnel *et al.* [8] have proposed an algorithm "for full 3d shape reconstruction of indoor and outdoor environments with mobile robots". Triebel *et al.*

[9] have built Multi-Level Surface (MLS) maps. Cole *et al.* [10] have proven that the traditional methods used to solve the SLAM problem in planar environments can also be extended to perform 6D SLAM in more difficult conditions, e.g. undulating outdoor areas. Magnusson and Duckett [11] have designed an alignment procedure based on the Normal Distributions Transform (NDT) [12].

The scan matching techniques can work with 2D or 3D maps. There are also local [13] and global [14] methods. While the local methods match single scans, the global methods consider the current scan and the global model. It is also possible to distinguish between feature-based, point-based, or mixed approaches. The first option requires a feature extraction before the scan matching. The point-based approach does not require any distinguishable structure in the environment and the mixed method seeks the correspondence between points and features.

The ICP method [15], which is the most common scan matching technique, is an algorithm that is used to minimize the spatial distance between two scans. This method is quite simple and its computational cost makes it possible to use it in real-time. It receives two clouds of points, an initial estimate of the translation and rotation, and the stopping criteria. It iteratively estimates the transformation (translation and rotation) that minimizes the distance between the clouds. Besl and Mac Kay [7] have implemented this method to register 3D shapes.

Many variants of this method have been proposed. A very interesting comparison of several methods depending on different parameters can be found in the work by Rusinkiewicz *et al.* [16]. The cost function is changed in order to match points that belong to similar surfaces in the variant proposed by Triebel *et al.* [9]. Bosse and Zlot [17] have proposed an improvement based on "the addition of robust optimization techniques to handle outliers and imperfect correspondences between the data". A traditional improvement applied to ICP is to incorporate additional information to the points of the clouds [18,19].

The Iterative Matching Range Point (IMRP) method, proposed by Lu and Milios [13], is based on the limitation of the maximum translation and rotation. The Iterative Dual Correspondence (IDC) method, also proposed by Lu and Milios [13], combines ICP and IMRP. The translation is computed by the ICP method and the rotation is estimated by the IMRP method. The Polar Scan Matching (PSM) method [20] does not need to find correspondences between points. It assumes that model and data are sorted in the same way and only points with the same bearing are matched. Thrun *et al.* [21] consider that the free space in the current model will remain free in the future. They increase the information that is extracted from the laser scan.

There are not many researchers who have played attention to visual information in scan matching. Henry *et al.* [5] have extracted visual features to establish correspondences in their scan matching method. Ramos *et al.* have created the CRF-matching [22], which uses conditional random fields for feature-based scan matching. May *et al.* [19] have used laser reflectance values to improve the ICP. However, they do not take full advantage of the application of colour-based features.

An evolutionary-based scan matching algorithm for 3D environments has been developed in our recent work. In this paper, this method has been improved in order to exploit colour properties in mapping. Our method tries to solve the same problem than ICP, which is the minimization of the distance between two point clouds. It also searches corresponding points between scans.

The rest of this paper is organized as follows. The colour properties that are used here are introduced in Section 2. After that, the registration method is explained in Section 3. The experimental results are presented in Section 4 and, finally, the most important conclusions are summarized in Section 5.

2 Colour Properties-Delta E

In recent years, the development of new sensors that provide colour and depth information, especially the Kinect developed by Microsoft, has caused a significant change in the research fields that take into account this type of information. One of the most influenced areas in robotics is mapping. Their main advantage is that the colour information is now available together with the depth point cloud. In addition, their low cost has enabled a rapid introduction into the research centers.

The data obtained from these sensors is a 3D depth scan where each measurement is accompanied by its colour information in RGB. That is why this information is traditionally called RGB-D. There are many applications that use this type of data in mapping. One of the most significative approaches is the KinectFusion project [23,24].

This paper seeks to analyze and exploit the colour characteristics for a particular application. In this section, a brief explanation about how to measure different colour properties is given. These properties will be used later to implement the cost function for the DE-based scan matching method.

In the RGB colour space, each colour is represented by three components corresponding to red, green, and blue:

$$Colour = (R, G, B). \tag{1}$$

Each value varies from zero to a defined maximum value. When the three components are equal to zero, the black colour is represented. The white colour is obtained when the maximum value is fixed in all the components.

The RGB model has been used in multiple fields such as electronics systems and photography. An important fact to be taken into account is that this colour space is highly dependent on the device. The same object will not produce the same RGB values in different devices. Therefore, a RGB value cannot be defined as an absolute colour. An additional disadvantage of this representation is that it depends on the lighting conditions.

Different interesting features can be extracted when each depth point has an associated RGB colour. One possibility is to consider the colour transitions (significant changes in colour of nearest neighbors in the scan) as the features

that will be matched. It is possible to compare the RGB values between adjacent points to measure colour differences:

$$d_c = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2}, \tag{2}$$

where ΔR, ΔG, and ΔB are the differences in each component between nearest neighbours in the scan, and d_c is the colour difference. If this measurement is compared to a fixed threshold, the most significant colour transitions in the depth point cloud can be discovered. These significant points can be used to match pairs of scans (only using the extracted transitions in the cost function, instead of the original point cloud without colour).

A different metric proposed to measure the difference between colours by the International Commission on Illumination (CIE, from the initials in french: *Commission Internationale de l'Éclairage*) is the *delta E* (ΔE), where E is the German word for sensation (*Empfindung*). The goal of this variable is to establish a ΔE value that means a Just Noticeable Difference (JND), which symbolizes colour differences that can be noticed by human beings. Different formulae have been proposed to compute this metric. All of them are based on the device independent CIELAB color space.

The Lab colour space can be defined as a *colour-opponent* space composed of three components: L represents the lightness, and a and b are *colour-opponent* dimensions. These dimensions are based on the CIE XYZ colour space, proposed by the CIE in 1931 and derived from the RGB space. The original coordinates of the Lab space were suggested by Hunter [25,26]. Nevertheless, these coordinates have evolved to the more recent CIE 1976 or CIELAB colour space, where the $L^*a^*b^*$ components are defined with asterisks to make a distinction between both spaces. The main difference between them is the type of transformation that is applied to calculate the final coordinates.

Two initial objectives were followed when defining the Lab colour space. The first one was to improve the perceptual uniformity. It is important that similar changes in the numerical values produce similar changes in the visual appearance. The second one was to create a space easy to compute from the XYZ coordinates.

The colour spectrum of the CIELAB space is larger than the RGB and CMYK spaces. It includes all perceivable colours, even outside the human vision gamut. One of the most important features of this space is that it does not depend on the device. Another important characteristic is that it has been designed to imitate the human vision system. The perceptual uniformity has been increased and the L^* component is closer to the human perception of lightness.

The CIELAB colour space is composed of three variables:

- L^*: lightness. From $L^* = 0$ (black) to $L^* = 100$ (diffuse white).
- a^*: position between red/magenta and green. From green (negative values) to magenta (positive values).
- b^*: position between yellow and blue. From blue (negative values) to yellow (positive values).

The variations of L^*, a^*, and b^* are nonlinear, trying to reproduce the non-linear response of the human eye. Furthermore, uniform changes in the values of

these variables cause uniform changes in the perceived colour. For this reason the colour difference between two points characterized by their coordinates in the CIELAB space can be approximated by the euclidean distance between them.

The conversion from RGB or CMYK to CIELAB is not an easy task because these models are dependent on the device. It is necessary to make an intermediate conversion to a device-independent space, which is the sRGB. After this conversion, it is possible to obtain the CIE 1931 XYZ coordinates and then the CIELAB values.

Going back to the *delta E* difference, several definitions have been proposed throughout the years (CIE76, CIE94, CIEDE2000, CMC l:c (1984)). The reason of these changes is to minimize or reduce the problems derived from the non-uniformity of the CIELAB space. An important problem to solve is that the human eye is more sensitive to some colours than others. The objective of these metrics is to measure a JND between two colours. Some typical thresholds used to define a JND are 1.0 or 2.3. Nevertheless, the objective of this work is to apply this metric when detecting relevant colour transitions, and this threshold is out of our interest.

In this work, the CIE76 version has been implemented for simplicity (more options will be implemented in a future work). The following equation extracts the ΔE difference between two colours, $L_1^* a_1^* b_1^*$ and $L_2^* a_2^* b_2^*$, in the $L^* a^* b^*$ space:

$$\Delta E_{ab}^* = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2}, \tag{3}$$

where $\Delta E_{ab}^* \approx 2.3$ establishes a JND. Equation 3 will be utilized to detect the most significant colour transitions in the scan, points of interest that will be used by the scan matching algorithm.

3 Six-Dimensional DE-Based Scan Matching

In this section, the main ideas about the DE-based scan matching method are briefly described. For a more detailed explanation, the reader can consult our previous work [27,28,3].

Starting from two independently acquired scans of 3D points (named model and data), this technique computes the rotation and translation that is necessary to apply to the data set in order to maximize the matching between them.

The stochastic search of the *matching pose* is based on the DE method for global optimization problems over continuous spaces. The main concepts of this method are given in Algorithm 1.

The initial population is composed of N_P candidates. Each candidate corresponds to a possible solution. The robot's pose has six degrees of freedom (DOF) because the robot is located in a 3D world:

$$pop_i^k = (x_i^k, y_i^k, z_i^k, \phi_i^k, \theta_i^k, \psi_i^k),$$

where pop_i^k represents element i at iteration k. The position is represented by the cartesian coordinates and the orientation is given by the Euler angles.

Algorithm 1. DE-based Scan Matching using Colour Properties

1: $model\_ct$, $data\_ct \leftarrow extract\_ct(model, data)$ ▷ Color transitions extraction
2: **for** $i = 1 : N_P$ **do**
3: $pop_i^1 \leftarrow init\_pop(data\_initial\_pose)$ ▷ First population generation
4: $e_i^0 \leftarrow cost(model\_ct, data\_ct, pop_i^1)$ ▷ Cost function calculation
5: **end for**
6: **for** $k = 1 : max$ **do**
7: **for** $i = 1 : N_P$ **do**
8: $v_i^k = pop_a^k + F(pop_b^k - pop_c^k)$ ▷ Mutation
9: **for** $j = 1 : D$ **do**
10: $u_{i,j}^k = v_{i,j}^k, \forall p_{i,j}^k < \delta$ ▷ Crossover
11: $u_{i,j}^k = pop_{i,j}^k, \forall p_{i,j}^k \geq \delta$
12: **end for**
13: $e_i^k \leftarrow cost(model\_ct, data\_ct, pop_i^k)$ ▷ Cost function calculation
14: **if** $e_i^k < e_i^{k-1} - \tau$ **then** ▷ Selection with Thresholding
15: $pop_i^{k+1} = u_{i,j}^k$
16: **else**
17: $pop_i^{k+1} = pop_i^k$
18: **end if**
19: **end for**
20: $pop^k = disc(pop^k)$ ▷ Discarding
21: $ind\_best \leftarrow min(e^k)$
22: $bestmem \leftarrow pop^k(ind\_best)$
23: **if** $convergence = true$ **then** ▷ Execution stops after convergence
24: $exit(bestmem)$
25: **end if**
26: **end for** ▷ Return best estimate

The first population will be randomly situated in a sphere close to the pose estimate provided by the odometry information. The scan matching problem consists of rectifying the robot's pose obtained by the odometry sensors, thus a population is created around the original pose.

The fitness function is executed for each member of the population (line 4 of Algorithm 1). This cost function is a key component of this procedure. It seeks corresponding points between significant colour transitions of both frames (only considering the points of interest, not the original depth point cloud) and calculates a cost value basing on the distance between these corresponding points. The scan matching method is based on the minimization of the cost value. The main concepts and the equations will be explained below in this section.

The principal loop starts in line 6 and it is repeated until one of the convergence conditions is met or the upper limit of iterations is reached. An additional loop that contains the evolutive search begins in line 7. It consists of the generation of a new population for the next generation. In a single iteration the algorithm is executed to obtain the next candidates, evolving in time to the correct pose.

The evolutive search starts with the mutation. Each population member is perturbed to generate a mutated vector v_i^k:

$$v_i^k = pop_a^k + F(pop_b^k - pop_c^k) \tag{4}$$

where pop_a^k, pop_b^k, and pop_c^k are three randomly chosen elements at iteration k and a, b, and c are different from running index i. F is a real and constant variable that controls the amplification of the differential variations.

To increase the diversity of the new generation, the crossover is introduced. The trial vector is defined by $u_i^k = (u_{i,1}^k, u_{i,2}^k, \ldots, u_{i,D}^k)^T$ and its parameters are selected according to the crossover probability:

$$u_{i,j}^k = \begin{cases} v_{i,j}^k; & \text{if } p_{i,j}^k < \delta, \\ pop_{i,j}^k; & \text{otherwise,} \end{cases} \tag{5}$$

where $p_{i,j}^k$ is a randomly chosen value from the interval $[0, 1]$ for each parameter j of the population member i at iteration k, and δ is the crossover probability that constitutes the crossover control variable. D represents the number of chromosomes (number of components of the population element) and is equal to six because the robot's pose has six DOF.

A comparison between the new candidate u_i^k and pop_i^k is done to decide which element should become a member of generation $k+1$. If u_i^k holds a better value for the fitness function than pop_i^k, then it is replaced by u_i^k; otherwise, pop_i^k is retained for the next generation.

A thresholding band has been combined with the selection mechanism to avoid the premature convergence in noisy optimization problems. The idea is to diminish the eagerness of the algorithm by rejecting those new solutions that do not improve the previous hypothesis in a pre-specified magnitude τ.

When the selection mechanism is combined with the thresholding band described before, the following condition must be checked:

$$pop_i^{k+1} = \begin{cases} u_i^k, & \text{if } e_{pop_i}^k - e_{u_i}^k > \tau \\ pop_i^k, & \text{otherwise} \end{cases} \tag{6}$$

where $e_{pop_i}^k$ is the cost function value of the current candidate and $e_{u_i}^k$ represents the fitness function value of the trial vector.

The previous stages (mutation, crossover, and selection with thresholding) are applied to the whole population, obtaining the next generation population $(k+1)$.

Since the convergence speed is reduced by the thresholding band, a discarding mechanism has been included to increase the algorithm speed while keeping the thresholding advantages. The worst members of the new population are replaced by new solutions that are situated close to better ones.

Finally, the algorithm returns the best solution, which is the robot's pose that minimizes the difference between model and data (scan matching solution). The data set is moved and rotated according to this solution, and the registration is successfully achieved.

3.1 Delta E-Based Cost Function

The cost function of each particle is obtained after extracting the most significant colour transitions.

First, the *delta E* divergence is calculated for each point of the cloud with respect to its closest neighbour according to Equation 3.

After that, the colour transitions are extracted. Only the points with a *delta E* divergence greater than a threshold are selected to be considered by the fitness function.

The cost function to minimize is the derived from the Euclidean distance between the corresponding points of the colour transitions:

$$e = \sum_{c=1}^{C} d(\mathbf{m\_ct}_{i_c}, \mathbf{d\_ct}_{j_c})^2, \tag{7}$$

where the number of correspondences is equal to C and the distance between two corresponding points is denoted by $d(\mathbf{m\_ct}_{i_c}, \mathbf{d\_ct}_{j_c})$.

The result of the scan matching method will be the relation between scans that minimizes the distance between corresponding points, but only considering colour transitions.

4 Experimental Results

The performance of the implemented method has been tested in different situations in a real environment using the the *Freiburg2* data set recorded from a Kinect mounted on a Pioneer robot[1].

Configuration parameters: *delta E* threshold to extract a transition: *min* $\Delta E^*_{ab} = 10$. Reduction factor of the original data: 0.25 (The original clouds are composed of 307200 points. They are reduced in a preprocessing step to increase the computational speed). DE parameters: $N_P = 20$, $F = 0.8$, $\delta = 0.75$.

First of all, the matching results in two different cases are shown in Figure 1 (named as Map 1 and Map 2 from now on). Both maps are composed of three scans. In Figure 2, the points of interest according to the *delta E* difference (top) and the original image (bottom) are presented. The errors are given in Table 1.

As can be observed in both figures, the algorithm presented in this paper is capable of solving the scan matching problem. Besides, we have tested our method more extensively with the data set, which is composed of 2900 different scans, in order to check its performance.

Analyzing the errors in Table 1, the translation errors are in the interval $[0.03, 5.90]$ cm, and the rotation errors are in the interval $[0.010, 0.156]$ rad. The accuracy is similar to that shown by other authors. Diosi and Kleeman [20] have reported an average error equal to 3.8 cm and 0.86° using PSM in planar maps.

[1] Thanks to Jürgen Sturm for making available this dataset (`http://vision.in.tum.de/data/datasets/rgbd-dataset/download#freiburg2_pioneer_slam`)

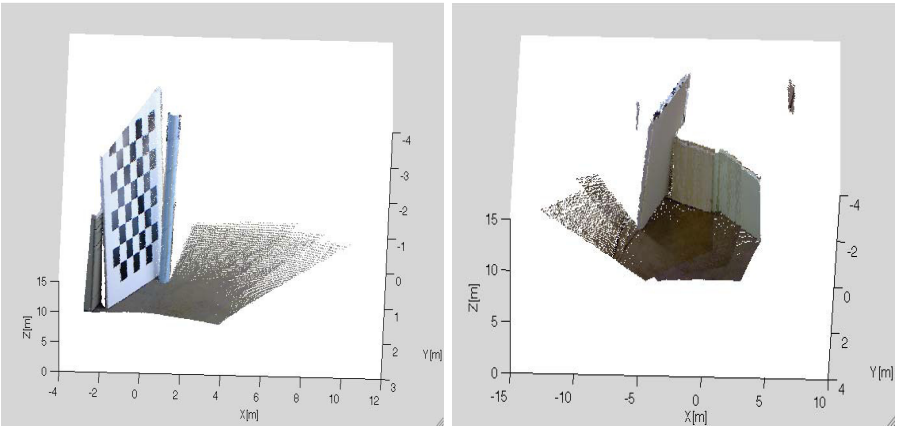


Fig. 1. Two examples of scan matching. Maps composed of three different frames.

Table 1. Scan matching errors in Maps 1 and 2. The ground truth is available. Frame 1 as origin.

	x(cm)	y(cm)	z(cm)	θ_x(rad)	θ_y(rad)	θ_z(rad)
Map 1 Frame 2	0.41	0.32	0.36	0.010	0.057	0.049
Map 1 Frame 3	1.41	0.68	0.68	0.023	0.109	0.103
Map 2 Frame 2	2.25	1.29	5.72	0.007	0.103	0.034
Map 2 Frame 3	3.32	5.90	2.73	0.034	0.156	0.044

The error obtained by Bosse and Zlot [17] is equal to 11.3 cm, also in planar maps.

An interesting property that is often studied in scan matching is the valley of convergence. It consists of running the algorithm when one of the scans is translated and rotated, checking the results under different offsets. A threshold has to be defined to determine when the matching results are successful. In this experiment, this threshold has been fixed to 0.2 m. It is particularly important to obtain satisfactory results when there are important changes in the orientation, because this is a typical shortcoming of the scan matching methods. Figure 3 shows the valley of convergence for the colour-based scan matching algorithm (left part) with translations from -1 to 1 m and rotations from $-80°$ to $80°$. Each semicircle corresponds to a translational offset. Each individual mark of the semicircle represents a rotational offset for a fixed translation. Successful cases are drawn in green and failures in red. The results using the original method without colour are also displayed in the right part of the figure.

Analyzing the results, the *delta E*-based version has a percentage of success of 86.42%. The percentage of success of the original version without colour is 80.25%. This percentage is slightly improved, but there is not a big difference between both implementations. Nevertheless, it is a single example and more experiments are needed to reach more conclusive results. Magnusson *et al.* [29]

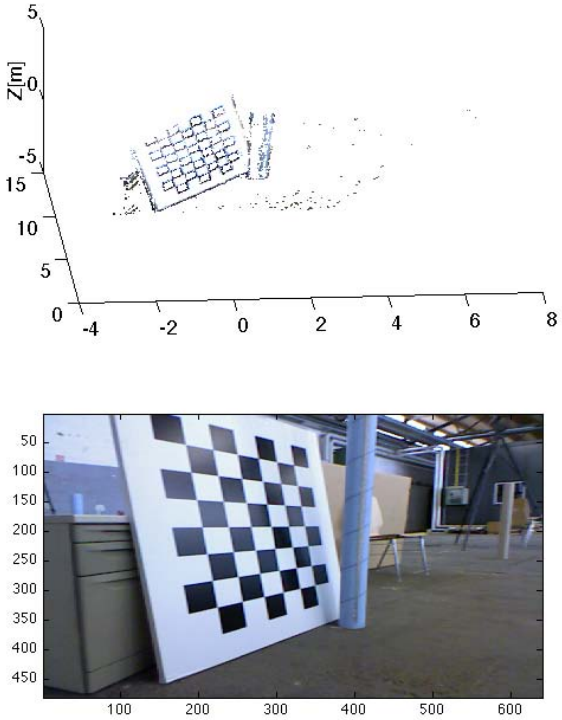


Fig. 2. Example of color transitions extraction: significant points according to the delta E between neighbors (top) and original image (bottom)

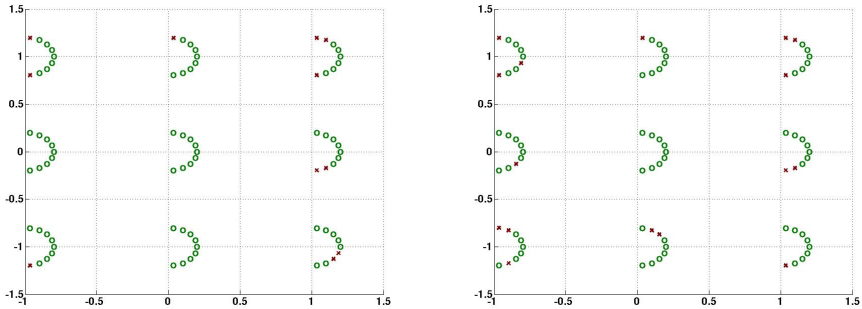


Fig. 3. Valley of convergence. Offsets in metres. Left: Version with *delta E*. Right: Version without colour. Successful matching in green. Wrong matching in red.

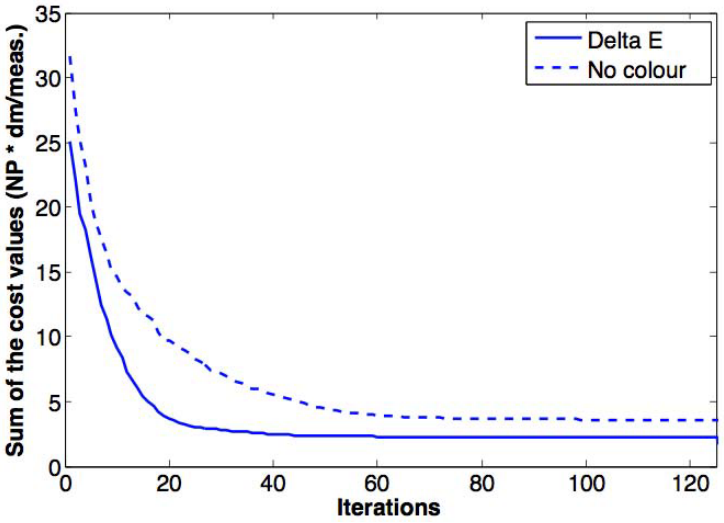


Fig. 4. Convergence comparison between the original method and the colour-based one

have compared ICP to NDT (they do not consider colour). They use two thresholds, a stricter one (0.2 m) and a weaker one (1 m). We only include their results with the strict threshold because we have chosen the same value for comparison. Their success rates for the strict translation threshold are: 13% for ICP, 37% for NDT, and 95% for trilinear NDT, which is an improved version of the NDT-based method. Our results are better when compared to the ICP implementation and the basic NDT algorithm, but the trilinear NDT registration algorithm outperforms our method. However, these are the results in particular cases. In order to have an adequate comparison, all methods have to be compared in detail using the same data set, which is an interesting work to be done in the future. This comparison will let us know if our method presents a better performance under sharp turns (robustness under rotation), which is one of the traditional shortcomings of the ICP method.

An additional fact has been observed in the valley of convergence experiment. The average of the fitness function final value for the *delta E*-based method is 1.1423 cm/measurement, and the average position error is equal to 0.0063 cm. In the matching version without colour, the cost function final value is equal to 1.8078 cm/measurement, and the average position error is 1.1117 cm. In this case, the accuracy is higher when the colour properties are included.

The last property analyzed in the experiments is the convergence speed. A comparison between the convergence of the new version and the original method without colour for a particular case is presented in Figure 4. The evolution of the sum of the cost values (for the whole population), in dm per measurement, against the number of iterations is displayed. In the original method, the algorithm converges after 100 iterations. In the new version, the final value is reached after 60 iterations. The convergence speed is significantly improved, which is an

important result because the computational cost depends directly on the number of iterations. Due to its relation with the computational cost, a more detailed study about the improvement in the convergence speed is a necessary work to be done in the future.

5 Conclusions

An initial version of a scan matching method based on colour properties has been proposed in this paper. In particular, we have used the *delta E* divergence to extract significant colour transitions of RGB-D scans. This colour information has been included in the cost function of the DE-based scan matching method, which only matches points that belong to these significant transitions.

The algorithm has been tested in a real environment, showing a good performance in a data set composed of multiple frames recorded from a Kinect. The errors obtained are similar to those obtained by other research groups.

Regarding the valley of convergence, this new version presents a slightly better performance. Although there are not significant differences when compared with the DE-based version without colour, both versions (with and without colour) outperform the classic ICP results shown by Magnusson *et al.* [29]. Since the lack of robustness when the change in the orientation is important is a key drawback in scan matching, a more detailed study about the valley of convergence of multiple methods/options (ΔE-DE, DE, ICP, NDT, etc.) is a challenging work to be accomplished in the future.

The convergence speed is significantly improved in the new version, which is an important advantage because the computational cost is decreased.

Finally, it is necessary to remark that this paper is an initial study about how to apply colour properties to improve the scan matching. More experiments are needed and more properties have to be considered and implemented to improve our method.

References

1. Leonard, J.J., Durrant-Whyte, H.: Mobile Robot Localization by Tracking Geometric Beacons. IEEE Transaction on Robotics and Automation 7, 376–382 (1991)
2. Smith, R., Self, M., Cheeseman, P.: Estimating Uncertain Spatial Relationships in Robotics. In: Proceedings of the Second Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI 1986), New York, NY, pp. 267–288. Elsevier Science (1986)
3. Martín, F., Triebel, R., Moreno, L., Siegwart, R.: Two different tools for three-dimensional mapping: DE-based scan matching and feature-based loop detection. Robotica (2013)
4. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
5. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. In: Proceedings of the Intl. Symp. on Experimental Robotics, ISER (2010)

6. Nüchter, A., Lingemann, K., Hertzberg, J.: 6D SLAM-3D Mapping Outdoor Environments. Journal of Field Robotics 24, 699–722 (2007)
7. Besl, P.J., McKay, N.D.: A method for registration of 3d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
8. Hähnel, D., Burgard, W., Thrun, S.: Learning compact 3d models of indoor and outdoor environments with a mobile robot. Robotics and Autonomous Systems 44, 15–27 (2003)
9. Triebel, R., Pfaff, P., Burgard, W.: Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006 (2006)
10. Cole, D.M., Newman, P.M.: Using Laser Range Data for 3D SLAM in Outdoor Environments. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2006 (2006)
11. Magnusson, M., Ducket, T.: A Comparison of 3D Registration Algorithms for Autonomous Underground Mining Vehicles. In: Proceedings of the Second European Conference on Mobile Robotics, Ancona, Italy (2005)
12. Biber, P., Straßer, W.: The Normal Distributions Transform: A New Approach to Laser Scan Matching. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003 (2003)
13. Lu, F., Milios, E.: Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. Journal of Intelligent and Robotic Systems 20, 249–275 (1997)
14. Tomono, M.: A scan matching method using euclidean invariant signature for global localization and map building. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2004 (2004)
15. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision 13, 119–152 (1994)
16. Rusinkiewicz, S., Levoy, M.: Efficient Variants of the ICP Algorithm. In: Proceedings of the Third International Conference on 3D Digital Imaging and Modeling (2001)
17. Bosse, M., Zlot, R.: Map Matching and Data Association for Large-Scale 2D Laser Scan-Based SLAM. The International Journal of Robotics Research 27, 667–691 (2008)
18. Johnson, A., Kang, S.B.: Registration and integration of textured 3-d data. In: International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM 1997) (1997)
19. May, S., Droeschel, D., Holz, D., Fuchs, S., Malis, E., Nüchter, A., Hertzberg, J.: Three-dimensional mapping with time-of-flight camers. Journal of Field Robotics 26, 11–12 (2009)
20. Diosi, A., Kleeman, L.: Laser Scan Matching in Polar Coordinates with Application to SLAM. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2005 (2005)
21. Thrun, S., Burgard, W., Fox, D.: A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2000 (2000)
22. Ramos, F., Fox, D., Durrant-Whyte, H.: CRF-matching: Conditional random fields for feature-based scan matching. In: Proceedings of Robotics: Science and Systems, RSS 2007 (2007)

23. Izadi, S., Kim, D., Molyneaux, O.H.D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST 2011), pp. 559–568 (2011)
24. Newcombe, R.A., Davison, A.J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Hodges, D.M.S., Kim, D., Fitzgibbon, A.: KinectFusion: Real-Time Dense Surface Mapping and Tracking. In: Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011 (2011)
25. Hunter, R.S.: Photoelectric color-difference meter. In: Proceedings of the Winter Meeting of the Optical Society of America (1948)
26. Hunter, R.S.: Accuracy, precision, and stability of new photo-electric color-difference meter. In: Proceedings of the Thirty-Third Annual Meeting of the Optical Society of America (1948)
27. Moreno, L., Garrido, S., Muñoz, M.L.: Evolutionary Filter for Robust Mobile Robot Localization. Robotics and Autonomous Systems 54(7), 590–600 (2006)
28. Martín, F., Moreno, L., Garrido, S., Blanco, D.: High-Accuracy Global Localization Filter for three-dimensional Environments. Robotica 30, 363–378 (2011)
29. Magnusson, M., Nüchter, A., Lörken, C., Lilienthal, A.J., Hertzberg, J.: Evaluation of 3D Registration Reliability and Speed – A Comparison of ICP and NDT. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2009 (2009)

Improving Sampling-Based Path Planning Methods with Fast Marching

Javier V. Gómez, David Álvarez, Santiago Garrido, and Luis Moreno

RoboticsLab, Carlos III University of Madrid, Avda. de la Universidad 30, 28911, Leganés, Madrid, Spain
{jvgomez,dasanche,sgarrido,moreno}@ing.uc3m.es
http://roboticslab.uc3m.es/

Abstract. Sampling-based path planning algorithms are well-known because they are able to find a path in a very short period of time, even in high-dimensional spaces. However, they are non-smooth, random paths far away from the optimum. In this paper we introduce a novel improving technique based on the Fast Marching Method which improves in a deterministic, non-iterative way the initial path provided by a sampling-based methods. Simulation results show that the computation time of the proposed method is low and that path length and smoothness are improved.

Keywords: Fast Marching, Path Planning, Path Improvement.

1 Introduction

Sampling-based path planning algorithms are nowadays one of the most powerful tools to solve planning problems, specially in high-dimensional spaces. Since the first versions of these algorithms appeared [1,2], they have been applied to many different problems and many different versions have appeared, improving more and more their performance [3].

However, the drawbacks of these methods are well-known. They are based on random (or pseudo-random) space sampling. This leads to non-optimal, stochastic paths which are far away from the optimal one (in terms of distance, obstacle clearance, etc). Many optimization techniques have been already proposed [4,5,6]. These optimization techniques, based on iterative processes, achieve the optimal path in terms of distance. Nevertheless, these optimal paths are still non-smooth and the obstacle clearance is minimum, which can be dangerous in real applications.

In this paper we introduce a method for improve the sampling-based paths by increasing the smoothness and decreasing the path length. Also, a *good enough* obstacles clearance is ensured. The proposed algorithm tries to imitate the Fast Marching Square (FM^2) algorithm [7] but applied over a triangular mesh. Modeling a continuous space as a triangular mesh and computing the paths as geodesics is already proposed in [8]. However, the requirements for the triangular mesh and

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_18,

the method used for computing the distances map makes this method computationally slow. In our case, we restrict the triangular mesh to a *region of interest* around an initial sampling-based path and compute the geodesic of the mesh with the Fast Marching Method (FMM) [9] speeding up the computation time and obtaining high-quality paths.

This paper is organised as follows. Next section explains the path initialization algorithm. Section 3 outlines the FMM applied to irregular triangle meshes. Following, section 4 details the proposed algorithm. The results are given in section 5 and finally the conclusions of the work are extracted in section 6.

2 Sampling-Based Path Initialization

The reason we choose to use a sampling-based algorithm as a path initialization is because these algorithms are known to be the fastest to provide a reliable path between two points when no previous experience is taken into account.

In this case, we will use the Rapidly-exploring Random Trees (RRT) algorithm which is one of the basic algorithms within the sampling-based group [2]. However, since most of the sampling-based methods share the same properties (fast response, non-smoothness, low obstacles clearance...) [3] any kind of sampling-based algorithm is suitable for the purpose of this paper.

The RRT algorithm is detailed in algorithm 1. Formalizing, the RRT creates a tree T, which is a set of N vertices V and $N-1$ edges connecting those vertices E, $T(V, E)$. Following, the main components of the algorithm are described:

- *Sample(i)* Provides uniformily random samples from the obstacles-free space.
- *Nearest*(V, x_{new}) Returns the closest vertex $v \in V$ to the point given in the argument x_{new}.
- *Steer*$(x_{new}, x_{nearest})$ Creates a path σ between the two points given. Usually, it is a straight line between those two points.
- *CollisionFree*(σ) Returns true if the path given σ is collision free.

The finishing condition in this case is to sample N times. However, this can cause that the algorithm ends and it does not find any possible path. Then, the finishing condition can be easily changed for any other, e.g. one of the vertices of the tree is close enough to the goal point. Figure 1 shows a tree expanding using as ending condition the number of vertices to expand N. Figure 2 plots the paths obtained with RRT when the ending condition is to reach a vertex close to the goal point.

3 Fast Marching Method

The FMM are a set of algorithms for computing consistent distance maps. The first approach was based on regular orthogonal grids [10,11]. Later, these algorithms were extended to general triangular meshes [12]. Since triangular meshes are more flexible when describin shapes, we will focus on this version of the FMM.

Algorithm 1. The RRT Algorithm

Input: Initial point x_0.
Output: Tree T.

```
V ← {x_0}; E ← ∅; T ← (V, E);
for i = 1 to N do
    x_new ← SAMPLE(i);
    x_nearest ← NEAREST(V, x_new);
    σ ← STEER(x_new, x_nearest);
    if COLLISIONFREE(σ) then
        V.ADD(x_new);
        E.ADD(x_new, x_nearest);
    end if
end for
return T = (V, E).
```

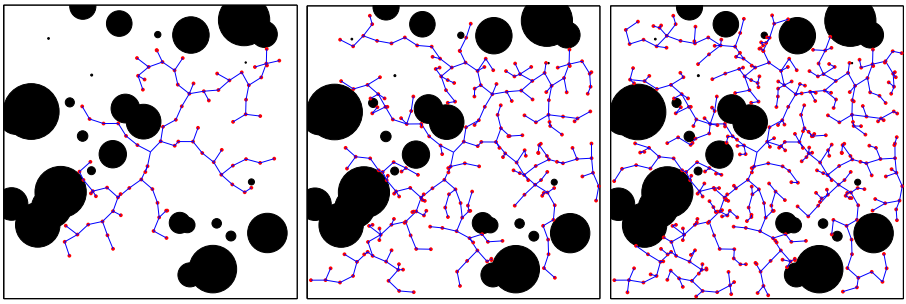


Fig. 1. Examples of a tree created with the RRT algorithm. From left to right: $N = 100, 300, 500$.

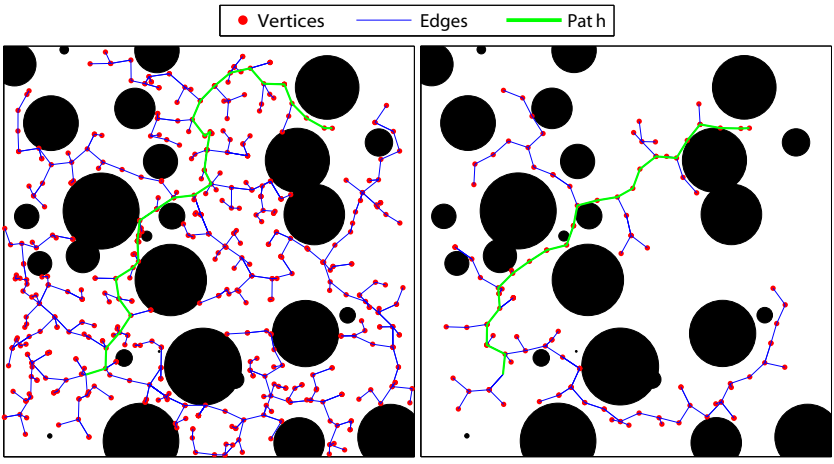


Fig. 2. Paths obtained when planning with the RRT algorithm

In this section we outline the FMM applied in the following sections of this paper. A detailed analysis of the FMM can be found in [9], whose notation we will use. Let X be the surface defined by a triangle mesh and x a coordinate parametrization of X, $x : U \to X$. FMM is a procedure to build a distance map $d(x) = d_X(x_0, x)$ by numerically solving the Eikonal equation:

$$||\nabla_X d(x)||_2 = 1 \tag{1}$$

where ∇_X represents the intrinsic gradient with the boundary condition $d(x_0) = 0$.

Intuitively, FMM simulates a wavefront propagation computing the time of arrival $d(x)$ for every point of the space when the wave propagates with constant, non-negative velocity. Let us suppose that a wave starts propagating at x_0 with $d(x_0) = 0$. This point of the mesh is already *frozen* (its value will never change). By *open* points we denote those points of the mesh which have not been visited yet by the wave, therefore $d(x) = \infty$. Finally, points in the *narrow band* are those belonging to the wave front, acting as an interface between frozen and open points. Algorithm 2 describes the FMM algorithm to compute the distance map $d(x)$.

Algorithm 2. Fast Marching Method

Input: non-obtuse triangular mesh (X, T), source point x_0.
Output: distance map $d : X \to \mathbb{R}$ from the source point.

Initialization.
1: **for all** $x \in X$ **do**
2: $d(x) \leftarrow \infty$;
3: **end for**
4: $d(x_0) \leftarrow 0$;
5: $frozen \leftarrow x_0$;
6: $narrow \leftarrow \mathcal{N}(x_0)$; ▷ Neighbours of x_0.
7: $open \leftarrow X \backslash (frozen \cup narrow)$;
Iteration.
8: **while** $frozen \neq X$ **do**
9: $x_1 \leftarrow \underset{x \in narrow}{\arg\min}\, d(x)$;
10: **for all** $t(x_1, x_2, x_3) \in \{(x_1, x_2, x_3) \in T : x_2 \in frozen \cup narrow, x_3 \in narrow \cup open\}$ **do**
11: $narrow \leftarrow narrow \cup \{x_3\}$;
12: UPDATE(x_1, x_2, x_3);
13: **end for**
14: $narrow \leftarrow narrow \backslash \{x_1\}$; ▷ Updating sets.
15: $frozen \leftarrow frozen \cup \{x_1\}$;
16: **end while**

The main difference between FMM and Dijsktra's algorithms [13] is the *Update* step in the line 12 of algorithm 2. This update step, detailed in algorithm 3

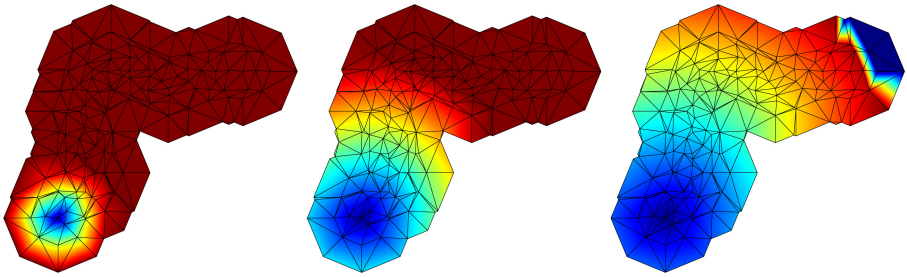


Fig. 3. Front wave propagation in a triangular mesh. The source point is at the bottom of the mesh.

requires two vertices of the same triangle in order to compute the time of arrival of the third vertex. This allows the geodesics to go for any place within the triangle mesh following the gradient of the distance map $d(x)$, which is approximated by the wave propagation direction n.

Algorithm 3. FMM Update Step

Input: non-obtuse triangle with vertices x_1, x_2, x_3 and the corresponding arrival times d_1, d_2, d_3.
Output: Updated d_3.

$V = (x_1 - x_3, x_2 - x_3)$;
$d = (d_1, d_2)^T$;
$Q = (V^T V)^{-1}$;
$p = \frac{1_{2\times1}^T Qd + \sqrt{(1_{2\times1}^T Qd)^2 - 1_{2\times1}^T Q 1_{2\times1} \cdot (d^T Qd - 1)}}{1_{2\times1}^T Q 1_{2\times1}}$;
$n = V^{-T}(d - p \cdot 1_{2\times1})$;
if $QV^T n < 0$ **then** ▷ Monotonicity condition.
    $d_3 \leftarrow \min\{d_3, p\}$;
else
    $d_3 \leftarrow \min\{d_3, d_1 + ||x_1||, d_2 + ||x_2||\}$;
end if

The aforementioned algorithm works for non-obtuse triangular meshes. In case there is any non-obstuse triangle in the mesh, it can be solved by connecting the vertex x_3 to another point on the mesh [12]. Figure 3 depicts the front wave propagation following the FMM. Also, Figure 4 includes the geodesic computed with FMM. In fact, it is not the optimal geodesic, since FMM carries out some approximations when computing $d(x)$. If necessary, it is possible to improve the accuracy of the geodesic by: 1) increasing the number of triangles of the mesh 2) applying other, more advanced FMM methods [14] ot increasing the order of the finite differences Eikonal solver [15]. In any case, the method would become more complex and slower.

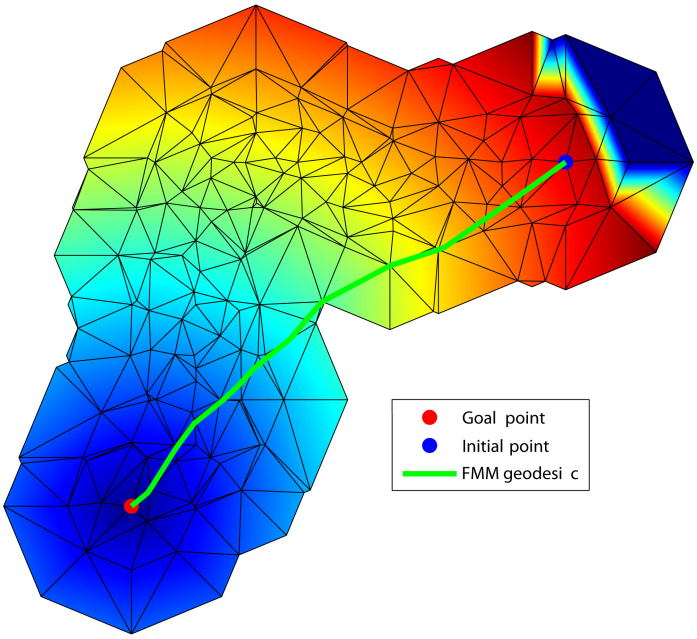


Fig. 4. Geodesic computed with FMM on a triangle mesh. Plotted together with the distances map $d(x)$.

4 Improving Path Quality with Fast Marching

In this section, we describe a novel technique for improving paths. We want to note that it is not an optimization technique since our method is not looking for an optimal solution. However, we propose a fast technique which improves the quality of a given path. In terms of path length, obstacle clearance and smoothness the final path will be closer to the optimal one. Although we cannot guarantee that the final solution is optimal, we provide a deterministic, non-iterative algorithm which will improve the initial path.

The proposed method is based on creating a tube around the initial path. Next, a triangular mesh is created within the tube and the FMM method is applied. However, in this case the velocity of the propagating wave will not be constant, but directly proportional to the distance to the closest obstacle. This is very close to the Fast Marching Square (FM^2) algorithm deeply studied in the recent years [7,16]. Computing geodesics in the distance map generated will provide collision-free, near-optimal paths in terms of path length (since FMM approximates the geodesics). The proposed algorithm, summarized in algorithm 4, is detailed in the following lines.

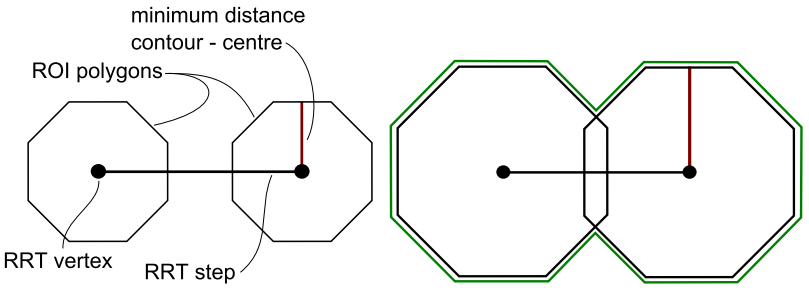
Algorithm 4. Proposed Path Improving Algorithm

Input: Initial path $p(V_p, E_p)$.
Output: Final, improved path q.

for all $v \in V_P$ **do**
  $(V_{ROI,v}, V_{M,v}) \leftarrow$ ComputeVertices(v);
  $V_{ROI} \leftarrow$ UpdateROI$(V_{ROI}, V_{ROI,v})$;
  $V_M \leftarrow (V_M \cup V_{M,v})$;
end for
$X \leftarrow (V_p \cup V_{ROI} \cup V_M)$;
$(X, T) \leftarrow \mathrm{CDT}(X, V_{ROI})$;
$d(x) \leftarrow \mathrm{FMM}(V_{P,0}, V_{P,g})$;
$q \leftarrow$ ComputeGeodesic$(d(x), V_{P,g})$;

4.1 Mesh Generation

Once the RRT (or any other sampling-based algorithm) has been applied, an initial path p is obtained, expressed as a subset of the tree T: $p(V_p, E_p) \subset T(V, E)$. A *region of interest* (ROI) has to be defined, so the mesh is confined around the initial path. This region of interest is created by placing a regular polygon with the center at every vertex V_p. It is important that the polygons cover the area needed to ensure connectivity of the polygons of adjacent vertices. This requirement is easy to overcome: when planning with RRT (or similar), the main parameter is the length of the step given in the *Steer* step. This parameter is the maximum possible distance between two vertices of the tree connected by an edge. Therefore, connectivity of the polygons will be ensured as long as the shortest line between the polygon contour and its center is larger than half of the RRT step parameter, as shown in figure 5.



(a) No connectivity. (b) Connectivity ensured.

Fig. 5. ROI connectivity according algorithms parameters

Once all the polygons have been created, the external contour, expressed as a set of vertices V_{ROI}, is computed by computing the union of all the polygons, as shown in figure 5 b).

With the ROI computed, the next step is to create a triangular mesh inside this ROI. Before creating the mesh, it is mandatory to define the set of vertices V_M that will be used when triangulating. Although a uniform random sampling could be a good option for this case, it could be complex and time consuming to ensure that all the points created randomly are in the ROI. For this reason, we have decided to create a structured sampling pattern.

When calculating the vertices of the polygons for the ROI, also the middle points between the center and vertices are computed and included in the set V_M. Therefore, the vertices to be used in the triangulation will be: $X = (V_p \cup V_{ROI} \cup V_M)$ (figure 6). Note that we are not restricting these vertices to be in the free space. Since the obstacles will be taken into account in the next step, it is not necessary to spend time checking the vertices generated.

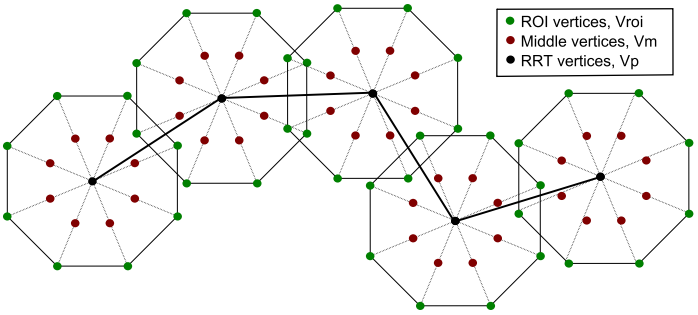


Fig. 6. Set of all the vertices X to be used in the triangulation

Finally, the Constrained Delaunay Triangulation algorithm (CDT) [17] is applied to the set of vertices X and the ROI contour defined by V_{ROI}, obtaining a triangular mesh (X, T). An example is given in figure 7. Any other triangulation algorithm could be chosen here. However, CDT maximizes the minimum angles of the triangles, so it tries to avoid skinny triangles, which is a desirable property for a FMM input mesh. Also, CDT is easily extensible to more dimensions.

4.2 Application of FMM

The next step is to compute the distances map $d(x)$ for the generated mesh (X, T). The FMM is applied from the goal point of the path $V_{p,g}$ until the front wave reaches the initial point of the path $V_{p,0}$. However, the propagation velocity used is not constant. The geodesics, when computing using FMM tend to go closer to the places where the wave can expand faster, following the *least action principle* [9]. Therefore, $d(x)$ is translated into a *times-of-arrival* map.

When updating vertex x_3 from (x_1, x_2), the distance to the initial point d_3 is updated as mentioned in section 3. However, when the propagation velocity

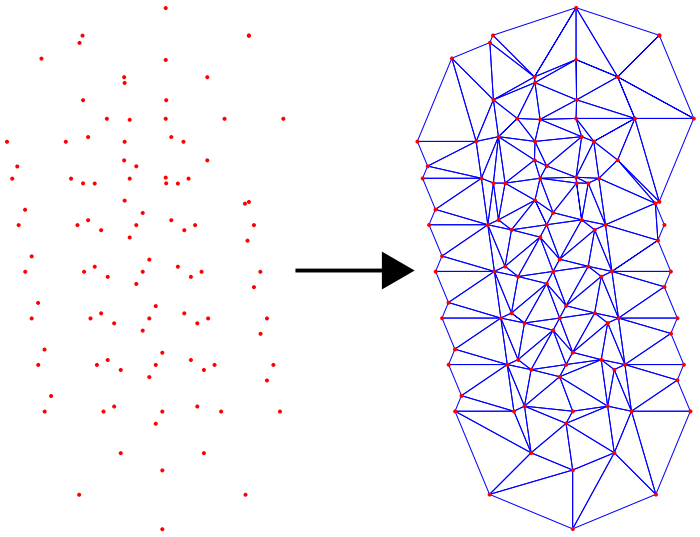


Fig. 7. Example of the CDT algorithm applied to a set of vertices X

is not uniform, after calculating d_3 is is necessary to update it, to compute the time of arrival approximated as:

$$d_3 = \min(d_1 + v_3 \cdot ||x_3 - x_1||, d_2 + v_3 \cdot ||x_2 - x_1||) \tag{2}$$

Where v_3 is the wave propagation velocity towards x_3. In figure 8 the velocities for every point of the mesh are shown. The closer a vertex is to an obstacle, the slower the wave propagates towards that vertex. The resulting wave propagation with this velocities map is shown in figure 9. Comparing this figure with figure 3 it is possible to appreciate the influence of the obstacles when computing the time of arrival value for each vertex.

Finally, in order to obtain the final, improved path gradient descent is applied from the point $V_{p,0}$. Since only one wave was expanded from the goal point $V_{p,g}$, gradient descent will always converge to this point since it is the only minimum in $d(x)$. The final path in comparison with the initial RRT path is shown in figure 10 (octagons were used in the mesh generation).

5 Results

The results will be expressed as a set of metrics and their variation for the initial RRT path and the improved path after applying FMM. The experimental setup comprises 3 different, random environments, 5 random path queries for every enviroment, and every query computed 10 times. Regarding the algorithms, the workspace size was 100×100 (the units are not important, since we are always using the same scale). 100 random obstacles were generated, with a maximum

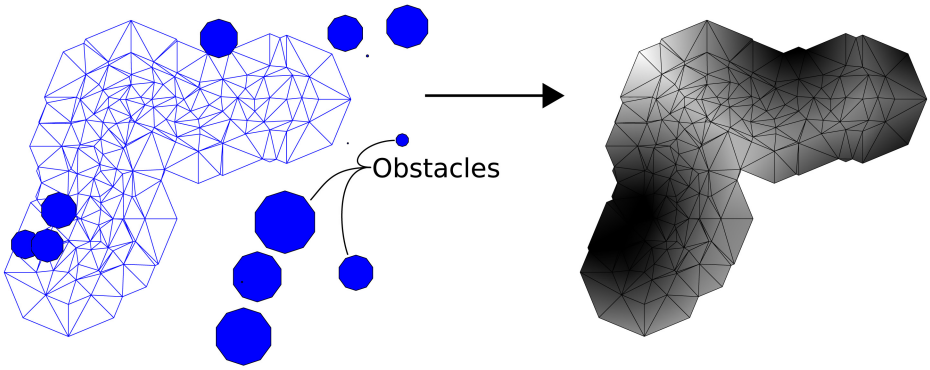


Fig. 8. Velocities map used when propagating the wave. The darker the slower the wave propagates.

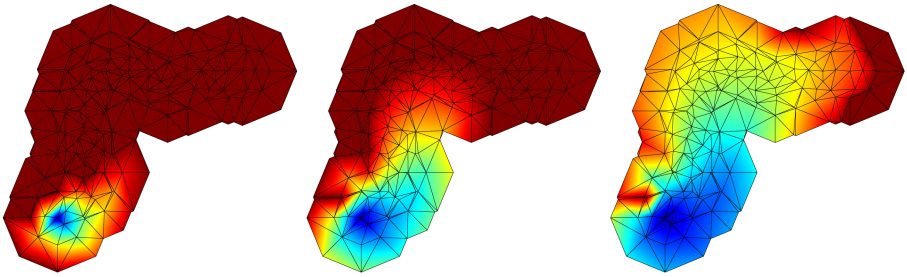


Fig. 9. Front wave propagation in a triangular mesh with varying velocity. The source point is at the bottom of the mesh.

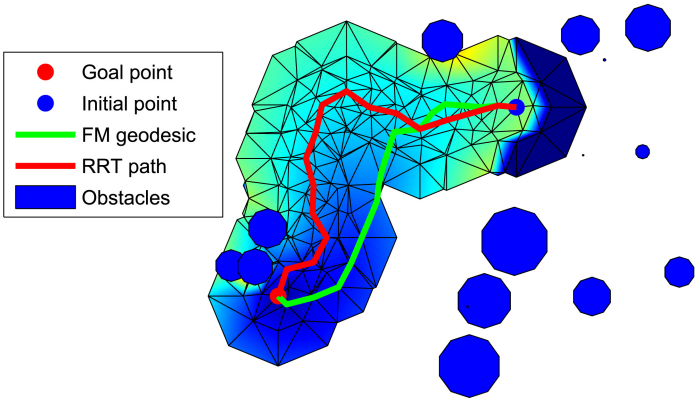


Fig. 10. Comparison of the initial RRT path the geodesic computed with the proposed method

radius of 2.5. The RRT maximum step size was set to 2. In FMM, octogons were used to create the mesh with a radius of 5.

The metrics computed are the following:

- **RRT Computation time** - The time t_{RRT} (in s) the RRT algorithm took to compute the initial path.
- **FM Computation time** - The time T_{FMM} (in s) the proposed algorithm took to improve the path.
- **Deviation in path smoothness** - The smoothness κ' can be measured in many different ways. We will use the smoothness metric given in [18], which represents the standard deviation of the angles along the path. Let α_i be the angle between two consecutive segments of a path divided into m segments. Therefore, $\kappa' = \sqrt{\frac{1}{m-1}\sum_{i=2}^{m}\alpha_i^2}$. The angle taken into account is illustrated in Figure 11. Since we are comparing the deviation between the FMM and RRT paths, we compute the smoothness ratio $\kappa' = \kappa'_{FMM}/\kappa'_{RRT}$.

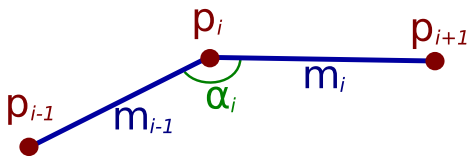


Fig. 11. The angle between two consecutive segments of a path

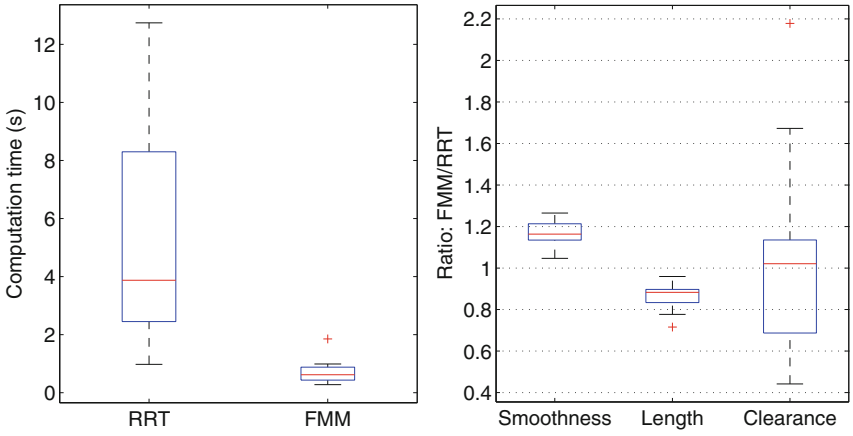
- **Deviation in path length** - The path length l is approximated by dividing the path into n points $P = \langle p_1, p_2, ..., p_n \rangle$ and computing $l = \sum_{i=1}^{n-1} d_E(p_i, p_{i+1})$, where d_E stands for the Euclidean distance. Therefore, we compute the deviation as the ration $l = l_{FMM}/l_{RRT}$.
- **Deviation in Minimum Obstacles clearance** - The metric d_n contains the deviation of the minimum distance of the points along the path to the closest obstacles of the environment. It is computed as the ratio $d_n = d_{n,FMM}/d_{n,RRT}$.

Figure 12 summarises the results obtained for the metrics aforementioned. The graph shows the distribution of results for the 15 experiments done (3 environemts, 5 queries per environment) using the means obtained per every query (computed 10 times). Figure 12 a) shows the computation times of both initialisation path obtained with RRT and the proposed, FMM-based algorithm. Although the absolute values are not meaningful (because of implementation issues), the conclusion from this graph is that the novel method computation time does not vary that much as RRT. The longer the initial RRT path is the longer the optimzation takes, but in a restricted period of time.

From figure 12 b) it is possible to evaluate the proposed algorithm. The smoothness ratio is close to 1.2, which means that the average smoothness improvement is around 20%. Also, note that for all the experiments, this ratio

is higher than 1. Analogously, the length ratio is always lower than 1 (average around 0.85) which supposes an average improvement of the path length of 15%. Lastly, the clearance ratio requires a deeper analysis. The RRT algorithm does not take into account the distance to obstacles when planning. Therefore, the minimum clearance is totally random (it could be, in fact, a safe trajectory). Therefore, it is not a significant result.

However, it is possible to ensure that the FMM method provides a good enough minimum clearance (where *good enough* depends on the application). In our case, the wave propagation velocity was proportional to the distance to the closest obstacle. If this proportion is made stronger (let us say, proportional to the square of the distance to the closest obstacle) the geodesic will be longer in terms of length but with a better minimum clearance.



(a) Computation times (b) Metrics ratios

Fig. 12. Results of the proposed FMM-based algorithm

Finally, figure 13 includes the visualisation of the application of the algorithm.

6 Conclusions

Along this paper we have proposed and analysed a novel method for improving initial paths computed with fast, non-optimal algorithms. The results show that the computation time is acceptable, but there is a huge margin for improvement since the implementation is not optimised.

The key properties of the proposed method are that it is deterministic and non-iterative. For a given initial path, it will always produce the same output and in one iteration (although the underlying FMM method is iterative, we are applying it as a *black box*). In some, very specific cases it is possible to have different outputs because of the CDT, which is unique for most of the meshes.

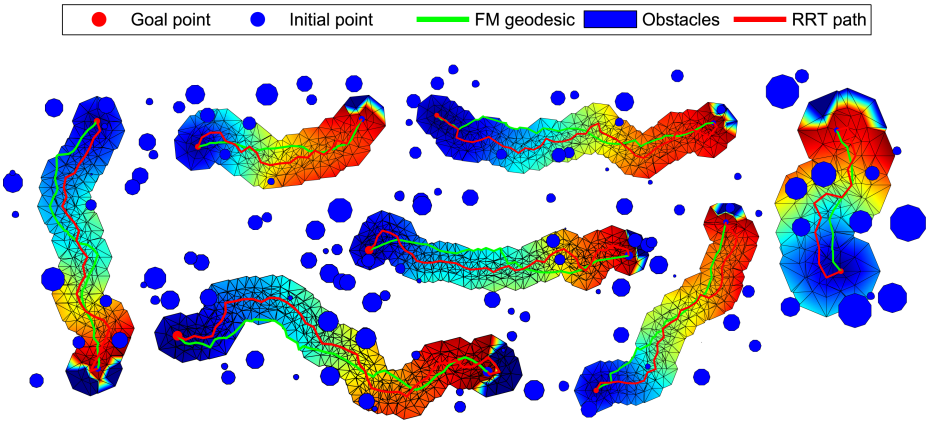


Fig. 13. Different executions of the algorithm and their results

As this work is still under development, there are many ways of improving the results. Testing different, more efficent velocities map can improve the quality of the final path. Also, the addition of other triangulation methods (or a different sampling for the mesh vertices) could improve the computation time.

Apart of the aforementioned, future work will focus on see how the parameters of both RRT and FMM affect the improvement and to try to apply it to more generic scenarios, where obstacles are not represented by circles.

Finally, this paper has focused on a 2D implementation. However, all the components of the algorithm are defined for n-dimensional space.

Acknowledgment. This work was supported by the projects number DPI2010-17772 and CSD2009-00067 HYPER-CONSOLIDER INGENIO 2010 of the spanish MICYT.

References

1. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. on Robotics and Automation 12, 566–580 (1996)
2. LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. Intl. J. of Robotics Research 20, 378–400 (2001)
3. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
4. Keraman, S., Farazolli, E.: Sampling-based algorithms for optimal motion planning. Intl. J. of Robotics Research 30, 846–894 (2010)
5. Perez, A., Karaman, S., Shkolnik, A.C., Frazzoli, E., Teller, S.J., Walter, M.R.: Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms. In: IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (2011)

6. Islam, F., Nasir, J., Malik, U., Ayax, Y., Hasan, O.: RRT*-Smart: Rapid convergence implementation of RRT* towards optimal solution. In: Intl. Conf. Mechatronics and Automation (2012)
7. Valero, A., Gómez, J.V., Garrido, S., Moreno, L.: Fast Marching Method for Safer, More Efficient Mobile Robot Trajectories. IEEE Robotics and Automation Magazine (in press)
8. Yershov, D.S., LaValle, S.M.: Simplicial Dijkstra and A* algorithms: from graphs to continuous spaces. Advanced Robotics 26, 2065–2085 (2012)
9. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Pringer Monographs in Computer Science (2008)
10. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. IEEE Trans. Automatic Control 40, 1528–1538 (1995)
11. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. Proc. National Academy of Sciences 93, 1591–1595 (1996)
12. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. National Academy of Sciences 95, 8431–8435 (1998)
13. Dijkstra, E.W.: A Note on Two Problems in Connexion With Graphs. Numerische Mathematik 1, 269–271 (1959)
14. Hassouna, M.S., Farag, A.A.: Multistencils Fast Marching methods: a highly accurate solution to the Eikonal equation on cartesian domains. IEEE Trans. Pattern Analysis and Machine Intelligence 29, 1563–1574 (2007)
15. Ahmed, S., Bak, S., McLaughlim, J., Renzi, D.: A third order accurate fast marching method for the Eikonal equation in two dimensions. J. on Scientific Computing 33, 2402–2420 (2011)
16. Garrido, S., Moreno, L., Abderrahimm, M., Blanco, D.: FM^2: A Real-time sensor-based feedback controller for mobile robots. Intl. J. of Robotics and Automation 24, 3169–3192 (2009)
17. Chew, L.P.: Constrained Delaunay Triangulations. In: Proc. of the Third Annual Symposium on Computational Geometry, pp. 215–222 (1987)
18. Cohen, B., Şucan, I.A., Chitta, S.: Generic Infrastructure for Benchmarking Motion Planners. In: IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (2012)

A State Lattice Approach for Motion Planning under Control and Sensor Uncertainty*

Adrián González-Sieira**, Manuel Mucientes***, and Alberto Bugarín

Centro de Investigación en Tecnoloxías da Información (CITIUS),
University of Santiago de Compostela, Spain
{adrian.gonzalez,manuel.mucientes,alberto.bugarin.diz}@usc.es

Abstract. Reliable motion planners have to take into account not only the kinematic constraints of the robot but, also, the uncertainty of both the motion and sensor models. In this way, it is possible to evaluate a motion plan based not just on the maximum likelihood path, but also in deviations from that path that have a non-negligible probability. As a result, motion plans are more robust and require a lower number corrections during the online implementation of the plan. In this paper we address the problem of motion planning under uncertainty in both motion and sensor models using a state lattice. The approach manages a very efficient representation of the state space, calculates on-demand the a-priori probability distributions of the most promising states with an Extended Kalman Filter, and executes an informed forward exploration of the state space with Anytime Dynamic A*. We provide results with a differential drive robot under different scenarios, showing the ability of the planner to calculate optimal solutions that minimize the probability of collision and the time to reach the goal state.

1 Introduction

Motion planning algorithms have experimented an impressive evolution in the last years. While the first approaches tried to solve planning and control problems separately, current proposals pose more realistic planners that take into account the kinematic restrictions of the vehicle dynamics and, therefore, can guarantee the generation of feasible solutions. The most successful approaches in this field are planners based on stochastic sampling sampling methods — probabilistic roadmaps (PRM), rapidly-exploring randomized trees (RRT) and their variants— and those based on deterministic sampling —state lattices.

* This work was supported by the Spanish Ministry of Economy and Competitiveness under grants TIN2011-22935 and TIN2011-29827-C02-02. This work was also supported in part by the European Regional Development Fund (ERDF/FEDER) under the project CN2012/151 of the Galician Ministry of Education.

** A. González-Sieira is supported by a FPU grant (ref. AP2012-5712) from the Spanish Ministry of Education, Culture and Sports.

*** M. Mucientes is supported by the Ramón y Cajal program of the Spanish Ministry of Economy and Competitiveness.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_19,

These planners assume a full knowledge of the state and motions, leaving uncertainty issues to the controller that implements the generated paths, which is typically a feedback controller.

Motion uncertainty in autonomous robots originates from the inaccuracy of control actions, unmodeled external influences, and the usage of partial or noisy information about the state. The evolution of the uncertainty depends on the executed motions and the current state of the vehicle, i.e., different generated paths can have very different uncertainties. Planning without taking into account the uncertainty produces unrealistic paths, as the full knowledge of the states and actions assumed by traditional planners can only be reproduced in simulation conditions. Uncertainty causes failures or deviations from the planned path and, frequently, it is necessary to obtain a new plan when the feedback controller cannot return the system to a state belonging to the planned path.

Under the assumption that both the motion and the sensor models are known, it is possible to predict the a-priori —at planning time— probability distributions of the states given the control commands to follow a path. This prediction can be done using a Kalman filter when the stochasticity of the models can be described with Gaussian distributions. With the information provided by these distributions, a planner can be designed to obtain solutions that optimize one or several objectives: minimal probability of collision, minimal covariance along the path, maximum likelihood of reaching the goal state, etc.

In this paper, we present a motion planning algorithm based on a deterministic sampling technique, the state lattice [1], that can obtain optimal and sub-optimal bounded paths taking into account the motion and sensing uncertainty. This approach extracts from the vehicle motion model a set of discrete actions connecting states belonging to the lattice. As the state lattice uses a regular sampling scheme, these actions are position-independent and can be used to connect every pair of states equally arranged. This provides a very efficient representation of the state space which cannot be achieved with stochastic sampling methods.

We use an Extended Kalman Filter (EKF) to estimate the a-priori distributions, as we assume that both the motion and sensor models follow Gaussian distributions. The estimated probability density functions (PDFs) are used to approximate the probability of collision along a path. Then, the algorithm obtains the safest and optimal path to the goal using a discrete search algorithm, Anytime Dynamic A* (AD*) [2], over the state lattice. Our proposal executes an informed forward exploration of the state space, calculating on-demand the a-priori distributions of the most promising states according to a heuristic function that takes into account the vehicle dynamics and the map information, improving the efficiency of the planner.

2 Related Work

The most successful approaches to motion planning are sample based techniques. Among all the approaches, three of them have been widely used: i) PRM [3], that

obtains random samples in free space, building a graph that connects the nearest ones; ii) RRT [4], that builds a tree in free space by iteratively obtaining random samples and connecting them with the nearest existing one; iii) state lattices [1], that exploit the benefits of a regular and deterministic sampling scheme to represent the problem as a directed graph. The first two approaches have many interesting properties, but they assume that the trajectories connecting every pair of states are generated online from the system dynamics —which depending on its complexity can become a difficult problem— while state lattices can work with a set of actions generated offline, which favours an efficient search.

Uncertainty in motion planning comes from different sources: control noise, sensor noise, partial information from the environment, and map uncertainty. Some planners only consider motion uncertainty, as [5], that tries to avoid rough terrain because the result of a control may differ from the expected. The proposal in [6] uses Markov Decision Process (MDP) theory to maximize the probabilities of collision avoidance and of successfully reaching the goal. A generalization of MDP, Partially Observable Markov Decision Process (POMDP), can be used to include sensor uncertainty, but this approach is not scalable to realistic problems due to its high computational complexity [7].

For systems that can be modelled with Gaussian PDFs, the state can be estimated using a Kalman filter, which has proved to be a successful approach for autonomous vehicles equipped with on-board sensors. This is the approach for Belief Roadmap (BRM) [8], which generates paths minimizing the state uncertainty at the goal, but regardless their total cost. Another approach is LQG-MP [9], which obtains the trajectory with the lowest cost and with a probability of collision under a threshold. However this proposal uses an RRT, which makes not possible to guarantee that the optimal path will be found [10]. Other proposals use variations of RRT modified with pruning strategies, but they also inherit the lack of guarantee of finding the optimal solution. The planner presented in [11] uses an extension of RRT which provably converges to the optimal solution when the number of samples tend to infinite, but as all random sampling techniques assumes that the action connecting every pair of states can be efficiently computed online.

3 Problem Formulation

The motion planner obtains optimal collision-free paths taking into account both control and sensor uncertainty. The robot dynamics (f) and sensor models (h) are explicitly given and they are linear or can be locally approximated by their linearization:

$$\begin{aligned} x_t &= f\left(x_{t-1}, u_t, m_t\right), & m_t &\sim (0, M_t) \\ z_t &= h\left(x_t, n_t\right), & n_t &\sim (0, N_t) \end{aligned} \tag{1}$$

where $x_t \in \mathcal{X} = \mathcal{X}^{free} \cup \mathcal{X}^{obs}$ is the state of the robot, $u_t \in \mathcal{U}$ is the control input, m_t is the random process noise, z_t is the sensor measurement and n_t its associated random noise. $\mathcal{X}^{obs}$ is formed by all states where the robot collides with obstacles in the environment.

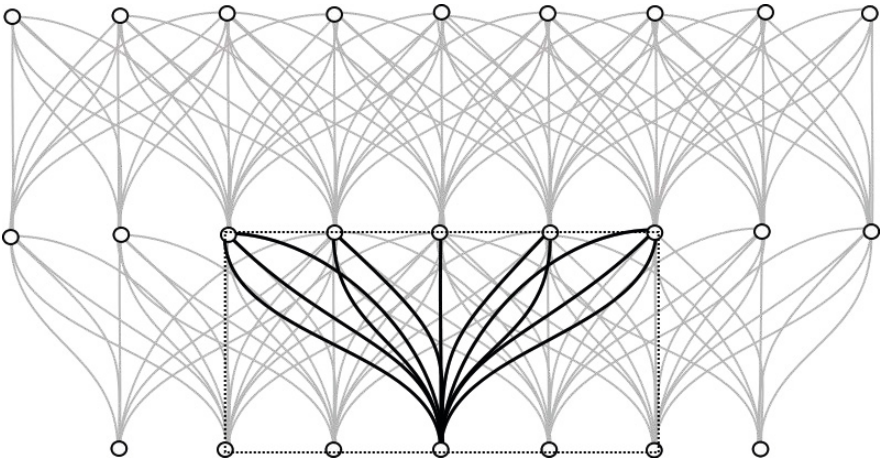


Fig. 1. Rectangular arrangement of the lattice. The canonical control set (black) is independent of the beginning position and is replicated in all equivalent states allowing a very efficient representation of the problem.

Planning relies on a state lattice, which samples the state space $\mathcal{X}$ in a regular form, obtaining a set of states $x_t \in \mathcal{X}^{lat} \subset \mathcal{X}$, named lattice states. These states are connected by a finite set of actions extracted from the vehicle dynamics, the canonical control set $\mathcal{U}$. The generation of $\mathcal{U}$ follows an iterative optimization method based on Newton-Raphson detailed in [12].

The state lattice is generated using a regular discretization scheme. Because of this regularity the canonical actions are independent of the beginning position and the same control commands connect every pair of states in $\mathcal{X}^{lat}$ equally arranged, as shown in Fig. 1.

Each canonical control $u^{a:b} \in \mathcal{U}$ is a composition of control commands that drive the robot from $x^a \in \mathcal{X}^{lat}$ to $x^b \in \mathcal{X}^{lat}$:

$$\begin{aligned} u^{a:b} &= \left(u_1^{a:b}, u_2^{a:b}, ..., u_{t^{a:b}}^{a:b}\right) \\ x^{a:b} &= \left(x_1^{a:b}, x_2^{a:b}, ..., x_{t^{a:b}}^{a:b}\right) \\ x_1^{a:b} &= x^a, x_{t^{a:b}}^{a:b} = x^b \\ x_t^{a:b} &= f(x_{t-1}^{a:b}, u_t^{a:b}, 0) \in \mathcal{X}, \forall t \in [1, t^{a:b}] \end{aligned} \tag{2}$$

where the intermediate states —which may not belong to $\mathcal{X}^{lat}$— are generated with the motion model f with no noise, and $t^{a:b}$ is the total time of the canonical action. Given an initial state of the vehicle ($x^{start} \in \mathcal{X}^{lat}$), and a goal state ($x^g \in \mathcal{X}^{lat}$), a valid path is a combination of N canonical actions, $(u^{start:1}, ..., u^{N-1:g})$, that drive the robot to the goal state without collisions.

4 Uncertainty Prediction

If both the prior probability and the motion and sensor noises are Gaussian, an EKF can be used to estimate the a-priori PDFs for each state $x_t, \forall t \in [1, t^{a:b}]$

given the control inputs along the followed path. The PDFs estimated with the EKF are also Gaussian, so a state is described as:

$$x_t \sim N(\bar{x}_t, \Sigma_t) \tag{3}$$

The a-priori PDFs are calculated when the search algorithm expands a new lattice state and its neighbourhood is generated. The complete process of uncertainty propagation is detailed in algorithm 1.

Algorithm 1. Uncertainty prediction: $uncertainty(x^a)$

for all $x^b \in succ(x^a)$ **do**
  $\bar{x}^0 = \bar{x}^a$
  $\Sigma^0 = \Sigma^a$
  for all $t \in [1, t^{a:b}]$ **do**
    $\widetilde{x}_t = f(\bar{x}_{t-1}^{a:b}, u_t^{a:b}, 0)$
    $\widetilde{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + V_t M_t V_t^T$
    $K = \widetilde{\Sigma}_t H_t^T (H_t \widetilde{\Sigma}_t H_t^T + W_t N_t W_t^T)^{-1}$
    $\bar{x}_t = \widetilde{x}_t + K(z_t - h(\widetilde{x}_t))$
    $\Sigma_t = (I - K H_t)\widetilde{\Sigma}_t$
  end for
  $\bar{x}^b = \bar{x}_t$
  $\Sigma^b = \Sigma_t$
end for
return $x_t^{a:b}, \forall t \in [1, t^{a:b}], \forall x^b \in succ(x^a)$

Given a state $x^a \in \mathcal{X}^{lat}$, and for each successor $x^b \in \mathcal{X}^{lat}$, the uncertainty is estimated applying an EKF iteratively to obtain the intermediate a-priori PDFs along the motion primitive. Each execution of the EKF performs two steps: a prediction step that uses the dynamics model and the current command $u_t^{a:b}$ of the canonical action, and an update step that incorporates the information given by the maximum likelihood measurement z_t. As the prediction of the PDFs is done at planning time without any information of the future execution of the path, the EKF is applied considering that both controls and measurements are those with maximum likelihood.

During the execution of the planned path the vehicle could deviate from the predicted trajectory. The optimal approach to minimize this deviation is to use a feedback controller based in a Linear Quadratic Regulator (LQR) in conjunction with a Kalman estimation of the state, a technique named Linear Quadratic Gaussian (LQG) controller.

5 Search Algorithm

Motion planning is a search problem that can be expressed as a directed graph where the nodes are the discrete states of the lattice ($\mathcal{X}^{lat}$), and the arcs connecting them are the actions of the canonical control set $\mathcal{U}$. As the canonical

control set is generated from the vehicle dynamics, it is clear that the state lattice observes by construction the differential constraints of the robot.

5.1 Anytime Dynamic A*

The search algorithm we have used in this proposal is Anytime Dynamic A* (AD*) [2]. This algorithm is very adequate for motion planning problems because it combines both replanning and anytime search. It is possible to calculate suboptimal bounded solutions adjusting an heuristic inflation parameter, ϵ, depending on the time available to obtain a solution.

The main operations done by the search algorithm are summarized in algorithm 2. While the solution is not found, the algorithm selects the most promising lattice state x^a. This state is the one that minimizes the cost of the the path from the beginning state $c(x^{start:a})$ plus the estimated cost to the goal $e(x^a)$, which is calculated by the heuristic function inflated by the parameter ϵ. Then, the expansion of the state x^a is performed, generating all the paths to its successor states $x^b \in succ(x^a)$ (each point in the path is represented by a Gaussian distribution) with algorithm 1.

Algorithm 2. Main loop of AD* with uncertainty

```
while solution not found do
  select min_{x^a ∈ X^lat} (c(x^{start:a}) + ε · e(x^a))
  uncertainty(x^a)
  for all x^b ∈ succ(x^a) do
    c(x^{start:b}) = c(x^{start:a}) + c(x^{a:b})
  end for
end while
```

In its typical form, AD* executes a backwards search. This makes possible to change the beginning state and obtain new solutions without replanning from scratch. Nevertheless, if we have to estimate the PDF of each state in the path at planning time, the PDFs are propagated along the outgoing trajectories of the most promising state x^a in each iteration of the algorithm, using the procedure introduced in algorithm 1. This function needs the prior PDF of x^a. This means that the distribution of the initial state $x^{start} \in \mathcal{X} \sim N(\bar{x}^{start}, \Sigma^{start})$ needs to be known in order to propagate it along the paths generated by the search algorithm. For this reason, a variant of AD* that executes a forward exploration of the state space was used.

After the PDFs are propagated, the cost of the path from the starting state to each successor state $x^b \in succ(x^a)$ must be updated. This operation just requires the calculation of the cost from x^a to x^b (algorithm 3).

In a planner without uncertainty the cost is the time to drive the robot throughout the path without collisions. However, if the planner takes into account the uncertainty, the paths have to be, at the same time, safe and optimal,

Algorithm 3. Cost evaluation: $c(x^{a:b})$

$c_s = 0$
for all $t \in [1, t^{a:b}]$ **do**
  $S =$ set of samples from $x_t^{a:b} \sim N(\bar{x}_t^{a:b}, \Sigma_t^{a:b})$
  $p_c = \dfrac{\sum_S checkCollision(S_i)}{|S|}$
  $c_s = c_s - \ln(1 - p_c)$
end for
return $\begin{bmatrix} c_s \\ t^{a:b} \\ \Sigma^b \end{bmatrix}$

i.e., the algorithm has to minimize the probability of collision and, at the same time, get an optimal cost —minimizing the time— to reach the goal state. For this reason, the cost function of a path between states $x^a \in \mathcal{X}^{lat}$ and $x^b \in \mathcal{X}^{lat}$ is defined as a vector of three elements that should be minimized: a safety cost, c_s, related to the collision probability along the path, the total time of the path, $t^{a:b}$ and the covariance at the final state, Σ^b.

The probability of collision of a path between the states x^a and x^b is obtained as a combination of the probabilities of collision for each intermediate state, $x_t^{a:b}$ (algorithm 3). Given a state $x_t^{a:b}$, its PDF is sampled, and for each sample in the set S, a collision check is performed. The collision check is done by applying the hyperplane separation theorem between the obstacles in the map and the vehicle shape at the sampled pose. This theorem requires the convexity of the checked forms to success. This is guaranteed by approximating the vehicle shape with a polygon and decomposing the obstacle information in squared cells stored in a grid map.

The motion planner propagates the PDFs and evaluates the cost of an action between a state and its successors on-demand when the search algorithm expands a new state x^a. As the expanded states are the most promising ones, this saves a lot of computation time without the need to applying external pruning strategies as in those approaches based on RRT, being simply integrated in the search.

Our proposal has to compare the cost of the different lattice states in order to select the most promising one, and this cost is a vector of three components. The comparison of the cost of two states, x^a and x^b is solved as:

$$\begin{aligned} c(x^{start:a}) < c(x^{start:b}) \Leftrightarrow (c_s^{start:a} = c_s^{start:b}) \vee \\ (c_s^{start:a} = c_s^{start:b} \wedge t^{start:a} < t^{start:b}) \vee \\ (c_s^{start:a} = c_s^{start:b} \wedge t^{start:a} = t^{start:b} \wedge \Sigma^a = \Sigma^b) \end{aligned} \tag{4}$$

thus, there is a hierarchy in the components of the cost vector: the most relevant element is the safety cost, then the total time of the path and, finally, the covariance of the final state.

The minimization of the cost function calculated in algorithm 3 and this comparison criterion generate safe paths with the lowest possible execution time

and, between paths with equal safety cost and execution time, the one with lowest uncertainty at the final state.

5.2 Heuristic

AD* executes an informed search over the state space. The estimation of the cost of the path between a lattice state and the goal state is generated by the heuristic function e, that influences the order in which the states are expanded. In [13] introduced an heuristic function for state lattices that was defined as the combination of two values: the cost of the path taking into account the kinematic constraints of the vehicle and considering free space (e_{FSH}), and the cost of the path regardless the motion model but using the obstacles information, e_{H2D}:

$$e(x_t) = \max(e_{H2D}(\bar{x}_t), e_{FSH}(\bar{x}_t)), \tag{5}$$

where the heuristic function uses the mean of the PDF ($\bar{x}$), which is a reasonable assumption attending to the optimistic nature of the heuristic.

e_{H2D} is calculated executing a Dijkstra search over an 8-connected 2D grid where the positions match the states in $\mathcal{X}^{lat}$. The exploration begins in the goal state and it is stopped when the explored cost reaches 1.5 times the cost between the initial state and the goal state. This process is executed only once, at the beginning of the planning process. For states not explored by the Dijkstra search the heuristic takes its maximum value, providing a stop condition for the planner when a solution up to the maximum Dijkstra explored cost cannot be found.

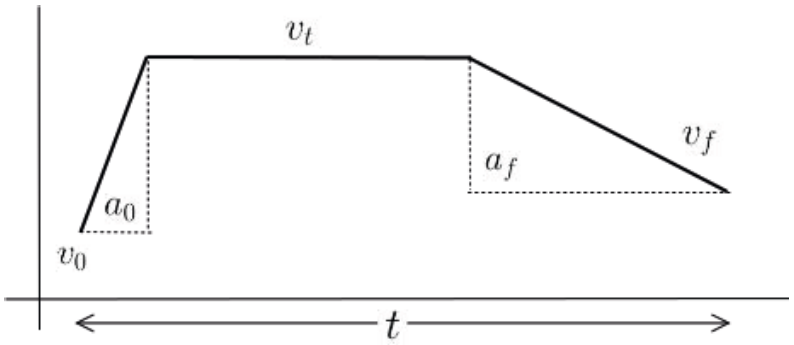
On the other hand, e_{FSH} uses the motion model to estimate the cost to the goal, which makes it a motion planning problem itself and, therefore it cannot be computed online. As this heuristic considers always free space, its values can be precomputed offline and stored in a Heuristic Look-Up Table, following the construction process detailed in [13].

6 Experimental Results

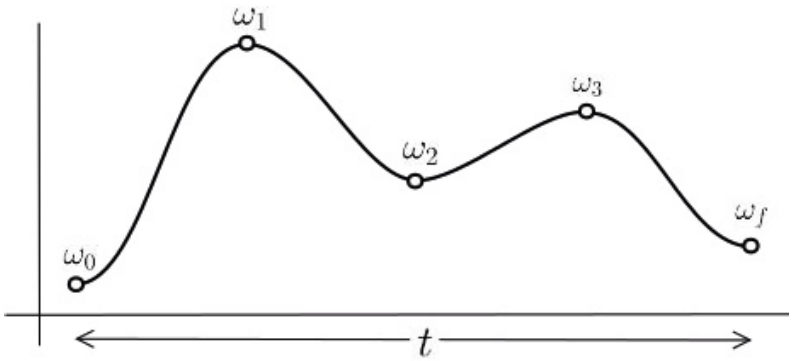
The proposed motion planner was tested with a differential drive robot in a 2D environment using Player/Stage. We have selected different scenarios (landmark positions) and, also, different uncertainty degrees for the sensor model.

The state $x_t = (x_x\, x_y\, x_\theta\, x_v\, x_\omega)^T$ is a 5-dimensional vector containing the vehicle pose and the current linear and angular speed, and the control command $u_t = (u_v\, u_\omega)^T$ is a 2-dimensional vector that defines the linear and angular speeds.

The lattice was built with the following discretization resolutions: 0.5 m in both x_x and x_y; x_θ contains the orientation values of the neighbours of a 16-connected grid, $x_v = \{0, 0.2, 0.5\}$ m/s, and $x_\omega = 0$ rad/s. This means that the lattice states can only take these values as part of the discretization scheme, but for any other state $x_t \notin \mathcal{X}^{lat}$, the state vector can take any value. In fact, the evolution of the linear and angular speeds in the canonical control set is defined by trapezoidal function for the linear velocity and cubic spline function for the



(a) Trapezoidal linear velocity profile.



(b) Cubic-spline angular velocity profile.

Fig. 2. Definition of the variable profiles of u_v and u_ω for a canonical control action

angular velocity. As it can be seen in Fig. 2(a) and Fig. 2(b), x_v and x_ω can take any intermediate value, while the beginning and final ones must observe the state lattice restrictions.

The canonical control set $\mathcal{U}$ used in the experiments was built connecting neighbours of distances (in number of states) 0, 1, 4 and 8, allowing motion primitives up to 4 m long.

The non-linear motion model $f(x_{t-1}, u_t)$ used in the tests is defined as:

$$\begin{bmatrix} x_x + \frac{u_v}{u_\omega}(-\sin(x_\theta) + \sin(x_\theta + u_\omega \Delta t)) \\ x_y + \frac{u_v}{u_\omega}\cos(x_\theta) - \frac{u_v}{u_\omega}\cos(x_\theta + u_\omega \Delta t) \\ x_\theta + u_\omega \Delta t \\ u_v \\ u_\omega \end{bmatrix} \tag{6}$$

where Δt is the time of application of the action, and the control noise covariance is calculated as a percentage of the control command u_t:

$$M_t = 0.1 I u_t \tag{7}$$

The sensor measurements come from the nearest landmark in the environment, $b = (b_x, b_y)$, according to the following observation model $h(x_t, b)$:

$$\begin{pmatrix} h_d \\ h_\theta \end{pmatrix} = \begin{pmatrix} \sqrt{(x_x - b_x)^2 + (x_y - b_y)^2} \\ \arctan(x_y - b_y, x_x - b_x) - x_\theta \end{pmatrix} \tag{8}$$

The measurement noise covariance is also defined as a percentage of the value of the measurement:

$$N = \begin{cases} 0.05 I h & \text{if } h_d <= 5 \\ 0.3 I h & \text{if } h_d > 5 \end{cases}, \tag{9}$$

which introduces a variable measurement noise depending on the distance between the landmark and the vehicle.

Collision check is a very frequent operation and could be very time consuming without a careful implementation (specially for a grid map with cell resolution of 0.125 m). To optimize this operation, offline we have centered the vehicle shape in a set of poses around the central point of a cell, calculating for each pose the list of adjacent cells occupied by the shape of the robot. When a collision check has to be performed online, the difference between the central position of the shape and the central point of the nearest cell is calculated. Given that difference, the closest sample is picked and the collision is checked with the list of adjacent cells that was generated offline. To guarantee that this loss of precision in the collision check does not affect the safety of the generated paths, the vehicle shape is enlarged with a safety margin that matches the sampling resolution of the offline cells list.

For the example in Fig. 3, a motion planner that obtains the optimal path without taking into account the motion and sensor uncertainty, obtains solutions with a high collision probability. This may cause failures when the vehicle deviates from the planned path, even in the case of executing it with a feedback controller.

The configuration of the landmarks in the environment directly affects the evolution of the covariance and, therefore, the solution obtained by the planner. In the example of Fig. 3(a) the localization is good near the upper doors, so the solution is similar to the optimal path without uncertainty but avoiding the proximity to the corners. When the landmarks are moved down, as in Fig. 3(b) and Fig. 3(c), the safest solution avoids the first narrow door, navigates through zones with good measurements and safely passes through the second one. More detailed results for these executions are given in Table 1, which contains the average values for 5 executions of the planner in the same conditions. Stochasticity comes from the sampling of the PDF to estimate the collision probability, p_c, which is obtained by combination of the collision probabilities of all states along the path. Even in the case of similar solutions as shown in Fig. 3(a), the proximity of a single state to obstacles significantly increases p_c.

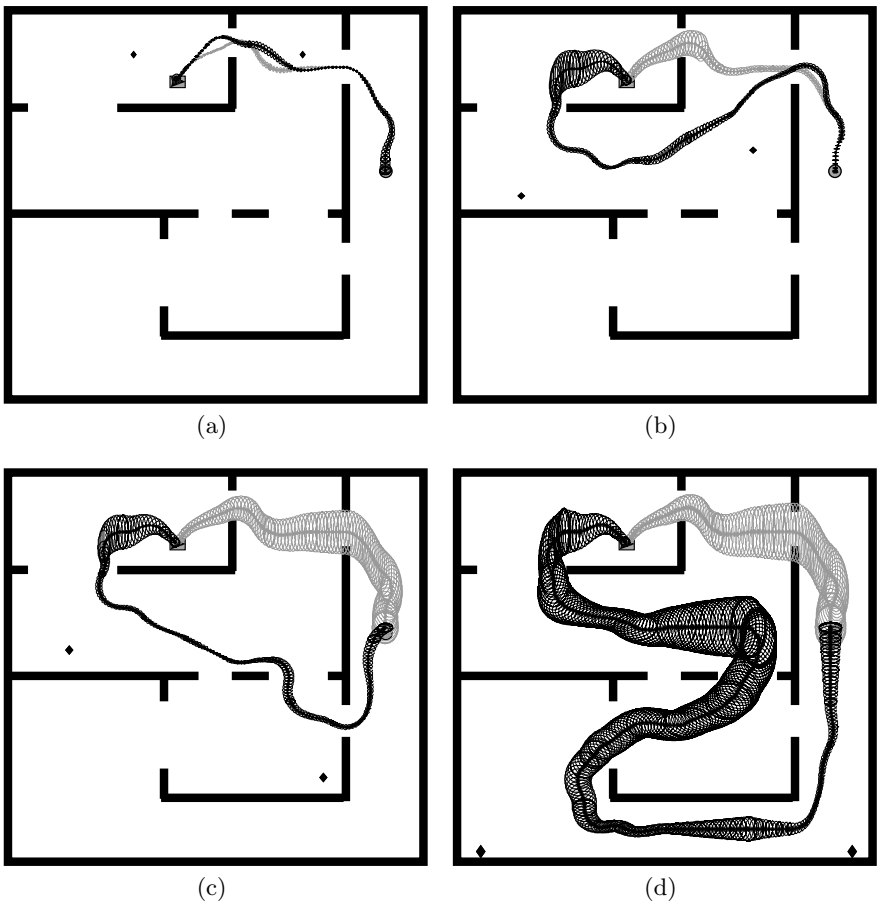


Fig. 3. Comparative between the best paths with (in black) and without (in gray) taking into account the uncertainty. Each figure shows a different positioning of the landmarks (diamond symbols). The ellipses represent the double of the deviation (of x_x and x_y) for the states on the path.

"O-" prefix denotes the optimal solution using a planner that does not take uncertainty into account, while "U-" indicates the solution obtained when planning with uncertainty.

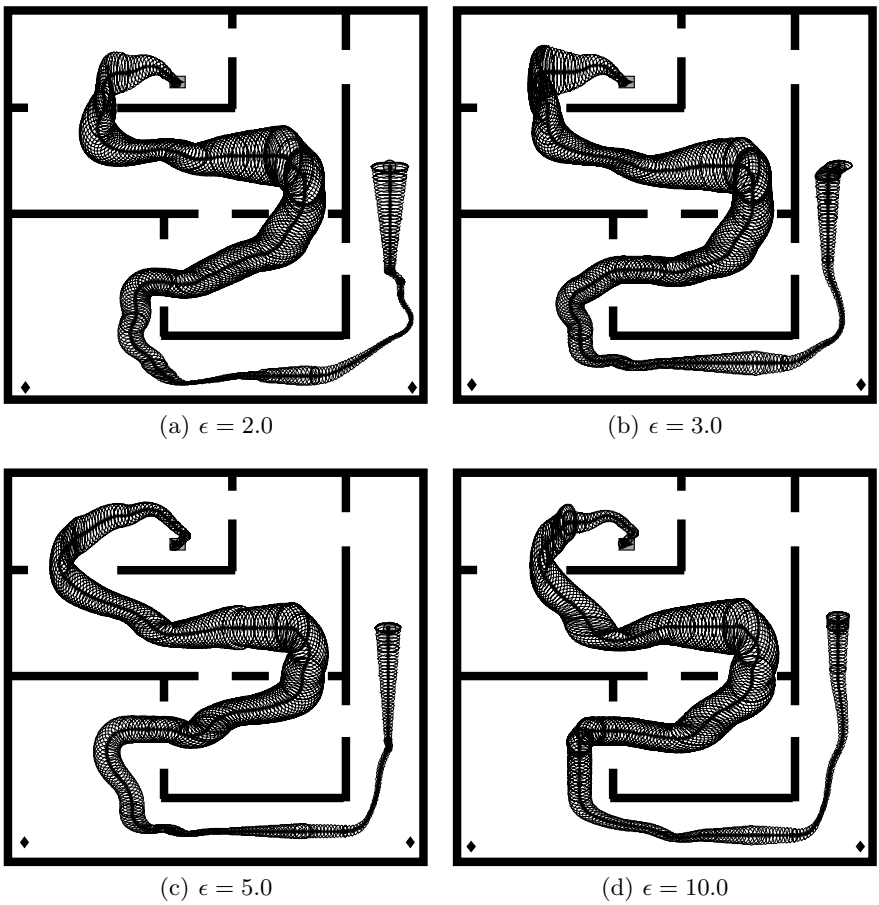
For the worst case, in Fig. 3(d), the measurements are very noisy in the upper region of the map, so the covariance grows fast and passing trough narrow areas has a high probability of collision. If the landmarks are set in the lower region of the environment, the safest solution is to take the longest path, as it obtains measurements with a lower uncertainty to safely reach the goal.

The number of lattice states expanded by the search algorithm and the planning time depend on the degree of uncertainty with respect to the optimal path without uncertainty. The reason is that the heuristic function does not contain information about the probability of collision or the covariance evolution, so if the optimal path is blocked due to a high probability of collision, the heuristic may underestimate the cost to the goal region, causing an increase in the number of expansions.

Figure 4 shows a comparative between different sub-optimal bounded solutions. The landmark positions are the same as in Fig. 3(d), but in this case the algorithm was run for different values of the heuristic inflation parameter, ϵ. As shown in Table 1, the higher the value of ϵ, the lower the execution time and the number of expansions of the algorithm. These values have an upper bound given by the loss of information caused by estimating the cost to the goal without uncertainty, i.e., for high values of ϵ the heuristic function does not provide useful information and, therefore, the positive effect of a lower number of state expansions disappears.

Table 1. Details of the experiments

Problem	**Planning**			**Solution**	
	ϵ	**Time (s)**	**Expansions**	p_c	**Cost (s)**
O-3(a)	1.0	1	146	0.51	31.02
U-3(a)	1.0	1	121	0	31.16
O-3(b)	1.0	1	146	0.97	31.02
U-3(b)	1.0	48	4,344	0	57.40
O-3(c)	1.0	1	146	1	31.02
U-3(c)	1.0	114	10,699	0	65.17
O-3(d)	1.0	1	146	1	31.02
	1.0	299	25,771	0	123.48
	2.0	204	16,183	0	132.82
U-3(d)	3.0	198	15,779	0	128.71
	5.0	198	15,856	0	139.29
	10.0	196	15,634	0	152.99



(a) $\epsilon = 2.0$ (b) $\epsilon = 3.0$

(c) $\epsilon = 5.0$ (d) $\epsilon = 10.0$

Fig. 4. Comparative of different sub-optimal bounded paths for several values of ϵ

7 Conclusions

We have presented a motion planning algorithm that takes into account the uncertainty of the motion and sensor models. The proposal is based on a search algorithm that obtains safe and optimal paths over a state lattice. The planner uses an EKF to predict the PDFs of the different states throughout the possible paths, and assigns them a probability of collision. Moreover, the planner can also get anytime solutions calculating safe sub-optimal bounded paths.

The performance of the motion planner was tested with several examples with different uncertainty conditions. All the results show a good performance both in terms of the probability of reaching the goal state without collisions and the time to travel throughout the path. Further work is still necessary to improve the PDFs prediction and, also to reduce the planning time. As many components of the planer are computed offline (like the canonical control set), the same

approach could be applied to the PDFs if they can be expressed as a function of the initial covariance, thus improving the efficiency.

References

1. Pivtoraiko, M., Knepper, R.A., Kelly, A.: Differentially constrained mobile robot motion planning in state lattices. Journal of Field Robotics 26(3), 308–333 (2009)
2. Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., Thrun, S.: Anytime dynamic A*: An anytime, replanning algorithm. In: Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS), pp. 262–271 (2005)
3. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
4. LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. The International Journal of Robotics Research 20(5), 378–400 (2001)
5. Melchior, N.A., Simmons, R.: Particle RRT for path planning with uncertainty. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1617–1624 (2007)
6. Alterovitz, R., Siméon, T., Goldberg, K.: The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty. In: Robotics: Science and Systems, pp. 246–253 (2007)
7. Papadimitriou, C.H., Tsitsiklis, J.N.: The complexity of Markov decision processes. Mathematics of Operations Research 12(3), 441–450 (1987)
8. Prentice, S., Roy, N.: The belief roadmap: Efficient planning in belief space by factoring the covariance. The International Journal of Robotics Research 28(11-12), 1448–1465 (2009)
9. Van Den Berg, J., Abbeel, P., Goldberg, K.: LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. The International Journal of Robotics Research 30(7), 895–913 (2011)
10. Karaman, S., Frazzoli, E.: Incremental sampling-based algorithms for optimal motion planning. In: Robotics: Science and Systems (2010)
11. Bry, A., Roy, N.: Rapidly-exploring random belief trees for motion planning under uncertainty. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 723–730 (2011)
12. González-Sieira, A., Mucientes, M., Bugarín, A.: Anytime Motion Replanning in State lattices for Wheeled Robots. In: Workshop on Physical Agents (WAF), pp. 217–224 (2012)
13. Likhachev, M., Ferguson, D.: Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles. The International Journal of Robotics Research 28(8), 933–945 (2009)

Fusion of the Odometer and External Position System with Different Coverage Areas

David Gualda, Daniel Ruiz, Jesús Ureña, and Juan C. García

Electronics Department, University of Alcala,
Alcalá de Henares, Spain
david.gualda@depeca.uah.es

Abstract. This paper presents an Extensive Local Positioning System for mobile robots (MRs) that fuses the information from a mobile robot (MR) internal odometer sensor and the measurements of distances from two external systems: an Ultrasonic Local Positioning System (ULPS) with high accuracy but a short coverage area and a set of WiFi beacons that create a WiFi Local Position System (WLPS) with long coverage area but low accuracy. When the MR navigates on the working area, it continuously obtains its relative position thanks to the internal odometer, and the accumulative errors are avoided fusing the WLPS information. When the robot is on a place where the positioning data from the ULPS are available, the mobile robot can obtain its position with high accuracy. The ULPS and the WLPS have been built and tested experimentally.

1 Introduction

Positioning of the mobile robot to accomplish a given task and achieve autonomous travel is an important technique and currently an important research field. Normally mobile robots use an odometer to calculate its position and orientation, these measurements are very accurate but it increases with time.

A solution to solve the accumulated positioning error is to use a Local Position System (LPS) [1,2]. Within these kinds of systems, Ultrasonic Local Positioning Systems (ULPS) provide high accuracy (with errors below the centimeter in some cases) at close ranges and allow the use of low cost sensors [3], the main problem of these systems is that they have a small cover area.

Alternatively LPS based on WiFi provide a high cover area but have low accuracy, normally around meters [4]. This low accuracy is due to the error in the transformation from the power of the WiFi signal received (RSS) to distance[5].

By fusing all the information from the different technologies (odometer, ultrasound, WiFi) it is possible to obtain a better estimation of the robot position and angle avoiding the odometer cumulative errors. The most common fusing algorithm is the Extended Kalman Filter (EKF) [6]; this filter is optimal for Gaussian noise as it minimizes the quadratic mean error between the different data. In addition to the EKF there are other Bayesian techniques to merge the information, such as particle filters [7] that use a Monte-Carlo method to describe the probability density function or the H-Infinite filter that minimizes the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_20,

worst case estimation error [8]. This last filter can be used in systems where the statistics errors are unknown or are not well characterized.

This paper introduces the combination of an ULPS and a WiFi Local Position System which allows a robot mobile to navigate in an extensive area using an H-infinite filter to merge the odometer, the WiFI and the ultrasonic information from the LPS. In order to improve the accuracy of the WiFi LPS in the calculation of the robot distance, a genetic algorithm it is used to calculate the formula to transform the RSS to distances inside the coverage area. When the robot is in a place where the ULPS is not available, the filter can use the WiFi information to avoid errors due to wheel slippage, small obstructions on the floor or related with the encoder pulse counting, but when it reaches to a place covered by the ULPS it uses the Ultrasonic information to improve the MR position.

This paper is organized as follow: Section II describes the genetic algorithm to obtain the WiFi propagation model; Section III introduces the Robot navigation algorithm with the H-Infinite filter to merge the data; Section IV shows different tests in a real system and, finally, some conclusions are discussed in Section V.

2 RSSI WiFi Distance Estimation

In order to built the WiFi LPS system, some Access Points (APs) of WLAN infrastructure of the Polytechnic School Building (EPS) have been used in this work. Figure 1 shows the location of the main access points in the workspace.

To integrate WiFi technology in the fusion algorithm is neccesary to know the distance from the point to each AP. The classical method to determinate these distances based on RSSI measures is by the traditional Hata-Okumura [9][10] Path Loss (PL) model:

$$PL = PL\,(d_0) + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + s \tag{1}$$

Where d is the distance to the AP, $PL(d_0)$ is the path loss value when the receiver is at d_0 meters of distance, usually this value is one meter; γ is equal to the path loss exponent that indicates how fast path loss increases with the distance and s is a zero mean Gaussian noise which has a high variance, in indoor enviornments is about several meters. This model is very innacurate, thus a new model has been proposed. This model calculates in a straighforward way the distances to the APs using several RSSI measured in the particular point. It is based on Genetic Programming Algorithm (GP)[11]. This algorithm is an evolution of Genetic Algorithms (GA) [12] and it was introduced by Koza in 1992.

GP allows to obtain a logical or math solution based on an empirical data set of the complex problem, this solution is composed by primitive arithmetic operators (sum, sin, cos, etc.).

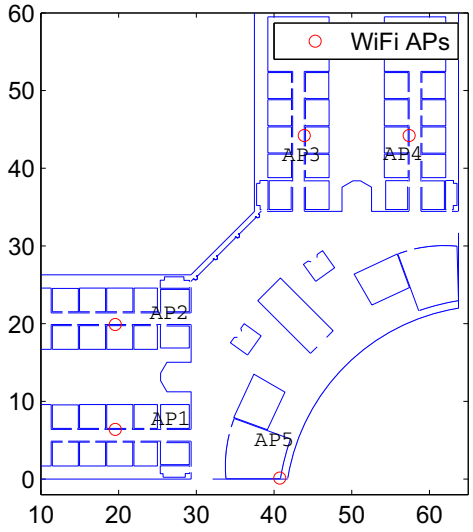


Fig. 1. WiFi APs

Populations of a lot of functions are genetically bred using Darwing principle of survival and reproduction. the funcion which solves the particular problem is created from the combination of Darwinian selection and genetic operations (reproduction, crossover and mutation). Each function in GP has a hierarchical structure in which, the size, the shape and the contents can dynamically change during the process. Before run the process, is neccesary to specify a set of input and output empirical data. The relation between these data will be calculated by GP obtaining a function. Furtermore, a fitness function must be defined in order to obtain the best individual. GP has been expanded in the last few years to solve problems in different areas. In [13]competitive results produced until 2010 by GP is discussed.

In our case, we have 38 training points, spread for all the working area as figure 2 shown.

For each training points, an average of 1000 RSSI values of APs is obtained. The ground truth of the positions have been obtained by a laser meter, thus the error is less than 5cm.

To determinate each expression GP requires a training data set. The input file is a matrix that contains an average of RSSI measures of P training points.

$$\begin{bmatrix} avgRSSI^1_{AP_1} & \dots & avgRSSI^1_{AP_5} \\ \dots & \dots & \dots \\ avgRSSI^P_{AP_1} & \dots & avgRSSI^P_{AP_5} \end{bmatrix} \tag{2}$$

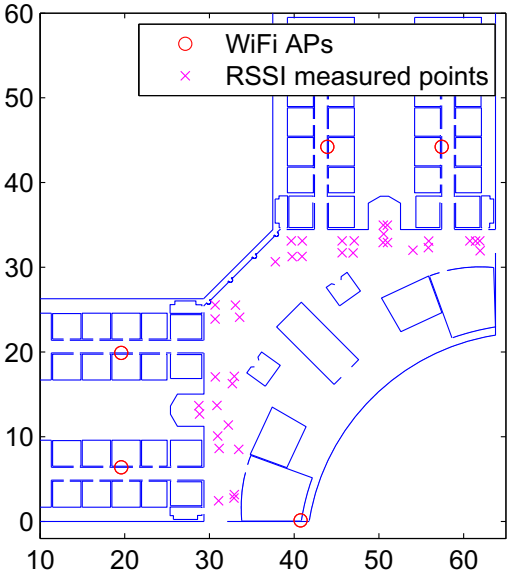


Fig. 2. RSSI measured points

Output file is a vector which is composed by all distances from training points to the particular AP_n which analytical equation wants to be found.

$$\left[d^1_{AP_n} \ldots d^P_{AP_n}\right]^T \tag{3}$$

To evaluate the functions obtained for each iteration of GP, a fitness function as been defined previously wich is the mean of relative distances between the estimate and the real distance. The best individual is which minimizes the expression 4:

$$g = \frac{1}{P}\sum_i^P \frac{|\hat{d^i_{AP_n}} - d^i_{AP_n}|}{d^i_{AP_n}} \tag{4}$$

where $\hat{d^i_{AP_n}}$ is the estimate distance between $AP_n$and the point i, and $d^i_{AP_n}$ is the real distance. This algorithm is repeated for each AP_n, so a model that estimates the distance for each AP is obtained by GP using all RSSI measures of the training points. Expression 5 shows the structure of the functions:

$$\begin{cases} d_{AP_1} = f\left(RSSI_{1:n}\right) \\ \ldots \\ d_{AP_m} = f\left(RSSI_{1:n}\right) \end{cases} \tag{5}$$

To evaluate the model, a set of test data have been captured in the work area. The mean relative error is 7.6 cm per meter for the training data and 17.2 cm per meter for the testing data. Figure 3 shows the Cumulative Distribution Function (CDF) of the relative error using the training data (red) and the testing data (blue) for GP model and Hata-Okumura propagation model (magenta). It can be observed that proposed model is more accurate than the classical model.

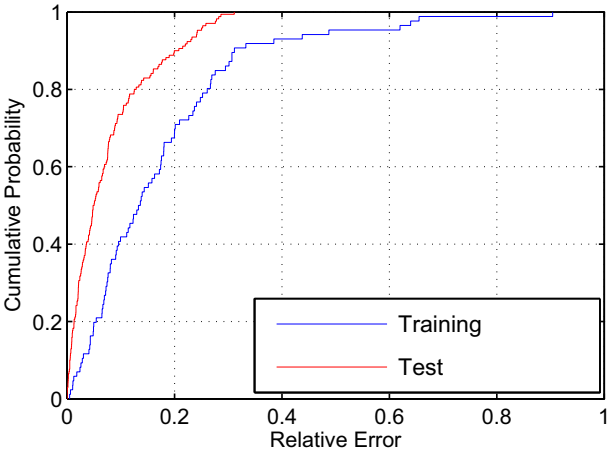


Fig. 3. CDF of relative errors

3 Robot Navigation

In our system the robot can use up to three different sources of information to calculate its position: odometer, ultrasound and WiFi signals, so a Minimax filter is used to fuse them, and thus obtain a more accurate robot location. The Minimax filter is similar to the Extended Kalman Filter but minimize the maximum error in the measurements instead of the mean quadratic error, this property make the filter more robust in systems where the noise is not very well characterized.

In our case we assume that the height (z) of the robot is known so we only obtain the two dimension robot position (x, y) and its orientation (θ), so the state vector of the filter at time k is:

$$\hat{\mathbf{q}}_k = [x_k,\, y_k,\, \theta_k]^T \tag{6}$$

The equations of the Minimax filter adapted to our system are:

Prediction stage:

$$\hat{\mathbf{q}}_{k,odo} = f(\hat{\mathbf{q}}_{k-1,odo}) \tag{7}$$

Update stage:

$$\begin{aligned}
\mathbf{L}_k &= \left(\mathbf{I} - \gamma \cdot \mathbf{P}_{k-1} + \mathbf{C}_k^T \cdot \mathbf{V}^{-1} \cdot \mathbf{C}_k \cdot \mathbf{P}_{k-1}\right)^{-1} \\
\mathbf{K}_k &= \mathbf{A}_k \cdot \mathbf{P}_{k-1} \cdot \mathbf{L}_k \cdot \mathbf{C}_k^T \cdot \mathbf{V}^{-1} \\
\hat{\mathbf{q}}_k &= \hat{\mathbf{q}}_{k,odo} + \mathbf{K}_k \left(\mathbf{m}_k - \hat{\mathbf{m}}_k\right) \\
\mathbf{P}_k &= \mathbf{A}_k \cdot \mathbf{P}_{k-1} \cdot \mathbf{L}_k \cdot \mathbf{A}_k \cdot \mathbf{R}
\end{aligned} \tag{8}$$

Where:

$f(\hat{\mathbf{q}}_{k-1,odo})$ is the function that obtain the robot position at time k, using the odometer information (eq. 9).

$\mathbf{A}_k$ is the Jacobian matrix of the robot regarding the state vector (section 3.1).

$\mathbf{C}_k$ is the Jacobian matrix of the measurement (Ultrasound and Wifi) regarding the state vector (section 3.5).

$\mathbf{K}_k$ is the filter gain.

$\mathbf{m}_k$ is the vector with the measurements from the Ultrasonic beacons and the WiFi stations at time k.

$\hat{\mathbf{m}}_k$ is the estimated measurements from the ultrasound beacons and the WiFi station to the estimated position of the robot.

$\mathbf{V}$ and $\mathbf{R}$ are the odometer and measurements noise matrices respectively.

$\mathbf{I}$ is the identity matrix.

$\mathbf{P}_k$is the filter covariance matrix.

γ is a filter parameter, in our case is set to 1e-3.

3.1 Odometer Dynamics

For the actual test, we use a pioneer DX5000 robot, the robot position using the odometer information in this model can be obtained using the following equations:

$$\hat{\mathbf{q}}_{k,\,odo} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \triangle D_k \cdot cos\left(\theta_k\right) \\ y_{k-1} + \triangle D_k \cdot sin\left(\theta_k\right) \\ \theta_{k-1} + \triangle \theta_k \end{bmatrix} \tag{9}$$

Where $\triangle D_k$ is the distance travelled by the robot between the instant $k-1$ and the instant k. $\triangle \theta_k$ is the angle increment in the robot orientation between the two instant of time.

The dynamic model of the robot can be obtained differenciating $\hat{\mathbf{q}}_{k,\,odo}$ regards the state vector of the filter.

$$\mathbf{A}_k = \frac{\partial \hat{\mathbf{q}}_{k,\,odo}}{\partial \mathbf{q}_k}_{\,|q_k = \hat{q}_k} = \begin{bmatrix} 1 & 0 & -\triangle D_k \cdot sin\left(\theta_k\right) \\ 0 & 1 & \triangle D_k \cdot cos\left(\theta_k\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

3.2 Ultrasound Dynamics

In the proposed system the mobile robot is not synchronized with the ultrasound system, so instead of knowing the absolute distance, it is measured the increments of distance between the receptions of the signals taking one of the signal as reference (Hyperbolic trilateration [14]).

Assuming that it is installed an Ultrasound system of N transducers and the first one is taking as the reference, the distance increment of the transducer i placed on (bx_i, by_i, bz_i) and the mobile robot located on the position (x_k, y_k, z_k) at time k can be expressed as:

$$\begin{aligned}\triangle d_{i1,k} = &\sqrt{(x_k - bx_i)^2 + (y_k - by_i)^2 + (z_k - bz_i)^2} \\ &- \sqrt{(x_k - bx_1)^2 + (y_k - by_1)^2 + (z_k - bz_1)^2}\end{aligned} \tag{11}$$

So the vector of measurements at time k is:

$$\mathbf{du}_k = [\triangle d_{21,k}, \, , \triangle d_{31,k}, \cdots , \triangle d_{N1,k},]^T \tag{12}$$

Differentiating (12) in regards to the state vector it is obtained the dynamic model of the ultrasound measurements:

$$\mathbf{U}_k = \frac{\partial \mathbf{du}_k}{\partial \mathbf{q}_k}_{|q_k = \hat{q}_k} = \begin{bmatrix} \frac{\partial \triangle d_{21,k}}{\partial x_k} & \frac{\partial \triangle d_{21,k}}{\partial y_k} & \frac{\partial \triangle d_{21,k}}{\partial \theta_k} \\ \vdots & \vdots & \vdots \\ \frac{\partial \triangle d_{N1,k}}{\partial x_k} & \frac{\partial \triangle d_{N1,k}}{\partial y_k} & \frac{\partial \triangle d_{N1,k}}{\partial \theta_k} \end{bmatrix} \tag{13}$$

Where:

$$\begin{aligned}\frac{\partial \triangle d_{i1,k}}{\partial x_k} &= \frac{(x_k - bx_i)}{\sqrt{(x_k - bx_i)^2 + (y_k - by_i)^2 + (z_k - bz_i)^2}} \\ &\quad - \frac{(x_k - bx_1)}{\sqrt{(x_k - bx_1)^2 + (y_k - by_1)^2 + (z_k - bz_1)^2}} \\ \frac{\partial \triangle d_{i1,k}}{\partial y_k} &= \frac{(y_k - by_i)}{\sqrt{(x_k - bx_i)^2 + (y_k - by_i)^2 + (z_k - bz_i)^2}} \\ &\quad - \frac{(y_k - by_1)}{\sqrt{(x_k - bx_1)^2 + (y_k - by_1)^2 + (z_k - bz_1)^2}} \\ \frac{\partial \triangle d_{i1,k}}{\partial \theta_k} &= 0\end{aligned} \tag{14}$$

3.3 WiFi Dynamics

In the section XX we have presented a formula to obtain the absolute distance between the WiFI stations and the mobile robot, so assuming a system with M

WiFi base stations, the distance between the WiFi station j placed on (bx_j, by_j, bz_j) and the mobile robot located on the position (x_k, y_k, z_k) at time k can be expressed as:

$$d_{j,k} = \sqrt{(x_k - bx_j)^2 + (y_k - by_j)^2 + (z_k - bz_j)^2} \tag{15}$$

So the vector of measurements from the WiFi base station at time k can be expressed as:

$$\mathbf{dw}_k = [d_{1,k}, , d_{2,k}, \cdots , d_{M,k},]^T \tag{16}$$

Differentiating (16) in regards to the state vector it is obtained the dynamic model of the WiFi measurements:

$$\mathbf{W}_k = \frac{\partial \mathbf{dw}_k}{\partial \mathbf{q}_k}_{|q_k=\hat{q}_k} = \begin{bmatrix} \frac{\partial d_{1,k}}{\partial x_k} & \frac{\partial d_{1,k}}{\partial y_k} & \frac{\partial d_{1,k}}{\partial \theta_k} \\ \vdots & \vdots & \vdots \\ \frac{\partial d_{M,k}}{\partial x_k} & \frac{\partial \triangle d_{M,k}}{\partial y_k} & \frac{\partial \triangle d_{M,k}}{\partial \theta_k} \end{bmatrix} \tag{17}$$

Where:

$$\begin{aligned} \frac{\partial d_{j,k}}{\partial x_k} &= \frac{(x_k - bx_j)}{\sqrt{(x_k - bx_j)^2 + (y_k - by_j)^2 + (z_k - bz_j)^2}} \\ \frac{\partial d_{j,k}}{\partial y_k} &= \frac{(y_k - by_j)}{\sqrt{(x_k - bx_j)^2 + (y_k - by_j)^2 + (z_k - bz_j)^2}} \\ \frac{\partial d_{j,k}}{\partial \theta_k} &= 0 \end{aligned} \tag{18}$$

3.4 System Noise

In the proposed system it is supposed that the main errors are in the odometer information and in the measured distances from the ultrasound and WiFi.

In the odometer case the Chenavier and Crowley model is used [15]. This model assumes that the errors are uncorrelated and follow this expression:

$$\mathbf{R} = \begin{bmatrix} K_{ss} \left|\triangle D_k cos(\theta_k)\right| & 0 & 0 \\ 0 & K_{ss} \left|\triangle D_k sin(\theta_k)\right| & 0 \\ 0 & 0 & K_{s\theta} \left|\triangle D_k\right| + K_{\theta\theta} \left|\triangle \theta_k\right| \end{bmatrix} \tag{19}$$

Where K_{ss} is the odometer drift coefficient when the robot follow a path, $K_{s\theta}$ is the angle drift coefficient when the robot follow a path and $K_{\theta\theta}$ is the angle drift coefficient when the robot only turns.

In the case of the WiFi signals it is assumed that the noise is Gaussian and is uncorrelated, so the WiFi noise matrix is a diagonal matrix whose elements are the standard deviation in the distance measurements:

$$\mathbf{N}_{WiFi} = \begin{bmatrix} \sigma_{w,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{w,M} \end{bmatrix} \tag{20}$$

For ultrasounds signals we assume also that the noise is Gaussian, but in this case the noise signa is correlated because we use hyperbolic trilateration so, supposing that all signals have the same standard deviation (σ_u) the ultrasound noise matrix is:

$$\mathbf{N}_{ultr} = \begin{bmatrix} \sigma_u & 0.5 \cdot \sigma_u & \cdots & 0.5 \cdot \sigma_u \\ 0.5 \cdot \sigma_u & \sigma_u & \cdots & 0.5 \cdot \sigma_u \\ 0.5 \cdot \sigma_u & 0.5 \cdot \sigma_u & \ddots & 0.5 \cdot \sigma_u \\ 0.5 \cdot \sigma_u & 0.5 \cdot \sigma_u & \cdots & \sigma_u \end{bmatrix} \tag{21}$$

3.5 Consideration about the Measurements

In the proposed system the robot can receive measurements only from the ultrasound system, only from the WiFi stations, from both of them or none of them. So the minimax filter has to adapt its equations depending of the measurements obtained. The matrices that depends on the received signal are $\mathbf{C_k}$ and $\mathbf{V}_k$.

In the case of no signal is recived the updated stage is not carry out, so $\hat{\mathbf{q}}_k = \hat{\mathbf{q}}_{k,odo}$ and $\mathbf{P}_k = \mathbf{P}_{k-1} + \mathbf{R}$

In the case ultrasound and WiFi signals are received $\mathbf{C}_k = [\mathbf{U}_k \, \mathbf{W}_k]^T$ and $\mathbf{V}_k = \left[\begin{array}{c|c} \mathbf{N}_{ultr} & \mathbf{Zeros} \\ \hline \mathbf{Zeros} & \mathbf{N}_{WiFi} \end{array}\right]^T$

If only ultrasound signal is received $\mathbf{C}_k = \mathbf{U}_k$ and $\mathbf{V}_k = \mathbf{N}_{ultr}$. If only the WiFi signals are received $\mathbf{C_k} = \mathbf{W}_k$and $\mathbf{V}_k = \mathbf{N}_{WiFi}$.

4 Results

The work has been developed in the second floor of the Polytechnic School Building (EPS) of Alcala de Henares. In [16] a fusion of ULPSs and odometry information is implemented on a mobile robot Pioneer 3dx. ULPS distribution in the real scenario is shown in figure 4.

Using the same scenario, RSSI information from WiFi APs has been introduced in the fusion algorithm. Figure 5 shows ULPSs, WiFi APs distribution and US coverage zones.

Fig. 4. ULPS distribution

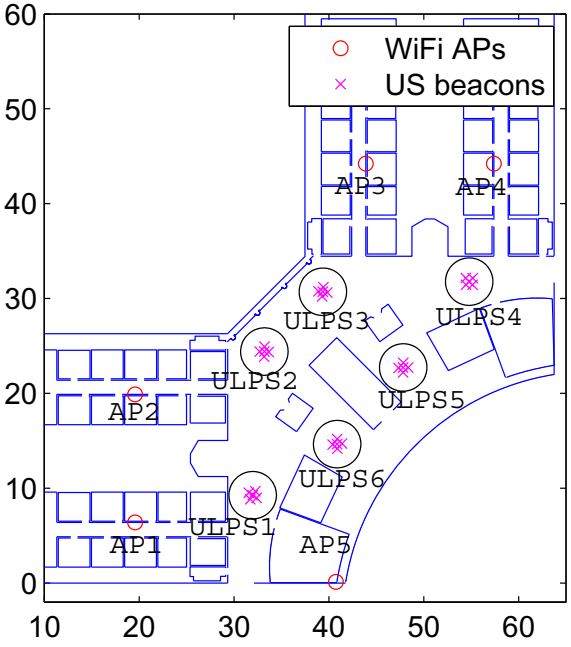


Fig. 5. APs and US beacons location

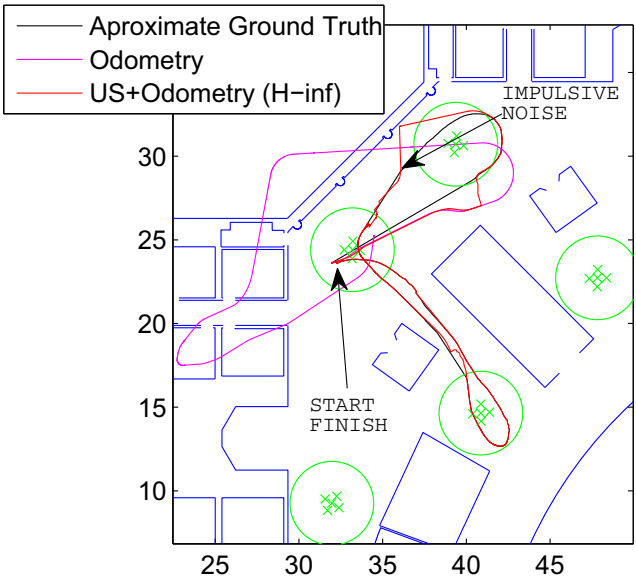


Fig. 6. US and Odometry fusion

To evaluate the fusion algorithm a trajectory has been realized by the robot. During the trajectory, the robot advance from the second ULPS to the third and sixth ULPS. During the process, the robot captures the odometry information, US signals if the robot moves in US coverage area and RSSI information from all APs. In addition, an impulsive noise has been introduced when the robot is in the blind US area between ULPS2 and ULPS3. Figure 6 shows the ground truth of the trajectory (black), the path using only the odometry information (magenta) and the fusion of US and odometry (red) by H-infinity algorithm described previously. Note that when the robot navigates out of ULPS coverage, it only moves using the odometry information and is not robust to impulsive noise and slides (see areas 1 and 2).

To solve the problem and make the system more robust, the inclusion of RSSI information from the APs of the WLAN is proposed. Figure 7 shows US, WiFi and Odometry fusion. In this case, the estimate trajectory is more accurate than the US and odometry fusion. WiFi information makes the estimation more robust when the robot is without US information and impulsive noises or slides are compensated with RSSI values (1 and 2).

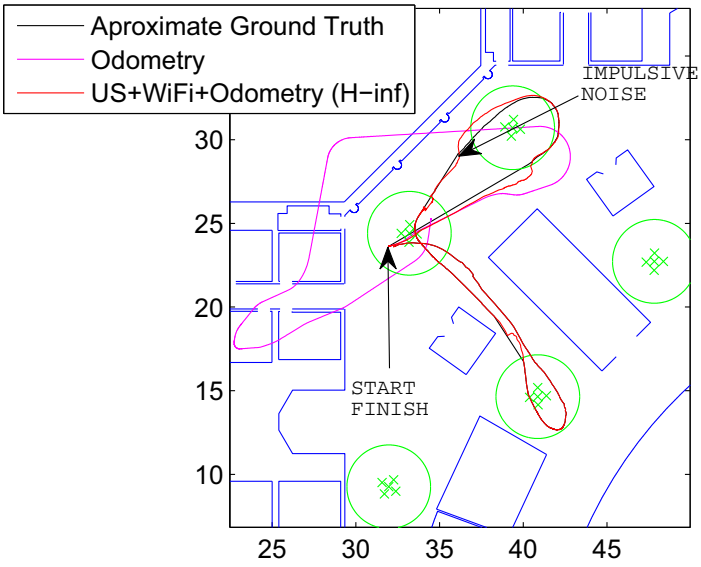


Fig. 7. US, WiFi and Odometry fusion

5 Conclusions

In this paper, a fusion of odometry, WiFi and US on a mobile robot is presented. The integration of RSSI measures in the H-infinity filter improves the accuracy of the position than the fusion of ULPS and Odometry. Furthermore, this integration makes the system more robust to impulsive errors when the robot navigates without US information as results shown. In adittion, a dynamic H-infinity filter has been implented which takes the advantage of use a variable number of US and WiFi information depending on the position of the robot.

Acknowledgments. This work has been possible thanks to the Spanish Ministry of Economy and Competitiveness (LORIS project, ref. TIN2012-38080-C04-01) and the FPI/UAH fellowship program.

References

1. Ureña, J., Hernández, A., Jiménez, A., Villadangos, J.M., Mazo, M., García, J.C., García, J.J., Álvarez, F.J., De Marziani, C., Pérez, M.C., Jiménez, J.A., Jiménez, A.R., Seco, F.: Advanced sensorial system for an acoustic LPS. Microprocessor and Microsystems 31, 393–401 (2007)

2. Bais, A., Sablatnig, R., Gu, J., Khawaja, Y.M.: Location tracker for a mobile robot. In: 2007 5th IEEE International Conference on Industrial Informatics, June 23-27, vol. 1, pp. 479–484 (2007)
3. Prieto, J.C., Jiménez, A.R., Guevara, J.I., Ealo, J., Seco, F., Roa, J., Ramos, F.: Subcentimeter-accuracy localization through broadband acoustic transducers. In: IEEE International Symposium on Intelligent Signal Processing, Alcalá de Henares, España, pp. 1–6 (October 2007)
4. Retscher, G., Moser, E., Vredeveld, D., Heberling, D., Pamp, J.: Performance and accuracy test of WiFi indoor postioning system. Journal of Applied Geodesy 1, 8 (2007)
5. Hatami, A., Pahlavan, K., Heidari, M., Akgul, F., On, R.S.S.: On RSS and TOA based indoor geolocation - a comparative performance evaluation. In: IEEE Wireless Communications and Networking Conference, WCNC 2006, April 3-6, vol. 4, pp. 2267–2272 (2006)
6. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 35–45 (1960)
7. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House (2004)
8. Simon, D.: To game theory approach to constrained minimax state estimation. Technical report, Department of Electrical Engineering, ClevelandState University (2005)
9. Hata, M.: Empirical formula for propagation loss in land mobile radio services. IEEE Trans. Veh. Technol. 29, 317–325 (1980)
10. Okumura, Y., Ohmori, E., Kawano, T., Fukua, K.: Field strength and its variability in UHF and VHF land-mobile radio service. Rev. Elec. Commun. Lab. 16(9) (1968)
11. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge (1992)
12. Holland, J.: Adaption in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor (1975)
13. Koza, J.R.: Human-competitive results produced by genetic programming. Genetic Programming and Evolvable Machines 11, 251–284 (2010)
14. Deffenbaugh, M., Bellingham, J.G., Schmidt, H.: The relationship between spherical and hyperbolic positioning. In: Proc. MTS/IEEE. Prospectsfor the 21st Century OCEANS 1996, vol. 2, pp. 590–595 (1996)
15. Chenavier, F., Crowley, J.L.: Position estimation for a mobile robot using vision and odometry. In: Proc. Conf. IEEE Int. Robotics and Automation, pp. 2588–2593 (1992)
16. Ruiz, D., Garcia, E., Urena, J., de Diego, D., Gualda, D., Garcia, J.C.: Extensive Ultrasonic Local Positioning System for navigating with mobile robots. In: 2013 10th Workshop on Positioning Navigation and Communication (WPNC), March 20-21, pp. 1–6 (2013)

Seamless Indoor-Outdoor Robust Localization for Robots*

P. Urcola, M. T. Lorente, J. L. Villarroel, and L. Montano

Instituto de Investigación en Ingeniería de Aragón
Universidad de Zaragoza
{urcola,mlorente,jlvilla,montano}@unizar.es

Abstract. In this paper we present a unified localization technique for indoor-outdoor environments that allows a seamless transition between a mapped zone using laser rangefinder on-board sensors and a GPS based localization zone. Different situations are detected during the indoor-outdoor transitions, in which the sensors used change and the localization estimator has to manage them properly for a continuous localization. The quality in the GPS measurements and the zone where the robot is localized are used to determine the best instant for switching the localization parameters for adapting to the situations.

1 Introduction

It is common in indoor robotics applications to assume a limited environment (a room, a building floor, etc) for localization purposes. This limitation is forced by the need of a finite map of the features in the zone to localize the robot. There are many map-based methods for robot localization (see [1] for a selection) that use different sensors achieving good results.

Due to the sparseness of the features needed to get a reliable localization system, the use of maps in outdoor scenarios is uncommon. Instead, outdoor applications usually utilize GPS based localization which avoids any limitation in the environment as it is accessible almost everywhere.

However, there are few systems that provide a continuous localization for both indoor and outdoor scenarios in such a way that the robot is not confined in a limited space.

Main difficulties come from the fact that very different sensors (odometry, IMU, rangefinders, GPS, etc) are needed to get a good estimation. The outputs of some of them completely change depending on the situation of the robot: the GPS gets no measurements indoors, open spaces make rangefinders useless, varying magnetic fields influence compass measurements, etc. Also, the use of different reference frames requires a special consideration of the measurements. All these problems become more obvious during the transitions as measurements are more imprecise.

* This work was partially supported by the Spanish project DPI2012-32100 and by Project DGA T04-FSE.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_21,

Some authors have adapted techniques from cell phones and use WPAN and WLAN based localization [2]. These approaches rely on a fixed and known infraestructure and provide results with an accuracy that may not be enough for autonomous robots.

Some approaches, such as [3], rely on the ability of a GPS based Kalman filter to deal with temporary loss of measurements. However, as the duration of this temporary loss increases, the estimation becomes more uncertain.

For transition detection, [4] uses a learning scheme using vision that allows to differenciate indoor and outdoor light characteristics.

In [5], the authors integrate, by means of a Kalman filter, a laser-based SLAM method with GPS, IMU and odometry measurements for a continuous localization. However, EKF scheme assumes independence between measurements but SLAM output and GPS measurements are highly correlated as they are both estimations of the same variables of the robot state. These independence assumptions, when real correlations are high, lead to incoherent estimations because they are overconfident. Thus, in our approach, we use the covariance intersection method [6] to avoid these problems.

In situations where compass cannot be used, the orientation of the robot cannot be directly measured from GPS and has to be derived from other measurements. [7] proposes a Kalman Filter to fuse odometry/IMU and GPS measurements, and the result is integrated into a Monte Carlo localization approach. To estimate the orientation of the robot they use consecutive GPS measurements, assuming that the robot is moving straight in between. This assumption is too strong to be considered in a general scenario. In [8], using the same estimation, the authors determine the likelihood of the estimated orientation by comparing it to the estimation obtained from the IMU. If they diverge, GPS based estimation is discarded. However, this technique would accept a bad estimation that is coherent with a bad IMU measurement. Thus, in our approach we propose a method to calculate the orientation of the robot from several GPS measurements whenever the robot moves straightforward to improve the estimation of the filter.

In real applications where indoor and outdoor scenarios are mixed, it is necessary to have a robust continuous localization without discontinuities when the scenario changes (indoor to outdoor, or vice versa) or when the most accurate sensors have to be chosen. In this paper, we contribute a unified framework for a seamless localization during navigation within different types of environments.

2 Scenario and Sensors

The continuous localization system for robots has to work in a scenario composed by zones with different characteristics:

- Indoor and outdoor mapped zones.
- Zones with no GPS coverage.
- Zones covered with imprecise GPS.
- Zones with good GPS coverage.

These zones overlap each other in such a way that the mapped area contains zones of different GPS coverage or not coverage at all. The scenario may not be completely mapped, but the position of all the mapped zones in GPS coordinates is known.

To estimate the localization and orientation of the robot in 2D we have three sources of information:

- Odometry plus IMU with high frequency in the whole scenario, in the sequel odometry/IMU.
- Laser rangefinder based localization in the mapped zone, as proposed in [9].
- GPS based localization.

3 Situations in the Environment

We have defined five different situations in which the robot may be, depending on the characteristics of the environment. Each of these situations requires a different measurement policy, as they present different characteristics.

3.1 Description of Situations

Indoors. When the robot is in an indoor situation, GPS is unavailable and the localization completely relies on the map-based technique.

Transition Indoors–Outdoors. Inside the Map. The robot starts receiving GPS signals, but they are too bad to be used. As the robot is inside the map, the map-based localization can still be used although the uncertainty will be bigger as the robot is leaving the map.

Transition Outdoors–Indoors. Inside the Map. In this situation, the GPS measurements become worse and they cannot be used anymore. However, as the robot is inside the map, the map-based localization can be used instead. But previously, we need to give a prior of the position of the robot to the indoor localization because after a while being out of the map, the indoor localization becomes corrupted due to the lack of rangefinder measurements. This action is called a *reset*. The prior information is taken from the current estimation of the robot localization computed using the GPS.

Outdoors. Outside the Map with Bad GPS. The robot is outside the map, so the indoor localization cannot be used anymore. But there is no GPS signal or it is not good enough to be used so that the robot cannot use any measurement but the prediction from the odometry/IMU.

Outdoors. Outside the Map with Good GPS. The robot receives good enough GPS measurements and thus they are used to estimate the position of the robot. Notice that the uncertainty of the GPS is still being considered during update. Depending on coverage conditions and on the kind of path, the GPS measurements can be used or not to compute orientation.

3.2 Detection of Situations

To detect in which situation the robot is, we use two variables: the GPS quality and the estimated position of the robot at that moment.

Receiving a GPS signal, even with an estimation of the measurement error, it is not enough to consider it as useful. In zones with bad GPS coverage (near high walls, below trees, etc), GPS provides awful estimations moreover if there were no previous good GPS estimations. In these cases, if the robot provides good enough prediction via odometry and IMU, it is better to discard the measurements rather than integrate them into the filter. To decide if a GPS measurement is useful, we define a quality value Q_{gps} which is computed from the variances of the latitude and the longitude measured using (1).

$$Q_{gps} = -log(\sigma_{lat}^2 \sigma_{lon}^2) \tag{1}$$

This quality measurement has been adopted after some experimental tests in different scenarios and it is related to the GPS covariance: the quality increases as the volume of uncertainty decreases. The number of satellites has also been considered as a quality measurement. However, the latter is more volatile than the first one and the uncertainty volume presents bigger discontinuities in the indoor-outdoor transitions and vice versa, as shown in Figure 1.

However, this is not enough to decide if the robot is indoors or not. It is possible that we receive no GPS at all (see Figure 1 on the left round $t = 700\,s$) or a really bad quality GPS (as in Figure 1 on the right after $t = 900\,s$) being outside a building. In these cases, we use the best estimation of the localization

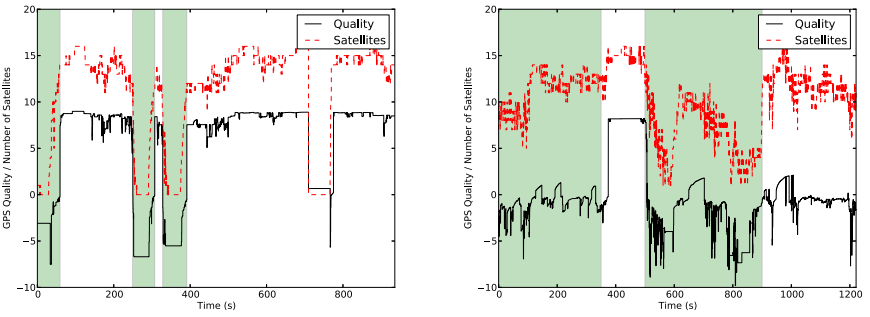


Fig. 1. GPS Quality and number of satellites during two experiments in different scenarios. Shadowed zoned represent indoor zones.

Table 1. Situation detection

	Good GPS	Bad GPS	No GPS
Inside Map	Outdoors	Transition	Indoors
Outside Map	Outdoors	Outdoors (No GPS)	

of the robot to know if it is inside a building or not. We are assuming that we know where the buildings are because they have been previously mapped.

Table 1 sums up the detection of the situations depending on the observed variables.

4 Continuous Localization

The first problem to deal with when considering different sources of measurements is that each source has a different frame of reference. Thus, we have defined a common frame where all the measurements are transformed into. This frame is defined in such a position that we can estimate its localization using both the map and the GPS.

The method proposed is based on a time discretized bayesian filter approach working in all those changing situations seamlessly. The measurements from different sources are weighted depending on the situation of the robot and on their own covariance. The localization of the robot at time t_k, $\mathbf{x}_k = (x, y, \theta)_k$ and its covariance $\mathbf{P}_k$ is tracked. The filter is divided in two steps, prediction and update.

4.1 Prediction

In the prediction step, the information from the odometry/IMU source $\mathbf{y}_{od}$ is used as the estimation of the movement of the robot since the last measurement from GPS or laser based localization. Odometry/IMU measurements accumulate error as time evolves so that we use $\mathbf{u}_{od}$ the relative movement with respect to a previous measurement instead of the absolute value. This phase corresponds to the prediction step of a Extended Kalman Filter [10], where $\mathbf{F}$ is the jacobian of the prediction function (2).

$$\begin{aligned} \mathbf{x}(k+1|k) &= \mathbf{x}(k|k) \oplus \mathbf{u}_{od}(k+1) \\ \mathbf{u}_{od}(k+1) &= \mathbf{y}_{od}(k+1) \oplus \mathbf{y}_{od}^{-1}(k) \\ \mathbf{P}(k+1|k) &= \mathbf{F}(k)\mathbf{P}(k|k)\mathbf{F}(k)' + \mathbf{Q}(k+1) \end{aligned} \tag{2}$$

Symbol $\oplus$ represents the composition of affine transformations in the plane (translation and rotation) in homogeneous coordinates. As odometry/IMU measurements are always available, the prediction step is performed at each time step. The uncertainty of the localization increases because of the odometry error $\mathbf{Q}$.

4.2 Update

In the update phase, the measurements from GPS $\langle \mathbf{y}_{gps}, \mathbf{R}_{gps} \rangle$ and from laser-based localization $\langle \mathbf{y}_{map}, \mathbf{R}_{map} \rangle$ are used to correct the predictions made in the previous phase. In case of the map-based localization measurements, the particle filter technique described in [9] is used. According with that approach, $\mathbf{y}_{map}$ is the mass center of the most promising cluster of particles and $\mathbf{R}_{map}$ is the population variance of the particles in the cluster.

All the measurements have three components $\mathbf{y} = (x, y, \theta)$, which corresponds to the components of the state being tracked. Odometry and map based localization provide orientation information but GPS data do not give it directly. In Section 5 we present a procedure to estimate it in some cases.

The general framework we use (3) is based on the covariance intersection filter [6]. This formulation permits the fusion of measurements from different sources that may be correlated, avoiding the statistical independence requirements of EKF approaches. The parameters γ_{map}, γ_{gps}, γ_{od} weight the covariances and their selection is described later, in section 4.3.

$$\begin{aligned} \mathbf{P}(k+1|k+1)^{-1} &= \gamma_{map}\mathbf{R}_{map}^{-1} + \gamma_{gps}\mathbf{R}_{gps}^{-1} + \gamma_{od}\mathbf{P}^{-1}(k+1|k) \\ \mathbf{x}(k+1|k+1) &= \mathbf{P}(k+1|k+1)(\gamma_{map}\mathbf{R}_{map}^{-1}\mathbf{y}_{map} + \\ &\qquad \gamma_{gps}\mathbf{R}_{gps}^{-1}\mathbf{y}_{gps} + \\ &\qquad \gamma_{od}\mathbf{P}^{-1}(k+1|k)\mathbf{x}(k+1|k)) \end{aligned} \tag{3}$$

4.3 Update Tuning

Depending on the situation of the robot, the weighting values $\gamma_{map}, \gamma_{gps}, \gamma_{od}$ in (3) take different values so that the final estimation relies on a mixture of the measurements and the prediction. In addition to that, the measurement covariances are also used to weight the estimation, giving more importance to smaller covariance estimations.

We set to 0.0 the weights of the sensors that are unreliable in each situation (e.g. GPS is set to 0.0 in indoor situations). The rest of the weights are equally set so that the sum of all of them is 1.0. This way, the only weighting factor for the measurements are the covariance matrices of the reliable sensors. The specific weights for the measurements in each situation are defined in Table 2.

5 Outdoor Orientation

Outdoors, the GPS provides the robot position. However, it does not directly provide the orientation of the robot. We propose a technique to estimate the orientation of the robot based on a set of GPS measurements.

During a distance L in which the robot moves straightforward, i.e. the steering angle is 0 and the orientation θ remains constant, different GPS measurements are stored. For each measurement, an estimation of the orientation $\hat{\theta}_j$ is calculated using (4). The covariances of the measurements $\mathbf{R}_{gps_0}$ and $\mathbf{R}_{gps_j}$ are

Table 2. Update weights in different situations

Situation	γ_{map}	γ_{gps}	γ_{od}	Reset
Indoors	0.5	0.0	0.5	No
In–Out (inside map)	0.5	0.0	0.5	No
Out–In (inside map)	0.5	0.0	0.5	Yes
Outdoors (no GPS)	0.0	0.0	1	No
Outdoors (good GPS)	0.0	0.5	0.5	No

propagated by means of the jacobian to get the covariance of the orientation σ_j. Then, once the robot has reached the distance L, a final estimation of the robot's orientation is calculated (5) as a weighted sum of the individual computed estimations (Figure 2), based on a mixture probability density function defined in [10]. The weights are valued depending on the covariance of each estimation (6).

$$\hat{\theta}_j = \arctan(\frac{\mathbf{y}_{gps\,j} - \mathbf{y}_{gps_0}}{\mathbf{x}_{gps\,j} - \mathbf{x}_{gps_0}}) \tag{4}$$

$$\hat{\theta} = \sum_{j=1}^{n} w_j \hat{\theta}_j,\ \sigma^2 = \sum_{j=1}^{n} w_j \left(\sigma_j^2 + \left(\hat{\theta}_j - \hat{\theta}\right)^2\right) \tag{5}$$

$$w_j = \frac{1}{n-1}\left(1 - \frac{\sigma_j^2}{\sum_{k=1}^{n}\sigma_k^2}\right) \tag{6}$$

Parameter L may be tuned for each scenario. As L increases, more GPS measurements are used to compute orientation, leading to a more accurate estimation. However, it requires that the robot moves long straight trajectories. After several experimental tests, we have set $L = 2.5\,m$ as a trade-off between accuracy and the need of long straight trajectories.

The measurement terms $\langle \mathbf{y}_{gps}, \mathbf{R}_{gps} \rangle$ used in the update phase in (3) are finally obtained as follows:

1. If the robot moves in a straight line (an orientation estimation can be computed),

$$\mathbf{y}_{gps} = \begin{pmatrix} x_{gps} \\ y_{gps} \\ \hat{\theta} \end{pmatrix},\ \mathbf{R}_{gps} = \begin{pmatrix} R_{x_{gps}} & 0 & -\frac{w_n}{L} R_{x_{gps}} \sin\hat{\theta} \\ 0 & R_{y_{gps}} & \frac{w_n}{L} R_{y_{gps}} \cos\hat{\theta} \\ -\frac{w_n}{L} R_{x_{gps}} \sin\hat{\theta} & \frac{w_n}{L} R_{y_{gps}} \cos\hat{\theta} & \sigma^2 \end{pmatrix}$$

2. If the robot does not move in straight line the orientation is taken from the prediction step (given by odometry/IMU), and then,

$$\mathbf{y}_{gps} = \begin{pmatrix} x_{gps} \\ y_{gps} \\ 0 \end{pmatrix},\ \mathbf{R}_{gps}^{-1} = \begin{pmatrix} \frac{1}{R_{x_{gps}}} & 0 & 0 \\ 0 & \frac{1}{R_{y_{gps}}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

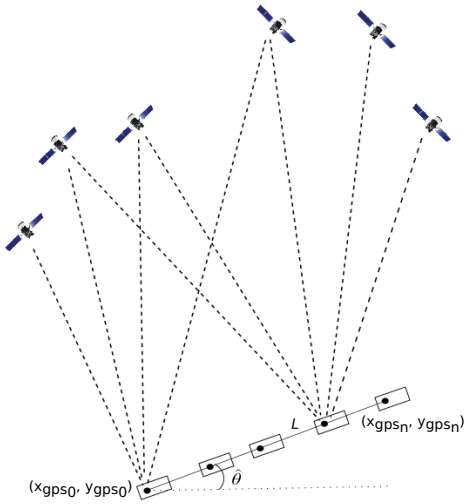


Fig. 2. Orientation of the robot obtained from GPS measurements

Note that the error in the position components (x_{gps}, y_{gps}) in GPS measurements are uncorrelated. However, as the orientation is computed from those components, there are non-zero correlation terms when the robot moves in straight line.

6 Results

In this section we present the experimental results obtained to show the robustness of the methods proposed above.

The platform used is a Robucar-TT[1] with all the sensors needed on-board and car-like motion capabilities (see Figure 3).

First, to show the capabilities of the outdoor localization method in good GPS measurement conditions, including orientation estimation, we designed an experiment in which the robot autonomously navigates describing a rectangle defined by four goals on its corners.

The navigation technique is an adaptation of the ORM technique [11] for car-like vehicles. It takes into account the kinodynamic constraints of the robot and permits maneuvers to avoid obstacles and to guide the robot towards the goal. The details of the navigation technique are out of the scope of this paper.

The goals are defined by GPS coordinates so that the localization should permit the robot to repeat the rectangular path with no drift although the odometry accumulates significant errors.

Figure 4 shows the trajectory followed by the robot completing 11 times the rectangular path for a total distance travelled of $2696\,m$ with an average speed of $0.95\,ms^{-1}$.

[1] `www.robosoft.com`

Fig. 3. Robucar-TT platform equipped with IMU, odometry, three range-finder sensors and GPS receiver

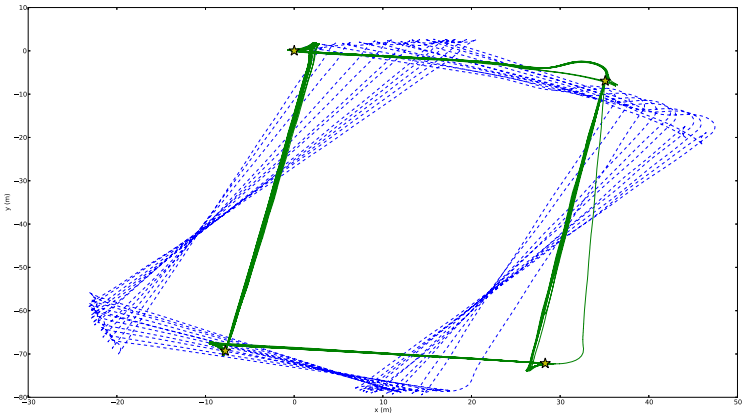


Fig. 4. Estimated localization (solid line) and odometry measurements (dashed line) for the rectangle experiment. The goals are represented by stars.

Figure 5 presents the evolution of the estimation drift during the experiment. We compare the filter localization estimation from the odometry/IMU and the GPS-based one. As known, odometry drift keeps growing with time, being the orientation error the main cause of the position error as well. At the end of the experiment, the odometry reaches a maximum of $18\,m$ of position error and more than $0.3\,rad$ of orientation error. Meanwhile, the estimation of the localization method is able to keep bounded the error to a small value.

Second, we designed a full experiment involving all kinds of situations and transitions. Figure 6 shows the trace of the robot during the experiment. The scenario covers an area of $12000\,m^2$ and the robot navigates for more than 1000 meters.

In the zones with good localization, the variance of the estimation was below $0.1\,m$ for x and y and below $0.05\,rads$ for the orientation. The maxima in the variances of the estimations occur when no measurements are available and during the transitions, specially if a *reset* is performed. This is because the proposed filter is underconfident to deal with the unknown correlations among different measurements.

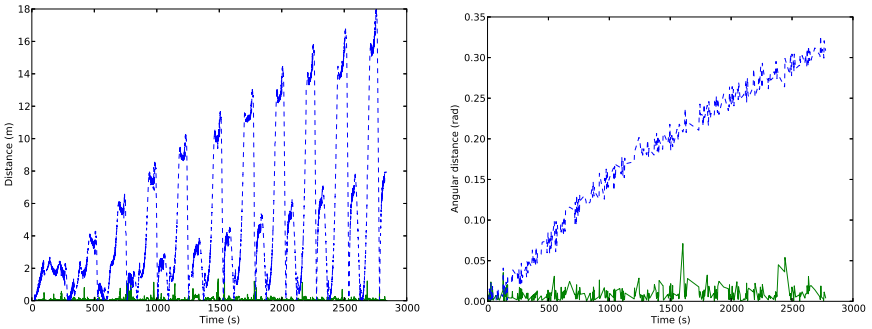


Fig. 5. Position (left) and orientation (right) error of the filter estimation (solid line) and the odometry (dashed line) during the experiment

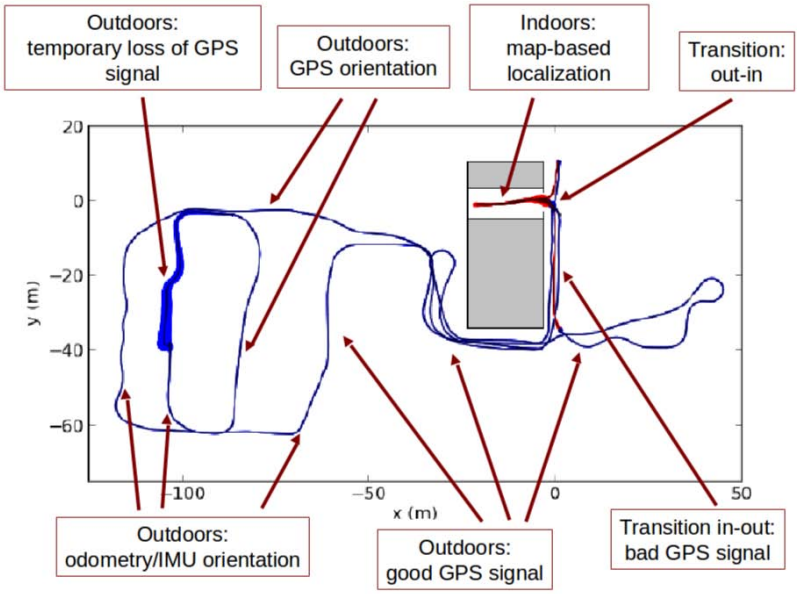


Fig. 6. A complete experiment. All the transitions are shown in which the filter works in the continuous localization.

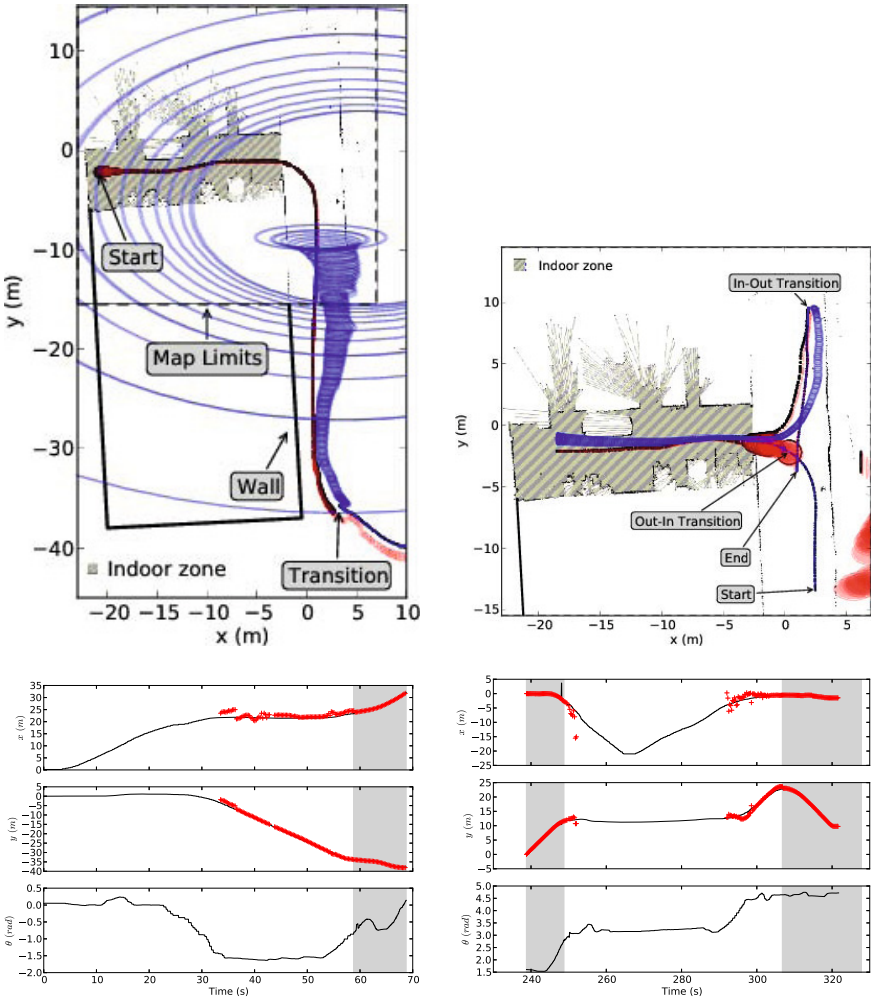


Fig. 7. In the first row, two examples of transitions where it is shown the GPS-based localization (blue), the laser-based localization (red) and the estimation using our method (black). The ellipses represent the uncertainty. In the second row the state variables (in black) during these transitions are depicted. Red crosses represent the GPS data received. The background color indicates whether the robot is using the GPS (gray) or not (white) for localization.

Figure 7 show in detail some transition examples of the experiment. On the top left figure, the robot starts inside the building with map-based localization. After leaving the building, the GPS data arrive but their quality is too low to be useful, mainly because the robot navigates close to a wall. The robot leaves the map but still the GPS is imprecise so the odometry/IMU prediction is used until a good enough GPS signal is received.

On the top right, all the trace remains inside the map limits. The robot navigates towards the building and when it is in front of the door, the GPS quality dramatically drops down and thus an outdoor-indoor transition is detected. In that moment, the map-based localization is reset, corresponding to a covariance enlargement. After a while inside the indoor zone, where the robot is continuously localized in the map by means of the laser and odometry/IMU measurements, the robot leaves the building and, when the GPS signal is good enough, an indoor-outdoor transition is detected and the localization system adapts to the outdoor situation.

As can be seen in the second row of Figure 7, localization continuity is kept along the experiment. Only around transitions, some discontinuities may be found as transitions take place in zones where some of the sensors start to provide bad measurements. However, the biggest gap detected is about $0.5\,m$ in x coordinate. A bigger discontinuity was found in $t = 250\,s$ (Figure 7 bottom right) caused by a spurious bad GPS measurement.

7 Conclusions

We present a mobile robot localization method that allows a robot to seamlessly switch between indoor and outdoor environments. We propose an outdoor GPS-based localization method which is able to estimate the position and orientation of the robot bounding the drifting error obtained from the odometry/IMU measurements.

By using a GPS quality estimation and the robot localization (in map or not), we can determine which is the situation of the robot and select the measurements to achieve the best continuous localization.

All the contributions have been tested in different experiments with a car-like platform. The data obtained in those experiments validates their functionality and robustness.

In future works, a smoother approach to the transition method can be studied, by using the weights in the covariance intersection framework to minimize the gaps during transitions. Also, an study of the limits of the GPS-based orientation method will be performed to determine the best parameters such as maximum curvature or minimum distance needed to get a good estimation.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press (2005)
2. Kang, J., Kim, D., Kim, E., Kim, Y., Yoo, S., Wi, D.: Seamless mobile robot localization service framework for integrated localization systems. In: 3rd International Symposium on Wireless Pervasive Computing (ISWPC), pp. 175–179 (May 2008)
3. Goel, P., Roumeliotis, S.I., Sukhatme, G.: Robust localization using relative and absolute position estimates. In: Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 1999, vol. 2, pp. 1134–1140 (1999)

4. Collier, J., Ramirez-Serrano, A.: Environment classification for indoor/outdoor robotic matching. In: Canadian Conference on Computer and Robot Vision, pp. 276–283 (2009)
5. Pacis, E.B., Sights, B., Ahuja, G., Kogut, G., Everett, H.R.: An adapting localization system for outdoor/indoor navigation. In: SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense Security Symposium, Orlando, EEUU (April 2006)
6. Julier, S.J., Uhlmann, J.K.: Using covariance intersection for SLAM. Robotics and Autonomous Systems (55), 3–20 (2007)
7. Hentschel, M., Wulf, O., Wagner, B.: A gps and laser-based localization for urban and non-urban outdoor environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 149–154 (2008)
8. Gu, D., El-Sheimy, N.: Heading accuracy improvement of mems imu/dgps integrated navigation system for land vehicle. In: Position, Location and Navigation Symposium, 2008 IEEE/ION, pp. 1292–1296 (2008)
9. Fox, D.: Adapting the sample size in particle filters through KLD-sampling. The International Journal of Robotics Research 22(12), 985–1003 (2003)
10. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with applications to tracking and navigation. John Wiley & Sons, Inc. (2001)
11. Minguez, J.: The obstacle-restriction method for robot obstacle avoidance in difficult environments. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 2284–2290 (2005)

Robust Person Guidance by Using Online POMDPs

Luis Merino[1], Joaquín Ballesteros[2], Noé Pérez-Higueras[1], Rafael Ramón Vigo[1], Javier Pérez-Lara[1], and Fernando Caballero[2]

[1] Robotics, Vision and Control Group
Universidad Pablo de Olavide, Sevilla, Spain
[2] Robotics, Vision and Control Group
Universidad de Sevilla, Spain

Abstract. The paper considers a guiding task in which a robot has to guide a person towards a destination. A robust operation requires to consider uncertain models on the person motion and intentions, as well as noise and occlusions in the sensors employed for the task. Partially Observable Markov Decision Processes (POMDPs) are used to model the task. The paper describes an enhancement on online POMDP solvers that allow to apply them to larger problems. The algorithm is used to control the robot in real-time for the guiding application. Results in simulation illustrate the approach.

Keywords: robust navigation, POMDP.

1 Introduction

In the last years, there has been an impressive development on localization, mapping, SLAM and navigation techniques. The scale of the scenarios to which this techniques can be applied has increased dramatically. But despite all these advances, achieving long-term autonomy, either in space and time, requires to enhance the robustness of all these skills. For this, a robot should cope with imperfect information, failure modes, partial models, etc.

In particular, we are interested in the robust operation of robots in human inhabited spaces. The FROG FP7 project[1] aims to deploy a guiding robot in touristic sites involving outdoor and partially outdoor scenarios. While robot guides has been developed since more than a decade [26,19], the project considers as new contributions the development of social behaviors and their adaptation by integrating social feedback, as well as the robust operation in outdoors crowded scenarios. Furthermore, it aims to demonstrate the operation of the robot for two weeks at the Royal Alcazar in Seville (see Fig. 1).

Among other activities, one particular application considered in the project is person guidance. Robustly navigating in these crowded scenarios (the Royal Alcazar may have around 5000 visits per day) requires to consider issues due

[1] `http://www.frogrobot.eu`

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_22,

Fig. 1. The FROG project aims to deploy a guiding robot with a fun personality, considering social feedback, in the Royal Alcazar of Seville and the Zoo of Lisbon. Two pictures of the first scenario are presented here. On the right, and initial design of the robot components can be seen.

to the high dynamics of the environment. Furthermore, guiding a person in this scenario involves not only ensuring a safe and efficient navigation but also social interaction and social awareness when achieving these goals.

For this and other applications, a robust operation, like robust navigation, requires reasoning about all the potential uncertainties present on the robotic system, due to imperfect models, limited information, errors, noise, etc. One framework that allows considering uncertainties in a principled way is Partially Observable Markov Decision Processes (POMDPs) [8]. By using POMDPs, the uncertainties in the sensors, as well as uncertainties on the models employed by the robot, are considered when planing the actions that the robot has to execute.

However, the broader application of planning under uncertainties to robotic systems requires to develop methods that cope with the curse of dimensionality: in this case the planning problems are posed not on the state space, but on the much harder information space (the space of all potential actions and observations histories that the robot may perform/gather).

In the paper, we present firstly a method for alleviating the complexity of online POMDPs, allowing to apply them to larger problems. Then, it is described how this method can be used to model the task of person guidance. Results in simulation show the applicability of the approach. The paper finalizes with some conclusions.

1.1 Related Work

POMDPs are increasingly used in robotics [11,14,6,4]. This is due to the development of more efficient offline POMDP solvers in the last decade [23,9,3].

However, their broader application to robotics has been precluded due to curse of dimensionality. Besides offline solvers, online POMDPs have been also considered [18,6], as they are more suitable for certain robotic applications. Still they are affected by the same issues.

Efficient and robust navigation in crowded environments requires taking into account human behavior models, social constraints and their uncertainties[7]. On one hand, besides the static obstacles the motion of the persons have to be considered. Typical approaches to human motion prediction simply assume a constant velocity, which is not valid in most cases. In general, human navigation intent will depend on the function and structure of the environment [25,2,12].

Furthermore, the interaction of the robot and the humans has to be taken into account to allow for an efficient navigation[27]. In the particular application of person guidance, in [5] probabilistic models of human-interaction are extracted from data for the purposes of person guiding. Finally, applications in which persons are also involved require to consider social constraints, like the human commitment and goals. Related to the work presented here, the authors in [16] employ Markov Decision Processes to predict the destination of the persons in guiding applications. However, the uncertainties on the observation processes are not considered.

Closest to our work, in [24] a POMDP is also used to infer the intentions of the person for wheelchair navigation. Similar ideas are considered, but in a different scenario. Moreover, offline POMDP models are used, so the system has to be re-planned offline if a different scenario is considered. Our online system allow to build the models on the fly. Furthermore, the proposed system is able to work with a larger scenario in terms of state space and observation outcomes.

2 POMDPs in a Nutshell

As commented above, in this paper we will analyze the use of POMDPs to model navigation tasks like person guiding. Formally, a discrete POMDP is defined by the tuple $\langle S, A, Z, T, O, R, D, \gamma \rangle$ [8]. The *state space* is the finite set of possible states $s \in S$; the *action space* is defined as the finite set of possible actions $a \in A$ that the robot can perform; and the *observation space* consists of the finite set of possible observations $z \in Z$. At every step, an action is performed by the robot, an observation is made and a reward is given.

After performing an action a, the state transition is modeled by the conditional probability function $T(s', a, s) = p(s'|a, s)$, allowing to model failures and uncertainties on the prediction models. In the same way, measurement process is modeled by the conditional probability function $O(z, a, s') = p(z|a, s')$, which permits to consider non-observability, occlusions, noise, etc.

The reward obtained at each step is $R(s, a)$. The planning objective is to maximize the sum of expected rewards, or *value*, for a planning horizon of D time steps. To ensure that the sum is finite when $D \to \infty$, rewards are weighted by a discount factor $\gamma \in [0, 1)$.

POMDPs consider that the state is non-observable; therefore, a belief function b is maintained by using Bayes rule. The belief obtained if we apply the action a

and get the observation z is $b'(s') = \tau(b, a, z) = \eta O(z, a, s') \sum_{s \in S} T(s', a, s) b(s)$. The normalization constant:

$$\eta = P(z|b, a) = \sum_{s' \in S} O(z, a, s') \sum_{s \in S} T(s', a, s) b(s) \tag{1}$$

gives the probability of obtaining a certain observation z after executing action a for a belief b. As said above, the objective is to plan a policy which indicates the action that has to be performed given the information available (summarized by the belief) $a = \pi(b)$ and that maximizes the cumulative expected reward, or value $V^{\pi}(b)$:

$$V^{\pi}(b) = R(b, \pi(b)) + \gamma \sum_{z \in Z} P(z|b, a) V^{\pi}(b^z_{\pi(b)})) \tag{2}$$

where $R(b, a) = \sum_s R(s, a) b(s)$ is the expected immediate reward[2]. The value of the optimal policy is usually denoted by $V^*(b)$ and associated to it is the optimal Q function:

$$Q^*(b, a) = R(b, a) + \gamma \sum_{z \in Z} P(z|b, a) V^*(b^z_a)) \tag{3}$$

2.1 POMDP Solver

Modeling a particular robotic task, like person guidance, implies to define the transition function $T(s', a, s)$ and the observation model $O(z, a, s')$. Notice that both models consider uncertainties, either in the outcome of actions and in the observation process. These models can be defined by hand or they can be learnt from data [5].

Once the models are defined, the task is codified into the POMDP by designing an appropriate reward function $R(s, a)$, which indicate the expected behavior of the robot. Again, this reward function can be defined by hand, or could be learnt from data [7].

Given the reward and the models, the optimal policy can be obtained by computing the policy $\pi(b)$ that maximizes (2) using dynamic programming. However, POMDPs are quite hard to solve (they are PSPACE-complete [15]). Current offline solvers, like [23,9,3], apply some approximations to obtain a solution. Furthermore, they try to solve the dynamic programming problem over the full belief state space. These algorithms require to recompute the full policy when there are changes on the environment dynamics or the sensor models.

As commented above, online solvers are more suitable for some robotic tasks. An online POMDP solver tries to determine the optimal action for the current belief point b, and not for all the possible beliefs. To achieve this, it creates an AND-OR tree by exploring the next beliefs for all possible actions and for all possible observations starting at the initial belief (see Fig. 2). The dynamic

[2] $b^z_a = \tau(b, a, z)$ will be used to obtain a more compact formula.

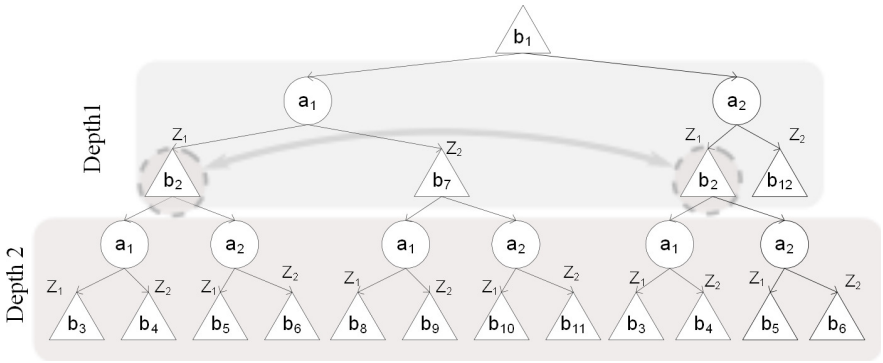


Fig. 2. An AND-OR Belief Tree with 2 actions and 2 observations. The OR-node (where the robot has to take an action) are represented by triangles and the AND-nodes by circle (when observations, which are not controlled, are received). If a repeated belief appears at the same depth, it means that it has an identical subtree and the same value (like belief b2).

programming solution to (2) is applied from the leaves to the root, obtaining the best action.

This tree is created on demand, and can be adapted online if the models or the reward function change. As we will see, this allows us to model tasks as the guiding task described below. Furthermore, this kind of structures can be used to create anytime algorithms [17], in which the time available is used to explore the tree as deep as possible.

The branching factor in AND-OR POMDP trees is $|A||Z|$ where $|A|$ is the number of actions and $|Z|$ is the number of observations, and the number of leaves nodes for a tree of depth D is $(|A||Z|)^D$, that is, exponential on the planning horizon. Exploring all the nodes to obtain the optimal action is, therefore, not a option. In [18], a classification of POMDP online algorithms can be found, and also three strategies that are employed to improve the computing time required to choose the best action:

- Monte Carlo sampling algorithms: minimize the branching factor by sampling a subset of observations.
- Heuristic search algorithms: guide the search of the most relevant branch nodes.
- Branch and Bound algorithms: Prune nodes that are suboptimal compared to other that have already been expanded.

We have developed an additional improvement that can be applied with any of the previous strategies. The main idea is to take advantage of the true topology of the belief space, which is actually a graph instead of a tree. For this, we introduce a pseudo-metric into the belief space to determine if two belief points are actually the same, that is, if we are revisiting a belief point, which then does not have to be expanded again within the AND-OR tree.

In this paper we employ a first idea in that direction. In the expansion of the AND-OR tree, if the next belief to be expanded is similar to a saved belief at the same depth, that belief will not be expanded because its value is already calculated (see Fig. 2). The similarity between two belief points in the tree b and b' is computed by using the Jensen-Shannon divergence $D_{JS}(b\|b')$ [1] (its square root is actually a true metric):

$$D_{JS}(b\|b') = \frac{1}{2}(D_{KL}(b\|\frac{b+b'}{2}) + D_{KL}(b'\|\frac{b+b'}{2})) \tag{4}$$

where

$$D_{\mathrm{KL}}(b\|b') = \sum_{s\in S} b(s)\ln\frac{b(s)}{b'(s)} \tag{5}$$

The Algorithm 1.1, called FSBS, proceeds via look-ahead search up to a fixed depth d. We use the structure of the RTBSS algorithm proposed by [18] to elaborate the FSBS, which therefore also prunes branches on the tree which are suboptimal by using lower and upper bounds on the optimal value function. In particular, the algorithm uses the max-planes lower bound [21] of the optimal value implemented in [22] (line 3). The δ function determines how the FSBS algorithm propagates this lower bound from the leaves up to the root.

$$\delta(b,0) = LowerBound(b) \tag{6}$$

$$\delta(b,d) = \max_{a\in A}(R(b,a) + \gamma\sum_{z\in Z} Pr(z|b,a)\delta(b_a^z, d-1)) \tag{7}$$

To determine the similarity between beliefs, we keep a node list for each depth d, $nodeList_d$. Every node contains the following items: the belief b; an action; and the accumulated reward if that action is applied to the belief, $\delta(b,d)$. We only keep the beliefs up to depth $D-1$ because the leaf nodes cannot be expanded.

The function *orderbyAccReward* in line 5 is used to find the accumulated rewards that are obtained when we apply each action to the current belief.

For each action at depth d, the *orderbyAccReward* function looks for a similar belief in $nodeList_d$, b', considering a similarity threshold *th*. If a belief is successfully found, the accumulated reward that was already stored, $\delta(b',d)$, is assigned to it, and if not, it is assigned as ∞ in order to get expanded first. The *orderbyAccReward* function returns a list, sorted by accumulated reward, in which each element contains the following items: the action associated; the accumulated reward if this action is applied to the input belief if exists; and a variable that indicates whether the resulting belief is already in a depth on the tree or not.

In line 12-13 we reuse the accumulated reward if a similar node is found at the same depth. In this case, we stop expanding along that path. If not, we expand and keep the node (line 15-18). To finish we choose the action that maximizes the accumulated reward (line 21).

Algorithm 1.1. FSBS Algorithm

```
1: function FSBS(b, d, th)
2:     if d == 0 then
3:         return LowerBound(b)
4:     end if
5:     {st_1, st_2, ..., st_|A|} ← orderbyAccReward(b, d, th)
6:     L_T(b) ← −∞
7:     i ← 0
8:     while i < |A| AND st_i.AccReward > L_T(b) do
9:         a ← st_i.IdAction
10:        rAcc ← st_i.AccReward
11:        L_T(b, a) ← −∞
12:        if st_i.isFoundSimilar then
13:            L_T(b, a) ← Reward(b, a) + γ rAcc
14:        else
15:            L_Z(b, a) ← Σ_{z∈Z} P(z|b, a) FSBS(b_a^z, d − 1, th)
16:            L_T(b, a) ← Reward(b, a) + γ L_Z(b, a)
17:            saveNode(b, a, d, L_Z(b, a))
18:        end if
19:        L_T(b) ← max{L_T(b), L_T(b, a)}
20:    end while
21:    return L_T(b)
22: end function
```

We calculate the optimal policy as:

$$\pi^*(b, D) = \arg\max_{a \in A}(R(b, a) + \gamma \sum_{z \in Z} P(z|b, a) FSBS(b_a^z, D - 1, th)) \quad (8)$$

This is applied in each planning iteration. At every iteration, the optimal action is applied and then a new forward search is performed, in a receding horizon fashion.

The algorithm has been tested in benchmark problems showing that using this similarity measurements it is possible to obtain a similar expected solution by expanding several orders of magnitude less nodes [1] than state of the art algorithms. In particular, Fig. 3 shows the comparison of the original RTBSS algorithm and the FSBS algorithm for the RockSample benchmark [20]. More details can be found at [1].

3 Modeling the Guiding Task as a POMDP

In a guiding application, the robot has to guide a person or group of persons towards a common destination. The typical solution to the problem is to plan a path towards the destination and then follow the path by controlling the speed of the robot by using some feedback on the person being guided, like laser, visual information or a combination of them [13].

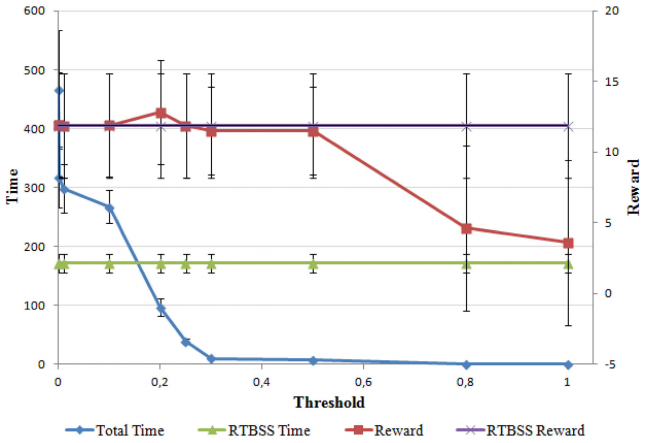


Fig. 3. Expected accumulated reward (right vertical axis) and execution time (left vertical axis) for RTBSS and FSBS using the JS divergence (RockSample). Different thresholds in the JS divergence (horizontal axis) are considered to determine if two belief points are the same in the tree. The larger the thresholds the less nodes are expanded. It can be seen that for thresholds between 0.3 and 0.5 in the similarity the expected reward is very similar, but the execution time is two orders of magnitude lower.

This requires having an estimation of the position of the person. However, imperfect sensors, false positives, occlusions, etc, may lead to uncertainties on the person position. In particular, vision algorithms can be affected by illumination changes, which makes it hard to track robustly a person being guided.

Moreover, one of the main sources of uncertainty in this problem is that the person may change his/her mind, or decide to stop for a while at a different interest point in his way to the destination. From a social point of view it is important that the robot considers the person goals while guiding the person, in order to wait for the person or even to change its goal accordingly. However, the robot cannot have a direct observation on this person goal.

As assumptions, we will assume that the robot has a map of the scenario. This map also includes a model of potential interest points for persons, determined beforehand by using information about the typical tours performed by tourists. This map can be also learnt by using data gathered by the robot [12].

Once the person selects a destination, a path planner determines a path towards this destination. In order to model the problem by using the POMDP framework described above, this path is then discretized into a set of points with a granularity that can be adjusted (1.5 meters in the current implementation). The map of the scenario is analyzed to discover if in the way towards the final destination there are points that may be of interest for the person and are considered as intermediate goals.

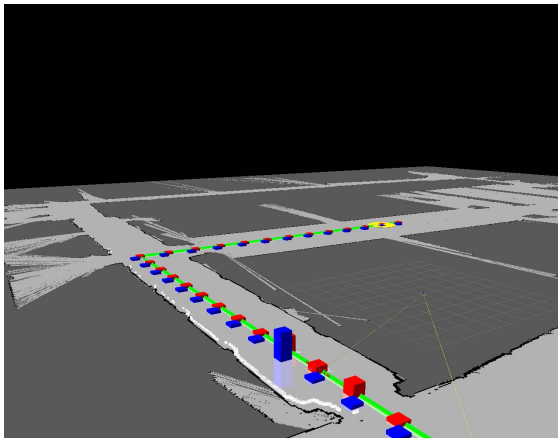


Fig. 4. The model employed: the planned path (in green) is discretized. At each point, the bars blue and red indicate the estimated belief on the robot and person positions respectively. Furthermore, the yellow circles indicate the current robot belief on the person goal. The sizes are proportional to the marginal probabilities.

The state space S is then composed by the position of the robot and the person within the discretized path, as well as the estimated goal of the person. The robot will maintain, thus, a belief state (a probability distribution) over its pose, the person position and the person intention (see Fig. 4).

The belief state is initialized to the initial position of robot and person in the tour, and the goal is initially set as the destination selected (the end of the path).

3.1 Observation Model

The robot sensors considered are the output of the localization system and a camera for person tracking.

The localization of the robot is provided by a map-based Monte-Carlo localization algorithm using the laser data of the robot. By using this algorithm, the robot can know its position within the path. The accuracy of the algorithm for different parts of the map is known beforehand and is included into the POMDP model. This model assigns some probability of obtaining a robot position measurement in the surroundings of the real position (for instance, in the corridor that can be seen in Fig. 4, the robot has some uncertainty in its position on the direction of the corridor because of the symmetry of the scenario; this can be included into the observation model).

For person guidance, a visual tracker is employed. The tracker is able to cope with illumination changes and it is quite robust [10]. However, the current robot system cannot orient the camera independently. Furthermore, it is only able to track the person if it is below a certain distance threshold from the camera. On

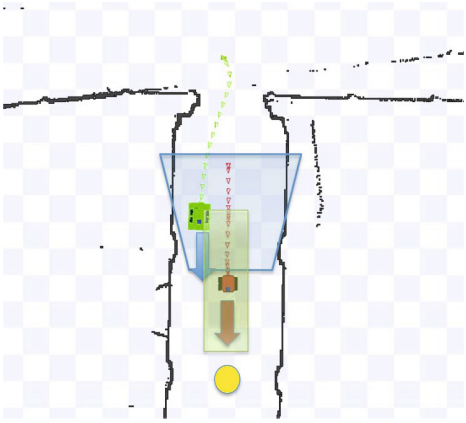


Fig. 5. Prediction models (arrows): the robot (red) and the person (green) can move forward or stay. If the robot decides to proceed forward it will reach the next point on the trajectory with certainty. The motion of the person is uncertain and will also depend on the presence of intermediate goals (yellow). Observation models: the robot can detect with its camera if the person is within its field of view (blue polygon) with high probability. This field of view is about 3 meters long. The position of the robot is obtained by the localization system with a precision depending on the place in the map (green rectangle).

the other hand, it is able to recognize persons that has already seen before, so if a person goes out and back into the field of view it will recognize him.

Therefore, the model considers a high probability of detection when the person is in the field of view of the camera (see Fig. 5), while accounting for a small possibility of misdetections.

3.2 Prediction Function

The action space of the robot considered here is to continue to the next discrete point on the trajectory or to wait for the person (it controls the speed in a on/off fashion, although this is implemented smoothed in the velocity controller).

The robot actions are modeled as deterministic (that is, the robot will reach the next point if commanded so, even though the possibility of crowded places in the scenario may be modeled).

The motion of the person is uncertain, though. The model employed considers that the person will adapt to the robot pace, but he may wander a bit, so there is some small probability that the person stays at his current position. Furthermore, the intentions of the person will depend on the place he is transversing. If the person is close to an intermediate goal, then the person may decide to stop by. Therefore, in those cases there is a larger probability of staying and changing the person goal.

3.3 Design of the Reward Function

The main design choice in a POMDP is the reward function, with which we encode the desired behavior of the robot. For this task, the objective is to follow the path, guiding the person and adapting to the person motion and intentions. Furthermore, it is important not to lose the person.

Therefore, the robot receives a positive reward if it goes towards the goal of the person and maintains the person below a certain distance. A larger reward is received if the final destination is reached than for the intermediate goals. However, the robot receives a penalization if it proceeds towards a different goal than that of the person (the person may stop for a while in one of the intermediate goals). This requires to reason about the potential intentions of the persons.

Besides, the actions are penalized so faster achievements are preferred.

4 Simulations

In order to analyze the modeling of the task by using POMDPs, in this section we present some results obtained by applying the mentioned algorithm in simulations. In the simulation, the robot is controlled by the algorithm, while

Fig. 6. Typical execution of the POMDP controller guiding the person (green) towards the destination

Table 1. Comparative table for a planning depth of 5

Algorithm	Number of nodes	Time (s.)	Averaged Reward
RTBSS	7776	6.75	241.52
FSBS (threshold 0.3)	108	0.1	238.34

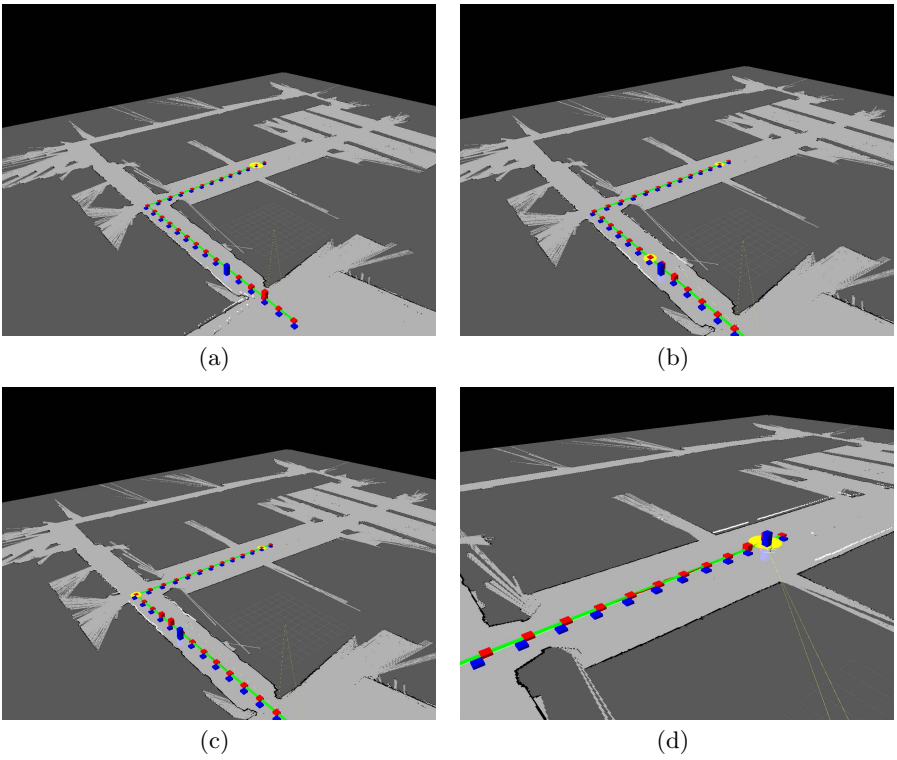


Fig. 7. a) The initial situation: the estimated goal (in yellow) is the initial destination. b) The robot and the person (the belief on their positions in blue and red, respectively) are approaching a potential intermediate goal. Therefore, the robot puts some probability mass on this goal (yellow), as well as the one at the final destination. This provokes that the robot slows down, as it considers that the person may stay at the intermediate goal. c) The person stays for a while in the intermediate goal, but then proceeds. When doing this he goes out of the field of view of the camera. However, the models and the Bayes filter allow to use this negative information to infer the motion of the person and the robot proceeds towards the final destination again. The belief on the potential goals change (yellow). d) The robot and person reach the final destination.

the person is simulated by a robot teleoperated by a person. Figure 6 shows the typical execution of the algorithm.

In the example, the trajectory selected is discretized into 27 zones, in which 2 intermediate goals, besides the final destination, are identified, giving a state space of dimension $|S| = 2187$ ($27 \times 27 \times 3$), with $|A| = 2$ potential actions and $|O| = 54$ potential observations (the position of the robot and if the person is see or not by the camera tracker). An AND-OR tree for this problem and depth 5 contains more than 10^{10} nodes.

Table 1 shows a comparison of the number of nodes and reward obtained by different methods and a planning depth of 5. As it is seen, the presented algorithm can run in real-time, obtaining a similar reward as the one obtained by the algorithm RTBSS [18].

Fig. 7 describe some of the situations encountered in the execution. It can be seen how the system is able to reason about its limitations in its sensor data and in the models in order to adapt itself to the situation. In the set of simulations, the average distance between the robot and the person is of 2.04 meters.

5 Conclusions

The paper has presented the application of POMDPs to navigation tasks, in particular a robot guiding task. POMDPs are a way of reasoning about the uncertainties of the system when controlling the robot, which allows to enhance the robustness of the robot operation. In order to apply these methods to robotics, it is needed to develop techniques able to cope with larger state, action and observation spaces. The paper presents one method to alleviate the complexity of POMDPs. Using this method it is possible to apply a POMDP model to the guiding task in real-time.

As a future work, we will perform actual experiments in a guiding setup, and compare the technique with other approaches for guiding [16].

Furthermore, the FSBS method will be extended to build belief graphs, which should reduce further the number of expanded nodes, allowing to cope with larger problems. Furthermore, nothing precludes to use the same ideas for continuous representations of the belief space, as well as considering continuous actions, which is very relevant for robotic applications.

Regarding the use of POMDPs for navigation tasks in pedestrian environments, one of the main issues is the design of the reward functions to define the particular task. A more interesting approach is to learn the reward function from demonstration or examples from humans [7]. This will also allow to consider social behaviors into the navigation stack, by transferring the way human guides behave to the robot.

Acknowledgments. This work is partially funded by the European Commission 7th Framework Programme under grant agreement no. 288235 (FROG), and the regional project ROMOCOG (P10-TEP-6412, P09-TEP-4479).

References

1. Ballesteros, J., Merino, L., Trujillo, M.A., Viguria, A., Ollero, A.: Improving the efficiency of online POMDPs by using belief similarity measures. In: Proc. International Conference on Robotics and Automation, ICRA (2013)
2. Bennewitz, M., Burgard, W., Cielniak, G., Thrun, S.: Learning motion patterns of people for compliant robot motion. I. J. Robotic Res. 24(1), 31–48 (2005)
3. Bonet, B., Geffner, H.: Solving pomdps: Rtdp-bel vs. point-based algorithms. In: Proceedings of the 21st International Jont Conference on Artifical Intelligence, IJCAI 2009, pp. 1641–1646. Morgan Kaufmann Publishers Inc., San Francisco (2009)
4. Capitan, J., Spaan, M., Merino, L., Ollero, A.: Decentralized Multi-Robot Cooperation with Auctioned POMDPs. The International Journal of Robotics Research 32, 650–671 (2013)
5. Feil-Seifer, D., Mataric, M.: People-aware navigation for goal-oriented behavior involving a human partner. In: Proceedings of the IEEE International Conference on Development and Learning, ICDL (2011)
6. He, R., Bachrach, A., Roy, N.: Efficient planning under uncertainty for a target-tracking micro-aerial vehicle. In: Proc. International Conference on Robotics and Automation, ICRA (2010)
7. Henry, P., Vollmer, C., Ferris, B., Fox, D.: Learning to navigate through crowded environments. In: ICRA 2010, pp. 981–986 (2010)
8. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence 101, 99–134 (1998)
9. Kurniawati, H., Hsu, D., Lee, W.: SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: Proceedings of the Robotics: Science and Systems Conference, Zurich, Switzerland (2008)
10. Liwicki, S., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. IEEE Transactions on Neural Networks and Learning Systems 23, 1624–1636 (2012)
11. López, M., Bergasa, L., Barea, R., Escudero, M.: A navigation system for assistant robots using visually augmented POMDPs. Autonomous Robots 19(1), 67–87 (2005)
12. Luber, M., Tipaldi, G.D., Arras, K.O.: Place-Dependent People Tracking. International Journal of Robotics Research 30(3) (March 2011)
13. Merino, L., Gilbert, A., Capitan, J., Bowden, R., Illingworth, J., Ollero, A.: Data Fusion in Ubiquitous Networked Robot Systems for Urban Services. Annals of Telecommunications, Special Issue Ubiquitous Robots 67 (2012)
14. Ong, S., Png, S.W., Hsu, D., Lee, W.S.: POMDPs for Robotic Tasks with Mixed Observability. In: Proc. Robotics: Science and Systems, RSS (2009)
15. Papadimitriou, C., Tsitsiklis, J.N.: The complexity of markov decision processes. Mathematics of Operations Research 12(3), 441–450 (1987)
16. Perrin, X., Colas, F., Pradalier, C., Siegwart, R.: Learning to identify users and predict their destination in a robotic guidance application. In: Howard, A., Iagnemma, K., Kelly, A. (eds.) Field and Service Robotics. STAR, vol. 62, pp. 377–387. Springer, Heidelberg (2010)
17. Pineau, J., Gordon, G., Thrun, S.: Point-based value iteration: An anytime algorithm for POMDPs. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico (2003)
18. Ross, S., Pineau, J., Paquet, S., Chaib-draa, B.: Online planning algorithms for POMDPs. Journal of Artificial Intelligence Research (2008)

19. Siegwart, R., Arras, K.O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., Philippsen, R., Piguet, R., Ramel, G., Terrien, G., Tomatis, N.: Robox at Expo.02: A large-scale installation of personal robots. Robotics and Autonomous Systems 42(3-4), 203–222 (2003)
20. Smith, T., Simmons, R.: Heuristic search value iteration for POMDPs. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pp. 520–527. AUAI Press (2004)
21. Smith, T.: Probabilistic Planning for Robotic Exploration. Ph.D. thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (July 2007)
22. Smith, T.: ZMDP software for POMDP and MDP planning (2012), `http://www.cs.cmu.edu/~trey/zmdp/`
23. Spaan, M.T.J., Vlassis, N.: Perseus: Randomized point-based value iteration for POMDPs. Journal of Artificial Intelligence Research 24, 195–220 (2005)
24. Taha, T., Miro, J.V., Dissanayake, G.: Pomdp-based long-term user intention prediction for wheelchair navigation. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 3920–3925 (2008)
25. Thompson, S., Horiuchi, T., Kagami, S.: A probabilistic model of human motion and navigation intent for mobile robot path planning. In: Gupta, G.S., Mukhopadhyay, S.C. (eds.) ICARA, pp. 663–668. IEEE (2009)
26. Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Hahnel, C.: Probabilistic algorithms and the interactive museum tour-guide robot minerva. The International Journal of Robotics Research 19, 972–999 (2000)
27. Trautman, P., Krause, A.: Unfreezing the robot: Navigation in dense, interacting crowds. In: IROS, pp. 797–803. IEEE (2010)

Part XIV

Perception for Mobile Robots

Learning-Based Floor Segmentation and Reconstruction

Jose Pardeiro, Javier V. Gómez, David Álvarez, and Luis Moreno

Robotics Lab, Department of Systems and Automation, University Carlos III of Madrid, Leganés, 28911, Spain
jose.pardeiro@gmail.com, {jvgomez,dasanche,moreno}@ing.uc3m.es
WWW home page: http://roboticslab.uc3m.es/

Abstract. This paper presents a comparison of the different colour spaces used in an environment modelling algorithm. This algorithm is based on the fusion of depth and colour information of a low-cost RGB-D camera to model an indoor environment. This modelling is based on creating and updating Gaussian models of the colour of the walkable floor. The analysis carried out tests the performance of three different colour spaces, obtaining the best choice to get a correct floor segmentation and reconstruction. As the results show, the algorithm performance highly depends on the colour space chosen. The method has been evaluated in a set of frames representing different environments captured with a RGB-D camera.

Keywords: Floor segmentation, colour spaces, RGB-D sensor.

1 Introduction

Nowadays a correct environment modelling is a very important technique to develop good mobile robots. To achieve a reliable autonomous motion the robot has to know which zones are walkable and which ones are occupied by obstacles. These obstacles can be static, such as paper bins, boxes or furniture, or dynamic like people walking, and the robot has to be able to separate these obstacles from the rest of the environment to detect the walkable zones.

Traditionally researchers have used different sensors trying to generate a tridimensional representation of the environment with different algorithms and techniques. In these techniques, the main purpose consists on detecting the free space immediately around the mobile robot rather than the specific wall-floor boundary. Some examples can be found in [1], where 4 stereo cameras are used to accomplish this task. An alternate approach was proposed in [2], using a combination of colour and gradient histograms to distinguish free space from obstacles.

Only a few of researchers have considered the floor segmentation a problem by itself. The techniques employed by these researchers want to utilise the ground plane restriction in different forms like planar holographs to optical flow vectors [3,4].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_23,

In 2005 the Department of Defense from USA organized a contest called DARPA Grand Challenge, consisting in moving a car from one point to another one without any human intervention. The Stanford University won this contest. Among the different systems implemented in the car, they used a segmentation of walkable floor algorithm which allowed their car to circulate at the highest average velocity [5]. For this task they used two different sensors: laser range finders and a standard RGB camera. Their solution combined the information of the sensors to be able to increase the walkable floor segmentation capability.

Our algorithm is based on the same idea. However, we focus on indoor environments and a low-cost RGB-D camera is used. This type of camera provides a modern technology at low cost, so it is not necessary to have a very expensive equipment to get a correct modelling. However, some researches about the performance of a RGB-D camera in environment modelling has been done with a successful results [6,7,8]. In this work, some common problems of image processing in indoor environments such as: brightness and colour changes caused by artificial lightning, similarity among different areas in the environment and so on, are addressed, and a measure of its performance its presented.

The next sections of the paper are organised as follows. In section 2 the description of the algorithm is done. Section 3 contains the experimental results and their analysis. Finally, in section 4 conclusions and future work are addressed.

2 Segmentation of Walkable Floors

In this algorithm a low-cost RGB-D camera has been used. These cameras consist of an standard RGB camera and a set of depth sensors able to measure distances. These sensors are capable of detecting the environment geometry in a range of about 7 meters, but this range may not be enough depending on the application and the velocity of the robot. For example, in figure 1 a capture of an RGB-D camera located in the middle of a corridor is shown. As it can be seen, the range measured by the RGB-D camera depth sensors is lower than the RGB image range. One of the main goals of this algorithm is to overcome this limitation merging the information provided by both sensors. The algorithm used for the fusion of this information is based on the research presented in [5].

The employed algorithm is based on learning the colour parameters of the floor segmented with the depth information. The estimation of the walkable floor out of range of the depth sensors is based on the assumption that the floor has the same colour properties than the previously segmented floor. Using this supposition the algorithm is able to estimate the walkable zones located outside of range of the depth sensors.

The algorithm starts working with the point cloud obtained by the RGB-D camera depth sensors. With this information a tridimensional representation of the environment can be obtained. First, these data is processed to be able to detect and segment the floor. Later, a transformation from points in the 3D space to pixels in the RGB image is needed to get colour parameters of the floor. To reduce the high quantity of information, a clustering technique is used to

(a) Original point cloud. (b) Original point cloud over RGB image.

Fig. 1. Example of image captured with RGB-D camera

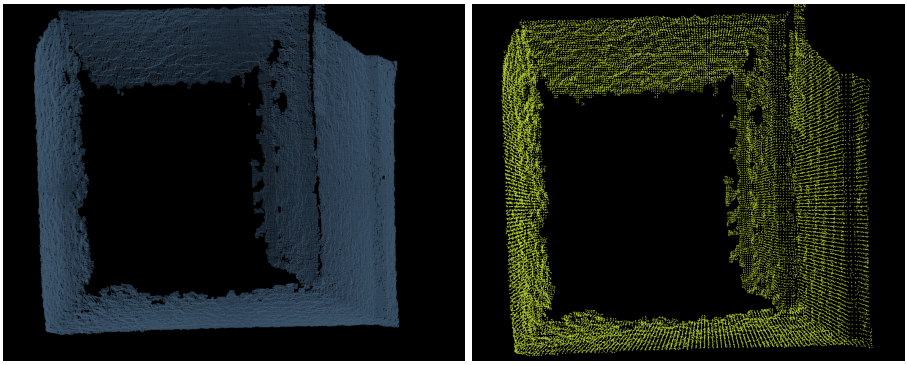
group it and the resulting groups are defined by a simple statistical function to reduce the time consumption of the next processes. Finally, a statistical measure of the distance between each pixel and the functions obtained before is used to classify the colour pixels. When the colour of a pixel is considered to be near the function that defines the colour parameters of the floor, this pixel is marked as part of the floor. In the opposite case, the pixel is discriminated and considered as an obstacle. All these steps are detailed in the following points.

2.1 Filtering

The algorithm begins with a filtering process. The filtering consists on reducing the amount of data obtained as much as possible while maintaining the desired features of the input. With this process the algorithm improves its speed and avoids some future problems due to the excess of information. The filter used is a voxel grid filter.

The voxel grid filter generates a grid of cubes which have an identical size. This filter is applied to all the data of the cloud. All the points which belong to the same voxel in the grid are replaced by their centroid. The output data keeps the original geometrical information of the environment but has fewer points and a more uniform density.

The data reduction obtained depends on the cube size, so the output grid obtained is bigger when the cube is smaller. Finding an appropriate voxel grid size with the objective to achieve an equilibrium between speed and filter quality is a critical, task dependent problem. An aggressive filtering can erase relevant information but a low filtering makes the algorithm to be excessively slow producing errors in the following steps. The decision on the size of the cube has been made empirically and the result shown in figure 2.



(a) Original point cloud with 208974 points. (b) Filtered cloud with cube size 3 cm with 36242 points.

Fig. 2. Input cloud filtering

2.2 Segmentation of the Point Cloud

The second step of the algorithm consists on the segmentation of the point cloud into the different main planes that belong to the cloud. To compute these planes the RANSAC [9], abbreviation of RANdom SAmple Consensus, algorithm is used. This method adjusts, in an iterative way, the parameters of the plane that contain the highest quantity of points. To minimise possible errors, RANSAC also evaluates the distance between different points, discriminating the points that overcome a limit.

Mathematically a plane can be described as:

$$ax + by + cz + d = 0 \tag{1}$$

When the coefficients of all the planes are obtained, they are analysed to determine their position in the 3D environment. For this purpose, the camera is assumed to be placed so that the z axis of the depth sensor is parallel to the floor. Using the camera as a reference, a plane is considered to be horizontal if $a, c \approx 0$ and $b \approx \pm 1$ since y is constant for every (x, z), and the height of the plane is $y \approx -d/b$. Therefore, if $y < 0$ the plane is the labelled as floor and as ceiling otherwise. The results of the segmentation of the planes are shown in figure 3 and table 1.

2.3 Pixels Which Pertain to the Floor

Once the planes have been analysed and the floor identified, the colour information of the floor is extracted to be analysed. This is done using the pin-hole model described on equation 2.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_i/Z_i \\ Y_i/Z_i \\ 1 \end{pmatrix} \tag{2}$$

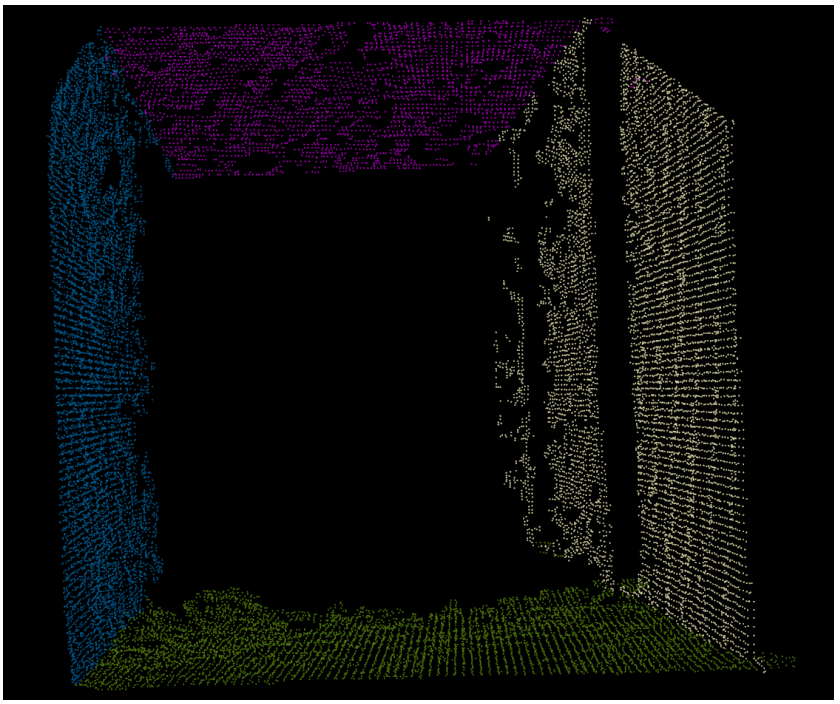


Fig. 3. Point clouds segmented

Table 1. Coefficients of generated planes

	a	b	c	d
Right wall	0.990	0.995	0.124	1.134
Left wall	0.995	-0.030	0.093	-1.204
Roof	-0.047	0.999	0.025	0.920
Floor	-0.044	-0.999	-0.01	-1.954

where u and v represent the pixel position, f_x and f_y the focal distance on x and y axes expressed in pixel-related units, C_x and C_y the principal point coordinates and X_i, Y_i and Z_i represent the real distances on the three Cartesian axes.

In the previous step the floor plane was obtained. Applying the formula described on 2 to the points belonging to the floor the corresponding pixel in the RGB image (u, v) can be obtained. From these pixels the colour information is extracted, and the result is shown on the figure 4

Fig. 4. Example of floor points projected as pixels

2.4 Colour Parameters Analysis

In the original algorithm [5], all the computation needed to compare the colour parameters is simplified due to the reduction of the information of the pixels to statistical functions. In our algorithm, among all the possibilities for clustering, the k-means [10] technique has been selected. This technique divides the input data into k different groups based on a distance function. When the clustering has finished, the obtained groups are reduced to 3-dimensional Gaussian distributions, where each dimension is a parameter of the colour space chosen. A Gaussian function can be defined using its mean and its covariance. To calculate these parameters, equations 3 and 4 are used.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{3}$$

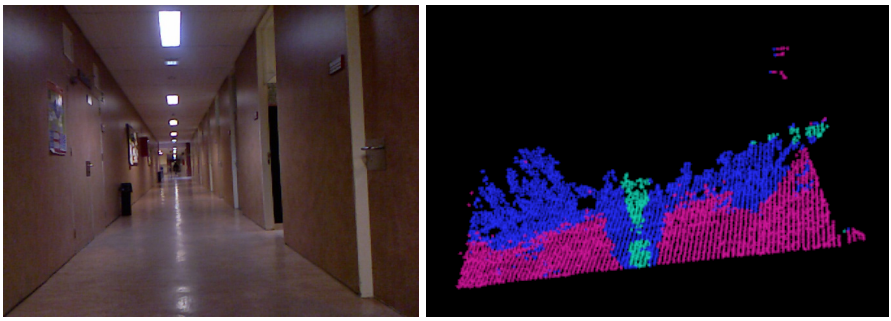
$$\sigma^2 = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} (x_i - \overline{x})^2 \tag{4}$$

The results of this process are shown in figure 5 and table 2. Each Gaussian function is assigned a mass which represent the number of points in the input

floor which are included in that Gaussian. This is useful as a measure of the importance of each Gaussian.

2.5 Region of Interest

Since the labelling of the pixels is colour-based, it could happen that some zones of the image are labelled as floor when they actually belong to walls, ceiling or obstacles. Thus, the introduction of a region of interest (ROI) when analysing the colour information is required. This region will delimit the area in which it is more likely to find pixels which belong to the floor.



(a) Original RGB image. (b) Floor k-means result with k=3.

Fig. 5. Example of the Gaussian distributions clustering

Table 2. Example of the Gaussian representation combined with K-means clustering

	R	G	B
Means 1	122.773	109.447	122.984
Variances 1	249.702	278.566	262.49
Mass 1		309	
Means 2	81.9109	62.0661	62.4672
Variances 2	30.2974	31.2947	77.3858
Mass 2		3221	
Means 3	94.288	78.1837	85.6717
Variances 3	35.3101	30.9852	60.289
Mass 3		309	

Considering simple corridors, the initial point cloud will be composed by 4 main planes: floor, walls and ceiling. As the parameters of the 3D planes have been extracted in previous steps, it is possible to find out the 3D line resulting of the intersection between the walls and the floor. Following, the computed lines are projected so the ROI parameters are expressed in terms of pixels, as shown in figure 6.

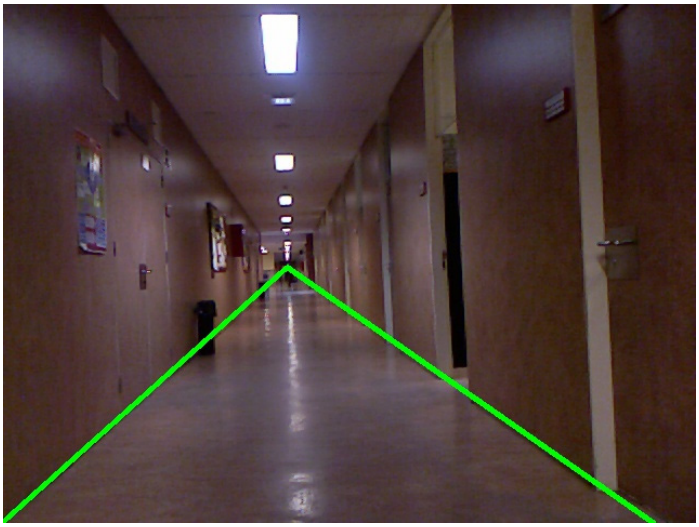


Fig. 6. Example of a region of interest

2.6 Floor Labelling

In this step all the pixels within the ROI are analysed in order to determine whether they belong to the floor or are obstacles. This labelling is done according to the distance between the Gaussian distributions of the learnt floors ($\mu_{L,i}$, $\Sigma_{L,i}$) and the colour of the pixel (u, v), denoted as $x_{(u,v)}$. In this case, the metric employed is the Mahalanobis distance $M_i(x_{(u,v)})$:

$$M_i(x_{(u,v)}) = \sqrt{(x_{(u,v)} - \mu_{L,i})^T \Sigma_{L,i}^{-1} (x_{(u,v)} - \mu_{L,i})} \tag{5}$$

Note that this model is independent of the colour space used. Eq. 5 is applied for every Gaussian distribution i obtained in section 2.4. If the minimum value $M_i(x_{(u,v)})$ is below a given threshold th, the corresponding pixel $x_{(u,v)}$ is labelled as floor i.

2.7 Map Reconstruction

This last step consists on reconstructing a map from the previous labelled data. Usually it is not possible to compute the inverse projection of the pixels because the depth information cannot be extracted with one single frame. However, in this case the pixels labelled as floor are known to lie in a 3D plane in the space. Together with the pinhole camera model, it is possible to extract the 3D position of every pixel. The pinhole camera model is defined by the equation 2 on section 2.3.

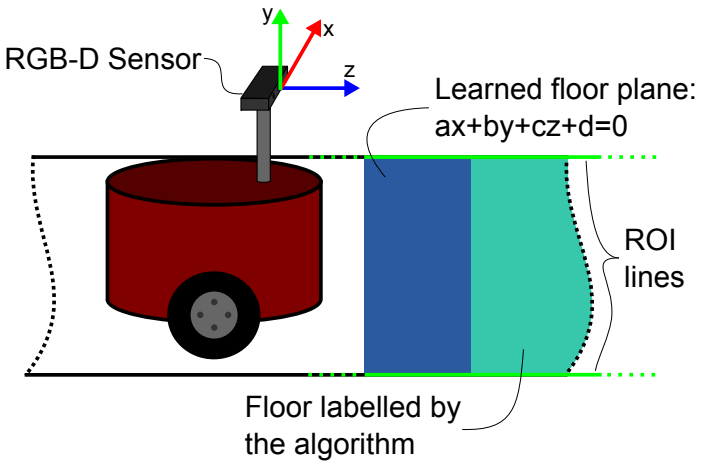


Fig. 7. Kinect coordinates

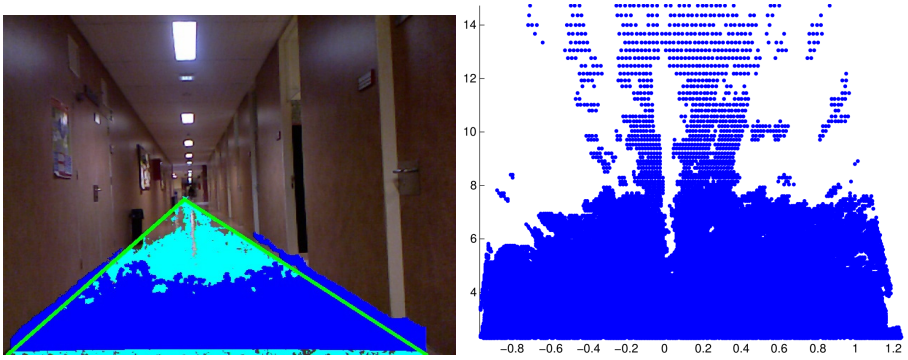
Assuming that the axis y is perpendicular to the plane OXZ, $\forall(x, z) \rightarrow z = K$, and that every point labelled as floor will lie in the plane $ax + by + cz + d = 0$, as detailed in figure 7. Therefore, it is possible to compute the inverse projection:

$$Y_i = -\frac{d}{b} \tag{6}$$

$$Z_i = f_y \frac{Y_i}{v - C_y} \tag{7}$$

$$X_i = \frac{(u - C_x)Z_i}{f_x} \tag{8}$$

The result is shown in the figure 8.



(a) Segmented floor. (b) Map Reconstruction.

Fig. 8. Example of map generation

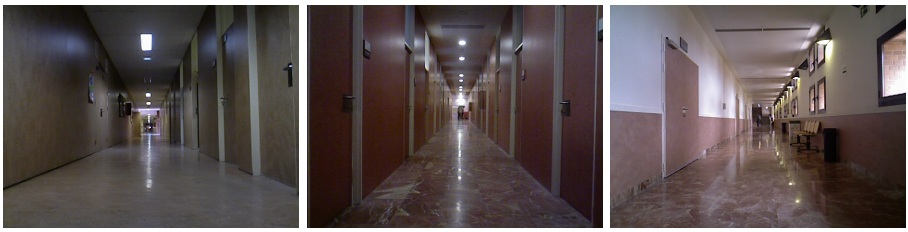
3 Experiments

In this section a comparison among three different colour spaces is carried out: RGB [11], HSV and CIELAB [12]. Although the RGB-D cameras work with RGB colour space by default, the conversions among these spaces are well known [13].

The experiments have been carried out using C++ together with the PCL [14] and OpenCV [15] libraries.

3.1 Experimental Setup

Three different scenarios (corridors) have been chosen, shown in figure 9. Obstacles have been included in all of them but only two of them have been analysed with both obstacles and clear corridor. In each case 15 frames adquired from the same point of view have been analysed. 10 different values for th are given from 2 to 5 in RGB and LAB experiments and from 20 to 50 in HSV cases.



(a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

Fig. 9. Environments employed in the experiments

The metrics employed for evaluation will be the Precision/Recall curves and also the F-score:

$$Precision = \frac{t_p}{t_p + f_p} \tag{9}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{10}$$

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{11}$$

where t_p stands for true positives, f_p for false positives and f_n means false negatives. The results are expressed in terms of means along all the frames for every value of th.

In this case, precision represents the rate of pixels labelled as floor correctly, among all the pixels actually considered as floor. Recall presents the sensitivity of the algorithm to detect floor pixels. And finally, the F term is a combination of the afore mentioned metrics evenly weighted.

3.2 Results Analysis

First, the algorithm is applied to the two obstacles-free environments in order to test the its performance in an ideal case. As figure 10 shows, the precision

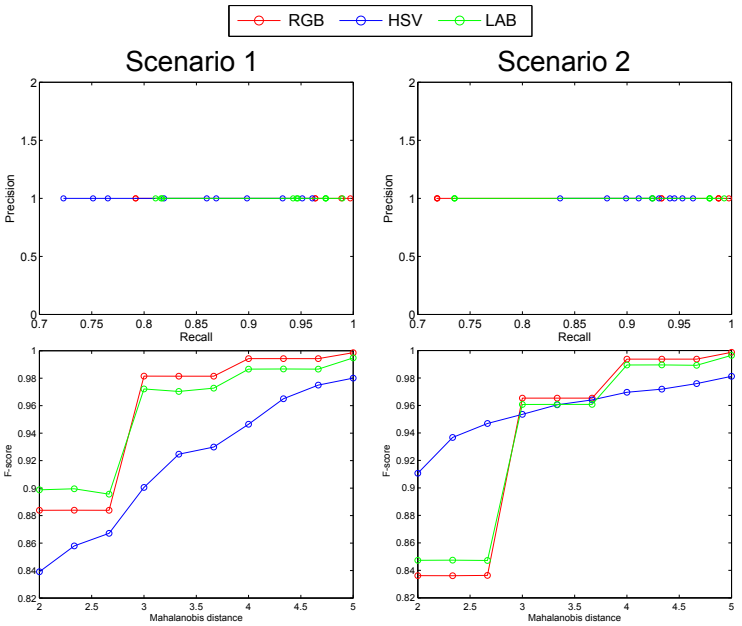


Fig. 10. Results in scenarios 1 and 2 with no obstacles

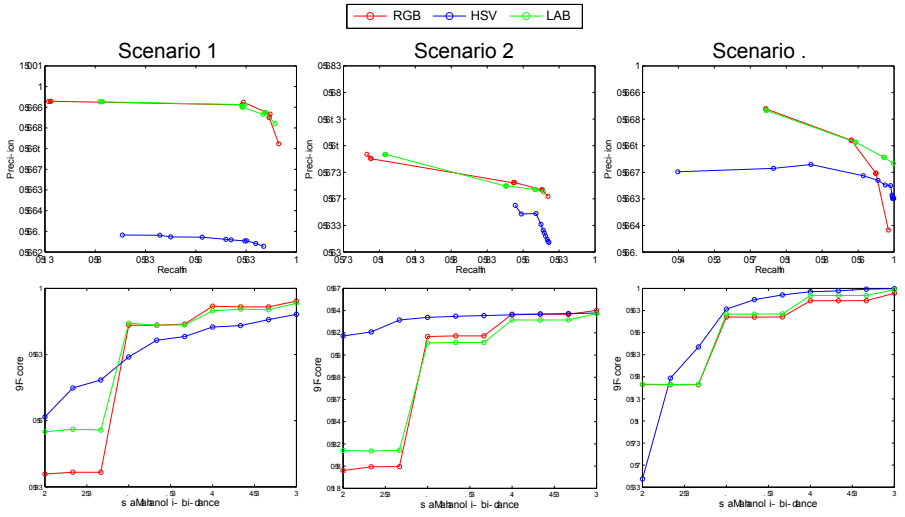
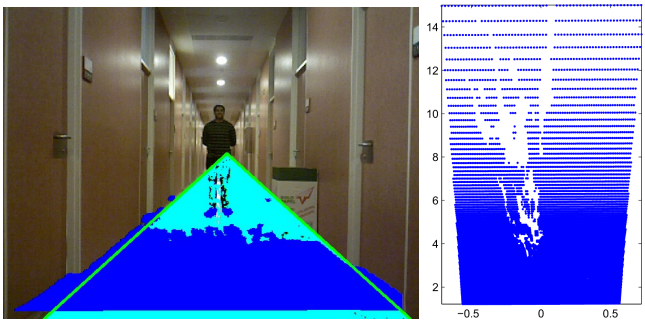
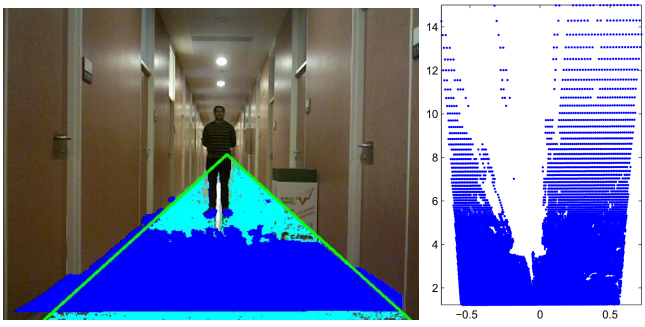


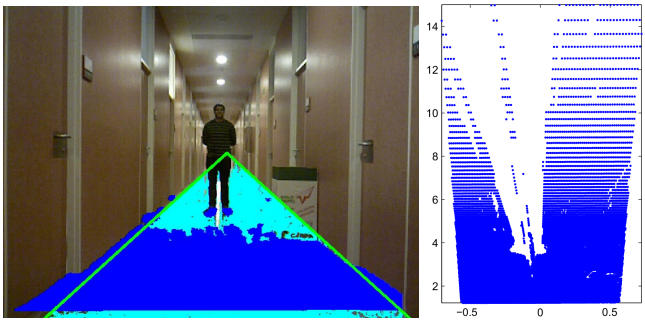
Fig. 11. Results in the three scenarios with obstacles in different positions. Mahalanobis values of HSV are multiplied by 10.



(a) HSV, $th = 43$.



(b) LAB, $th = 4.3$.



(c) RGB, $th = 4.3$.

Fig. 12. Example of the results for the environment 2. Left: colour space view. Right: 2D map reconstruction, scales are in meters.

is always 1 because all the pixels within the ROI are floor, $f_p = 0$. The recall improves since an increment in the th value makes the algorithm to be more permissive.

After, the algorithm is applied on three different environments with obstacles, such as people or paper bins, and the results are showed in figure 11.

In all cases, the precision of LAB and RGB are very similar, both of them higher than HSV. It means that HSV labels more obstacles as floor than the others. However, the HSV recall value is usually higher. As the th value becomes more permissive, the precision falls while the recall value increases. In other words, the algorithm reduces the number of f_n but the amount of f_p is higher.

Focusing on the F-score, for restrictive values of th the HSV colour space is more reliable. But if a balanced algorithm is desired, RGB and LAB are the better choices.

A visual example of the results is given in figure 12. In these images, dark blue represents the area labelled as floor by the depth sensors, and light blue represents the zone labelled as floor by the algorithm, all the images are evaluated with a permisive th value.

Besides, a 2D map representing the segmented floor is shown. In the latter, a good visual feedback of the different performance between the colour spaces is clearly seen. HSV example hardly avoids labellin the person in the picture as and obstacle, while the others present a good performance in close and far distances. However, all of them label the paper bin on the right as floor.

4 Conclusions and Future Work

In this paper the performance of three different colour spaces has been analysed using an environment modelling algorithm. The algorithm has been applied on different environments, with and without obstacles, and Precision vs Recall and F-score graphics has been used to evaluate the performance.

LAB and RGB colour spaces have produced the best results on the algorithm with a moderate th value obtaining an equilibrium between floor labelled and obstacles labelled. HSV colour space is too sensible on the definition of a colour. For this reason this colour space obtains the best results when the th value is restrictive.

Some problems have been produced on different environments, which are a bit difficult to solve. When a bright area is in the depth sensor range, it usually considered as an independent Gaussian distribution, therefore this colour is labelled as floor and it will lead to errors. The same problem appears when an obstacle has the same colour than the floor, because the algorithm detects different pixels with a very similar colour. In theses cases it is not possible to tell the difference between floor and obstacles by using only colour pixel information.

With this contribution a better colour space selection can be done. These results could be enough to choose the correct space colour in environment modelling applications based on colour data.

Future work is focused on improving the algorithm performance introducing new parameters related to the planes information to try to avoid some problems if the object has been detected by the depth sensor.

Acknowledgement This work was supported by the projects number DPI2010-17772 and CSD2009-00067 HYPER-CONSOLIDER INGENIO 2010 of the spanish MICYT.

References

1. Sabe, K., Fukuchi, M., Gutmann, J.-S., Ohashi, T., Kawamoto, K., Yoshigahara, T.: Obstacle avoidance and path planning for humanoid robots using stereo vision. In: Proceedings of the IEEE International Conference on Robotics and Automation (2004)
2. Lorigo, L.M., Brooks, R.A., Grimson, W.E.L.: Visually-guided obstacle avoidance in unstructured environments. In: IEEE Conference on Intelligent Robots and Systems, vol. 1, pp. 373–379 (1997)
3. Kim, Y.G., Kim, H.: Layered ground floor detection for vision-based mobile robot navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation (2004).
4. Zhou, J., Li., B.: Robust ground plane detection with normalised homography in monocular sequences from a robot platform. In: Proceedings of the International Conference on Image Processing (2006)
5. Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., Bradski, G.: Self-supervised monocular road detection in desert terrain. In: Proceedings of Robotics: Science and Systems, Piladelphia, USA (2006)
6. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D Mapping: Using Depth Cameras for Dense 3D Modelling of Indoor Environments. In: International Symposium on Experimental Robotics (2010)
7. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: Using Kinect-style depth cameras for dense 3D modelling of indoor environments. The International Journal of Robotics Research, 1–17 (2012)
8. Pardeiro, J., Gómez, J.V., Álvarez, D., Moreno, L.: Estimación de suelos navegables para interiores. Robocity 2030: Robots sociales. Leganés, Spain (2013)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395 (1981)
10. Kanungo, T.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 881–892 (2002)
11. Pascale, D.: A Review of RGB colour spaces...from xyY to R'G'B'. The Babel-Colour Company (2008)
12. Schanda, J.: COLORIMETRY: Understanding the CIE System. John Wiley and Sons, Inc. (2007).
13. Tkalcic, M., Tasic, J.F.: Colour spaces - perceptual, historical and applicational background. In: EUROCON 2003 (2003); Computer as a Tool. The IEEE Region 8, 304–308 (2003)
14. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: Proc. Int. Conference on Robotics and Automation, ICRA (2011)
15. Baggio, D.L., Emami, S., Millán Escrivá, D., Ievgen, K., Mahmood, N., Saragih, J., Shilkrot, R.: Mastering OpenCV with Practical Computer Vision Projects. Packt Publishing, Birmingham (2012)

Perception and Navigation in Unknown Environments: The DARPA Robotics Challenge

Eduardo J. Molinos, Ángel Llamazares, Noelia Hernández, Roberto Arroyo, Andrés Cela, José Javier Yebes, Manuel Ocaña, and Luis Miguel Bergasa

Department of Electronics, University of Alcalá, Madrid, Spain
{emolinos,allamazares,nhernandez,rarroyo,acela,jyebes,
mocana,bergasa}@depeca.uah.es

Abstract. This paper presents different techniques to achieve the tasks proposed in the DARPA (Defense Advanced Research Projects Agency) VRC (Virtual Robotics Challenge), which entails the recognition of objects, the robot localization and the mapping of the simulated environments in the Challenge. Data acquisition relies on several sensors such as a stereo camera, a 2D laser, an IMU (Inertial Motion Unit) and stress sensors. Using the map and the position of the robot inside it, we propose a safe path planning to navigate through the environment using an Atlas humanoid robot.

Keywords: Humanoid Robots, Navigation, Mapping, Perception, DARPA Robotics Challenge.

1 Introduction to Virtual Robotics Challenge

The DARPA Robotics Challenge [1] is an open competition focused on the development of humanoid robots with the aim of operating in rough terrain and austere conditions using human tools commonly available in real environments. The Virtual Robotics Challenge is a preliminary phase of the competition carried out in simulated scenarios and using open source tools like ROS (Robot Operating System) [2] and Gazebo [3]. Three different scenarios are proposed: get into a vehicle and drive it, walk through rough environments and manipulate unknown objects.

We have participated in this challenge as SARBOT Team [4], which is composed of three universities (Universidad Politécnica de Madrid, Universidad Carlos III and Universidad de Alcalá) and a research centre (Centro de Automática y Robótica). Universidad de Alcalá group is focused on the perception of the environment, mapping abilities and navigation planning.

2 VRC Proposal

The Virtual Robotics Challenge is divided into three scenarios with multiple tasks that have to be solved. We have divided the scenarios into simpler tasks so our multidisciplinary team could work on it in an efficient way.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_24,

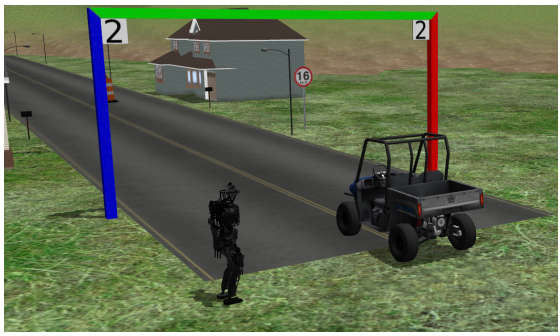


Fig. 1. Scenario 1. Atlas robot facing the car

The Atlas Robot [5] is used to fulfil the tasks in all the three scenarios. This humanoid robot is equipped with several sensors, such as: stereo camera on the head of the robot, a Hokuyo UTM-30LX laser mounted into a rotating platform that can turn 360 degrees with an integrated IMU, an IMU in the pelvis of the robot, force sensors in the robot's foot plants, stress sensors in all the joints of the robot and a pair of Sandia hands (each of one equipped with a camera and force sensors).

Scenario 1 is mainly focused on the task of driving a car. In order to complete this task, we have to localize the robot into the starting pen, find the way out, detect the car and get close to its driver door, get into the car, start the car, drive it through the gates of the road, stop the car, get out the car and walk through the last gate. Figure 1 shows the Atlas Robot in the environment facing the vehicle.

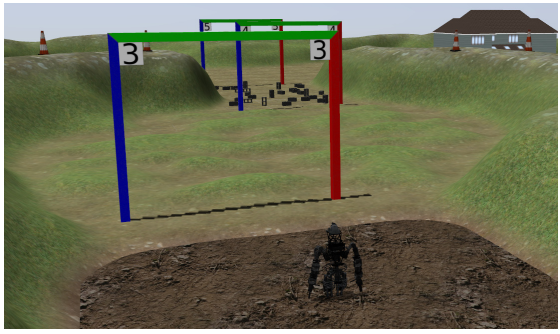


Fig. 2. Scenario 2. Atlas robot walking across the mud pit

Scenario 2 is mainly focused on the task of walking in difficult terrain. As in the first scenario, the robot is initially into a starting pen and has to walk through the exit. Then, it has to detect the first gate and walk through it, go

down a ramp into the mud pit, walk across the mud pit in the direction of the second gate, go up the ramp to get out the mud pit, walk in a rough terrain, cross the third gate and finally walk across a terrain full of obstacles. Figure 2 shows the Atlas Robot walking across the mud pit.

Scenario 3 is mainly focused on the manipulation task. The robot must get out the starting pen, detect a table with a hose on it, approach to the table, grab the hose, connect it to a standpipe located on the wall and open a circular valve. Figure 3 shows the Atlas Robot facing the table to grasp the hose.

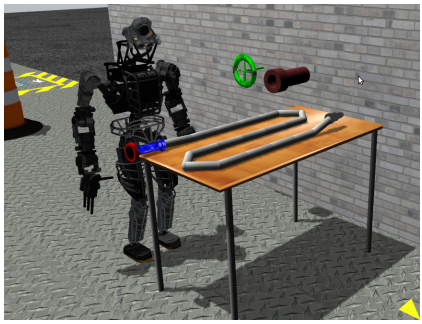


Fig. 3. Scenario 3. Atlas robot looking at the hose

3 SARBOT Team Proposal: Perception and Navigation

Some tasks are common for all the proposed scenarios by the Virtual Robotics Challenge such as navigation and localization, although the goals of the navigation are different for each of them. The challenge sets some objects that can be recognised by analysing some of its characteristics: the gates, the starting pen, the vehicle, the table, and so on. We propose a combination of different methods based on camera and laser to identify the objects in the environment and plan the goals to go.

Hereafter, we explain different methods that we use to achieve the different tasks:

3.1 Mapping

We use the laser mounted on the head of the robot for the mapping task. We use a 3-D laser point cloud representation of the environment instead of the commonly used 2-D representation. This 3-D representation of the environment is obtained by rotating the laser 360 degrees and knowing its position from the data of the IMU mounted on the laser. Usually, the point cloud could be obtained employing the disparity map available if a stereo camera is considered,

but in VRC the accuracy of this point cloud is acceptable enough only when the object has rich textures. However, this is not the case for the VRC simulated environments. Then, we discard the disparity map in favor of the 3D data from the laser.

To ensure the consistency of the 3-D maps, we transform the point cloud obtained from each measurement into a discrete representation based on Octomap [6], which is a probabilistic volumetric representation of space based on Octrees that allows to represent a full 3-D environment minimizing memory requirements. This representation along with an effective movement model of the robot (obtained by the IMU data and stress sensors) give us the robustness needed to work without a SLAM (Simultaneous Localization And Mapping) system [7]. Consequently, the computational cost is also reduced with this approach. Figure 4 depicts a 3-D map of the task 2 environment and an image of the environment.

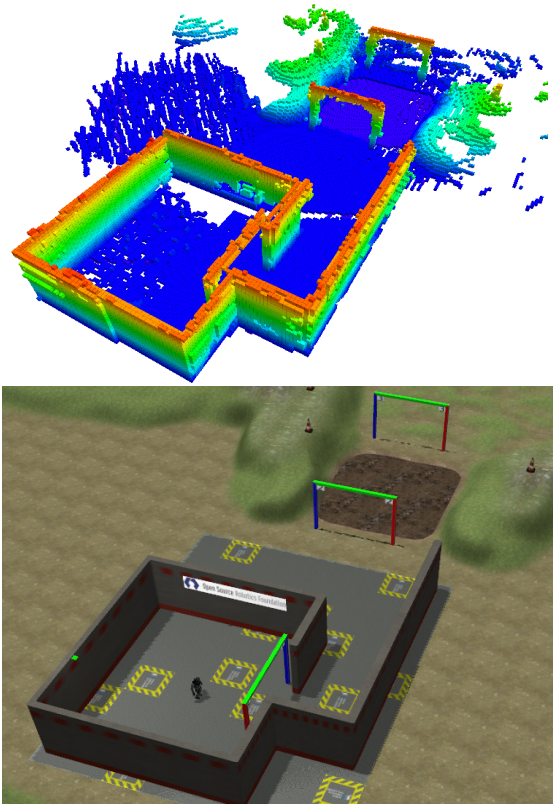


Fig. 4. 3-D Map of Task 2 Environment

3.2 Navigation

Since navigation into a 3-D environment is a computationally expensive task and the Atlas Robot has a limited range of movement in the z axis, we can transform the built 3-D map into a 2-D representation, which allows to plan the optimal way to be followed by the robot saving computational cost. We use the heights difference from one 3-D block to another to mark a 2-D cell as occupied or free. The minimum height difference to be marked as occupied is calculated knowing the prior abilities of the robot to walk through different heights.

Once we have built the 2-D map from the 3-D map, we are able to calculate the optimal route to reach the goal. To do so, we use a navigation algorithm for humanoid robots [8] which uses an ARA* (Anytime A*) Planner [9] to plan the path to be followed by the robot. This algorithm searches the path from the robot to the goal avoiding rough turns at the starting of the movement and employs a cost function based on the Euclidean distance between initial pose and goal and the number of steps that the robot has to follow. We configure this algorithm with a discretisation of steps that the robot can follow allowing it to walk straight, turn and even walk backwards. Figure 5 shows the 2-D map representation where the steps to follow are marked in different colours: steps for left foot in red and steps for right foot in green. The goal is represented as a red arrow.

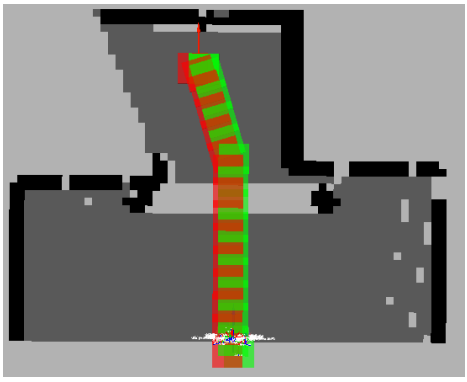


Fig. 5. Navigation. Step-by-step route

3.3 Goal Recognition

Once the navigation framework is set, recognition of different goals is needed to complete the tasks. There are different objects that can be recognised in the environment of VRC scenarios, following we show the techniques that we use to recognise them and plan the robot's navigation:

Gate Recognition: an algorithm based on a mono-camera colour detection is employed to recognise the different gates in the environment, taking into account that the gate's upper post is green, the left post is blue and the right post is red.

We train a Bayesian classifier [10] from a set of sampled images in a resolution of 800 x 800 pixels. The colour differentiation is made by employing an optimization based on Mahalanobis distance using HSV (Hue, Saturation, Value) channels, because this colour space allows a better discrimination between each type of post, attending to the hue and saturation characteristics. Then, we apply a skeleton function to ensure that all the posts are connected and the gate is fully recognised. Moreover, we use a SVM (Support Vector Machine) classifier implemented in LibLinear [11] and based on HOG (Histogram of Oriented Gradients) [12] features to identify the number of the gate and ensure we are detecting the proper gate. In this case, HOG is applied as a powerful shape descriptor which allows to differentiate the varied number silhouettes and classify them correctly. Figure 6 shows the gate recognition, its orientation with respect the robot and the number 2 detected on the gate.

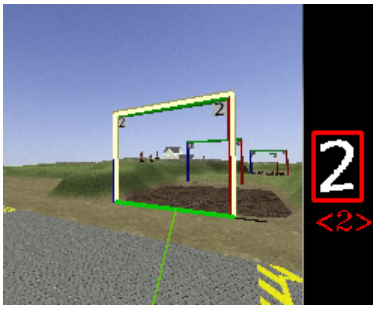


Fig. 6. Gate and number recognition

Getting Out of the Starting Pen: we have considered two different scenarios: depending on whether the first gate and the exit is on the line of sight of the camera or not. If the robot can see both sides of the gate (red and blue posts) in the image it means that the robot is looking at the exit, so the goal is the furthest point in the map in the robot's field of view. In case that only one or none of the posts of the gate are detected in the same image the robot turns until both posts are recognised. If both posts cannot be detected in the same image the robot is at the side of the exit and it moves toward the centre of the starting pen and turns until is facing the exit.

Once the robot is in the exiting corridor, we need to plan a path to exit it. Knowing that the corridor has only one exit and the entrance to the starting pen, we use an occupancy map where the upper posts of the gates and the walls are represented as occupied. This way, the entrance to the starting pen is virtually blocked and the goal is calculated as the half distance from the furthest reachable point. Figure 7 shows the map built while exiting the starting pen where black cells represent obstacles detected above the robot, dark grey cells represent detections at other heights and light grey cells represent unknown cells. The outgoing path is depicted in green and red steps, the initial pose of the robot is represented as a pink rectangle and the pink arrow represents the final goal.

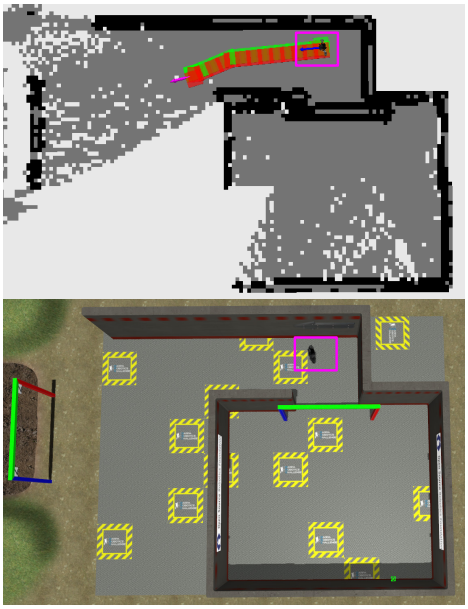


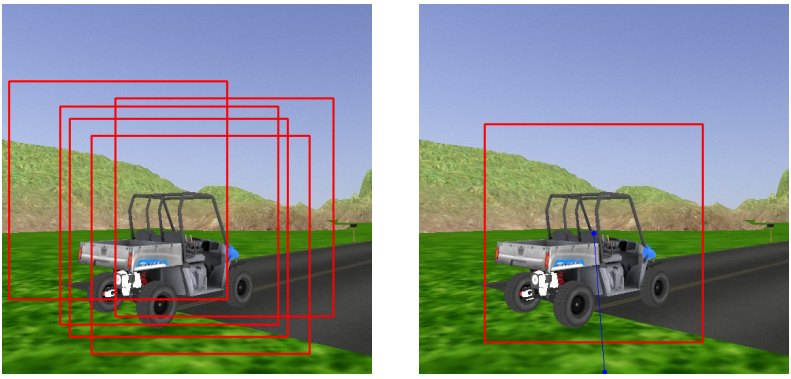
Fig. 7. Atlas robot getting out of the starting pen

Approaching to a Gate: the gate is detected using the previously explained gate recognition algorithm and the robot turns until it faces it. To be able to calculate the goal and plan the robot movement, the correspondence between the gate detected with the camera and the 2-D map representation is needed. To do so, occupancy map is built where occupied cells represent heights above the robot (to detect the gate upper posts). These cells are labeled looking for connected blobs in the robot's field of view corresponding to the gate. The goal is calculated as the blob's centroid and its direction as the normal vector of the gate.

Getting Close to the Table: assuming that the table is about one meter above the floor, we build a map where occupied cells represent objects between 0.8 and 1.2 meters high. The connected objects are classified rejecting the smallest ones. Then, the nearest object to the robot similar in size to the table is selected. The centroid of the table is used as a goal to get close to it. Once the robot is near the table the hose can be recognised by the cameras and its position can be obtained to grasp it.

Approaching to the Car: an SVM classifier based on LibLinear [11] is employed with the aim of detecting the car and its orientation in the camera image. This classifier uses a combination of appearance descriptors based on HOG and GCH (Global Color Histogram) [13], with the aim of taking into account the shape and color of the car for the classification. At first, a grid is created for classifying the different parts of the image and a sliding window algorithm is

applied to obtain the hypothetical places where the car could be located, as shown in Figure 8(a). After this, an average of the hypothetical positions of the car is carried out, obtaining the final position and its orientation in the image, as can be observed in the example presented in Figure 8(b). Once the car position is identified, the robot turns until it faces it. A map is created where the occupied cells represent objects above the floor and under 2 meters height (the height of the car) and the goal is set as the obstacle centroid to face the door and be able to get into the car.



(a) Hypothetical places detected after the sliding window classification.

(b) Final car position detected (in red) and orientation estimation (in blue).

Fig. 8. Car detection and orientation estimation

4 Conclusions

We have competed in the DARPA VRC by means of dividing complex tasks into simpler ones. In this work different techniques to navigate and map an unknown environment using heterogeneous sensors to recognise objects are presented. We have also demonstrated that a full 3-D environment can be efficiently reduced to a 2-D map to plan an optimal path that a humanoid robot can follow.

Acknowledgements. This work has been possible thanks to SARBOT Team, formed by Centro de Automática y Robótica, Universidad Politécnica de Madrid, Universidad Carlos III and Universidad de Alcalá. This work has been funded by the "Ministerio de Economía y Competitividad" through TIN2011-29824-C02-01 and TIN2011-29824-C02-02 (ABSYNTHE project) as well as by the Comunidad de Madrid through the project Robocity2030 (CAM-5-0505/DPI000176).

References

1. The Robotics Challenge, `http://www.theroboticschallenge.org/`
2. Robot Operating System (ROS), `http://www.ros.org`

3. Gazebo Simulator, http://www.gazebosim.org
4. SARBOT team, http://www.sarbot-team.es
5. Atlas Robot, http://www.bostondynamics.com/robot_atlas.html
6. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. Autonomous Robots (2013)
7. Durrant-Whyte, H., Bailey, T.: Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. IEEE Robotics and Automation Magazine 2 (2006)
8. Hornung, A., Dornbush, A., Likhachev, M., Bennewitz, M.: Anytime Search-Based Footstep Planning with Suboptimality Bounds. In: Proc. of the IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS (2012)
9. Likhachev, M., Gordon, G., Thrun, S.: ARA*: Anytime A* with Provable Bounds on Sub-Optimality. In: Neural Information Processing Systems 16: Proceedings of the 2003 Conference (NIPS 2003). MIT Press (2004)
10. Zhang, H.: The optimality of Naive Bayes. In: FLAIRS Conference, pp. 562–567 (2004)
11. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 886–893 (2005)
13. Novak, C.L., Shafer, S.A.: Anatomy of a color histogram. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 599–605 (1992)

Facilitating Human-Robot Interaction: A Formal Logic for Task Description

Takehiko Nakama*, Enrique Muñoz, Kevin LeBlanc, and Enrique Ruspini

European Center for Soft Computing
c/ Gonzalo Gutiérrez Quirós, s/n, 33600 Mieres, Spain
{takehiko.nakama,enrique.munoz,kevin.leblanc,enrique.ruspini}@softcomputing.es
http://www.softcomputing.es/

Abstract. We develop a formal logic for task descriptions that are easy to interpret for both humans and robots. Tasks are described in propositional forms that reflect syntactic structures observed in natural language so that all the resulting task descriptions can be easily understood by humans. At the same time, these propositional forms ensure that each task description is interpretable by robots. Infinitely many task descriptions can be created in this formal logic so that the formal logic can support complex human-robot interactions. We establish a hierarchy of propositions that enhances the expressive power, the interactivity, and the deductive apparatus of our formal logic. We also examine how to systematically evaluate the feasibility of each task description using the formal logic.

Keywords: formal logic, propositional logic, predicate logic, quantificational logic, fuzzy logic, human-robot interaction, peer-to-peer communication, task description.

1 Introduction and Summary

By interacting with humans, robots can effectively perform a wide range of practical tasks—for instance, assistance to people with disabilities (e.g., [25], [32], [12], [24]), search and rescue (e.g., [21], [7], [8], [30]), and space exploration (e.g., [36], [15], [14], [13]). The mode and degree of human-robot interaction vary considerably (see [18] for review). In the teleoperation mode of human-robot communication, robot operations require continuous low-level inputs from humans; consequently, even slight lapses in communications can degrade performance substantially, and the workload of the human operator can be undesirably high (e.g., [3], [6]). At the other extreme, autonomous robots that operate with no human input often perform complex tasks poorly compared to robots that are capable of collaborating with humans interactively (e.g., [1], [11], [5]). To overcome the disadvantages of the fully teleoperational and autonomous approaches, many studies have been conducted recently to develop robotic systems in which robots and humans collaborate as true team members through peer-to-peer

* Research for the authors supported by the Spanish Ministry of Economy and Competitiveness through the project TIN2011-29824-C02-02 (ABSYNTHE)

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_25,

human-robot interactions (e.g., [28], [10], [9], [19], [20]). One of the major challenges of this approach is the increased complexity of the human-robot interactions in the peer-to-peer mode (e.g., [18]).

Various schemes have been developed for the peer-to-peer mode of human-robot interaction. Many studies have proposed schemes based on natural language (e.g., [27], [2], [26]). Although natural language is clearly an effective medium for human communications, it presents several major problems when used for human-robot communications. Natural language descriptions tend to be notoriously underspecified, diverse, vague, or ambiguous, so they often lead to error spirals that are hard to overcome (e.g., [37], [35], [33], [16], [17]). Restricting natural language to a small finite set of linguistic expressions mitigates these problems but does not fully support the extensive human-robot interactions required for complex tasks. Highly mathematical or symbolic expressions may be easy for experts (e.g., mathematicians, logicians, programmers, and system administrators) to understand, but naive users cannot be expected to be able to understand them; consequently, they are unsuitable for the variety of situations in which robotic systems need to interact with naive users as well as experts. Low-level sensory and motor signals and executable code are easy for robots to interpret, but they are cumbersome for humans and thus cannot, on their own, create an effective human-robot interface.

In this paper, we take a first step toward establishing a formal logic to effectively mediate the peer-to-peer mode of human-robot interaction. Our formal logic serves as a middle ground between the natural-language-based mode of human communication and the low-level mode of robotic communication. In this scheme, each task description is a formal proposition, called a propositional task description, and it can be easily understood by humans because it resembles the corresponding natural-language sentence. At the same time, we ensure that each propositional task description can be unambiguously interpreted and executed by robots. Our formal logic has high expressive power; it can create infinitely many propositional task descriptions, just as a natural language can create infinitely many sentences. Thus, the formal logic can support highly complex human-robot interactions required by peer-to-peer human-robot collaborations. As in other formal logics, we can infer and reason in our formal logic. We also establish a hierarchy of propositional task descriptions that enhances the expressive power, the interactivity, and the deductive apparatus of our formal logic.

This study is closely related to two lines of research: situation calculus (and other logical formalisms such as event calculus and fluent calculus) and precisiation language (or precisiated natural language). Situation calculus (e.g., [29], [31]) is a formal logic for representing actions, situations, and fluents and for reasoning about them (see also [23] for event calculus and [34] for fluent calculus). While it has contributed significantly to formalizing planning and programming in robotics, its logical expressions are rather technical and thus fail our requirement of comprehensibility for naive users. In contrast, we focus on developing a descriptive and deductive apparatus to establish a hierarchy of task descriptions that results in high expressive power and interactivity for robots and a variety of users, including those who have no expert knowledge on robotic systems. For this purpose, our formal logic employs propositional forms that reflect syntactic structures observed in natural language. In this regard, our major departure from situation

calculus brings us closer to precisiation language, which is an integral part of Zadeh's computational theory of perceptions (e.g., [40], [38], [39]). In this theory, Zadeh introduced the concept of precisiated natural language (PNL), which refers to a set of natural-language propositions that can be treated as objects of computation and deduction. The propositions in PNL are assumed to describe human perceptions, and they allow artificial intelligence to operate on and reason with perception-based information, which is intrinsically imprecise, uncertain, or vague. Each proposition in PNL is translated into a precisiation language, which then expresses it as a set or a sequence of computational objects that can be effectively processed by machines (e.g., [40], [38], [39]). Our study significantly extends this line of research in two regards. To our knowledge, our study is the first to formulate the architecture or structure of the precisiation language; we accomplish this formulation by establishing a formal logic as a precisiation language. Also, our study substantially generalizes Zadeh's precisiation language. Zadeh considered the primary function of natural language as describing human perceptions, and his PNL and precisiation language only deal with perceptual propositions ([40], [38], [39]). However, the importance of natural language is not limited to describing human perceptions. In describing tasks, we need not only perceptual propositions but also action-related propositions. Our formal logic can be considered a generalized precisiation language that can handle both perceptual and action-related propositions.

The remainder of this paper is organized as follows. In Section 2, we explain propositional task descriptions that constitute our formal logic. In Section 3, we present a recursive definition of the syntax of the formal logic. In Section 4, we describe inference and reasoning in our formal logic. In Section 5, we develop a hierarchy of task descriptions. In Section 6, we discuss the generality of our scheme. In Section 7, we examine the semantics of our formal logic, which is used to represent and determine the degree of feasibility of each propositional task description.

2 Propositional Task Descriptions

In this section, we develop propositional task descriptions that enable effective human-robot interactions. As described in Section 1, we describe each task in a propositional form observed in natural language, and the resulting task descriptions form a formal logic. Thus, all well-formed formulas of the formal logic are easy to interpret for a variety of users.

In Section 2.1, we describe the components of such propositional task descriptions. In Section 2.2, we describe how to form an atomic proposition. In Section 2.3, we describe how to form a compound proposition.

In these sections, we will provide examples of rather simple task descriptions and explain our scheme in an intuitive manner. However, our scheme can be easily applied to systems that require more intricate task descriptions. In Section 6, we will discuss how our scheme can be used to mediate human-robot interactions with various degrees of complexity.

2.1 Components of Propositional Task Descriptions

To establish our propositional task descriptions, we first define sets whose elements constitute propositions. These sets will be called component sets. Consider the following component sets:

- S denotes the set of all agents that can perform a task. For instance, we can have

$$S = \{robot\ 1, robot\ 2, robot\ 3, user\}. \tag{1}$$

- V denotes the set of all verbs that characterize actions required by tasks. For instance, we can have

$$V = \{call, find, deliver, go, move, press, put\}. \tag{2}$$

- O denotes the set of all objects that may receive an action in V or compose an adverbial phrase [see (4)]. For example, we can have

$$O = \{box, button, table, room\ 1, room\ 2, building, \\ robot\ 1, robot\ 2, robot\ 3, user, null\}. \tag{3}$$

  Here the three robots and the user in S are also in O because they can also receive an action in V. The element denoted by "null" will be called the null element. In Section 2.2, we will explain how it is used in forming a task description.
- A denotes the set of all adverbial phrases that can be included in task descriptions. For instance, we can have

$$A = \{in\ \gamma, from\ \gamma, from\ \gamma_1\ to\ \gamma_2, to\ \gamma, null \\ |\ \gamma, \gamma_1, \gamma_2 \in O\}. \tag{4}$$

  Again, the null element is denoted by "null," and its use is explained in Section 2.2.
- C denotes the set of all connectives that can be used to combine multiple propositions in forming compound propositions (see Section 2.3 for compound propositions). For instance, we can have

$$C = \{and, if, or, then, until, whenever\}. \tag{5}$$

The elements in these component sets are combined in a specified manner (see Section 2.2) to form propositions (propositional task descriptions) in our formal logic.

2.2 Atomic Propositions

In our formal logic, an atomic proposition is defined to be a tuple that consists of elements in component sets. Each admissible tuple structure is specified in the form of a cartesian product of component sets. To develop task descriptions that are easy to interpret for a wide variety of users, including naive users, we employ tuple structures that reflect syntactic structures observed in natural language. For the component sets described in Section 2.1, we can define each atomic proposition to be an SVOA clause

(The S, V, O, and A in SVOA stand for subject, verb, object, and adverbial phrase, respectively) by setting the admissible tuple structure to $S \times V \times O \times A$. The SVOA structure is observed in many languages, including English, Russian, and Mandarin. Using the null element in O and A, we can also generate SVO, SVA, and SV clauses. See the following examples of atomic propositions resulting from the component sets (1)–(4).

- $\frac{robot\ 1}{S}\ \frac{move}{V}\ \frac{null}{O}\ \frac{null}{A}.$
  We will omit instances of the null element and express it more simply as $\frac{robot\ 1}{S}\ \frac{move}{V}.$

- $\frac{robot\ 2}{S}\ \frac{find}{V}\ \frac{ball}{O}.\ \left(\frac{robot\ 2}{S}\ \frac{find}{V}\ \frac{ball}{O}\ \frac{null}{A}.\right)$

- $\frac{robot\ 1}{S}\ \frac{deliver}{V}\ \frac{box}{O}\ \frac{from\ room\ 1\ to\ room\ 2}{A}.$

A task will be described atomic if it is represented by a single atomic proposition. For humans, these task descriptions are easy to specify and understand. Meanwhile, the structural and lexical constraints substantially limit the diversity and flexibility of everyday language, so we can ensure that the resulting task descriptions are unambiguously interpretable for robots. As will be explained in Section 5, we establish a hierarchy of task descriptions, and at the lowest level of the hierarchy, each atomic task description is directly associated with an indecomposable, self-contained executable code. Thus, these atomic propositions can be considered building blocks of all task descriptions, and they constitute each task description at higher levels of the hierarchy.

There are several ways to deal with the undesirable or nonsensical atomic propositions that can be formed in $S \times V \times O \times A$. (Note that in formal logics, there can be well-formed formulas that are self-contradictory.) We can remove all such propositions from the cartesian product to ensure that each resulting atomic proposition is acceptable. (In this case, we abuse the notation and let $S \times V \times O \times A$ denote the "cleaned" cartesian product.) Also, we can consider them as always false so that they will never be executed in practice (see Section 7).

2.3 Compound Propositions

A compound proposition consists of multiple atomic propositions combined by one or more connectives in the component set C. With the examples (1)–(5) in Section 2.1, we can form the following compound propositions:

- $\underbrace{\frac{robot\ 1}{S}\ \frac{put}{V}\ \frac{ball}{O}\ \frac{in\ box}{A}}_{\text{atomic proposition}}\ \underbrace{\frac{if}{C}}\ \underbrace{\frac{user}{S}\ \frac{call}{V}\ \frac{robot\ 1}{O}}_{\text{atomic proposition}}.$

- $\underbrace{\frac{robot\ 2}{S}\ \frac{go}{V}\ \frac{to\ room\ 1}{A}}_{\text{atomic proposition}}\ \underbrace{\frac{whenever}{C}}\ \underbrace{\frac{user}{S}\ \frac{press}{V}\ \frac{button}{O}}_{\text{atomic proposition}}.$

In formal logic, parentheses are used to indicate the scope of each connective. In our formal logic, they disambiguate the manner in which tasks are performed:

$$\bullet\ \frac{\underset{S}{\underline{robot\ 1}}\ \underset{V}{\underline{go}}\ \underset{A}{\underline{to\ room\ 1}}}{\text{atomic proposition}}\ \frac{if}{C}\ \left(\frac{\underset{S}{\underline{user}}\ \underset{V}{\underline{call}}\ \underset{O}{\underline{robot\ 1}}}{\text{atomic proposition}}\right.$$

$$\left.\frac{or}{C}\ \frac{\underset{S}{\underline{user}}\ \underset{V}{\underline{press}}\ \underset{O}{\underline{button}}}{\text{atomic proposition}}\right). \tag{6}$$

Tasks described by compound propositions will be referred to as compound tasks. These compound task descriptions are still quite easy for humans to specify and understand, and the syntactic structures imposed on the clauses and the compositions ensure effective interpretation and execution by robots.

There are many tasks that we can describe using conditions and imperatives. For instance, consider the compound task (6). In this description, the clause "*user call robot 1*" or "*user press button*" represents a condition that must be checked in determining whether to send the robot to room 1, and the clause "*robot go to room 1*" is an imperative. Our task descriptions consist of atomic propositions described in Section 2.2, and each atomic proposition will be either a condition or an imperative. By definition, an atomic task consists of a single atomic proposition, and the proposition is always an imperative. The type of each atomic proposition in a compound task description is unambiguously determined by the logical connective that connects it and by the location of the atomic proposition relative to the connective. For instance, an atomic proposition that forms a subordinate clause immediately following the connective "if" is considered a condition, whereas an atomic proposition that forms a clause immediately preceding the connective is considered an imperative. If the proposition is a condition, the robotic system monitors the described condition. If it is an imperative, the system executes the described action provided that all the required conditions are satisfied.

3 Well-Formed Formulas of the Formal Logic

We examine how to define the syntax of our formal logic for task description. We will provide a recursive definition of the syntax using the framework discussed in Section 2, but it can easily be generalized; we will address this issue in Section 6.

Using the component sets S, V, O, A, and C described in Section 2.1, we can define the syntax of a propositional logic as follows:

1. Any $x \in S \times V \times O \times A$ is an atomic well-formed formula.
2. If α and β are well-formed formulas, then $\alpha\ c\ \beta$, where $c \in C$, is also a well-formed formula.
3. Nothing else is a well-formed formula.

This recursive definition allows us to generate infinitely many well-formed formulas, i.e., propositional task descriptions that are unambiguously interpretable for humans and robots. With sufficiently rich component sets, we can describe any task that must be performed by the robotic system, and, using the syntax, we can ensure that the robotic

system does not operate on any ill-formed task descriptions (i.e., task descriptions that are not interpretable).

With more extensive component sets, a predicate logic will be more appropriate as a formal logic for task description. To establish a predicate logic, we can consider each element in V as a predicate representing a ternary relation defined on $S \times O \times A$. Obviously, quantifiers can be incorporated in this formal logic. See Section 6.

4 Inference and Reasoning in the Formal Logic

As in other formal logics, we can infer and reason in our formal logic by adding a deductive apparatus to it. For inference and reasoning, we treat the connectives in the component set C as logic primitives (see, for instance, [35]). Syntactically, a set of inference rules can be constructed, and axioms can also be established. (The hierarchy described in Section 5 represents non-logical, domain-dependent axioms.) Typical introduction- and elimination-rules in formal logics, such as modus ponens and modus tollens, can be easily incorporated in our formal logic. Consequently, we can form a sequent, which consists of a finite (possibly empty) set of well-formed formulas (the premises) and a single well-formed formula (the conclusion), and we can examine its provability (derivability) using proof theory; we can determine if a conclusion follows logically from a set of premises by examining whether there is a proof of that conclusion from just those premises in the formal logic.

As will be described in Section 7, we can employ fuzzy relations to establish the semantics of our formal logic. This semantics allows us to investigate the truth conditions and the semantic validity of each proposition or sequent. As in other formal logics, comparative truth tables can be used to determine semantic validity.

5 Hierarchy of Task Descriptions

Using the scheme described in Sections 2–3, we develop a hierarchy of task descriptions. The importance of the hierarchy is threefold; (i) It enhances the expressive power of our task description scheme by building up its vocabulary while ensuring the interpretability of each resulting proposition; (ii) It enhances the interactivity of our scheme by allowing human-robot interactions to take place at various levels of detail; (iii) It strengthens the deductive apparatus of our formal logic by establishing domain-dependent axioms that can be used for inference and reasoning.

To explain this hierarchy, we first explore it with simple, concrete examples that appeal to ordinary practices and intuitions, and then move on to formal characterizations. A complex task can be described rather concisely, i.e., it can be expressed by an atomic proposition. In many cases, naive users will prefer specifying a complex task using an atomic proposition as opposed to a more lengthy compound proposition. On the other hand, in order for a robotic system to actually execute a complex task, the task must be broken down into simpler subtasks, and the manner in which the subtasks are performed must be specified. Consequently, the atomic proposition describing a complex task can be reexpressed as a compound proposition that consists of atomic propositions describing the required subtasks. Some of the subtasks may have to be further decomposed in

order to fully specify how to execute them. Expert users may want to establish and combine these subtasks carefully so that the robotic system can perform the task effectively and efficiently. This flexibility in the level of detail gives naive users an efficient, user-friendly interface with robots while giving expert users the power to customize tasks. Since our interaction scheme must mediate interactions between robots and a variety of users, it should allow interactions to take place at various levels of detail. We develop a hierarchy of task descriptions to achieve this objective. The hierarchy determines the appropriate level of detail for each human-robot interaction, i.e., it determines whether, from the point of view of a given user, a given task is "atomic" or "compound."

Consider the following task description:

$$\frac{robot}{S}\ \frac{bring}{V}\ \frac{box}{O}\ \frac{from\ room\ 1\ to\ room\ 2}{A}. \tag{7}$$

This atomic proposition can be reexpressed as a compound proposition that consists of three atomic propositions representing the subtasks that must be performed; the task (7) can be defined as

$$\frac{\frac{robot}{S}\ \frac{find}{V}\ \frac{box}{O}\ \frac{in\ room\ 1}{A}}{\text{atomic proposition 1}}$$

$$\frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{collect}{V}\ \frac{box}{O}}{\text{atomic proposition 2}}$$

$$\frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 2}{A}}{\text{atomic proposition 3}}. \tag{8}$$

Atomic propositions 1 and 2 in (8) can also be reexpressed as compound propositions that clarify how they are performed; atomic proposition 1 in (8) can be defined as

$$\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 1}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{search}{V}\ \frac{box}{O}}{\text{atomic proposition}}, \tag{9}$$

and atomic proposition 2 in (8) can be defined as

$$\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ box}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{grasp}{V}\ \frac{box}{O}}{\text{atomic proposition}}. \tag{10}$$

Therefore, using (9)–(10) and atomic proposition 3 in (8), we can reexpress (7) as

$$\frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 1}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{search}{V}\ \frac{box}{O}}{\text{atomic proposition}}$$

$$\frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ box}{A}}{\text{atomic proposition}}\ \frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{grasp}{V}\ \frac{box}{O}}{\text{atomic proposition}}$$

$$\frac{then}{C}\ \frac{\frac{robot}{S}\ \frac{go}{V}\ \frac{to\ room\ 2}{A}}{\text{atomic proposition}}. \tag{11}$$

Figure 1 visualizes the resulting hierarchy, which consists of three levels (levels 0, 1, and 2). For simplicity, each atomic proposition is represented by its verb; for instance, the atomic proposition at the highest level (level 2), "Robot *bring* box from room 1 to room 2," is represented by "bring." The task expressed by the atomic proposition at level 2 is described in more detail at the intermediate level (level 1), where the atomic propositions that involve the verbs "find," "collect," and "go" describe the subtasks that constitute the task. These subtasks are described in more detail at the lowest level (level 0), where they are expressed by the atomic propositions that involve the verbs "go," "search," and "grasp."

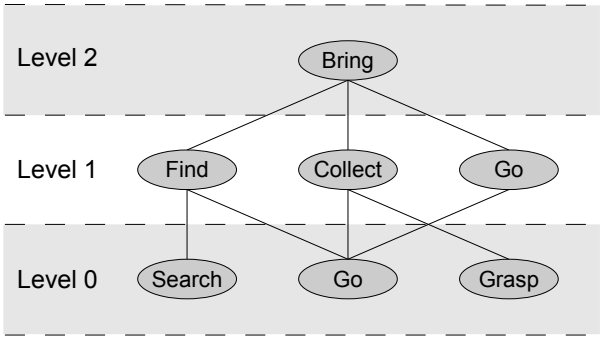


Fig. 1. Hierarchical task description. At the highest level (level 2), the task is expressed as atomic proposition (7), represented by "bring." At the intermediate level (level 1), the task is expressed as compound proposition (8), which consists of atomic propositions represented by "find," "collect," and "go." At the lowest level (level 0), the task is expressed as compound proposition (11), which consists of atomic propositions represented by "go," "search," and "grasp."

The suitability of a given task description depends on the level of granularity required for it. Naive users will most likely prefer describing tasks at level 2, thus preferring (7). For expert users, there may be situations where they prefer specifying a given task step by step or reconfiguring its subtasks according to various circumstances; in such cases, interacting with robots at level 1 using (8) or at level 0 using (11) will be desirable. Thus, the hierarchy allows a variety of users to interact with robots at various levels of detail. At level 0, we have atomic propositions that are not decomposable; each of them is directly associated with a self-contained executable code that is run to perform the corresponding task.

The hierarchy of task descriptions can be characterized more formally as follows. Let S_i, V_i, O_i, A_i, and C_i denote the component sets for level i of the hierarchy ($i \geqslant 0$). The elements in these sets reflect the degree of detail suitable for level i. Then at level i, we establish a formal logic with these component sets, as described in Sections 2–3. Atomic propositions in $S_0 \times V_0 \times O_0 \times A_0$ are the building blocks of all propositional task descriptions; each of them is considered indecomposable, and it is directly associated with a self-contained executable code. For each $i \geqslant 1$, every atomic proposition in $S_i \times V_i \times O_i \times A_i$ can be decomposed into atomic propositions in $S_{i-1} \times V_{i-1} \times$

$O_{i-1} \times A_{i-1}$. Figure 2 visualizes a typical form of this hierarchy. For each i and j, $p_j^{(i)}$ denotes an atomic proposition at level i, and e_j denotes an indecomposable, self-contained executable code associated with $p_j^{(0)}$. In the figure, $p_j^{(0)}$ is connected to e_j for each j, and for each $i \geqslant 1$ and j, $p_j^{(i)}$ is connected to atomic propositions at level $i-1$ that describe the subtasks associated with $p_j^{(i)}$. The figure shows that, for instance, $p_2^{(2)}$ can be expressed as a compound proposition consisting of two atomic propositions ($p_1^{(1)}$ and $p_4^{(1)}$) at level 1 and also as a compound proposition consisting of four atomic propositions ($p_1^{(0)}$, $p_2^{(0)}$, $p_3^{(0)}$, and $p_4^{(0)}$) at level 0.

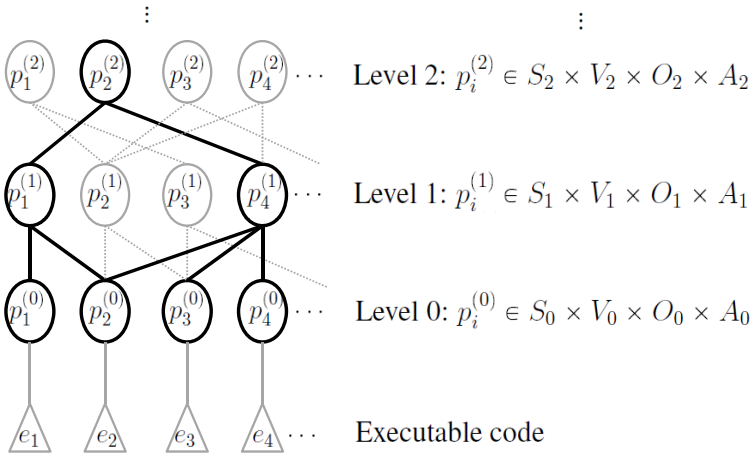


Fig. 2. Hierarchy of task descriptions. At level i, the formal logic described in Sections 2–3 is established with component sets S_i, V_i, O_i, A_i, and C_i. Atomic propositions in $S_0 \times V_0 \times O_0 \times A_0$ are the building blocks of all propositional task descriptions, and each of them is directly associated with an indecomposable, self-contained executable code. For each $i \geqslant 1$, every atomic proposition at level i is connected to atomic propositions at level $i-1$ that describe the subtasks associated with it.

Different levels of granularity may require different component sets, but the same syntactic structure is enforced at all levels. Using the hierarchy, we can ensure that all the resulting task descriptions remain interpretable for both humans and robots at each level, and we can achieve a high degree of efficiency in human-robot interaction by selecting the appropriate level of detail. As regards formal logic, the hierarchy represents non-logical, domain-dependent axioms that can be used for inference and reasoning, and these axioms can be used in examining the provability of a sequent in our formal logic (see Section 4).

6 Generality of Our Framework

Although we considered rather simple component sets and syntactic structures in Sections 2–5 to facilitate the exposition of our formal logic, we can easily extend our

scheme to more sophisticated component sets and syntactic structures so that our formal logic can mediate human-robot interactions for complex robotic systems. Each component set can be made as large as necessary, and other component sets or clause structures can be incorporated in the formal logic. For instance, in addition to the SV, SVO, SVA, and SVOA structures described and used in Sections 2–5, we can also use other commonly observed clause structures (see, for instance, [4]), such as the SVC, SVOC, and SVOO structures, in forming atomic propositions. Furthermore, we can extend the clause structure so that a phrase can be used as the subject or the object in an atomic proposition. Negation, a unary logical connective, can certainly be included in C and incorporated in the formal logic.

As briefly mentioned in Section 3, we can establish not only a propositional logic but also a quantificational logic, which fully incorporates quantifiers and predicates in well-formed formulas. Since propositions that describe perceptions often include quantifiers (see, for instance, [38, 39]), a quantificational logic will be necessary to mediate extensive human-robot interactions regarding perceptions. Human-robot interactions are not limited to task descriptions, and our formal logic can mediate any form of human-robot interaction.

7 Semantics of the Formal Logic: Feasibility Evaluation

By establishing the formal logic for task descriptions, we can also develop a formal and efficient scheme for systematically evaluating the feasibility of each propositional task description. In fact, this scheme establishes the semantics of our formal logic. In formal logic, each proposition is evaluated and assigned a truth value—in two-valued logics, for instance, it is either 1 (true) or 0 (false). In our framework, we evaluate each proposition and let its truth value reflect the feasibility of the corresponding task specification; 1 indicates that the task certainly can be carried out whereas 0 indicates that it certainly cannot be. We maintain that it is more realistic and practical to let the degree of feasibility take on not only the values 0 and 1 but also other values between 0 and 1, so we use a many-valued logic in our formulation. In real-world problems, it can be highly practical to evaluate the feasibility of a task description before any serious attempt is made to execute it. In our case, for instance, we expect our robotic systems to interact with a variety of users, including users who have no knowledge of these systems, and some users may describe tasks that are virtually impossible to accomplish. It is more desirable to disregard such highly infeasible tasks immediately than to waste resources by attempting to realize them. Also, the user may want to be informed of the degree of feasibility of the task that he specifies before the system attempts to perform it. When multiple options are available for performing a specified task, the user may want to compare their degrees of feasibility before determining which option to take.

To evaluate the feasibility of each propositional task description systematically, we use a fuzzy relation, which is a generalization of a classical ("crisp") relation (see, for instance, [22]). It is a mapping from a Cartesian product to the set of real numbers between 0 and 1. While a classical relation only expresses the presence or absence of some form of association between the elements of factors in a Cartesian product, a fuzzy relation can express various degrees or strengths of association between them. (Hence

a classical relation can be considered a "crisp" case of a fuzzy relation.) In our study, each propositional task description consists of pre-specified components, so a function that assigns a degree of feasibility to each proposition can be represented by a fuzzy relation on a Cartesian product of the components. Using some of the operations defined on fuzzy relations, we can systematically and economically determine the degree of feasibility of each task. We will provide details of the semantics in our full-length paper.

8 Discussions

We have taken a first step toward establishing a formal logic for task description that facilitates human-robot interaction. Various syntactic structures in natural language can be incorporated in our formal logic so that the resulting formal language serves as a middle ground between the natural-language-based mode of human communication and the low-level mode of machine communication. The high expressive power of the formal logic is achieved by its syntax, which enables us to create infinitely many task descriptions while ensuring that each of them is unambiguously interpretable. As in other formal logics, we can infer and reason in our formal logic. The hierarchy of propositions enhances the expressive power, the interactivity, and the deductive apparatus of our formal logic. Using the semantics of the formal logic, we can evaluate the feasibility of each task description systematically and efficiently.

We are currently implementing our task description scheme in a multi-agent robotic system. As described in Section 6, our formal logic is not limited to describing tasks. For instance, it can be applied to establishing perceptual propositions that allow artificial intelligence to operate on perceptual information obtained by humans. We are developing a quantificational logic that deals with such propositions.

References

1. Abbott, K.A., Slotte, S.M., Stimson, D.K.: Federal aviation administration human factors team report on: The interfaces between flightcrews and modern flight deck systems (June 1996)
2. Allen, J.F., Byron, D.K., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A.: Toward conversational human-computer interaction. AI Magazine 22, 27–37 (2001)
3. Anderson, M., Conner, C., Daniel, V., McKay, M., Yancey, N.: Demonstration of the robotic gamma locating and isotopic identification device. In: Proceedings of the He American Nuclear Society Spectrum (2002)
4. Biber, D., Conrad, S., Leech, G.: A student grammar of spoken and written English. Pearson ESL (2002)
5. Bruemmer, D.J., Few, D.A., Boring, R.L., Marble, J.L., Walton, M.C., Nielsen, C.W.: Shared understanding for collaborative control. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 35(4), 494–504 (2005)
6. Bruemmer, D.J., Marble, J.L., Dudenhoeffer, D.D., Anderson, M., Mckay, M.D.: Mixed-initiative control for remote characterization of hazardous environments. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (2003)
7. Casper, J., Murphy, R.R.: Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. IEEE Transactions on Systems, Man, and Cybernetics, Part B 33, 367–385 (2003)

8. Dias, M.B., Kannan, B., Browning, B., Jones, E.G., Argall, B., Dias, M.F., Zinck, M., Veloso, M.M., Stentz, A.J.: Evaluation of human-robot interaction awareness in search and rescue. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2327–2332 (2008)
9. Dias, M.B., Kannan, B., Browning, B., Jones, E.G., Argall, B., Dias, M.F., Zinck, M., Veloso, M.M., Stentz, A.J.: Sliding autonomy for peer-to-peer human-robot teams. In: Proceedings of the 10th International Conference on Intelligent Autonomous Systems (2008)
10. Bernardine Dias, M., Harris, T.K., Browning, B., Jones, E.G., Argall, B., Veloso, M.M., Stentz, A., Rudnicky, A.I.: Dynamically formed human-robot teams performing coordinated tasks. In: AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before, pp. 30–38 (2006)
11. Espinosa, J.A., Cadiz, J., Rico-Gutierrez, L., Kraut, R.E., Scherlis, W., Lautenbacher, G.: Coming to the wrong decision quickly: Why awareness tools must be matched with appropriate tasks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 392–399. ACM (2000)
12. Feil-Seifer, D., Mataric, M.J.: Defining socially assistive robotics. In: Proceedings of the International Conference on Rehabilitation Robotics, pp. 465–468 (2008)
13. Ferketic, J., Goldblatt, L., Hodgson, E., Murray, S., Wichowski, R., Bradley, A., Chun, W., Evans, J., Fong, T., Goodrich, M., Steinfeld, A., Stiles, R.: Toward human-robot interface standards: Use of standardization and intelligent subsystems for advancing human-robotic competency in space exploration. In: Proceedings of the SAE 36th International Conference on Environmental Systems (2006)
14. Fong, T., Nourbakhsh, I., Kunz, C., Flückiger, L., Schreiner, J., Ambrose, R., Burridge, R., Simmons, R., Hiatt, L.M., Schultz, A.: The peer-to-peer human-robot interaction project. Space 6750 (2005)
15. Fong, T., Thorpe, C.: Vehicle teleoperation interfaces. Autonomous Robots 11, 9–18 (2001)
16. Forsberg, M.: Why is speech recognition difficult. Chalmers University of Technology (2003)
17. Gieselmann, P., Stenneken, P.: How to talk to robots: Evidence from user studies on human-robot communication. How People Talk to Computers, Robots, and Other Artificial Communication Partners, 68 (2006)
18. Goodrich, M.A., Schultz, A.C.: Human-robot interaction: a survey. Found. Trends Hum.-Comput. Interact. 1, 203–275 (2007)
19. Johnson, M., Feltovich, P.J., Bradshaw, J.M., Bunch, L.: Human-robot coordination through dynamic regulation. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 2159–2164 (2008)
20. Johnson, M., Intlekofer, K.: Coordinated operations in mixed teams of humans and robots. In: Proceedings of the IEEE International Conference on Distributed Human-Machine Systems (2008)
21. Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahsahi, T., Shinjou, A., Shimada, S.: RoboCup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 739–743 (1999)
22. Klir, G.J., Folger, T.A.: Fuzzy sets, uncertainty, and information. Prentice Hall (1988)
23. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Generation Computing 4(1), 67–95 (1986)
24. Kulyukin, V., Gharpure, C., Nicholson, J., Osborne, G.: Robot-assisted wayfinding for the visually impaired in structured indoor environments. Autonomous Robots 21, 29–41 (2006)
25. Lacey, G., Dawson-Howe, K.M.: The application of robotics to a mobility aid for the elderly blind. Robotics and Autonomous Systems 23, 245–252 (1998)
26. Lauria, S., Bugmann, G., Kyriacou, T., Klein, E.: Mobile robot programming using natural language. Robotics and Autonomous Systems 38, 171–181 (2002)

27. Lueth, T.C., Laengle, T., Herzog, G., Stopp, E., Rembold, U.: KANTRA: human-machine interaction for intelligent robots using natural language. In: Proceedings of the IEEE International Workshop on Robotand HumanCommunication, pp. 106–111 (1994)
28. Marble, J., Bruemmer, D., Few, D., Dudenhoeffer, D.: Evaluation of supervisory vs. peer-peer interaction with human-robot teams. In: Proceedings of the Hawaii International Conference on System Sciences (2004)
29. McCarthy, J.: Situations, actions and causal laws. In: Minsky, M. (ed.) Semantic Information Processing, pp. 410–417. MIT press, Cambridge (1968)
30. Norbakhsh, I.R., Sycara, K., Koes, M., Yong, M., Lewis, M., Burion, S.: Human-robot teaming for search and rescue. Pervasive Computing, 72–79 (2005)
31. Reiter, R.: Proving properties of states in the situation calculus. Artificial Intelligence 64, 337–351 (1993)
32. Shim, I., Yoon, J., Yoh, M.: A human robot interactive system "RoJi". International Journal of Control, Automation, and Systems 2, 398–405 (2004)
33. Shneiderman, B.: The limits of speech recognition. Communications of the ACM 43(9), 63–65 (2000)
34. Thielscher, M.: Introduction to the fluent calculus. Electronic Transactions on Artificial Intelligence 3 (1998)
35. Tomassi, P.: Logic. Routledge (1999)
36. Wilcox, B., Nguyen, T.: Sojourner on mars and lessons learned for future planetary rovers. In: Proceedings of the SAE International Conference on Environmental Systems (1998)
37. Winograd, T., Flores, F.: Understanding computers and cognition: A new foundation for design. Ablex Pub. (1986)
38. Zadeh, L.A.: Some reflections on information granulation and its centrality in granular computing, computing with words, the computational theory of perceptions and precisiated natural language. Studies in Fuzziness and Soft Computing 95, 3–22 (2002)
39. Zadeh, L.A.: Precisiated natural language (PNL). AI Magazine 25(3), 74–92 (2004)
40. Zadeh, L.A.: A new direction in ai: Toward a computational theory of perceptions. AI Magazine 22(1), 73 (2001)

Internet-Based Supervisory Teleoperation of a Virtual Humanoid Robot

Francisco Suárez-Ruiz, Alexander Owen-Hill, and Manuel Ferre

Centre for Automation and Robotics (UPM-CSIC)
Universidad Politécnica de Madrid, 28006 Madrid, Spain
{fa.suarez,alex.owenhill,m.ferre}@upm.es
http://www.car.upm-csic.es/

Abstract. This paper describes the approach used by the Sarbot-Team[1] for controlling the Atlas humanoid robot during the DARPA Virtual Robotics Challenge that took place in June 2013. Herein we present a proposal for overcoming the restrictions on performance caused by limited bandwidth, high latency and the effects of signal degradation induced by beyond line of sight (BLOS) conditions, RF interference, and other related circumstances.

Experimental evaluation confirmed the effectiveness of our approach and present an alternative for coping with constrained communication conditions during the control of humanoid robot deployed at unattended areas.

Keywords: Teleoperation, internet-based control, humanoid robot, robot simulation.

1 Introduction

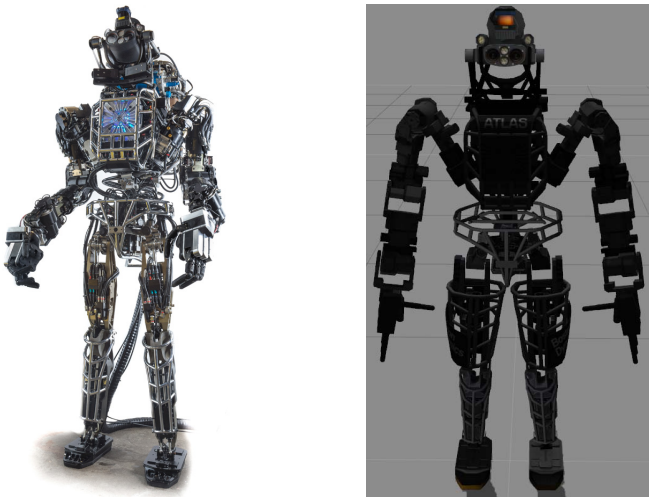
The DARPA Robotics Challenge (DRC)[2] program plans to conduct three competitions:

- The Virtual Robotics Challenge (VRC) in June 2013
- The DARPA Robotics Challenge Trials (DRC Trials) in December 2013
- The DARPA Robotics Challenge Finals (DRC Finals) in December 2014

The primary technical goal of the DRC is to develop ground robots capable of executing complex tasks in dangerous, degraded, human-engineered environments. Competitors in the DRC are expected to focus on robots that can use standard tools and equipment commonly available in human environments, ranging from hand tools to vehicles, with an emphasis on adaptability to tools with diverse specification.

[1] http://www.sarbot-team.es/
[2] http://www.theroboticschallenge.org/

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_26,



(a) Real robot (b) Simulation in gazebo

Fig. 1. The Atlas humanoid robot (Boston Dynamics)

1.1 The Atlas Robot

Figure 1 shows the Atlas robot which is one of the seven robots used for the DARPA Robotics challenge. Atlas is a bipedal human sized robot, with a height of 2.25 m. and a mass of 150 Kg.

It has been designed for rough terrains and is capable of coordinate actions such as climbing. For that it can use hands and feet to go through difficult spaces.

Atlas has two hands, arms, legs feet and a torso, 28 hydraulically-actuated degrees of freedom (6 at each extremity, 3 at the torso and 1 at the neck).

It has stereo cameras and a laser range finder at the head that allows 3D perception for complex tasks like navigation and manipulation.

1.2 The DRC Simulator

Gazebo [11] and ROS [16] are the basis for the DRC Simulator. The DRC Simulator[3] (DRCSim) comprises additional worlds, models, and code that is specifically required to simulate the DRC domain.

The VRC competition was comprised of three tasks that are representative of the challenges planned for the DRC Trials and DRC Finals, these are:

1. Walk a short distance to and climb into a utility vehicle, drive along a roadway at no greater than 16 Km/h, climb out of the utility vehicle, and walk to the finish area.

[3] `http://gazebosim.org/wiki/DRC/`

2. Walk over varied terrain. To complete the course, the robot exits the starting pen, walks across flat pavement, crosses a mud pit, crosses uneven hills, and traverses a rubble pile.
3. Connect and operate a hose. The robot begins in a starting pen, walks to a table on which the hose is located. The task is to pick up the hose so that the hose connector lifts above the table surface, mate the hose connector to the standpipe so that the two are aligned, attach the hose to the standpipe by rotating the hose connector so that it threads onto the standpipe (approximately 2.5 revolutions) and open the valve by rotating it one revolution (360 degrees).

In this paper we present an internet-based supervisory teleoperation approach and the experimental results on the remote control between Madrid, Spain and Dallas, TX, USA.

This paper is organized as follows. In Section 2, we introduce the related work in the field of humanoids robots navigation and manipulation. In Section 3 an overview of the system is given and the computing environment and communications parameters are described. In Section 4 the proposed method is depicted and the implementation and the resulting user interfaces are presented. In Section 5 the experimental results from the VRC competition between Madrid, Spain and Dallas, TX, USA are provided to show the relevance of the proposed internet-based supervisory teleoperation. Concluding remarks follow in Section 6.

2 Related Work

Recently, humanoid robots have shown great capabilities performing complex task associated with the human-based design of facilities, tools, etc. [4,10,2].

Controlling such as complex humanoid robots (e.g the atlas robot) to autonomously perform difficult tasks in unknown and unstructured environments presents various unique challenges.

Despite these challenges, performing intricate tasks under hazardous environmental conditions is a promising area for the use of humanoids due to safety concerns for humans.

The Fukushima Daiichi Tsunami event in 2011 presented a scenario where quick human response was extremely difficult. The disaster magnitude and the radiation contamination [18] seem to be a situation where humanoid robots technology can act in safe and timely manner.

Several works had addressed challenges in different aspects of the total problem that represents task-oriented whole body motion for humanoid robots [6]. Dang *et al.* have presented a simulation framework for humanoid grasping and object manipulation [5].

Robot navigation (autonomous locomotion) is a very mature area of research. Multiple approaches have been proposed to provide mapping and navigation capabilities to mobile robots [14,19,12,13]. Collision avoidance into trajectory optimization as a method of solving robotic motion planning problems was recently addressed by Schulman and Lee [17]. Waard *et al.* have analyzed the capabilities of the Atlas robot for locomotion over flat terrain [20].

Concerning internet-based remote control of humanoid robots, experimental results have been gathered during operation of a personal robot between Daejeon, South Korea and Davis, CA, USA [8]. An extensive survey of human-robot interaction can be found in [7]

This paper focuses on supervisory control, since the latency and bandwidth limitations dismiss the possibility of using bilateral teleoperation which requires high bandwidth to ensure stability in the haptic feedback loops, typically running at 1 kHz [3,9].

3 System Description

Figure 2 shows a block diagram of the computing environment for the Virtual Robotics Challenge. Each run during the competition uses 4 cloud computers: 1 running the simulator, 2 running Field Code, and 1 shaping traffic [1].

The DRC Simulator, running on 1 computer in the cloud (Dallas, TX, USA), simulated the robot and the environment. During the VRC competition, a separate instance of the DRC Simulator were run on a separate, dedicated machine for each team.

Field Code runs on the two Field Computers, Field1 and Field2. These are two computers in the cloud, serving as surrogates for the computer(s) that will be tethered to the Atlas robot in the DRC Trials. During the VRC our team ran perception, planning, and control code on the Field1 and Field2. This code handled processes requiring high-bandwidth communications with the robot. This approach reduced the data required to flow across the internet link. OCU (Operator Control Unit) Code ran on one computer at the team facility (Madrid, Spain). This computer system served as the operator control station that allowed command and control of the robot. The human machine interface ran here (see section 4.2). The machine containing OCU Code were on the other side of the data-metered link from the Field Code.

The computer running the DRC Simulator and the field computer 1 were equipped with two (2) Intel Xeon E5- 2690 eight-core chips, 16 GB RAM, 1 TB HD, and a NVIDIA TESLA M2090 6GB GPU, running on a 64-bit architecture. The field computer 2 was equipped with two (2) Intel Xeon E5-2690 eight-core chips, 32 GB of memory, and 128GB SSD, running on a 64-bit architecture.

3.1 Communications Parameters

To simulate the effects of signal degradation induced by beyond line of sight (BLOS) conditions, RF interference, or other related circumstances, the communications parameters were varied by the OSRF-controlled VPN [1] (see VPN in Fig. 2). Each run began with the following conditions of time, latency, uplink bits limit and downlink bits limit:

- Time: The time for each run shall be 30 minutes.
- Latency: To achieve a target latency value of 500 ms, the router induces latency on both the incoming and outgoing links as shown in Fig. 3.

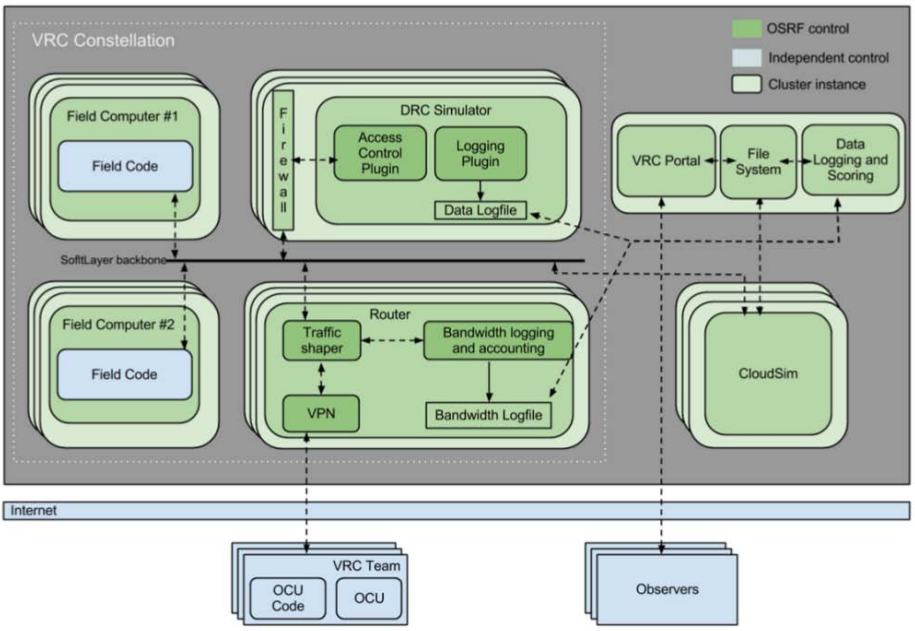


Fig. 2. VRC constellation computing environment block diagram [1]. Each block represents a computer. Thick black line represents a high speed communication bus. Dashed lines represent data flow.

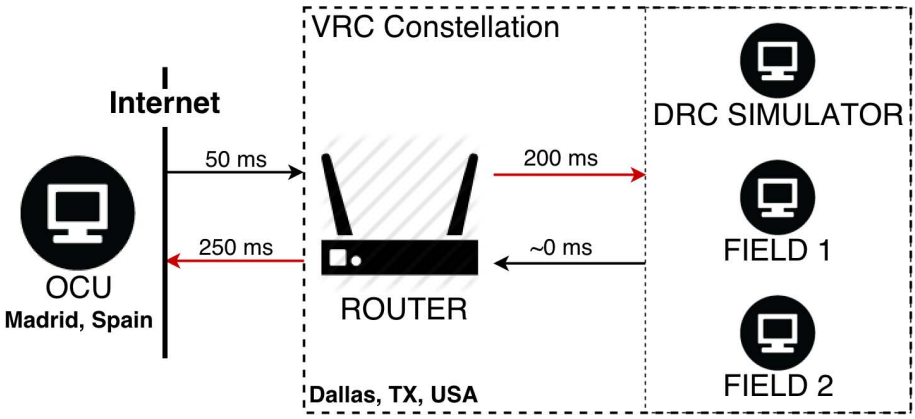


Fig. 3. Latency with a target value of 500 ms divided evenly between inbound and outbound network links

– Uplink bits: Table 1 shows the assumed average uplink bandwidths, and the corresponding number of uplink bits and bytes allotted. Note that Table 1 lists the cases in descending order from least challenging to most challenging. Each case differs from neighboring cases by a factor of four.

Table 1. Number of uplink data allotted per run

Run	Uplink		
	Bandwidth (bps)	Bits allotted	Bytes allotted
1	16,384	29,491,200	3,686,400
2	4,096	7,372,800	921,600
3	1,024	1,843,200	230,400
4	256	460,800	57,600
5	64	115,200	14,400

– Downlink bits: Table 2 shows the assumed average downlink bandwidths, and the corresponding number of downlink bits and bytes allotted. Note that Table 2 lists the cases in descending order from least challenging to most challenging. Each case differs from neighboring cases by a factor of two.

Table 2. Number of downlink data allotted per run

Run	Downlink		
	Bandwidth (bps)	Bits allotted	Bytes allotted
1	524,288	943,718,400	117,964,800
2	262,144	471,859,200	58,982,400
3	131,072	235,929,600	29,491,200
4	65,536	117,964,800	14,745,600
5	32,768	58,982,400	7,372,800

4 Method

Some choices were made to enable fast implementation of the supervisory control and to make the most efficient use of the constrained communication parameters.

Our approach is based in connecting two roscores. One running in the VRC constellation and the other running in the local machine. This allows us to use existing ROS tools for visualization, `rviz`[4] in particular.

Using `rviz` we can visualize the robot state, the occupancy map and the poses of objects of interest amount other data of interest for our application (see section 4.2).

[4] `http://ros.org/wiki/rviz`

For connecting the two roscores the `multimaster`[5] package can be used, but due to its problems handling connection lost We decided to use sockets implemented with the ZeroMQ[6] library (`python-zmq`) as shown in Fig. 4.

Some of the main features [15] why we choose this library are:

- Performance: It is orders of magnitude faster than most AMQP messaging systems.
- Simplicity: The API is plausibly simple which makes sending messages easy compared with raw socket implementation. The integration with python allows serialization of objects straight forward which for our application is crucial.
- Scalability: ZeroMQ follows a broke-less design so that there is no single point of failure.
- Transport variety: Supports a range of transport mechanisms for messages including UDP and multicast (INPROC, IPC, MULTICAST, TCP).

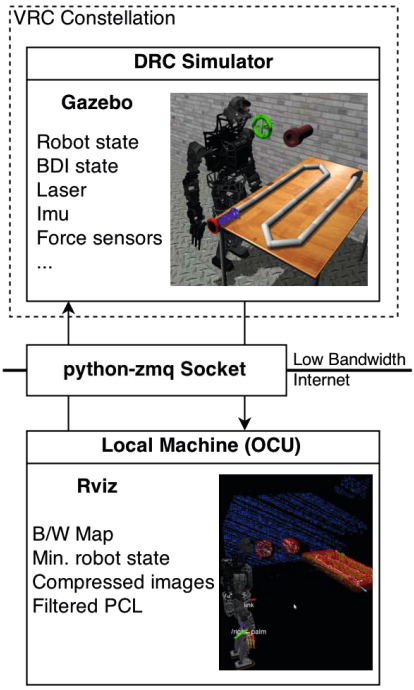


Fig. 4. Socket implementation connecting two roscores

[5] `http://ros.org/wiki/multimaster`
[6] `http://www.zeromq.org/`

4.1 Implementation

To tackle the execution of the three VRC tasks we propose a supervisory teleoperation approach. The main idea of such approach is sending high level commands that the robot will perform.

In the local machine, the goal is to have enough feedback so that the operator can verify the correct execution of the commands.

Both conditions have to be met while staying under the uplink (Table 1) and downlink (Table 2) limits.

This approach has been implemented taking advantage of the ROS [16] communication layer, specially the *topics* published by the DRCSim. Table 3 summarizes the messages required for our application and its maximum size in bytes.

Table 3. Topics received in the OCU for the supervisory teleoperation

Name	Type	Tag	Size (bytes)
DRC Simulator Published Topics			
/atlas/atlas_state	atlas_msgs/AtlasState	S_1	1,160
/atlas/atlas_sim_interface_state	atlas_msgs/AtlasSimInterfaceState	S_2	905
/multisense_sl/camera/left	sensor_msgs/CompressedImage	S_3	4,795
/multisense_sl/camera/right	sensor_msgs/CompressedImage	S_4	4,795
Custom Published Topics			
/sarbot/occupancy_grid	nav_msgs/OccupancyGrid	C_1	3,482
/sarbot/filtered_pcl	sensor_msgs/PointCloud2	C_2	3,925
/sarbot/hose_pose	geometry_msgs/PoseStamped	C_3	240
/sarbot/standpipe_pose	geometry_msgs/PoseStamped	C_4	240
/sarbot/valve_pose	geometry_msgs/PoseStamped	C_5	240
/sarbot/robot_position	geometry_msgs/PoseStamped	C_6	240
	Total		**20,022**

4.2 User Interface

Server. Figure 5 shows the interface used during the robot teleoperation in the server side running in the field computer 1. The simplicity of this interface is due to the datalink limitations mentioned above.

During the VRC competition the fields Remaining Downlink and Uplink show the operator the remaining bytes in each case. This additional information can help the operator to decided to stop or start the sending of the data (by pressing the D key).

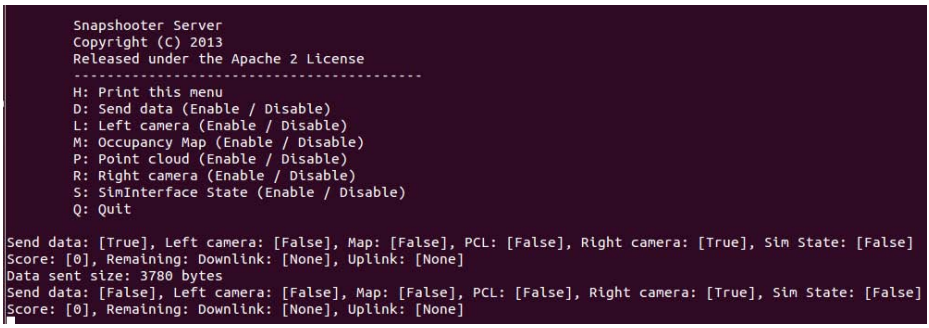


Fig. 5. Server side interface

In addition, before sending the data, the application calculates the packet size, p as shown in Equation (1).

$$p = \sum_{i=1}^{a} \alpha_i S_i + \sum_{i=1}^{b} \beta_i C_i \tag{1}$$

Where S_i and C_i are the data sizes shown in Table 3 and $\alpha_i \wedge \beta_i \in \{0, 1\}$ are set through the user interface shown in Fig. 5. This information is displayed in the console.

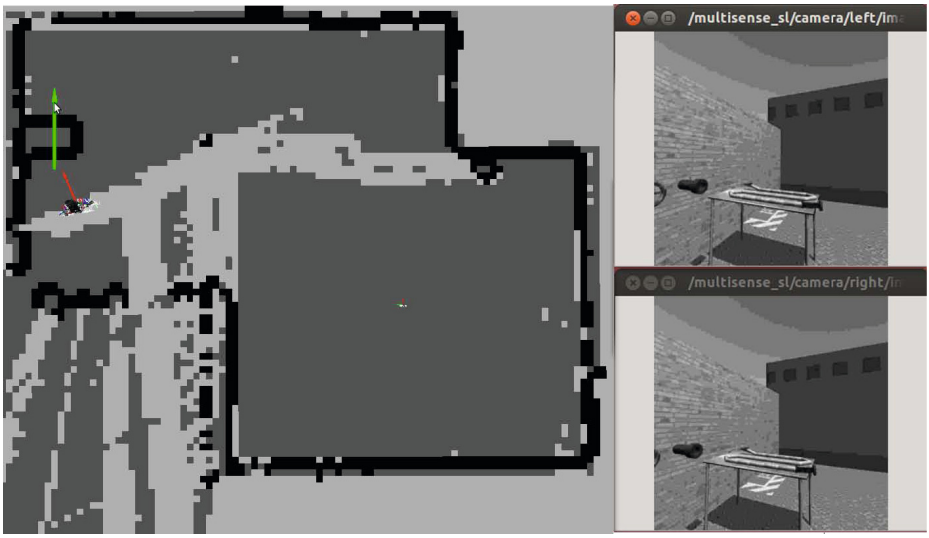


Fig. 6. Client side interface running in the OCU during navigation

Navigation. Figure 6 shows the user interface used during the navigation stage of teleoperation.

The operator controls the motion of the robot by selecting a goal pose (position + orientation) using the mouse, see the green arrow in Fig. 6. The robot generates automatically a trajectory and moves towards the goal while the operator has visual feedback of the robot position with respect to the map.

The map is built with a grid size of 20 cm., it's the resolution for assigning probabilities of occupancy in the areas that the robot can see. This resolution was chosen as a trade off between precision and speed in the mapping and navigation. The probability of occupancy for each square is given as a value between 0 and 100%.

The color convention for the map is as follow:

- **Black**: Indicates the **occupied** areas.
- **Gray**: Indicates the **unknown** areas.
- **White**: Indicates the **unoccupied** areas.

The operator can only select goals in the unoccupied and unknown areas (white and gray), and that in the unknown areas, it's the operator who deals with the uncertainty.

Manipulation. Figure 7 shows the user interface used during the manipulation tasks. The manipulation interface is intended only for visualization proposes and for particular use during the VRC task 3.

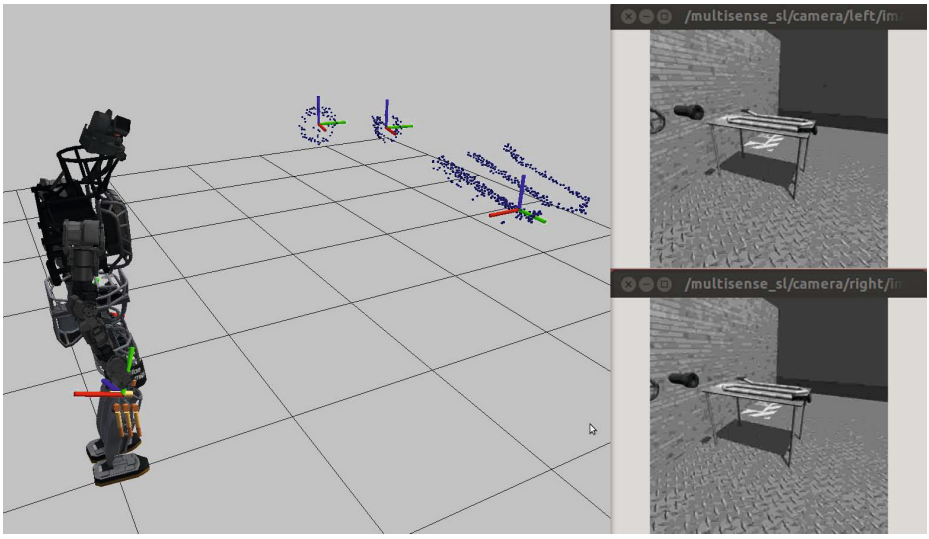


Fig. 7. Client side interface running in the OCU during manipulation

As shown, the operator can have visual information about the perception pipeline for manipulation. Figure 7 shows the robot state, the pose of the robot hands, the pose of the objects of interest (hose connector, standpipe and valve), right and left camera images jpeg compressed with a 5% ratio and the filtered point cloud (PCL) assembled from the readings of the head mounted laser.

5 Results

To evaluate the proposed method, We present the results obtained performing 15 trials during the VRC. This trials were divided in five (5) runs for each of the three (3) tasks described in Section 1.2. For each tas, the most challenging conditions are presented at the run 5, this is due to the limited uplink bytes allotted (14 kB, see Section 3.1).

5.1 Uplink

Figure 8 shows the percentage of uplink data consumed for the 15 trials. As can be seen, the uplink data used for runs 1 to 3 was under the 2%, indicating a very low consumption of total data allotted.

For the three tasks, in run 4, the percentage of data consumption had important values, task 1 and task 2 were under the 8% but for the task 3 that value goes up until 20.5%.

In the case of the run 5, is were the importance of the proposed approach can be seen. Out of the 14 kB. available, during task 1 (44.4% used) and task 3 (55.8% used) we banked bytes. Only in the task 2 the total allotted bytes were consumed.

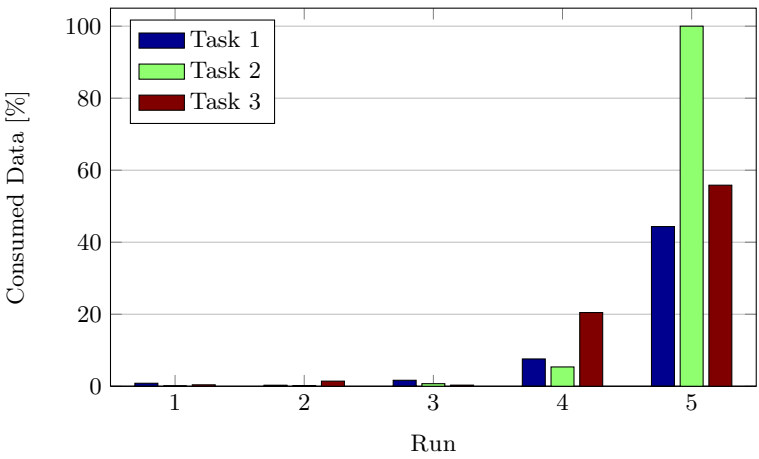


Fig. 8. Percentage of uplink data consumed in different 15 trials

5.2 Downlink

Figure 9 shows the percentage of downlink data consumed for the 15 trials. In this case, due to a high amount of bytes available and the proposed approach the data consumed was always under safe limits. For all the task in runs 1 to 3, the downlink data required was under the 7% of total allotted.

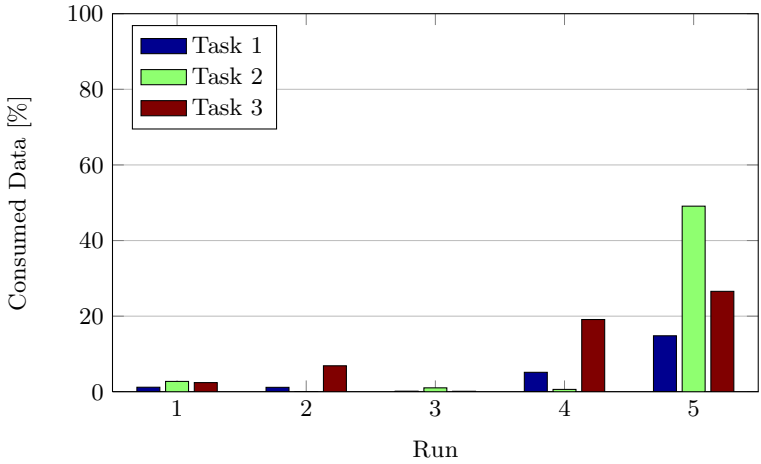


Fig. 9. Percentage of downlink data consumed in different 15 trials

It's in the run 5, where the total bytes were ∼7 MB (see Table 2, the data used reached 49.1% for the task 2, 26.6% for the task 3 and only 14.8% for the task 1.

This results can be assigned to the fact that the images where compressed using a very low ratio (5% compression ratio).

6 Conclusions and Future Work

Allowing the operator to have control over the data rate can reduce the required bytes for different type of tasks while teleoperating a humanoid robot in supervisory mode. From the results obtained during the VRC, we can come to the conclusion that additional work need to be done to improve tools that offer similar solutions to the problem that represents connecting two roscores under limited communication conditions.

The use of ROS communication topology (connecting all the systems to the same roscore) is adaptable but has communication overhead and was not suitable for the VRC conditions. Using the existing package, `multimaster`, neither was appropriate due its problems under data loss conditions and not stable link connections.

The proposed method allowed us to have the required feedback from the robot sensors (DRC Simulator) without exceeding the downlink and uplink limitations imposed for the challenge. This has proven the possibilities for deploying humanoid robots in unknown and unstructured environments using and internet-based supervisory teleoperation and has confirmed the effectiveness of our approach.

Concerning the image transport between remote locations, as mentioned in Section 4.2, using jpeg compression was enough for fulfilling the requirements from the operator and the communication constraints. Nevertheless, the use of recent compression techniques like VP8 or theora can reduce even more the data required for video transmission.

Further work may include investigating alternatives to extend the proposed method as a generic tool for connecting two roscores. Such tool can be an alternative to the `multimaster` package.

Acknowledgement. This work has been funded by the "Ministerio de Economía y Competitividad" under grant DPI2009-12283 (TEMAR project). The authors would like to thank all the members of the SARBOT[7] (Search And Rescue roBOT) team for their efforts and help during the Virtual Robotics Challenge. SARBOT is a multidisciplinary team formed by Centre for Automation and Robotics (UPM-CSIC), Universidad Politécnica de Madrid, Universidad Carlos III and Universidad de Alcalá.

References

1. Aguero, C., Gerkey, B., Krotkov, E.: Technical Guide, Virtual Robotics Challenge, DISTAR Case 21251, Release 2 (2013)
2. Asfour, T., Yokoi, K., Lee, C.S.G., Kuffner, J.: Humanoid Robotics [TC Spotlight]. IEEE Robot. Automat. Mag. 19(1), 108–118 (2012)
3. Brady, K.: Internet-based remote teleoperation. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 65–70. IEEE (1998)
4. Calinon, S., Guenter, F., Billard, A.: On Learning, Representing, and Generalizing a Task in a Humanoid Robot. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 37(2), 286–298 (2007)
5. Dang, H., Jun, Y., Oh, P., Allen, P.K.: Planning Complex Physical Tasks for Disaster Response with a Humanoid Robot. In: IEEE International Conference on Technologies for Practical Robot Applications, TePRA (2013)
6. Gienger, M., Janssen, H., Goerick, C.: Task-oriented whole body motion for humanoid robots. In: 5th IEEE-RAS International Conference on Humanoid Robots, 2005, pp. 238–244. IEEE (2005)
7. Goodrich, M.A., Schultz, A.C.: Human-Robot Interaction: A Survey. Foundations and Trends in Human-Computer Interaction 1(3), 203–275 (2007)
8. Han, K.H., Kim, Y.J., Kim, J.H., Hsia, S.: Internet control of personal robot between KAIST and UC Davis. In: IEEE International Conference on Robotics and Automation, vol. 2, pp. 2184–2189. IEEE (2002)

[7] `http://www.sarbot-team.es/`

9. Hokayem, P.F., Spong, M.W.: Bilateral teleoperation: An historical survey. Automatica 42(12), 2035–2057 (2006)
10. Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., Akachi, K.: Humanoid robot HRP-3. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2471–2478. IEEE (September 2008)
11. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp. 2149–2154. IEEE (2004)
12. Konolige, K., Agrawal, M.: FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. IEEE Transactions on Robotics 24(5), 1066–1077 (2008)
13. Konolige, K., Agrawal, M., Bolles, R., Cowan, C., Fischler, M., Gerkey, B.: Outdoor Mapping and Navigation Using Stereo Vision. In: Khatib, O., Kumar, V., Rus, D. (eds.) Experimental Robotics, Springer Tracts in Advanced Robotics, vol. 39, pp. 179–190. Springer, Heidelberg (2008)
14. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: AAAI 2002 Proceedings, pp. 593–598 (2002)
15. Piël, N.: ZeroMQ an introduction (2013), `http://nichol.as/zeromq-an-introduction`
16. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source Robot Operating System. In: ICRA Workshop on Open Source Software (2009)
17. Schulman, J., Lee, A.: Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization. In: Draft Submitted at Robotics: Science and Systems 2013, Berlin, Germany (2013)
18. Suárez-Ruiz, F., Ferre, M., Aracil, R.: Telerobotics for Nuclear Fusion Environment. In: III Workshop de Robótica: Robótica Experimental, Sevilla, Spain, pp. 595–600 (2011)
19. Thrun, S.: The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. The International Journal of Robotics Research 25(5-6), 403–429 (2006)
20. de Waard, M., Inja, M., Visser, A.: Analysis of flat terrain for the Atlas robot. In: Proceedings of the RoboCup IranOpen 2013 Symposium, RIOS 2013 (2013)

3-Dimensional Object Perception for Manipulation Tasks Using the Atlas Robot

Silvia Rodríguez-Jiménez and Mohamed Abderrahim

Department of Systems Engineering and Automation,
Carlos III University of Madrid, Leganés, Spain
{srjimene,mohamed}@ing.uc3m.es

Abstract. The goal of the DARPA Robotics Challenge (DRC) is the development of ground robots capable of executing complex tasks in disaster relief environments. The Virtual Robotics Challenge (VRC) was the first phase of the competition, in which teams develop their own software to maneuver a simulated robot through a virtual obstacle course and perform a set of complex tasks. In these scenarios, grasping and manipulation tasks are required; therefore this paper provides a description of the visual 3-Dimensional (3D) perception functionality to grasp a hose lying on a table, one of the tasks of the challenge. The sensor head of the used Atlas robot is equipped with a laser range scanner and a stereo camera, which provide the 3D point cloud to be processed. The presented approach processes the point cloud; a plane detector is used to segment the scene and a tabletop assumption is then considered to detect the objects of interest. The hose connector is recognized using color-based region growing segmentation and then its pose is estimated by a cylinder model fitting. The experimental results demonstrate the efficiency of the proposed 3D perception pipeline in the scenario under study.

1 Introduction

Natural or man-made disasters cause grave risks to the health of human and endanger their life from the moment they occur. The goal of the DARPA Robotics Challenge (DRC) [1] is to develop advanced robots capable of assisting humans in performing dangerous tasks and manipulation of objects and debris in disaster scenarios. These robots require multiple layers of software and tools to interact with the human environments and perform the most hazardous activities in disaster response operations. The Virtual Robotics Challenge (VRC) was the first stage of this competition. In the VRC, teams tested developed software in locomotion, perception and manipulation within a virtual environment, to complete a series of tasks that are prerequisites for more complex activities. The robot provided to this challenge is Atlas (Fig. 1a and Fig. 1b), developed by Boston Dynamics, Inc., and based on its Petman humanoid robot platform. The robot has 28 degrees of freedom (DoF), Sandia hands, and an articulated sensor head equipped with a stereo camera and a laser range finder (LRF). The robot sensor head is MultiSense-SL[2] (Fig. 2) provided by a company called Carnegie Robots. Its characteristics from the datasheet are the following:

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_27,

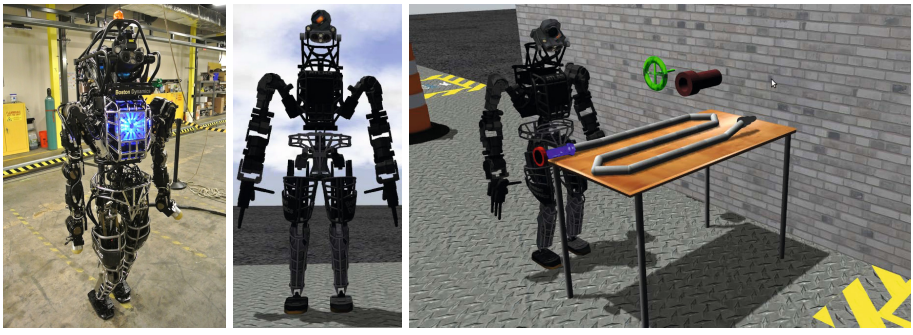
- Stereo camera details: range from 0.4 m to 18 m. Depth resolution: ± 0.31 mm@1 meter; ± 30.00 mm@10 meter.
- Laser details: range from 0.1 m to 30 m. Accuracy: ± 30 mm@< 10 m; ± 50 mm@>10 m.

The Atlas robot and scenarios are simulated in the DARPA Robotics Challenge Simulator (DRCSim), an open-source platform building upon on Gazebo[1] environment under Robot Operating System (ROS)[2] [3]. Software simulation of the sensor head is available within the DRCSim.

The software competition consisted in three challenges, which included driving a utility vehicle, walking across a series of terrain and manipulation tasks. Our work was performed as part of the third VRC task, in which the robot has to grasp a hose lying on a table, attach the hose connector to a standpipe and turn a valve (Fig. 1c). This paper focuses on the 3-Dimensional (3D) perception approach developed to interpret the scenario with the available sensors and it enables the robot to grasp the hose and perform such complex task.

In order to achieve robotic manipulation in complex scenarios, 3D object perception system is required for modeling the surrounding environment and the objects. Systems for performing everyday mobile manipulation in cluttered environments are popular and form a growing area of interest in robotics research.

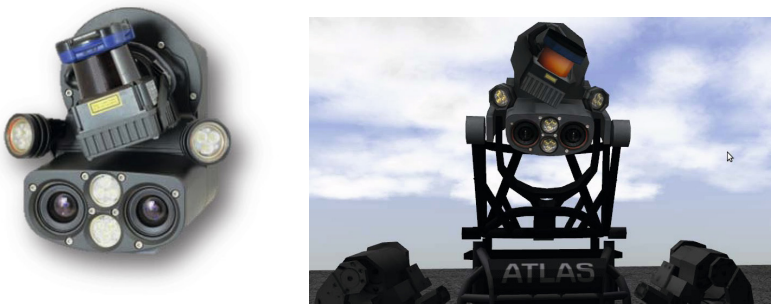
For this reason, they are currently being developed by many research groups [4,5,6,7,8]. In the majority of these approaches to object perception, a natural restriction of space is exploited: objects are usually constrained to tabletops, horizontal or vertical planes.



(a) (b) (c)

Fig. 1. Atlas robot. (a) Real robot (Courtesy Boston Dynamics); (b) Atlas URDF model in Gazebo; (c) Screenshot in Gazebo of the third task of the Virtual Robotics Challenge.

[1] http://gazebosim.org/
[2] http://www.ros.org/wiki/



(a) (b)

Fig. 2. Robot sensor head of the Atlas. (a) Real head; (b) Virtual head in Gazebo.

In the third VRC task (Fig. 1c), the hose lies on a table, the standpipe and the valve are placed on a wall. Therefore, the natural restriction of location in human environments could be applied and the analysis can be based on a common perception system scheme with 3D point cloud [9,10]:

1. Detection of the horizontal support planes, e. g., a table.
2. Extraction and clustering of the measurements on top of these planes to obtain segmentation into objects.
3. Further processing such as, for instance, recognizing, classifying or tracking the clusters of interest.

Following this scheme, in Section 2 our proposed visual perception system is introduced. Results are presented in Section 3 and the discussion on these results and future work are elaborated in Section 4.

2 3D Object Perception

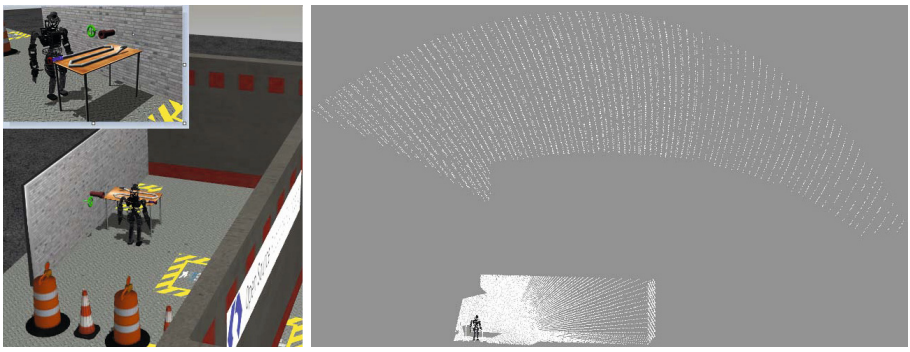
As described in the VRC Rules [11], the task three is decomposed into four subtasks: grasping and lifting up the hose, transporting it to the standpipe opening, connecting the hose to the standpipe, and opening the valve. To complete a task autonomously, the humanoid robot has to understand its surrounding environment. Thus, the first goal to achieve is to identify and localize relevant workspace objects, and in particular the hose and its connector. The hose is modeled with up to ten rigid segments. The weight and the stiffness of the hose, the thread size, and the hose connector handle size are different between competitive runs, which mean that they are parameters which are not known a priori. As initial work, in this paper we focus on the 3D perception system for the hose connector detection and its pose estimation, providing a collision map to ensure that the grasping motion is not in collision. The grasping itself is beyond the scope of the material of this paper, and here the techniques employed for these perception requirements are explained.

Fig. 3a shows the workspace of the world: the hose lies on a table, the connector appears in blue; the standpipe appears in red and the valve in green color. Although the color between the connector and the rest of the hose will be always different, the object color could differ between runs. Therefore, the completed models of the objects in the world are not known during the competition.

Our approach to object detection is based on processing 3D point clouds from laser and stereo camera, using the Point Cloud Library (PCL) [12] in the ROS environment coupled to the DRCSim. To obtain a larger 3D Cartesian coordinate (XYZ) point cloud from a rotary laser range scanner, the articulated stream of scans are assembled as is shown in Fig. 3b.

The laser scans contain all points returned from the scanner. For manipulation task, only points from close-range scene are necessary, therefore, the scans are filtered removing unnecessary distant points (Fig. 4a). The point cloud from stereo camera (Fig. 4b) and filtered laser scans (Fig. 4c) are calibrated, in order to minimize the shift between them in the overlapped areas (Fig. 4d). The laser scanner provides a much wider field of view, whereas the point cloud is sparser in some areas. However, the stereo camera barely provides data of the table as is shown in Fig. 4b.

Therefore, since the adopted perception software is comprised of several steps, the approach exploits the capabilities of each sensor in each one. The filtered laser scan points are used to estimate the wall and table which is then employed to filter out the RGB-Depth (RGB-D) point cloud generated by the stereo camera.

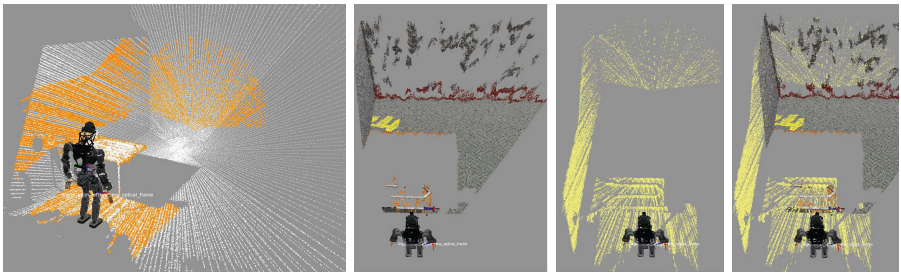


(a) (b)

Fig. 3. The workspace of the third VRC task. (a) Screenshot of Gazebo world from different points of view; (b) Point cloud from the rotary laser range scanner for 10 seconds, displayed in RViz (3D visualization tool for ROS).

2.1 Overview of the Perception Approach

To achieve the manipulation autonomously, the robot should be able to identify and localize relevant workspace objects for performing top-grasps along object's



(a) (b) (c) (d)

Fig. 4. Visualization of 3D point clouds provided by the laser range scanner and stereo camera, displayed in RViz. (a) Filtered point cloud is shown in orange color, and the complete laser scans in white; (b) RGB-D point cloud from stereo camera; (c) filtered point cloud from rotary laser range scanner; (d) comparison between point cloud from stereo camera and laser scanner.

principal axis. The proposed perception process is consists of several steps: segmentation, detection, recognition and localization.

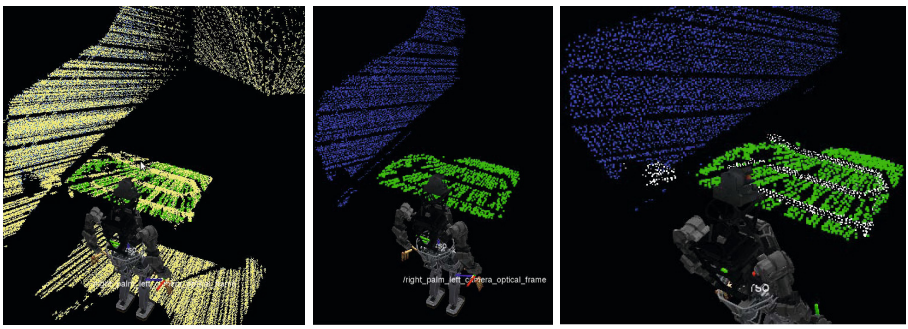
1. Segmentation: The sensor head is positioned to survey the workspace by neck actuators. The 3D point cloud from the filtered assembled laser scans is analyzed for plane segmentation using RANdom Sampling Consensus (RANSAC) [13], finding points within a point cloud that support a 3D plane model. The table is considered as the dominant horizontal plane over the ground and the wall is the vertical plane closest to the table. The table and wall are added to the collision map of the environment.
2. Detection: To detect the hose, a table top object detector similar to [14] is run on the depth data; points lying outside of a box around the table plane are eliminated. Remaining points are clustered into individual objects using Euclidean distances with certain thresholds. A 3D minimum bounding box rectangular to enclose all the clusters is determined by its centroid and simple boolean minimum or maximum operations on the XYZ coordinates of all its points. These bounding boxes contain object candidates to be the hose.
3. Recognition: The hose is formed by rigid segments and the hose connector. Although their properties could be different during the competition, the color between the connector and the rest of the hose will be always different. The adopted approach gives us the possibility of recognizing the hose connector in RGB-D point cloud from the stereo camera using color-based region growing segmentation [15]. The input to the algorithm is the RGB-D point cloud which is inside the bounding boxes lying on the table plane. The region growing process uses k-d tree to search the nearest neighbors of each point. The regions with a small difference between average colors are then merged. The hose connector is the region which is located between two areas of different color (the rigid elements and the end of the hose).

4. Localization: once the hose connector has been detected, its 3D coordinates is calculated with respect to the robot's global coordinate frame. The 3D shape of the hose could be approximate by a cylinder, thus its position and its axis are calculated by fitting a cylinder model to the point cloud that belongs to the hose connector.

3 Results

In this work, the following versions have been used under Ubuntu 12.04 (64 bits): ROS fuerte, DRCSim 2.6, Gazebo 1.8 and PCL 1.7. The proposed algorithms have been encapsulated in several ROS nodes which have been tested in the environment provided for VRC Qualification Task 3. The terrain and the objects that constitute the environment are available as SDF world files. Thus, for testing, in the beginning the robot is initialized to a predefined pose that is adapted to allow an advantageous reach of the workspace as is shown in Fig. 3.

The neck joint is moved to direct the head towards the workspace and the laser range scanner starts to rotate. The laser scans are assembled during 10 seconds into one point cloud which is filtered to consider only the close-range scene. Because of the gaps in the RGB-D point cloud from the stereo system, the collected point cloud data from the rotary laser scanner is used for the plane segmentation (Fig. 5a). The table is detected as the dominant horizontal plane over the ground (green point cloud shown in Fig. 5b) and the wall (with the valve and standpipe) are retrieved as the closest vertical plane (blue point cloud shown in Fig. 5b).



(a) (b) (c)

Fig. 5. Overview of the plane detection: filtered point cloud is shown in yellow, table in green, wall in blue and clusters of interest in white. (a) Filtered point cloud, detected table and wall; (b) Detail of the detected table and wall; (c) Detail of the clusters of interest over the planes.

Once the table is detected, the points above the plane are extracted (white points shown in Fig. 5c) and clustered into individual set of points and object candidates. The bounding boxes for all detected objects in the workspace are

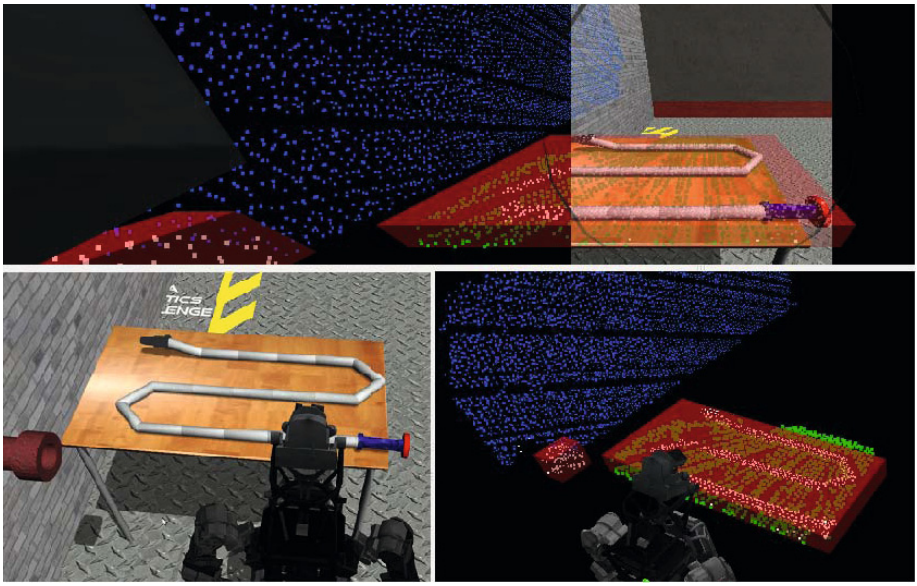
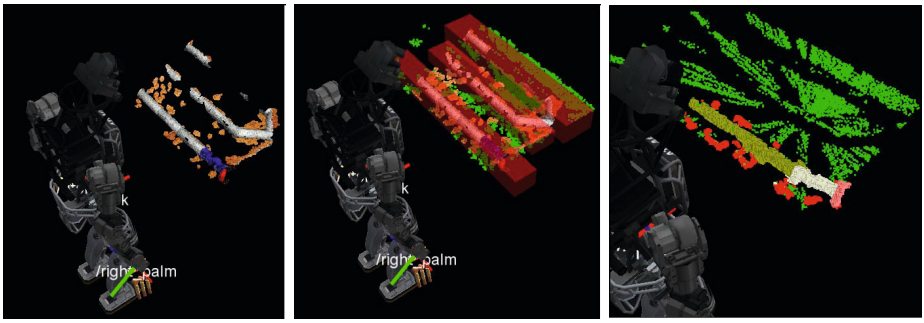


Fig. 6. Bounding box estimation of the relevant workspace objects.

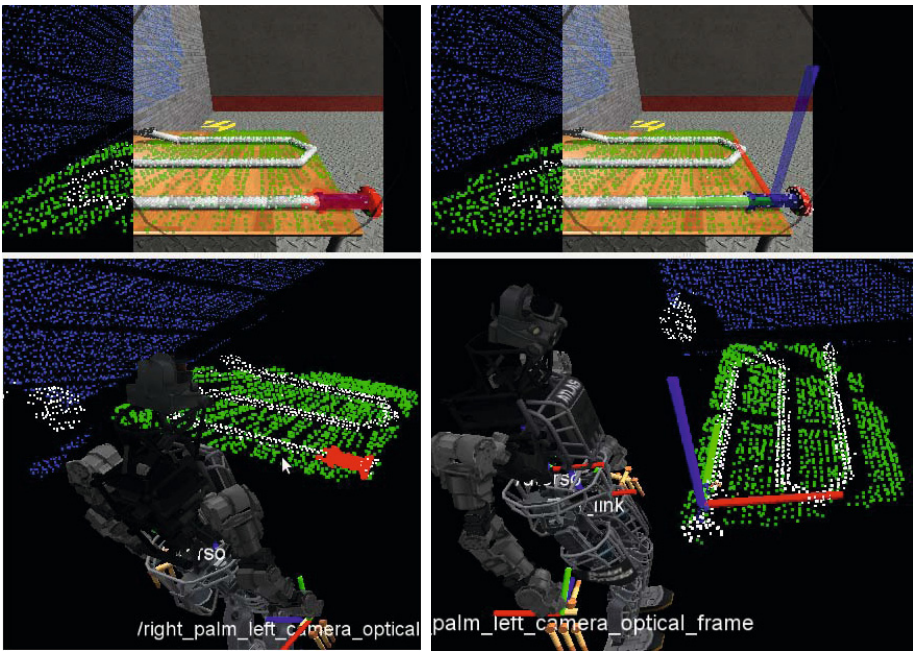


(a) (b) (c)

Fig. 7. Detail of the hose detection. The green point cloud corresponds to the table extracted from the laser data. (a) RGB-D point cloud from the stereo system inside the bounding box on the table shown in Fig. 6; (b) Due to the gaps of the point cloud, the hose is considered as three parts; (c) The color-based segmentation is applied and is shown the part corresponds to the cluster of interest: red points belongs to the table, dark green point to rigid elements, beige points to hose connector and pink points to the end of the connector hose.

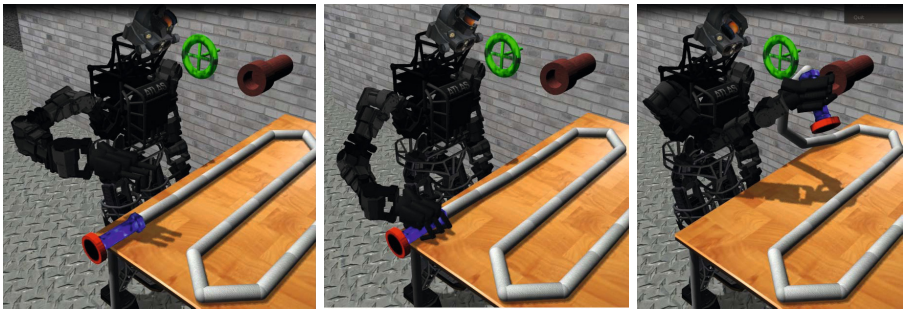
computed. Fig. 6 shows two bounding boxes which contain the point clouds corresponding to the standpipe and the hose.

The axis-aligned bounding box on top of the table encloses fully the hose. Its bounds are used to filter the RGB-D point cloud from the stereo camera



(a) (b)

Fig. 8. Detail of the hose pose estimation. (a) Cylinder model detection is applied to the detected hose connector. The fitted model is shown as red points; (b) The axis and the centroid pose of the hose connector is shown respect to the pelvis frame.



(a) (b) (c)

Fig. 9. Example of top-grasp along the principal axis of the hose connector. (a) Pre-grasp position; (b) Grasp pose; (c) Resulting grasp of the hose.

(Fig. 7a). The proposed approach computes filtered point cloud, lowering running time. This runtime efficiency is fundamental due to the fact that this perception system is used for a software competition. Because of the characteristics of the

camera, the RGB-D point cloud contains gaps and less information of the whole hose. For this reason, the points are clustered into individual units (Fig. 7b), and color-based region growing segmentation is applied to each unit. As a result, the hose is segmented into three areas: rigid elements, connector and end of the connector (Fig. 7c). The hose connector is detected taking into account that this is the blue region which is located between two areas of different color (Fig. 8a). In this scenario, the proposed approach was successfully executed and achieved 48 hose connector detections out of 50 trials. To obtain the position and the axis of the hose connector, a cylinder model is fitted to the point cloud corresponding to the hose connector previously detected (Fig. 8b). Its axis is referenced to the pelvis coordinate frame.

Since grasping itself is not the main scope of this work, the suitability of the hose connector pose for grasping has been tested as a representative way. The posterior task of attaching the hose to the standpipe and the manipulation will be considered in future works. Our approach extracts the connector's principle axes in the horizontal plane. This horizontal orientation is used by grasp planning to approach the object from the top and grasp it along such orientation (Fig. 9b). The pre-grasp position (Fig. 9a) is placed at several centimeters behind the grasp pose along the approach direction. As is shown in Fig. 9c, the robot performs successfully top-grasps along the object's principal axis using its right hand.

4 Discussion and Future Work

Humanoid disaster response robot to autonomously perform a complex task is challenging. In this paper we have described a 3D perception system to detect a hose in the simulation environment of the third VRC task. While this paper addresses one subtask of the overall hose task, the integration of this phase with the other steps is in progress.

The perception algorithm successfully segments the planes, detecting the table and the wall, as well as, the relevant objects in the workspace. For hose connector detection, a combination of color and shape has been efficiently used. The perception pipeline yields its robust results in a very short-time, which is considered as a basic requirement for a software competition. Although grasping is not the scope of this work, a representative top-grasp along the principal axis of the hose connector has been performed. It demonstrated that the estimated pose of the hose is precise enough to compute reliable grasping.

In future work, a more exhaustive evaluation of grasping, manipulation and transport tasks are considered. To increase the robustness of grasping and manipulation tasks we consider exploiting the fusion of visual and tactile data.

Acknowledgments. The authors of this paper would like to thank all the members of the SARBOT (Search And Rescue roBOT) team for their efforts and help in carrying out the tests during the Virtual Robotics Challenge. SARBOT (http://www.sarbot-team.es/) is a multidisciplinary team formed by researchers from three Universities (Universidad Politécnica de Madrid, Universidad de

Alcalá and Universidad Carlos III de Madrid) and one Public Research Centre (CAR-CSIC) to participate in the first DARPA Robotics Challenge (DRC).

References

1. DARPA: Drc. (2012), http://www.darpa.mil/Our_Work/TTO/Programs/DARPA_Robotics_Challenge.asp
2. Robotics, C.: MultiSense-SL Datasheet (2013), http://www.theroboticschallenge.org/local/documents/MultiSense_SL.pdf/
3. 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: ICRA Workshop on Open Source Software (2009)
4. Srinivasa, S., Ferguson, D., Helfrich, C., Berenson, D., Romea, A.C., Diankov, R., Gallagher, G., Hollinger, G., Kuffner, J., Vandeweghe, J.M.: Herb: a home exploring robotic butler. Autonomous Robots 28(1), 5–20 (2010)
5. Srinivasa, S., Ferguson, D., Vandeweghe, J.M., Diankov, R., Berenson, D., Helfrich, C., Strasdat, K.: The robotic busboy: Steps towards developing a mobile robotic home assistant. In: International Conference on Intelligent Autonomous Systems (July 2008)
6. Kojima, M., Okada, K., Inaba, M.: Manipulation and recognition of objects incorporating joints by a humanoid robot for daily assistive tasks. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 1564–1569 (2008)
7. Bagnell, J.A., Cavalcanti, F., Cui, L., Galluzzo, T., Hebert, M., Kazemi, M., Klingensmith, M., Libby, J., Liu, T.Y., Pollard, N.S., Pivtoraiko, M., Valois, J.S., Zhu, R.: An integrated system for autonomous robotics manipulation. In: IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 2955–2962. IEEE (2012)
8. Stückler, J., Steffens, R., Holz, D., Behnke, S.: Efficient 3d object perception and grasp planning for mobile manipulation in domestic environments. In: Robotics and Autonomous Systems (2012)
9. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in human environments. In: The 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA (October 2009)
10. Holz, D., Schnabel, R., Droeschel, D., Stückler, J., Behnke, S.: Robocup 2010, pp. 121–132. Springer, Heidelberg (2011)
11. DARPA: VRC Technical Guide (2013), http://www.theroboticschallenge.org/
12. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: International Conference on Robotics and Automation, Shanghai, China (2011)
13. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
14. Rusu, R., Bradski, G., Thibaux, R., Hsu, J.: Fast 3d recognition and pose using the viewpoint feature histogram. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2155–2162. IEEE (2010)
15. Zhan, Q., Liang, Y., Xiao, Y.: Color-based segmentation of point clouds. In: ISPRS Laser Scanning Workshop, vol. XXXVII, pp. 248–252 (2009)

Estimation of Gaussian Plume Model Parameters Using the Simulated Annealing Algorithm

Gonçalo Cabrita and Lino Marques

Institute of Systems and Robotics,
University of Coimbra,
3030-290 Coimbra, Portugal
{goncabrita,lino}@isr.uc.pt

Abstract. This article presents a novel cost function for estimating the parameters of the Gaussian plume model using simulated annealing. The novel cost function takes into account the meandering and intermittency phenomena found on dispersing plumes. The proposed method was validated using real gas sensor data sampled by a swarm of 5 robots performing the Decentralized Asynchronous Particle Swarm Optimization for plume tracing under a controlled environment.

Keywords: Gaussian plume model, simulated annealing, odor plume estimation, swarm robotics.

1 Introduction

Robotic odor-sensing technology or simply robotic olfaction has attracted substantial interest by the research community in recent years [1] [2] [3] [4]. This interest is driven by the developments in the robotics and sensing technologies along with the vast number of areas and applications of robot olfactory systems including safety, security, and environmental inspection to name a few. Robots equipped with gas sensors can be used instead of humans in areas with odor contamination for purposes such as inspection, detection of leakages leading to the contamination source, providing continuous monitoring of the contaminated environment, for specific characterization of the odor, for building the gas distribution map of the environment and others.

One of the challenges that is at the forefront of this field of research is how to deal with multiple odor sources. We recently proposed a novel methodology which targets this problem called virtual cancelation plume [5]. In a nutshell, the goal of virtual cancelation plume is to make an odor source invisible to the robot or robots, allowing the pursuit of multiple odor sources by successively cancelling the sources already found. In order to achieve this goal once a known odor source is found a model of the plume being created by that source is generated. In order to do this the robot or robots must be able to estimate the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_28,

necessary parameters. This model will then be used to affect the readings of the gas sensors equipped on the robots. Needless to say that achieving a model that represents the actual plume is of the utmost importance, thus this article tackles the problem of estimating the parameters of a plume represented by a Gaussian model, the model used in [5], using the data gathered by the robots.

Naturally this is not a new problem. In fact the process of estimating the parameters of a plume model from a sensor generated gas distribution map is known in the literature as the inverse method. In 1994 Lehning et al. used this technique to estimate gas emissions and source locations in land-fills [6]. In 2005 Flescha et al. used the same principle to achieve the same results for an agricultural scenario [7]. More recently [8] applied the inverse problem approach to locate a known gas source in a desert setting from simultaneous measurements of gas concentration and wind data. In [8] they use a simulated annealing algorithm to generate candidate distributions and present and evaluate three different cost functions with different regularisation terms. The common element in these works is the fact that these were not developed having mobile robotics in mind, in fact some of the strategies proposed to deal with the chaotic nature of odor plumes are not even applicable to a mobile robotics scenario. As the entire dataset is extracted prior to analysis a series of filtering and preprocessing is performed, in [6] unstable periods are removed from the dataset whereas in [8] the average of each location was calculated prior to parameter estimation. There are some examples of the estimation of odor plume parameters in a mobile robot context, e.g. in [9] a group of quadcopters uses a non-linear least squares approach to estimate the parameters of a Gaussian Puff model using the data gathered by their gas sensors.

In order to properly understand the challenges that we are facing when trying to estimate the parameters of a Gaussian plume model from gas sensor readings we must first take a look at the model itself and how it differs from an actual odor plume.

2 The Gaussian Plume Model

The Gaussian model is probably one of the simplest plume models commonly used. Let $\bar{c}$ be the mean chemical concentration at any given x and y given by Equation 1 where Q is the emission rate of the source, $\bar{u}$ is the mean transport velocity and σ_y and σ_z are the Gaussian plume dispersion parameters. Note that σ_y and σ_z depend on x and for non-buoyant releases can be approximated by power laws represented by equations 2 and 3. The three main requirements for a Gaussian plume model to hold are a continuous odor source, uniform wind flow and homogeneous turbulence [10].

$$\bar{c}(x,y) = \frac{Q}{\pi\sigma_y\sigma_z\bar{u}} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \tag{1}$$

$$\sigma_y = ax^p \tag{2}$$

$$\sigma_z = bx^q \tag{3}$$

Equation 1 can be derived from the assumption of a Gaussian concentration distribution in the y and z axis (considering a dominant transport velocity along the x axis) at any cross section in the plume downind of the source and the integral mass-conservation condition in Equation 4.

$$\int_0^\infty \int_{-\infty}^\infty \bar{c}\bar{u}\,\mathrm{d}y\,\mathrm{d}z = Q \tag{4}$$

A number of early diffusion experiments were conducted with the goal of characterising the values of σ_y and σ_z for different conditions. One of these experiments was carried out at the Brookhaven National Laboratory (BNL) [10]. Based on more than 15 years of diffusion data collected at the BNL site in central Long Island, New York, using nonbouyant, passive tracers, the dispersion parameters σ_y and σ_z were determined from the analysis of extensive ground-level concentration measurements for four classes of wind turbulence (Table 1). These values provide us an insight into the range of values that σ_y and σ_z can assume for an actual odor release, however these are not applicable to all locations.

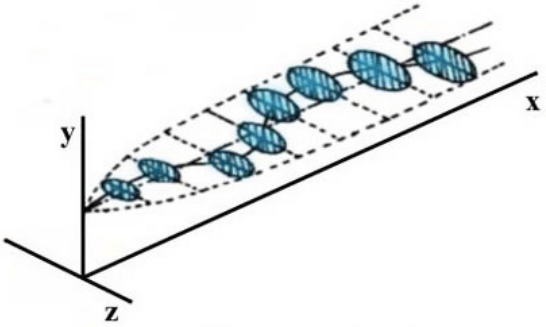
Table 1. The BNL dispersion parameterization scheme [10]

Turbulence Type	σ_y	σ_z
Unstable	$0.40x^{0.91}$	$0.41x^{0.91}$
Neutral	$0.36x^{0.86}$	$0.33x^{0.86}$
Stable	$0.32x^{0.78}$	$0.22x^{0.78}$
Very Stable	$0.31x^{0.71}$	$0.06x^{0.71}$

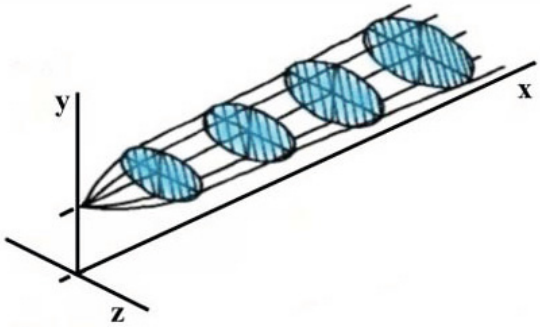
2.1 Instantaneous vs. Average

The dispersion of gaseous and particles in the atmosphere is a complex process that is dependent on the entire spectrum of turbulent motions in the atmosphere, ranging from the micro-scale (the smallest being the random motion of molecules that cause molecular diffusion) up to the macro-scale (e.g., arising from synoptic events and zonal currents). The random nature of turbulent motions of different scales produces fluctuations of the instantaneous concentration with the probabilistic and statistical characteristics of the field determined by the turbulent motions of the boundary-layer flow [11].

Concentration fluctuations are a ubiquitous feature of dispersing plumes, and the recognition of this feature has practical importance in the estimation of the parameters of a plume model based on instantaneous gas sensor readings. This is mainly due to the fact that most plume models are in fact average models, representing a plume as if its concentration values were sampled and averaged for a



(a) A meandering plume model, where the clouds both increase in size and follow a meandering path as they move away from the source.



(b) The basic Gaussian plume model, in which the spread of the cloud increases with distance from the source, estimated with a Gaussian standard deviation.

Fig. 1. Adapted from [12]

large amount of time. On the other hand mobile robots instantaneously sample the plume across their trajectory, providing a picture that represents the status of the plume at multiple places in sequential times. The two main phenomena commonly observable in a dispersing plume, meandering and intermittency are discussed next.

The meandering effect can be viewed in Figure 1. A meandering plume (Figure 1(a)), where the clouds both increase in size and follow a meandering path as they move away from the source can be averaged by a Gaussian plume model (Figure 1(b)). However an observer standing still inside the plume might not be able to detect odor at all times [12].

Intermittency is a characteristic of odor propagation that is also reflected in the probability of occurrence of zero concentration inside the plume. This is however a concept with which people are often more familiar with as it can be easily observed in a rising column of smoke. The smoke will not generate a smooth plume, it will in fact rise in patches, thus if an observer were to stand still inside the plume it would read intermittent values of odor. This effect is more prominent near the source, since as the odor propagates the patches tend to mix and the plume becomes more homogeneous as the distance from the source increases. There are some statistical analysis of how intermittency varies inside a plume [13], however these results do not apply to a broader scope of transport conditions.

3 Simulated Annealing

Simulated annealing is a metaheuristic optimization algorithm inspired in the physical process of the annealing of a solid to low energy states. This process consist of heating a solid until it reaches its fusion temperature, so that matter shifts from the solid to the liquid state. Posteriorly the temperature is slowly lowered to avoid meta-stable states. The goal of this process is to obtain a crystal-like state, i.e. a minimum energy state. On its liquid state matter contains great amounts of energy and its particles are randomly placed. On the other hand, on its crystal solid state particles are extremely structured, resulting in minimum energy. It is extremely important that the passage from the liquid to the solid state is performed slowly so that matter does not solidify in an intermediate non-structured state.

The concept of simulated annealing was introduced as an optimization algorithm during the 80s by Kirkpatrick et al. [14] [15]. However it was in the 50s that Metropolis presented an article [16] in which the process of annealing was modelled using an algorithm based on Monte Carlo techniques [17]. Not only does the simulated annealing algorithm accepts changing state if the cost function is minimized, it might also move to states which increase the cost function. It is this ability that allows this algorithm to escape local minima and find the global cost function minimum. As the algorithm progresses and the temperature cools down these worse states get increasingly less likely to be accepted. At the same time the step between states also decreases.

The process of estimating the Gaussian plume model parameters is treated as an optimization problem, where the goal is to minimize a cost function that depends on the model generated for each stage and the gas sensor readings provided by the robots. The simulated annealing was chosen for this purpose since metaheuristic and evolutionary algorithms have proven in the past to be able to cope with the chaotic nature of odor dispersion [18] [19] [20] [8]. The pseudocode of the simulated annealing algorithm used in this work is presented next.

Simulated Annealing

```
function simulatedAnnealing

    state = initial_state
    best_state = state

    cost = meanderingIntermittencyCostFunction(readings, state)
    best_cost = cost

    k = 0

    while k < k_max and cost > target_cost

       T = T_start*(k_max-k)/k_max

       new_state = generateState(best_state, T)
       new_cost = meanderingIntermittencyCostFunction(readings, new_state)

       if exp((cost - new_cost)/(T)) > random(0, 1)

           state = new_state
           cost = new_cost
        end

       if new_cost < best_cost

           best_state = new_state
           best_cost = new_cost
       end

       k = k+1
   end
end
```

The cost function is at the core of the simulated annealing algorithm. In the next section we describe a novel cost function designed having in mind the characteristics of gas dispersions discussed in 2.1.

3.1 The Meandering-Intermittency Cost Function

A common approach to define a cost associated with a certain state is to calculate the squared error between the generated model and the set of input data. However in this case it might not be reasonable to penalize a reading that is inside the plume and deviates from the value provided by the average model when we know *a priori* that such an occurrence is not only possible, but extremely likely to happen.

We developed a novel cost function that will degrade with the presence of chemical readings above the clean air threshold outside of the plume model. On

the other hand chemical readings inside the plume model will not contribute towards the cost unless they are above a concentration value that depends on the average value as determined by the plume model. The main goal is to find the shape that can fit all the sensor readings inside while at the same time representing the decay of chemical concentration along the centerline properly. The proposed cost function is shown next where c_th is the clean air threshold and *alpha* is a tunning value that allows to adjust the magnitude of the error.

The meandering-intermittency cost function

```
function meanderingIntermittencyCostFunction

    for each gas sensor reading

        error = 0
        c = gaussianPlumeModel(reading_x, reading_y)

        if c > c_th and reading > c
            error = error + alpha*exp(pow(reading - c, 2))
        end
    end
end
```

4 Experimental Setup

The experiments described next were performed inside a controlled environment, an arena designed specifically for odor experiments represented by the schematic in Figure 2 and shown in Figure 3(b). The $3m \times 4m \times 0.5m$ arena is an enclosed environment delimited by four walls where two extremities are made of honeycomb-like plastic, allowing for the air to circulate. It includes an array of controllable fans thus making it possible to control the airflow inside the arena. The top is covered by a large transparent acrylic cover. This setup allows to generate laminar and constant wind-flow. As a result the requirements for the Gaussian plume to be applicable are met.

Experiments were performed using a swarm of 5 miniQ robots (shown in Figure 3(a)) running the Decentralized Asynchronous Particle Swarm Optimization (DAPSO) algorithm for plume tracing [21]. The miniQs are small and cheap robots based on the popular Arduino platform. They were modified to achieve olfactory swarming mainly due to the e2v MiSC5524 gas sensors that they carry. Moreover, two LEDs (one red and one blue) were also integrated in the robots for usage with an Arecont MegaVideo IP camera for correcting the odometry of the robots and to provide global localization. Odometry correction and global localization is achieved using SwisTrack [22], a software designed for tracking robots.

All experiments were performed under an average wind speed of $0.1\,m/s$ with a chemical release rate of approximately $1\,mg/s$ at a height of approximately

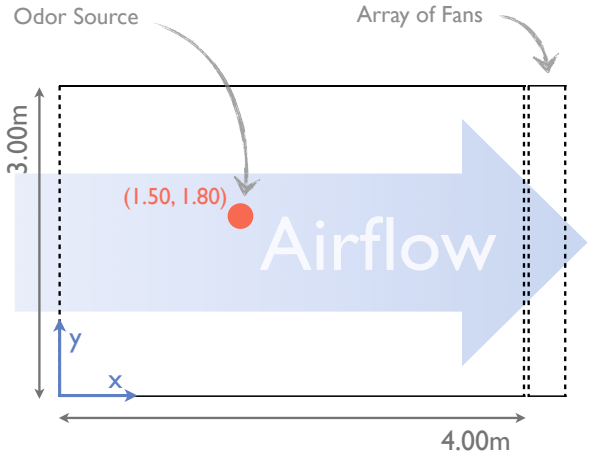


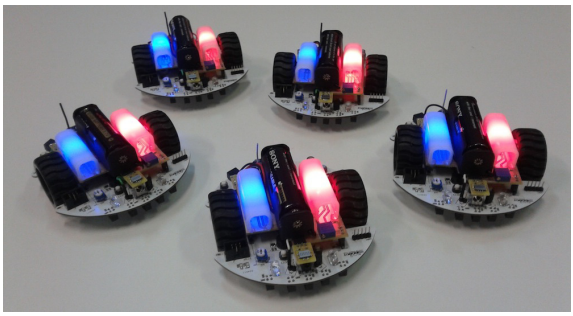
Fig. 2. The arena used for the experiments

0.07 m (about 0.04 m above the height at which the gas sensors are located). A total of 3 experiments was performed. The data collected by the robots was used to estimate the parameters of the Gaussian plume model in MATLAB where the simulated annealing algorithm using the proposed cost function was implemented. The values of c_{th} and α used were respectively 0.1 and 1.2. Notice that the chemical data collected by the robots was normalized between 0 and 1, meaning that we consider that the plume ends wherever the chemical concentration falls bellow 10% of the maximum value. We are also admitting that model-wise the concentration is allowed to peack 20% above the average value.

5 Experimental Results

The graphics in Figures 4(a), 5(a) and 6(a) contain the gas distribution map sampled by the robots during the experiments. Notice that these maps were obtained by interpolating the data sampled by the robots using Krigging [23] and are shown here for visualization purposes only. Krigging, a method commonly used in geostatistics, is based on the notion that the value at an unknown point in space should be the average of the known values at its neighbours weighted by the variogram of the distance to the unknown point. The data used by the simulated annealing algorithm to estimate the parameters of the Gaussian plume models was not pre-processed in any way.

Figures 4(b), 5(b) and 6(b) contain the Gaussian plume models after each experiment. Table 2 contains the parameters of the Gaussian plume model estimated by the simulated annealing algorithm for each experiment.



(a) The miniQs.

(b) The test arena.

Fig. 3. The miniQ robots used in the experiments and the test environment

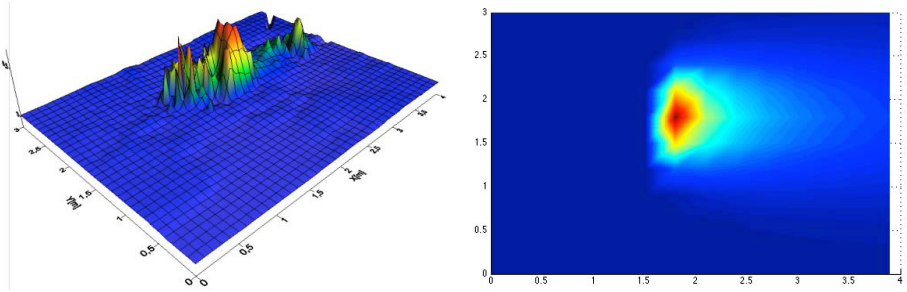
6 Discussion

Although the robots performed all 3 experiments under the same conditions it is normal that the data extracted differs from experiment to experiment due to both the chaotic nature of the odor plume and the random component present in the DAPSO algorithm. The latest results in completely different paths taken by each robot during each experiment, thus resulting in an entirely different dataset space-wise.

Nevertheless the resulting plume models resemble one another in shape. The plume model generated for experiment 2 presents the value of $\bar{u}$ closest to the

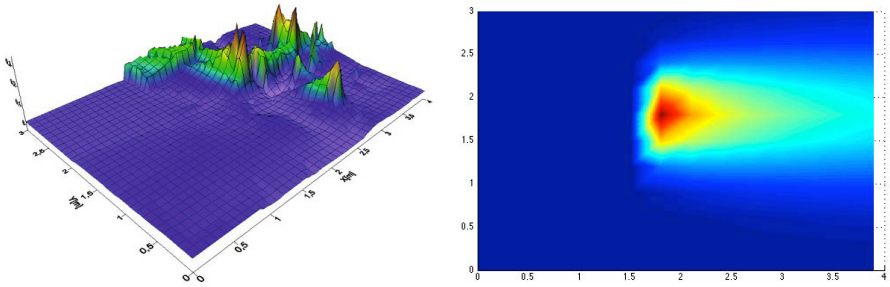
Table 2. The Gaussian plume parameters estimated for each experiment

Experiment	Q	$\bar{u}$	σ_y	σ_z
1	0.7039	0.6762	$0.5013x^{0.3713}$	$0.3499x^{0.5774}$
2	0.3425	0.1992	$0.4602x^{0.2534}$	$0.3956x^{0.2089}$
3	1.7305	0.6511	$0.8156x^{0.7215}$	$0.3643x^{0.5381}$



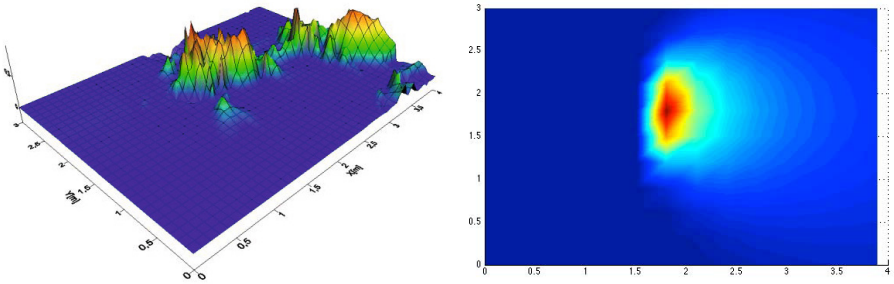
(a) Odor map (b) Estimated Gaussian plume model.

Fig. 4. Results for experiment 1



(a) Odor map (b) Estimated Gaussian plume model.

Fig. 5. Results for experiment 2



(a) Odor map (b) Estimated Gaussian plume model.

Fig. 6. Results for experiment 3

measured value. The plume model generated during experiment 3 is the widest, having to accommodate the sensor readings present at $x = 4\,m$.

Comparing the estimated values of σ_y and σ_z with those in Table 1 it is possible to see that these are well inside the expected range. There is however no identifiable pattern in relation to the conditions in Table 1. In spite of this no conclusions can be drawn in this regard as the values of σ_y and σ_z in Table 1 might not necessarily be applicable for different scenarios, particularly a small-scale scenario as the experimental setup used in this work. These values (Table 1) were obtained from data collected in large field experiments, some extending up to 60 km.

7 Conclusions

The meandering-intermittency cost function allowed to successfully estimate the parameters of the Gaussian plume model for real world experiments. The resulting plume models are adequate in the contexts of the virtual cancelation plume algorithm.

Future work will focus on re-running the experiments with a sensor network in place to allow for the measurement of ground truth data and the accurate estimation of the odor plume. This procedure will allow assessing the accuracy of the proposed method.

Acknowledgments This work was partially carried-out in the framework of TIRAMISU project. This project is funded by the European Community's Seventh Framework Program (FP7/2007-2013) under grant 284747.

References

1. Ishida, H., Wada, Y., Matsukura, H.: Chemical sensing in robotic applications: A review (2012)
2. Lochmatter, T., Aydın Göl, E., Navarro, I., Martinoli, A.: A plume tracking algorithm based on crosswind formations. In: Martinoli, A., Mondada, F., Correll, N., Mermoud, G., Egerstedt, M., Hsieh, M.A., Parker, L.E., Støy, K. (eds.) Distributed Autonomous Robotic Systems. STAR, vol. 83, pp. 91–102. Springer, Heidelberg (2013)
3. Neumann, P.P., Hernandez Bennetts, V., Lilienthal, A.J., Bartholmai, M., Schiller, J.H.: Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms. Advanced Robotics 27(9), 725–738 (2013)
4. Marques, L., Nunes, U., de Almeida, A.: Olfaction-based mobile robot navigation. Thin Solid Films 418(1), 51–58 (2002); 1st Int. School on Gas Sensors
5. Cabrita, G., Marques, L., Gazi, V.: Virtual Cancelation Plume for Multiple Odor Source Localization. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2013 (2013)
6. Lehning, M., Shonnard, D.R., Chang, D.P., Bell, R.L.: An inversion algorithm for determining area-source emissions from downwind concentration measurements. Air & Waste 44(10), 1204–1213 (1994)

7. Flesch, T.K., Wilson, J.D., Harper, L.A., Crenna, B.P.: Estimating gas emissions from a farm with an inverse-dispersion technique. Atmospheric Environment 39(27), 4863–4874 (2005)
8. Thomson, L.C., Hirst, B., Gibson, G., Gillespie, S., Jonathan, P., Skeldon, K.D., Padgett, M.J.: An improved algorithm for locating a gas source using inverse methods. Atmospheric Environment 41(6), 1128–1134 (2007)
9. Euler, J., Stryk, O.V.: Optimal Cooperative Control of Mobile Sensors for Dynamic Process Estimation. In: RSS213 Workshop on Robotics for Environmental Monitoring (2013)
10. Arya, S.P.: Air Pollution Meteorology and Dispersion. Oxford University Press (1999)
11. Yee, E., Chan, R., Kosteniuk, P., Chandler, G., Biltoft, C., Bowers, J.: The vertical structure of concentration fluctuation statistics in plumes dispersing in the atmospheric surface layer. Boundary-Layer Meteorology 76(1-2), 41–67 (1995)
12. Murlis, J., Elkinton, J., Card, R.: Odor plumes and how insects use them. Annu. Rev. Entomol. 37, 505–532 (1992)
13. Yee, E., Chan, R.: A simple model for the probability density function of concentration fluctuations in atmospheric plumes. Atmospheric Environment 31(7), 991–1002 (1997)
14. Kirkpatrick, S., Gelatt Jr., D., Vecchi, M.P.: Optimization by simmulated annealing. Science 220(4598), 671–680 (1983)
15. Kirkpatrick, S.: Optimization by simulated annealing: Quantitative studies. Journal of statistical physics 34(5-6), 975–986 (1984)
16. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. The Journal of Chemical Physics 21, 1087 (1953)
17. Gilks, W.R., Richardson, S., Spiegelhalter, D.J.: Markov chain Monte Carlo in practice, vol. 2. CRC Press (1996)
18. Marques, L., Nunes, U., de Almeida, A.: Cooperative odour field exploration with genetic algorithms. In: Proc. 5th Portuguese Conf. on Automatic Control (CONTROLO 2002), pp. 138–143 (2002)
19. Marques, L., Nunes, U., De Almeida, A.: Odour searching with autonomous mobile robots: An evolutionary-based approach. In: Proceedings of the IEEE Int. Conf. on Advanced Robotics, pp. 494–500 (2003)
20. Marques, L., Nunes, U., De Almeida, A.: Particle swarm-based olfactory guided search. Autonomous Robots 20(3), 277–287 (2006)
21. Turduev, M., Cabrita, G., Kırtay, M., Gazi, V., Marques, L.: Experimental studies on chemical concentration map building by a multi-robot system using bio-inspired algorithms. Journal of Autonomous Agents and Multi-Agent Systems (2013)
22. Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J., Martinoli, A.: Swistrack-a flexible open source tracking software for multi-agent systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 4004–4010. IEEE (2008)
23. Hawkins, D.M., Cressie, N.: Robust kriging - a proposal. Journal of the International Association for Mathematical Geology 16(1), 3–18 (1984)

Competing in the DARPA Virtual Robotics Challenge as the SARBOT Team

Elena Garcia[1⋆], Manuel Ocaña[2], Luis Miguel Bergasa[2], Manuel Ferre[3], Mohamed Abderrahim[4], Juan C. Arevalo[1], Daniel Sanz-Merodio[1], Eduardo J. Molinos[2], Noelia Hernandez[2], Ángel Llamazares[2], Francisco Suarez[3], and Silvia Rodriguez[4]

[1] Centre for Automation and Robotics, CSIC-UPM, 28500 Arganda del Rey, Madrid, Spain
[2] Universidad de Alcala, University Campus 28805 Alcala de Henares, Madrid, Spain
[3] Universidad Politecnica de Madrid, Jose Gutierrez Abascal 2, Madrid, Spain
[4] Universidad Carlos III de Madrid, Av. Universidad, 30 28911 Leganés, Madrid, Spain

Abstract. On October 25^{th} 2012 DARPA anounced the DARPA Robotics Challenge (DRC) kick off. The DRC is part of the US Department of Defense's strategic plan to conduct humanitarian, disaster relief and related operations. The DRC particularly aims to promoting innovation in robotic technology (software and hardware) to improve the capability of ground robotic systems for disaster-response operations. Designed as a three-stage competition, the DRC started with a Virtual Robotics Challenge (VRC), where a selection of 26 teams proved their algorithms in a cloud-based simulated platform. The six best-performing teams in the VRC competition won an ATLAS humanoid robot to continue to the second stage of the DRC. This paper describes the work performed by one of the VRC competitors, the SARBOT Team.

Keywords: DARPA Robotics Challenge, Virtual Robotics Challenge, Disaster Response Operations, Humanoids.

1 Introduction to the DRC

The DARPA Robotics Challenge [1] is an attempt to improve humanitarian assistance and disaster relief missions by introducing ground robots into responders teams able to rapidly enter those hazardous scenarios where human personnel cannot reach. The current state of robotics is task-specific and not flexible, adaptable and robust enough for a real-world application. The unpredictable scenario of a disaster requires robot designs compatible with environments engineered for humans, even if they are degraded. Robots being part of responders teams should have the ability to use a diverse assortment of tools engineered for humans. Besides, these robots should be supervised by personnel who have little or no robotics training.

⋆ The SARBOT Team participated in the VRC on track C without funding from DARPA. http://www.sarbot-team.es

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_29,

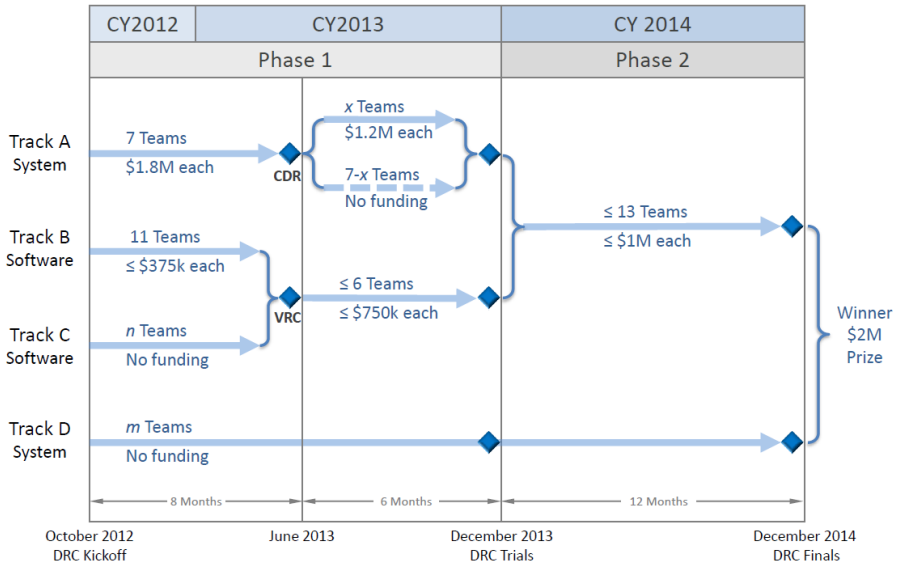


Fig. 1. DRC schedule and tracks

The DRC will give a boost to the topic by collecting know-how and effort of the best robotics teams worldwide in a three-stage competition where robot designs and supervised-autonomy control approaches must be merged together to approach a given number of disaster-response tasks. Figure 1 shows the DRC scheduling. After the challenge kick off, the fist stage was the Virtual Robotics Challenge, June 18-20 2013, where 11 Track B Teams, funded by DARPA and a bunch of more than a hundred unfunded Track C Teams competed to solve three tasks in a simulated platform. Only the best teams continue to the second stage where they will compete with 7 Track A teams, these bringing their own robot designs in the DRC Trials on December 2013.

This paper is focused on the VRC competition and it presents the team strategy and supervised-autonomy approach developed by one of the VRC competitors, the SARBOT Team. Section 2 describes the VRC rules, virtual platform, tasks and scoring procedure. Section 3 presents the SARBOT Team's partners, overall strategy and supervised-autonomy approach. Finally Section 4 shows the team's results in the VRC competition.

2 The VRC Competition

The first stage of the DRC was a Virtual Challenge, conceived by DARPA to select teams developing best-performing algorithms for robotics applications before testing them on real robots. Teams competed with their own code on a common simulated robot and environment. Prior to the VRC, teams from Tracks B and C had to qualify for the VRC by demonstrating their ability to complete two tasks

using the simulator provided by DARPA. The 26 qualifying teams (10 Track B and 16 Track C teams) competed in the VRC to see which teams could most successfully perform three tasks in the Virtual Platform, a cloud-based DARPA simulator. After the VRC up to 6 winning teams would receive funding and an ATLAS robot to compete in the DRC Trials.

2.1 Virtual Platform

The VRC took place in the cloud, making use of a constellation of 4 cloud computers for each team: one running the simulation, two running the field code and one shaping traffic. During the VRC competition, a separate constellation of machines was dedicated for each team. Figure 2 shows a block diagram of the complete virtual platform.

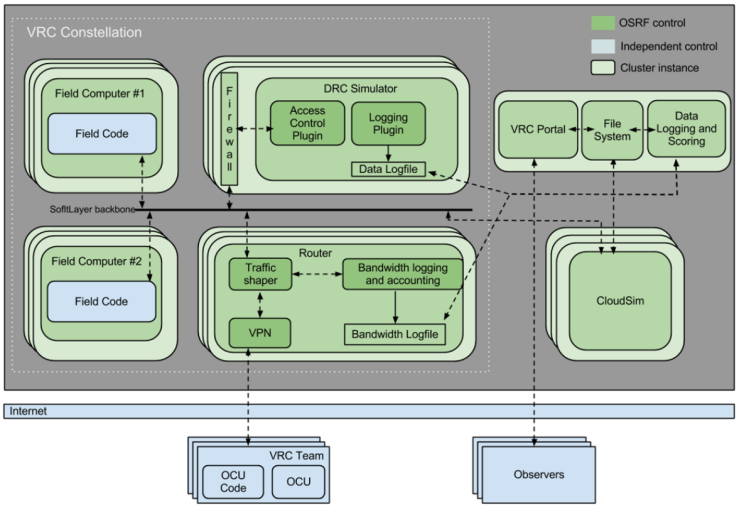


Fig. 2. Block diagram of the virtual platform in the cloud for VRC competition

The DRC Simulator runs on a computer in the cloud (see DRC Simulator in the block diagram) equipped with two Intel Xeon E5- 2690 eight-core chips, 16 GB RAM, 1 TB HD, and a NVIDIA TESLA M2090 6GB GPU, running on a 64-bit architecture. The DRC Simulator was developed by Open Source Robotics Foundation (OSRF) and it simulates, monitors and displays the physical and sensory behaviors of the robot in a 3D environment in real time. The simulator, Gazebo based on ROS Fuerte, uses physics-based models of inertia, actuation, contact and environment dynamics to simulate the robot's motion through an environment.

Field code runs on the two Field Computers, Field1 and Field2. Field1 is the same type as the computer running the DRC Simulator. Field2 is equipped with two Intel Xeon E5-2690 eight-core chips, 32 GB of memory, and 128GB SSD, running on a 64-bit architecture. For the VRC these are two computers in the cloud, serving as surrogates for the computer(s) that will be tethered to the Atlas robot in the DRC Trials. The teams run their perception, planning, and control code on the Field1 and Field2 computers, that is, all the processes requiring high-bandwidth communications with the robot.

Team's operator code runs on a system of one or more computers at the team's facility (see OCU Code within VRC Team in the block diagram). This computer system serves as the operator control station that allows command and control of the robot.

The Router is a lightweight cloud machine that runs a VPN server and scripts to measure latency, shape network traffic, control latency, and record bandwidth usage during VRC runs. The OSRF-controlled VPN handles all data between the OCU Code and the Field Code instances. During the VRC bandwith and latency are varied between runs to simulate communication disturbances in real crisis scenarios.

2.2 The ATLAS Robot

During the VRC the simulated robot was modelled on the ATLAS humanoid robot by Boston Dynamics [2]. It is 1.88m tall biped robot, weighing 150 kg (see Fig. 3(a). The robot motion is performed by means of 28 actuated degrees of freedom: 6 per leg, 6 per arm, 1 at the neck, 3 at the torso, shown in Fig. 3(b). All the joints include position sensors, and there is a 6-axis force/torque sensor in each foot. There is an IMU in the pelvis of the robot.

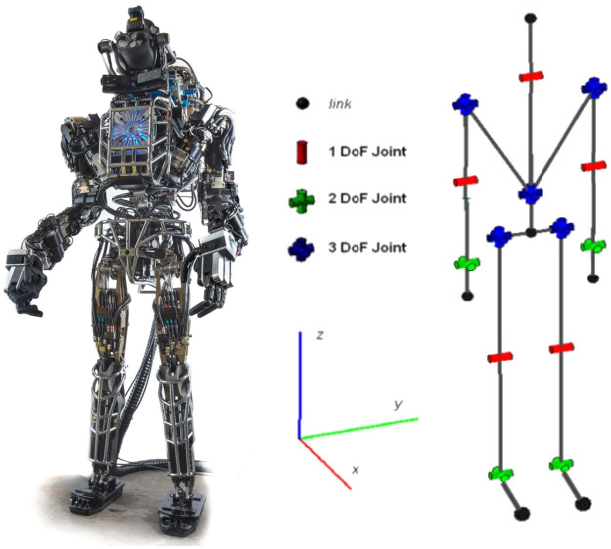
The arms feature modular wrists that accept 3rd party hands. For the DRC, two sets of dextrous hands are available: one pair by iRobot [3] and another pair by Sandia National Labs [4]. Both hand models are strong enough to hold a payload above 5 kg. However, during the VRC only the hands by Sandia Labs were modelled in Gazebo, equipped with a camera and tactile sensors.

ATLAS includes a head-mounted MultiSense-SL sensor package with LIDAR, stereo sensors, dedicated sensor electronics and perception algorithms, developed by Carnegie Robotics, featuring a range of 0.4 to 18m [5].

2.3 VRC Tasks Description

During the VRC, teams run five instances of three different tasks:

Task 1: Drive a utility vehicle
Task 2: Walk along varying terrain
Task 3: Connect a hose to a valve and open the valve



(a) (b)

Fig. 3. (a) ATLAS humanoid robot by Boston Dynamics; (b) Kinematic scheme[1]

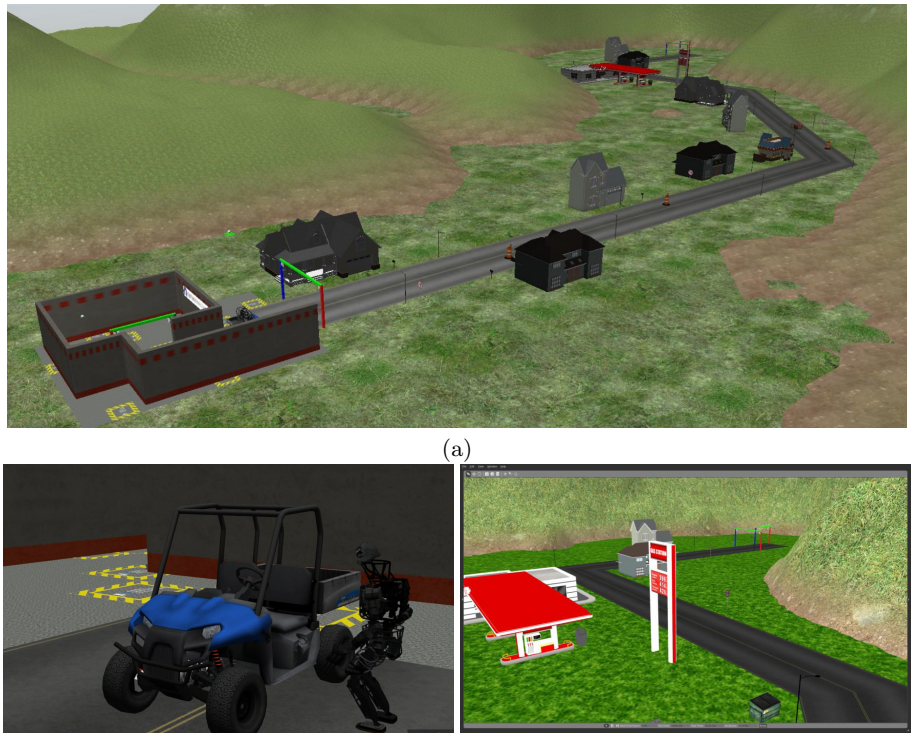
All teams will run the same 15 instances, where DARPA will vary communication parameters (mainly latency and bandwith), will limit uplink bits sent from the OCU to the Field Computers, and will vary the physics of the simulated environments. The maximum time for all runs is 30 min.

Task 1: Drive a Utility Vehicle. The robot begins in the starting pen, walks to the vehicle, drives on the road through the final road gate, gets out of the vehicle, and walks through the last off-road gate. Figure 4 shows a perspective view of the course with construction barrels on the road, and closer views of the course with the utility vehicle parked on the road near the starting pen, and the ending area. The road curvature may vary between runs and barrel placement may vary between runs.

The utility vehicle is modeled on the Polaris Ranger EV equipped with a forward/reverse button. The robot must get into the vehicle, remove hand brake, press forward button and drive the vehicle using the steering wheel and throttle pedal. The top vehicle speed is 16 km/h. The roadway is level and relatively smooth.

For the purposes of scoring, the driving task consists of four checkpoints, receiving the team one point per checkpoint got:

Checkpoint 1: Get pelvis in seat of vehicle
Checkpoint 2: Drive through first on-road gate
Checkpoint 3: Drive through final on-road gate
Checkpoint 4: Walk through final off-road gate (after getting out of vehicle)



(a)

(b) (c)

Fig. 4. (a) Perspective view of Task 1 arena; (b) Vehicle near the starting pen; (c) Ending area

The run ends when the robot has crossed the final off-road gate or when the maximum time has elapsed.

Task 2: Walk along Varying Terrain. The robot walks across flat pavement, descends to a mud pit, crosses it and ascends from it, crosses uneven hills, and traverses rubble (see Fig. 5).

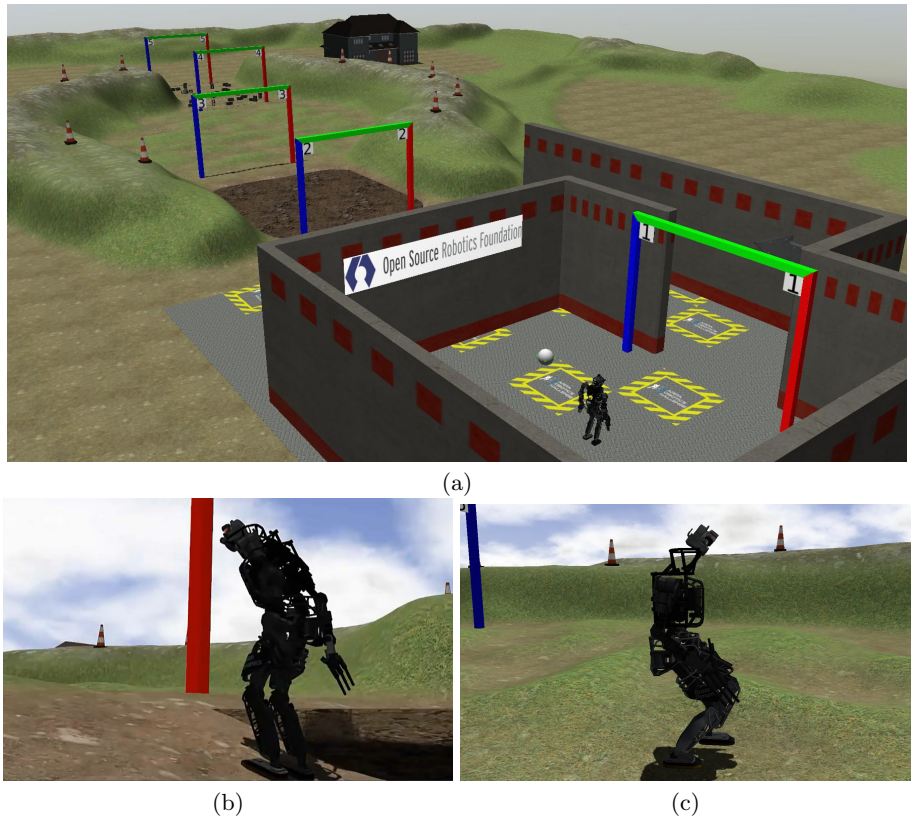


Fig. 5. (a) Perspective view of Task 2 arena; (b) Slope up the mud pit; (c) Uneven hills

Gates are numbered and they must be crossed in order. The mud pit is located between Gate 2 and Gate 3. The consistency of the mud may vary between runs, making it more or less slippery by changing the coefficient of friction, and making it more or less soft by changing the viscous coefficient. Besides, the ground profile shows a continuously varying slope degree both descending to and ascending from the mud pit. The consistency of the mud is such that the robot cannot perceive the ground profile.

Uneven hills are found between Gate 3 and Gate 4. The terrain includes a slope up and a slope down. The terrain constituting the uneven hills may vary between runs.

Finally, the rubble is found on flat ground between Gate 4 and Gate 5. The rubble consists of cinder blocks. Individual rubble elements are loose (not embedded) and subject to gravity, making individual pieces of rubble unstable.

For the purposes of scoring, the walking task shall consist of four checkpoints:

Checkpoint 1: Cross gate 2 after level ground
Checkpoint 2: Cross gate 3 after mud
Checkpoint 3: Cross gate 4 after uneven hills
Checkpoint 4: Cross gate 5 after rubble

The run ends when the robot has crossed the final gate or when the maximum time has elapsed.

Task 3: Connect a Hose to a Valve and Open the Valve. The robot walks to reach a hose on a table, lifts the hose, mates the hose to a standpipe opening by aligning hose connector and standpipe, attaches them by threading the hose connector onto the standpipe approximately 2.5 revolutions, and opens the valve by rotating it one revolution. The hose was modeled as up to ten rigid elements. Figure 6 shows the arena, with the hose connector in blue on the table, the standpipe in red and valve in green on the wall.

For the purposes of scoring, the hose task shall consist of four checkpoints:

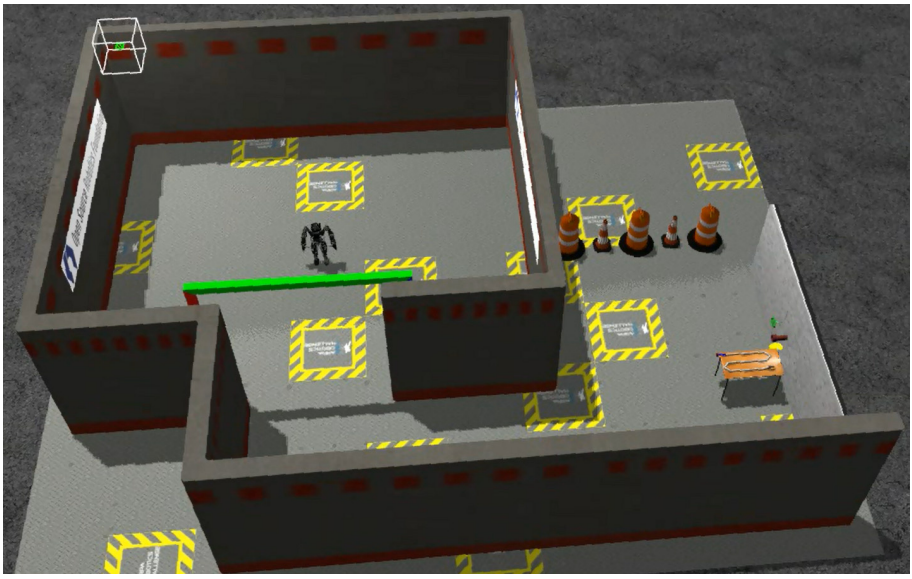
Checkpoint 1: Lift the hose connector off the table
Checkpoint 2: Mate the hose connector to the standpipe
Checkpoint 3: Attach the hose to the standpipe
Checkpoint 4: Open the valve

The run ends when the robot has completed the task or when the maximum time has elapsed.

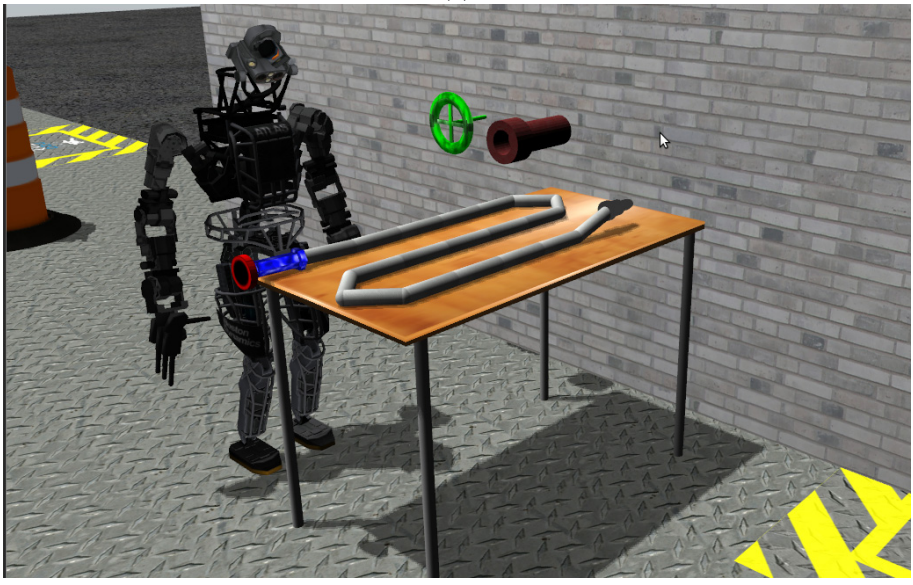
2.4 Conducting Runs

Each team run a total of 15 instances, five for each task. Each run had different configurations in terms of environment physical properties and communications. No two teams conducted the same sequence of runs. At the beginning of a run the team was informed about what task to perform. The robot started the run in the starting pen. The run time started when the robot leaved the starting pen by passing through the first gate.

The VRC traffic shaping enforced a round-trip latency of 500 ms between the Field Computers and the OCU Computer aimed to encourage autonomy, which is less sensitive than teleoperation to communications latency. All teams had equal latency, except for teams with underlying roundtrip internet latency in excess of 500 ms.



(a)



(b)

Fig. 6. (a)Perspective view of VRC Task 3 arena; (b) Closer view of hose, standpipe and valve

The number of bits allotted per run was limited by an assumed average bandwidth over the entire run time. The assumed average bandwidth was different for each of the five runs for a given task. Table 1 shows the assumed average uplink (from OCU to Field Computer) and downlink (from Field Computer to OCU) bandwidths, and the corresponding number of uplink and downlink bits allotted for each run of a task. The uplink and downlink was halted upon depletion of the Uplink Bits and Downlink Bits allotment respectively, which in turn halts the corresponding communications between OCU and Field Computer.

Table 1. Average bandwidths and corresponding number of bits allotted for each run

Run Id	Assumed Average Uplink Bandwidth (bps)	Number of Uplink Bits Allotted	Assumed Average Downlink Bandwidth (bps)	Number of Downlink Bits Allotted
1	16,384	29,491,200	524,288	943,718,400
2	4,096	7,373,800	262,144	471,859,200
3	1,024	1,843,200	131,072	235,929,600
4	256	460,800	65,536	117,964,800
5	64	115,200	32,768	58,982,400

A run terminates if the robot sustains damage. In this context, damage was defined as three occurrences of exceeding a threshold value for the maximum absolute acceleration of the center of mass.

2.5 Ranking and Scoring

Once all teams had performed the total of 15 runs, teams were ranked based on the following rules:

Let C represent the total number of checkpoints a team achieves over all 15 runs. Let B represent the percentage of uplink bits remaining after all 15 runs. Let T represent the total number of seconds remaining from the initial allotment after all 15 runs.

Teams were ranked initially by their C values. Teams with the same C value were ranked in an alternating pattern. The team with the largest B value ranked first.

Team scoring was the team's C value.

3 The SARBOT Team

The SARBOT team was one of the 115 unfunded Track C teams that registered to participate, only 16 of which qualified to compete in the VRC. SARBOT was one of the three European Teams that qualified and competed in the VRC. We designed a strategy based on the decomposition of each VRC task into subtasks. Each of these subtasks were programmed to be performed autonomously by

the robot, so the supervised autonomy consisted in the operator providing the sequence from one subtask to the next, and the transition mainly depended on the state of the robot and degree of accomplishment of each subtask. Of course, if the execution of the task was ideal, the operator could be omitted and a programmed sequence of subtasks would have done, but the complexity of the environments and the realism of the simulator provided enough uncertainty to make the operator necessary.

3.1 The Consortium

The completion of real-world actions required the integration of world perception, decision making and action mechanisms, so to ensure VRC task completion the SARBOT team was composed of four different groups, each one skilled in one of the required topics:

1. Robe-Safe group from the University of Alcala: World perception and robot navigation, including vehicle driving
2. Field and Service Robotics Group from the Centre for Automation and Robotics - CSIC: Robot legged locomotion
3. Robotics Lab - Carlos III University of Madrid: Perception for handling
4. Robots and Intelligent Machines Group - Universidad Politecnica de Madrid: Robot manipulation and grasping

Robotics and eSafety Research Group, RobeSafe (UAH). The RobeSafe Research Group at the University of Alcala is focused on perception systems applied to the fields of Service Robotics (with a focus in human-robot interfaces, multi-sensorial perception, location, navigation, SLAM and tele-assistance services [6]) and e-Safety (application of Advanced Driver Assistance Systems (ADAS) to road safety [7]).

This group brings into the SARBOT team its extensive expertise in perception and autonomous navigation. The main tasks of the RobeSafe group for the project are:

1. To perceive from laser and stereo camera sensors, that provides the MultiSense-SL sensor,
2. To process the information that comes from the sensors and elaborate a comprehensible knowledge to guide the robot in a safe way for complex terrains,
3. Identify the main elements (portals, walls, gaps, elevation of the terrain, the car, the table, etc.) to be used by the planning stage, and
4. To provide a safe navigation path that could be used by the locomotion group FSR (CSIC).

Field and Service Robotics, CAR (CSIC). The Field and Service Robotics group at the Centre for Automation and Robotics (joint centre by CSIC and UPM) has been involved in RTD activity in the field of legged locomotion since

1989, contributing to topics such as gait generation, stability [8], optimization of energy consumption and impedance control [9]. The group has designed, developed and controlled more than 10 legged platforms for real-world applications (humanitarian demining, civil construction, and human gait assistance).

The FSR group brings into the SARBOT team its extensive expertise in artificial legged locomotion. The main tasks of the FSR (CSIC) group for the project are:

1. To act as the team Coordinator,
2. To lead and guide VRC Task 2: walking in progressively more complex terrain,
3. To establish technical system requirements for locomotion, and
4. To develop gait generation algorithms and control algorithms for robot balance, legged locomotion and ground adaptation for any of the VRC tasks.

Robotics Lab, Carlos III University of Madrid (UC3M). Researchers at Carlos III University of Madrid's (UC3M) Robotics Lab goal is to endow a robotic hand system with advanced perception capabilities, high level feedback control and elements of intelligence to grasp and manipulate objects. The group's main technical and scientific contributions to the field are 3D object reconstruction and recognition [10] and learning approaches for in-hand manipulation and complex robot motor control [11].

Because of the extensive expertise in perception for manipulation, the UC3M group has collaborated SARBOT in the VRC Task 3:

1. To design the visual perception system, which allows to model the surrounding environment and the objects,
2. To develop algorithms for segmentation, object recognition and object pose estimation,
3. To provide the information for robotic hand's planning to ensure a collision-free motion for grasping and robotic manipulation.

Robots and Intelligent Machines (UPM). The group of 'Robots and Intelligent Machines' (ROMIN) at the Centre for Automation and Robotics (joint centre by CSIC and UPM) has a long tradition in the design and development of new robotic systems. The main research are focused on: telerobotics, haptic devices, parallel robots, and visual servoing. This group has participated in several national and international projects and has a great activity in publications in the mentioned areas.

This group brings into the SARBOT team its extensive expertise in object telemanipulation:

1. To design a frame for the robot teleoperation, and
2. To develop algorithms for remote object manipulation and grasping of the VRC tasks.

3.2 Team Strategy for Supervised Autonomy

SARBOT's approach to addressing the VRC challenges was based on the decomposition of each VRC task into subtasks. All these subtasks were programmed to be performed autonomously by the robot, so the supervised autonomy consisted in the operator providing upper level commands to transition from one subtask to the next.

Each autonomous subtask was designed as the integration of perception + decision making + action mechanisms, thus taking advantage of our team components, each skilled in one of these topics. Figure 7 shows a block diagram of a generic subtask, made up of the sequential execution of perception, decision and action mechanisms.

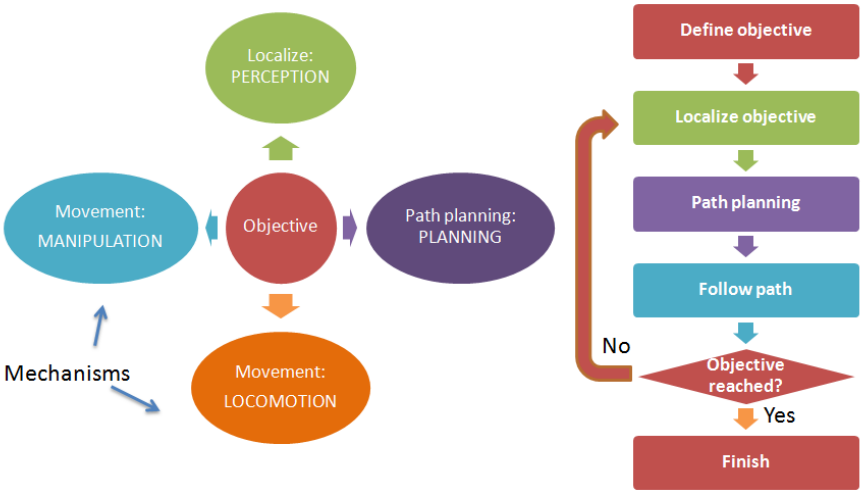


Fig. 7. Mechanisms composing a generic subtask, and block diagram of the sequential development

Tables 2 to 4 list autonomous subtasks and their perception-decision-action mechanisms designed by SARBOT to accomplish each of the three VRC tasks. The group that manages the mechanism is shown.

4 VRC Results

The SARBOT team competed in the VRC and ranked 17th of 26 competitors. We designed a robot that was supposed to walk bipedally, and to perceive the world and build 3D maps, and make decisions, to have a realistic competitive robot, and we never tried to execute the tasks with the only goal of scoring. Considering the VRC as a preliminary stage to the DRC with a real robot, all the code and experience of the VRC should be useful for the DRC competition

Table 2. Task 1 decomposition into autonomous subtasks and related perception-decision-action mechanisms

Task	Task Leader: UAH			Objective	Localization	Path planning	Walking	Manipulation
1	**Climb into a utility vehicle, drive along a roadway, and climb out of the vehicle**							
	STL	**1.1. Getting into the vehicle**			Mechanism leader			
	UAH	Exit starting pen		Gate 1	UAH	UAH	CSIC	
	UAH	Walk to the vehicle		Vehicle	UAH	UAH	CSIC	
	UAH	Walk to driver door		Driver door	UAH	UAH	CSIC	
	UPM	Grasp handles		Handles	UC3M	UPM		UPM
	CSIC	Get into vehicle		Seat	UC3M	UPM	CSIC	UPM
		1.2. Driving the vehicle						
	UC3M	**DRIVE ()**	Remove hand brake	Hand brake	UC3M	UPM		UPM
	UPM		Grasp steering wheel	Steering wheel	UC3M	UPM		UPM
	UPM		Rotate steering wheel	Angle				UPM
	CSIC		Throttle	Pedal	UC3M	CSIC	CSIC	
	UAH	Drive to Gate 2		Gate 2	UAH	UAH	**DRIVE()**	
	UAH	Follow the road		Carretera	UAH	UAH	**DRIVE ()**	
	UAH	Avoid barrels		Obstáculo	UAH	UAH	**DRIVE()**	
	UAH	Drive to Gate 3		Gate 3	UAH	UAH	**DRIVE)**	
		1.3. Getting out of the vehicle						
	UPM	Grasp Handles		Handles	UC3M	UPM		UPM
	CSIC	Exit vehicle		Goal position	UC3M	UPM	CSIC	UPM
	UAH	Walk through Gate 4		Gate 4	UAH	UAH	CSIC	

Table 3. Task 2 decomposition into autonomous subtasks and related perception-decision-action mechanisms

Task	Task Leader: CSIC			Objective	Localization	Path planning	Walking	Manipulation
2	**Walk over varied terrain**							
	STL	SubTask			Mechanism Leader			
	UAH	Exit starting pen		Gate 1	UAH	UAH	CSIC	
	UAH	Avoid wall		Wall	UAH	UAH	CSIC	
	CSIC	**Generate Gait ()**	Turn	Angle			CSIC	
	CSIC		Straight line	CG path			CSIC	
	CSIC		Ground impedance control	Kxy Kz Bxy Bz			CSIC	
	CSIC		Slope	Ground inclination			CSIC	
	CSIC		Gait forbidden areas	Forbidden areas			CSIC	
	UAH	**Walk ()**	Get ground profile	terrain	UAH	UAH	CSIC	
	CSIC		Select gait	Ground profile + CG path		CSIC	**Generate Gait()**	
	CSIC		Run gait	Gait		CSIC		
	UAH	Walk through Gate i		Gate i	UAH	UAH	**Walk ()**	

Table 4. Task 3 decomposition into autonomous subtasks and related perception-decision-action mechanisms

Task	Task Leader: UPM		Objetivo	Localization	Path planning	Walking	Manipulation
3	Connect Hose an open spigot						
	FL	**3.1. Mating**		Mechanism leader			
	UAH	Exit starting pen	Gate 1	UAH	UAH	CSIC	
	UAH	Avoid wall	Wall	UAH	UAH	CSIC	
	UAH	Walk to table	Table	UAH	UAH	CSIC	
	UPM	Grasp hose	Hose connector	UC3M	UPM		UPM
	UPM	Lift hose	Clearance		UPM	CSIC	UPM
	UC3M	Reach pipe & Mate axis	Hose and pipe axes	UC3M	UPM	CSIC	UPM
		3.2. Threading					
	UPM	Thread	Thread angle		UPM	CSIC	UPM
		3.3. Open valve					
	UC3M	Reach & Grasp valve	Valve	UC3M	UPM	CSIC	UPM
	UPM	Open valve	Angle	UC3M	UPM	CSIC	UPM

with a real robot, so we dedicated our efforts to solve the tasks as realistically as possible. In fact our team was one of those teams with a low number of robot falls during the VRC tasks.

During the VRC we found that the most difficult subtask was always at the beginning of the VRC task, which completely blocked the task. For example, we were able to drive the vehicle, but did not have the time or the personnel to teach the robot to enter the vehicle, so we did not have a chance in Task 1. Similarly, getting out of the mud pit in Task 2 was so complicated because the ankle joint range was limited, besides, walking in the mud was difficult, much more than the rest of terrains in Task 2.

Other difficulties were found in VRC arenas and robot gazebo models all the time during preparation for the VRC. We spent several efforts to step over the 45 cm gap of the Qualification walking task, and when we finally succeeded, the gap was reduced by DARPA. We experienced also many problems with the points cloud from the cameras simulated in gazebo, which made it so complicated to perceive the table and hose in VRC Task 3, until the last version of gazebo was released. Joint controllers were not tunable and excessive noise in IMUs and other sensors made us spend so much time.

Now, after the VRC has finished, we have won the experience of taking part in one of the most amazing robotics challenges ever, and we have developed algorithms that are valid to control the operation of any robotic platform in a real-world environment. Our developments are valid for a wide variety of applications: search and rescue, surveillance, exploration, etc.

Acknowledgements. This work has been performed by the SARBOT Team, composed by the University of Alcala, Polithecnic University of Madrid, University Carlos III of Madrid, and Spanish National Research Council (CSIC). This work has been partially funded by the Spanish Ministry for Economy and Competitiveness through grant DPI2010-18702.

References

1. The DARPA Robotics Challenge (2013), http://www.theroboticschallenge.org
2. Boston Dynamics: Atlas - the agile anthropomorphic robot (2013), http://www.bostondynamics.com/robot_Atlas.html
3. Guizzo, E.: iRobot smashes its new robotic hand with baseball bat (2012), IEEE Spectrum blog, http://spectrum.ieee.org/automaton/robotics/robotics-hardware/irobot-smashes-its-new-robotic-hand-with-baseball-bat
4. Ackerman, E.: Sandia National Labs gives roboticists a hand (2012), IEEE Spectrum blog, http://spectrum.ieee.org/automaton/robotics/humanoids/sandia-labs-robotic-hand-
5. Carnegie Robotics: MultiSense-SL (2013), https://support.carnegierobotics.com/attachments/token/ndxeazyuhj2sgd7/?name=Carnegie_Robotics_MultiSense_SL-v2.pdf
6. Llamazares, Á., Ivan, V., Molinos, E., Ocaña, M., Vijayakumar, S.: Dynamic Obstacle Avoidance Using Bayesian Occupancy Filter and Approximate Inference. Sensors 13(3), 2929–2944 (2013)
7. Gonzalez, A., Bergasa, L., Yebes, J.: Text detection and recognition on traffic panels from street-level imagery using visual appearance. IEEE Transactions on Intelligent Transportation Systems (2013)
8. Gonzalez de Santos, P., Garcia, E., Estremera, J.: Quadrupedal Locomotion: An Introduction to the Control of Four-Legged Robots. Springer, London (2006)
9. Arevalo, J., Garcia, E.: Impedance control for legged robots: An insight into the concepts involved. IEEE Transactions on Systems Man and Cybernetic Part C-Applications and Reviews 42(6), 1400–1411 (2012)
10. Rodríguez-Jiménez, S., Burrus, N., Abderrahim, M.: 3d object reconstruction with a single rgb-depth image. In: Proceedings of the International Conference on Computer Vision Theory and Applications, VISAPP (2013)
11. González-Quijano, J., Abderrahim, M., Fernández, F., Bensalah, C.: A kinodynamic planning-learning algorithm for complex robot motor control. In: Proceedings of the 2012 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp. 80–83 (2012)

Reactive Humanoid Walking Algorithm for Occluded Terrain

J.C. Arevalo, Daniel Sanz-Merodio, and Elena Garcia*

Centre for Automation and Robotics CSIC-UPM Carretera Campo Real Km 0,200
28500 Madrid, Spain
juan.arevalo@csic.es

Abstract. Humanoid robots are a good option for rescue missions in disaster scenarios when the safety of the rescue team can be compromised. Thus, allowing saving lives without the need of risking the safety of the rescue team. In this study we propose a reactive walking algorithm that provides statically stable locomotion when negotiating difficult terrains (combinations of inclined planes and flat ground) in the case that computer vision is not an option due to occluded terrains. Furthermore, we have tested the algorithm in simulation using the DRCSim software package obtaining good results.

1 Introduction

Rescue robots can reduce the number of fatal victims in disaster scenarios where environmental conditions are hazardous for human and animal rescue teams, such as Fukushima's nuclear disaster [3]. However, to be able to do this, they need to negotiate different types of terrain and possibly use tools designed for humans. For that reason, in the case of a disaster scenario, a humanoid robot is among the best choices.

Nevertheless, humanoid robots (and bipeds in general) are less stable than other types of legged robots such as quadrupeds or hexapods. For that reason, they require a more careful motion planning than other types of multilegged machines. Ideally, to perform such planning, information of the terrain should be available. This information could be given either a priori or obtained by means of artificial perception. However, there is no reason for this information to be known a priori (especially in a disaster scenario) which makes it the most logical choice to obtain the information through perception.

Though perception can improve motion planning [2], there is also the issue of occlusion. Whenever the terrain is occluded it is impossible to analyze by means of artificial perception (Computer vision). In this work, we plan to tackle this problem in the worst case scenario: where perception is not available and the terrain is unknown. In this case, it is impossible to use planning algorithms

* This work has been performed by the SARBOT Team, composed by the University of Alcala, Polithecnic University of Madrid, University Carlos III of Madrid, and Spanish National Research Council (CSIC).

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_30,

to obtain the necessary footholds. To solve this problem we propose a reactive algorithm which provides a statically stable walking gait.

To evaluate the proposed algorithm we make use of the DARPA Robotics Challenge Simulator (DRCSim) software. This software was developed to be used in an international competition where a humanoid robot needs to perform a variety of tasks expected to be present in disaster scenarios, including walking through difficult occluded terrain. Fig. 1 shows a photo of the Atlas robot and a screen capture of the simulation used for this work.

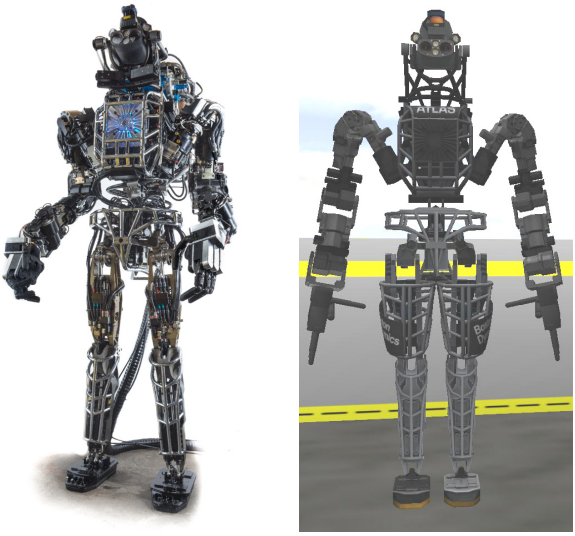


Fig. 1. Atlas robot (left) and Atlas robot simulation (right)

The rest of this paper is organized as follows: Section 2 presents the details of the controller used for the robot's gait; Section 3 details the stability control approach; Section 4 describes the modifications needed to the gait and stability control to improve adaptability to various terrains; Section 5 describes the simulation setup in detail; Section 6 shows the simulation results and Section 7 draws the conclusions.

2 Gait Control

The Atlas robot is controlled with a position PID controller with a proportional velocity controller and a feed-forward torque command at each joint:

$$\tau_j = \left(K_p + K_d s + \frac{K_i}{s}\right)(x_d - x) + K_v(v_d - v) + \tau_d \tag{1}$$

where τ_j is the joint torque; K_p, K_d and K_i are the proportional, derivative and integral constants of the PID controller respectively; x_d, v_d and τ_d are the desired position, velocity and torque respectively; x and v are the measured position and velocity of the joint; and s is the Laplace transform variable.

The joint commands where computed by means of inverse kinematics while the velocity was set to zero thus emulating a damping element on each joint. Fig. 2 shows a kinematic diagram of the legs of the Atlas robot. As it can be seen in the figure, the robot has 6 degrees of freedom (DOF) at each leg ($q_{i,j}$ with $i = 1, 2, .., 6$ and $j = r, l$ where r and l stand for right and left respectively). Moreover, each leg has three consecutive joints parallel to each other, which is a sufficient condition to the existence of closed form solutions [8] for the inverse kinematics problem.

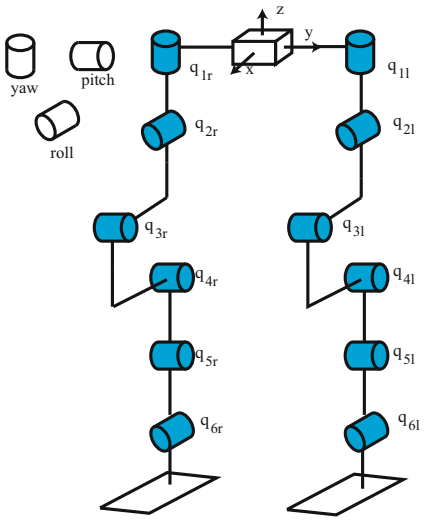


Fig. 2. Kinematic diagram of the legs of the Atlas Robot

Moreover, the robot has position, velocity and torque sensors at each joint; a 6 DOF torque sensor at each foot (placed at the ankles); and one inertial measurement unit (IMU) placed at the center of the hip link (the base of the robot). It is important to note that the robot also has stereoscopic vision, a laser and one additional IMU at the head, however due to the nature of this work these last sensors are not used.

Because we are assuming that the terrain is occluded and we have no information of its geometrical properties, the trajectories are computed to achieve a statically stable gait [6], which requires less planning and has no need for a dynamical model since the motion is stable at all times during the locomotion cycle. The walking gait is programmed as a two phase discontinuous gait (see Fig. 3) [7].

- In the first phase all the weight is put into the next support leg (left leg in the figure). Then, the other leg is lifted from the ground and placed at its next foothold position (swing). Afterwards, the center of mass's horizontal projection is moved into the leg that has just finished its swing phase. More generally, the center of mass is placed inside the polygon of the leg that will be support leg in the next single support phase.
- The second phase consists on repeating the aforementioned motions starting with the other leg (see Fig. 3 bottom). These two steps are repeated cyclically to form a periodical walking gait.

The above described procedure can be written in the following set of equations for each leg in double support.

$$^{n}X_{sw} = X_{sp} + (X_{sw} - X_{sp})(0.5 - cos(\omega)/2) \tag{2}$$

$$^{n}X_{sp} = X_{sw} + (0 - X_{sw})(0.5 - cos(\omega)/2) \tag{3}$$

where X_{sw} and X_{sp} are the positions of the legs at the end of a single support phase in the base coordinate frame located in the Atlas pelvis. The superscript n is used to indicate that at the end of this double support phase the leg will be in the state indicated by the subscript; and $\omega = t/t_{ds}$, where t_{ds} is the duration of the double support phase. Note that the trajectories described in the equations are straight lines multiplied by cosines in order to smooth the transitions between each phase.Moreover, all the positions in the equations are vectors of dimension 2, i.e. they are of the form: $X = [x, y]^T$.

Regarding single support, the equations can be written as:

$$^{n}X_{sw} = X_{sw} + (^{n}X_{sp} - X_{sw})(0.5 - cos(\omega)/2) \tag{4}$$

$$^{n}z_{sw} = z_{sw} + h(0.5 - cos(\omega * 2)/2) \tag{5}$$

$$^{n}X_{sp} = X_{sp} \tag{6}$$

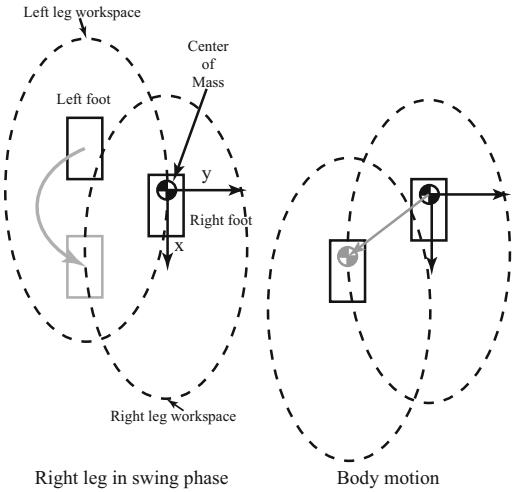
where h is the step height and z is the vertical coordinate of the swing leg's foot, while all the other parameters are the same as the previous equations.

3 Stability Control

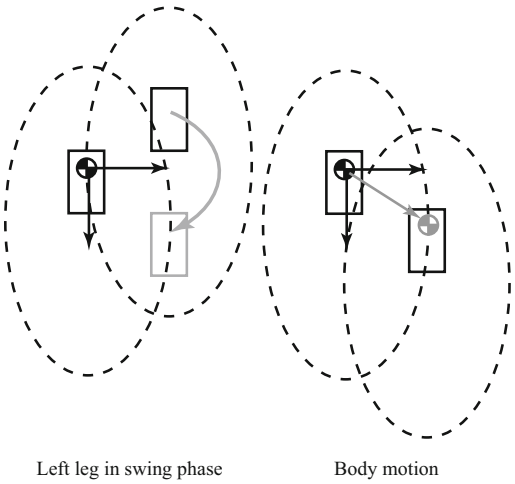
The stability control was divided into two controllers: one for the single support phase and other for the double support phase. The reason to do this is because when both legs are in support phase the horizontal components of the reaction forces from the ground make opposite contributions and may cancel each other out.

3.1 Single Support Controller

To maintain the balance while the robot is in single support we assume that the robot standing on one leg is equivalent to an inverted pendulum. However



Phase 1



Phase 2

Fig. 3. Depiction of a two phase discontinuous gait for biped robots

since the robot is capable of 3 dimensional movements, we decouple the problem by planes, that is we use the ankle joint capable of roll (q_6) to control lateral stability and the ankle joint capable of pitch (q_5) to control sagittal stability. To do that, we block the hip and knee joints of the support leg and use the ankles degrees of freedom to maintain the center of mass in a vertical position, i.e. pitch ($\phi = 0$) and roll ($\theta = 0$). It is important to say that because this leg is in single support the COM should not move. The measurements of the pitch

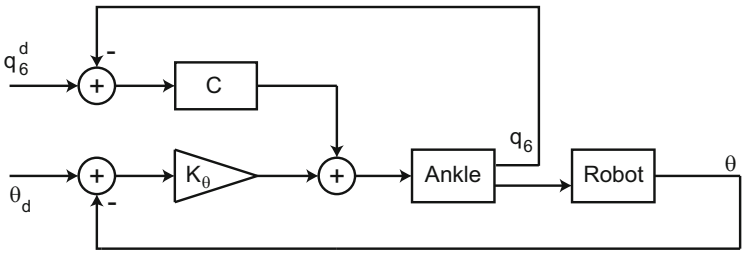


Fig. 4. Block diagram of the control scheme to maintain balance in single support

and roll angle are taken from the IMU present at the hip. The feedback loop is closed by means of the feedforward torque at the corresponding ankle joint. The control scheme to control the roll angle is depicted in Fig. 4.

In the figure, q_6 and θ are the measured ankle angle and roll angle respectively; q_6^d and θ_d are the desired ankle position and robot's roll angle respectively; while C is the part of equation 1 that does not include the feed forward torque τ_d, i.e. $C = (K_p + K_d s + K_i/s)(x_d - x) + K_v(v_d - v)$; τ_d is then replaced by $\theta_d - \theta$; and K_θ is a controller gain.

The robot's pitch angle (ϕ) is controlled using the same scheme as the roll angle (see Fig. 4) replacing θ by ϕ and q_6 by q_5.

3.2 Double Support Controller

The double support controller is based on the same approximation as the single support controller. Nevertheless, as both legs are now on the ground their contributions to the center of mass angular momentum might cancel each other. This situation is because both legs are connected to the hip link at opposite ends, in other words, the legs exert forces at opposite ends of a rigid bar. Therefore, to compute in an accurate manner the net contribution to the center of mass' angular momentum it is needed to know as precisely as possible the position of each leg and their individual contribution to the angular momentum.

To avoid the complicated computation of the net angular momentum, while taking into account the aforementioned situation, we use a simple fuzzy logic controller [1] to couple the controllers from each leg to maintain the center of mass in a vertical position.

Fig. 5 shows the block diagram of the controller used in double support to control the roll angle (θ). In the figure, q_{6l} and q_{6r} indicate the left and right leg respectively; the subscript d indicates the desired position; C has the same form as the control scheme for single support; and the fuzzy controller has one input, two rules ($\theta < 0$ and $\theta > 0$) and two outputs. This means if $\theta < 0$ then the right leg is used to compensate for the deviation and if $\theta < 0$ then the left leg is used to correct the robot orientation.

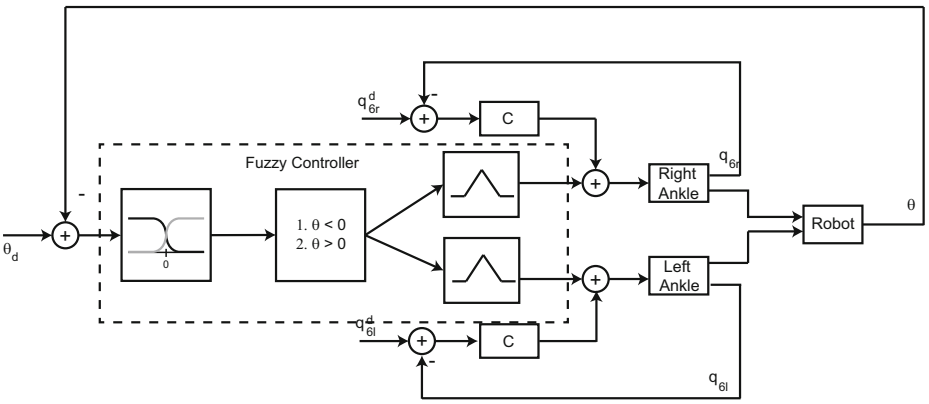


Fig. 5. Block diagram of the controller used to maintain balance in double support phase

The control scheme for correcting the pitch angle (ϕ) was not modified during double support, because the force contributions of each leg had the same direction. In other words, because of the manner in which the gait is designed, both force contribution to the angular momentum around the pitch axis are always parallel. For that reason, the only modification needed was reducing the magnitude of the controller gain K (see Fig. 4).

4 Adaptability to the Terrain

As we mentioned earlier, the adaptability to the terrain when no information of its geometrical properties is available requires a reactive approach. For that reason, the gait control approach described needs to be modified to take into consideration the terrain variations. The first thing to consider is that the foot needs to be placed on the ground and the contact must be held. Nonetheless, in certain extreme cases a very compliant foot might be needed or a good foothold might not be possible at all. For that reason we restrict this study to horizontal and inclined planes (whether they have positive or negative inclination) and combinations of the two.

Fig. 6 shows the state diagram of the implemented controller. As it can be seen in the figure, the algorithm is the same for each foot, switching between the two when the COM is moved. Moreover, to ensure foot contact we make use of the force sensor present at each foot. The reason for this is that if a fixed foot orientation were to be commanded the robot might not be able to ensure contact. Thus, to accommodate for any possible terrain (within the limitations of this work) we make the position PID constants of the swing leg equal to zero and only leave active the damping constant in order to avoid high impact forces that might destabilize the robot. In this way, we modify the foot trajectory generation explained above (see equations (4)-(6)) to stop the motion of the swing leg when

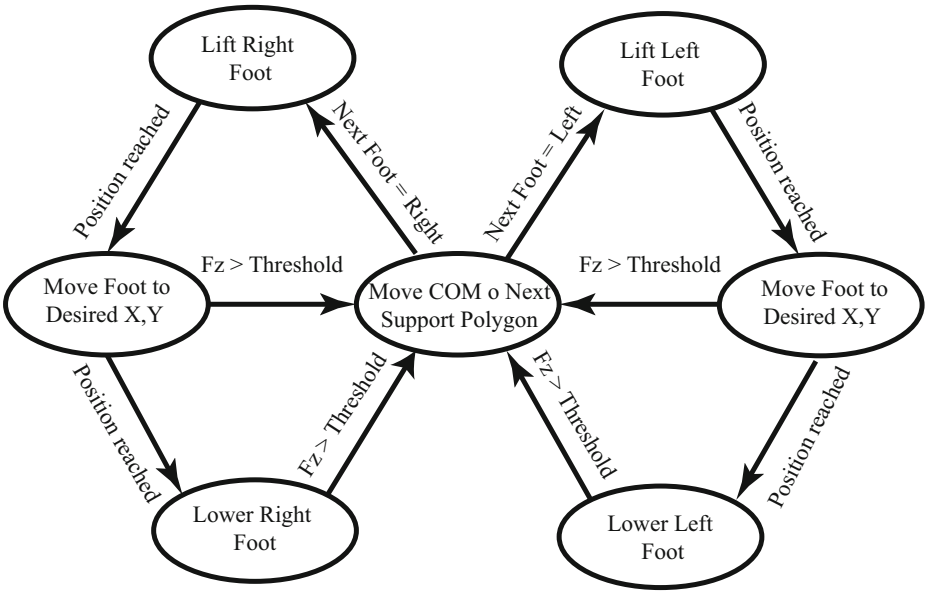


Fig. 6. Flow diagram of the reactive walking algorithm

the foot forces reach a certain threshold level. Furthermore, when the threshold is crossed (in this case 10 N), the leg is commanded to move downwards until the ground reaction force reaches a higher threshold (150 N)to make the contact stable.

Once the contact is established, the center of mass is commanded to move within the support polygon to the area occupied by the foot of the next support leg.

5 Simulation Setup

To test the walking algorithm we used the DARPA Robotics Challenge Simulator (DRCSim). The DRCSim is a collection of models, environments, plugins, and tools that customize the Gazebo simulator for use in the DARPA Robotics Challenge (DRC) [4]. Gazebo is a multi-robot simulator for outdoor environments. It is capable of simulating a population of robots, sensors and objects in a three-dimensional world. It generates both realistic sensor feedback and physically plausible interactions between objects [5].

The user interface of the simulator is shown in Fig. 7. This interface is mostly for visualization of the robot behavior, the view of the robot can be rotated zoomed in or zoomed out. In addition, the robot can be moved and rotated and components of the scenario can be deleted or added, although this is not recommended because it can interfere with the simulation behavior.

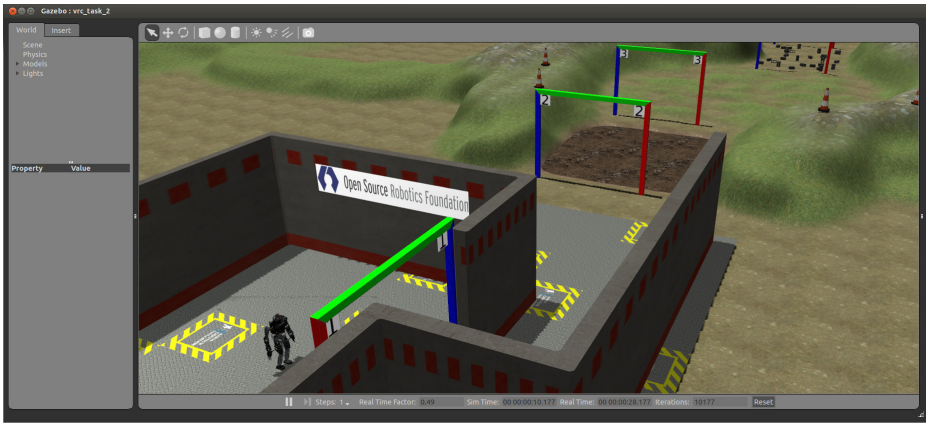


Fig. 7. Screen capture of the DRCSim graphical interface

The DRCSim includes many scenarios and objects; nevertheless the most important is the ATLAS robot. The robot model is included in the simulator with a behavior plugin that has a control interface using ROS (Robot Operating System). In other words, to control the robot it is necessary to use a series of ROS topics to send commands to the joints and to retrieve sensor information.

6 Simulation Results

The algorithm proposed in section 4 was implemented on the Atlas robot as a ROS node programed in C++, with an execution frequency of 1 kHz. Fig. 8 shows the foot trajectories in the planes XY (horizontal) and ZX (sagittal) respect to the base (the hip joint). It is important to note that the robot is advancing in the negative X direction.

In the figure, the XY plane (top) shows the foot trajectories of both the right (dark line) and the left (light line) foot. As it can be seen the trajectory of each foot describes a triangle, accordingly to the two phase discontinuous gait. The trajectory of the right foot in the sagittal plane is shown in the bottom of Fig. 8. The circle indicates the starting point of the trajectory and the arrows the direction followed by the foot.

Another important remark is that the trajectory is depicted when the movement is periodical, in other words, the first step is not included. Moreover, in the figure it can be seen that first the foot moves backwards to move the center of mass forward. Then, once the center of mass is in the right position the foot elevates from the ground and goes back to its initial position, but lifted from the ground. Afterwards, it moves forward according to the commanded step length, after the foot reaches the desired horizontal position then it moves downwards until it detects contact. Then, the foot moves backwards again to move the center of mass in the direction of motion and finally the foot is commanded to go back to its original vertical position, thus completing a gait cycle.

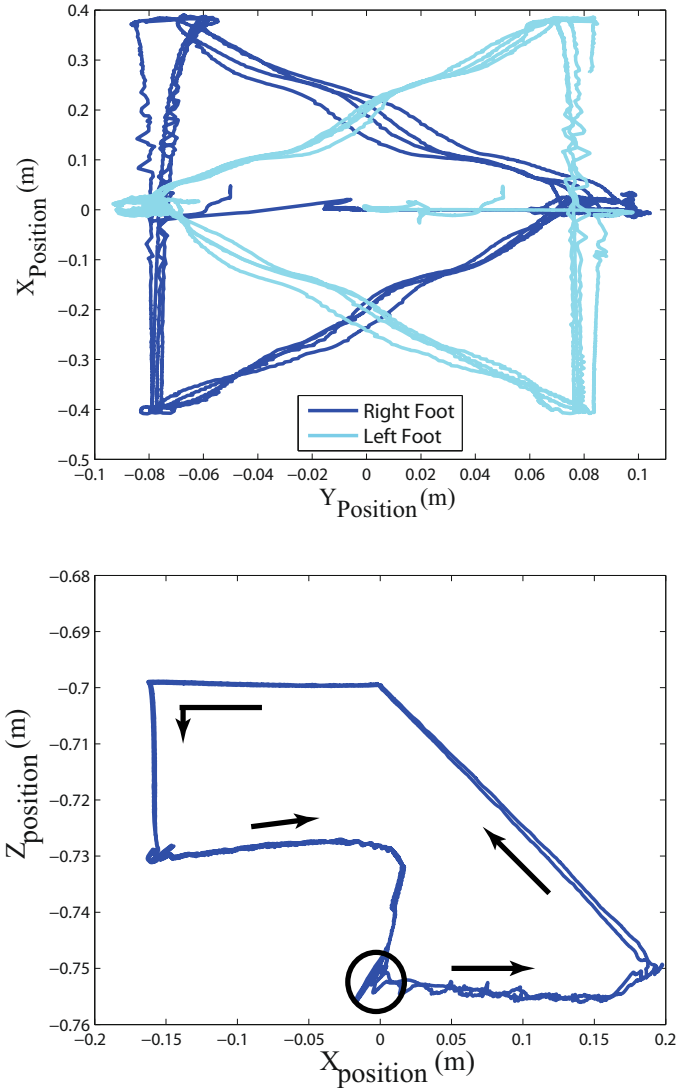


Fig. 8. Foot trajectories in the planes XY (top) and XZ (bottom) when the Atlas robot is walking on level ground

Figure 9 shows the XZ plane view of the right foot trajectory while walking downhill (top) and uphill (bottom) with variable slopes overlapped with the foot trajectory while walking in flat ground. As it can be seen in Fig. 9 (left), the foot moves downwards more distance than in the case of flat ground. Moreover, as the terrain has variable slopes, it moves further down with each step.

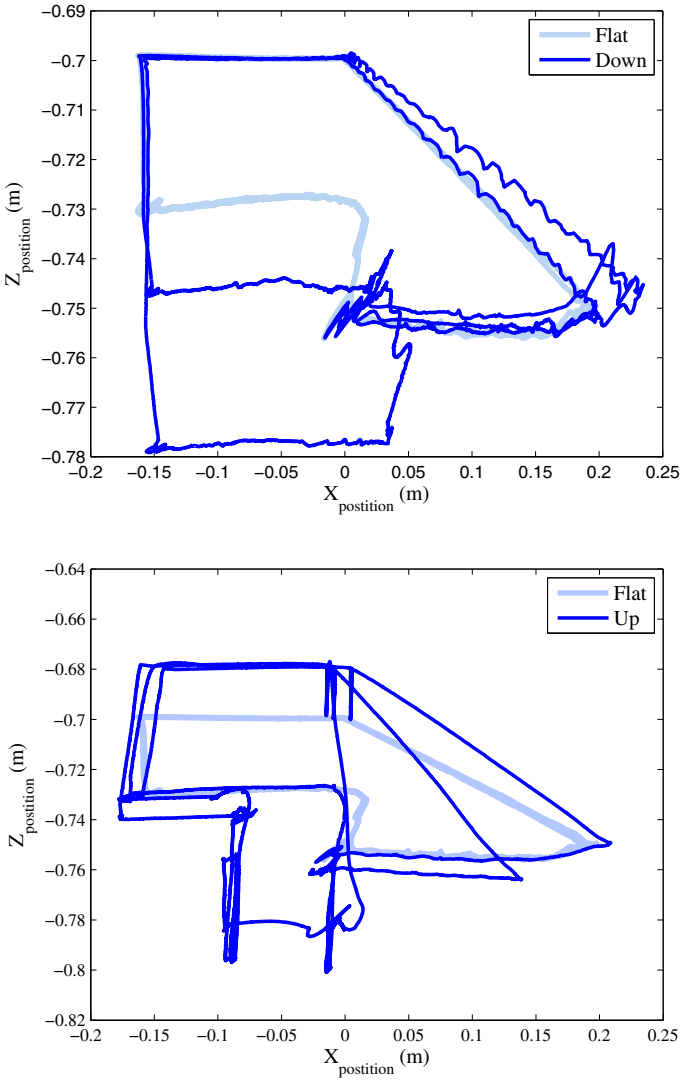


Fig. 9. Foot trajectories in the XY plane when the Atlas robot is walking on inclined planes. Downhill (top) and uphill (bottom) overlapped with the trajectories on flat terrain (light line).

In the right side of Fig. 9 the foot trajectory used when walking uphill is shown. As it can be seen in the figure, the step length is gradually reduced while the height of the starting point of the gait is increased. This is because as the ground is higher than when the foot's initial position it is forced to stop its movement (depending on the force threshold) before reaching the desired position. The increase of the starting point height is because if this height is not

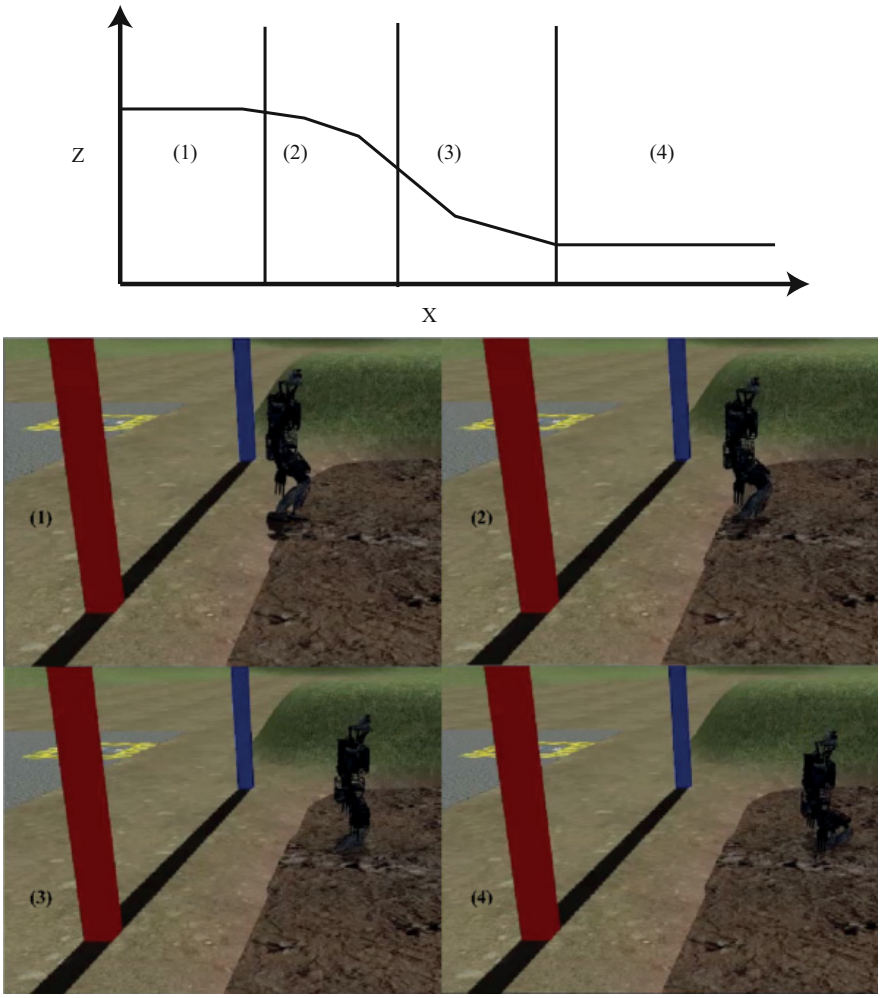


Fig. 10. Snapshots of the Atlas robot walking into a mud-pit descending a ramp and then walking on level ground

increased the robot would not be able to continue its path as eventually the next step will be out of the leg's workspace.

To further test the algorithm, we programmed the Atlas robot to descend a mud-pit which had a ramp with variable inclination followed by a flat ground. Fig. 10 shows snapshots of the Atlas robot walking into a mud pit descending a ramp and then walking on the level ground. In the top part of the figure is a graphical representation of the terrain profile of the mud pit. Notice that there are numbered areas; each number corresponds to a snapshot of the robot (bottom of Fig. 10 stepping on a determined region of the mud pit, finally it is important

to say that in the competition the slope angles are established randomly, though the angles of the region 3 are usually higher than the angles in region 2.

7 Conclusion

Legged robots are a good choice in rescue missions since they can help save lives especially when conditions are hazardous for the rescue team (e.g. Fukushima nuclear disaster), since we avoid risking one life to save another. Among legged robots, bipeds and more specifically humanoid robots are the most efficient choice if the disaster zone was designed for humans due to the anatomical similarities.

For a humanoid robot to navigate effectively in a disaster scenario it is necessary to have information of the terrain, this information is not normally available beforehand so it is necessary to obtain it through artificial perception. Nonetheless, sometimes the terrain might be occluded making perception useless. In this case the only alternative left is to use a purely reactive approach for locomotion.

In this work we have developed a reactive algorithm which is able to walk statically stable trough combinations of inclined plains and flat ground when the terrain is not known and perception is not an option. The algorithm is based on reaction forces and uses a force threshold to detect the ground, afterwards when contact is established it corrects the basic gait pattern to account for inclination of the terrain.

The method has been tested in simulation with the DRCSim software package and the Atlas robot model showing good qualitative performance.

Acknowledgement. This work has been performed by the SARBOT Team, composed by the University of Alcala, Polithecnic University of Madrid, University Carlos III of Madrid, and Spanish National Research Council (CSIC).

This work has been partially funded by the Spanish National Plan for Research, Development and Innovation through grant DPI2010-18702 which has funded personnel costs.

Mr. Juan Carlos Arevalo would like to thank the Spanish National Research Council and the Spanish Ministry of Economy and Competitivity for funding his PhD research.

References

1. Fuzzy logic in control systems: fuzzy logic controller. i 20
2. Belter, D.: Perception-based motion planning for a walking robot in rugged terrain. In: Kozłowski, K. (ed.) Robot Motion and Control 2011. LNCIS, vol. 422, pp. 127–136. Springer, Heidelberg (2012)
3. Davids, A.: Urban search and rescue robots: from tragedy to technology. IEEE Intelligent Systems 17(2), 81–83 (2002), doi:10.1109/MIS, 999224
4. gazebosim.org: DRCSim user guide, `http://gazebosim.org/wiki/DRC/UserGuide` (accessed July 1, 2013)
5. gazebosim.org: GAZEBO - about gazebo, `http://gazebosim.org/about.html` (accessed July 1, 2013)

6. Nagy, P., Desa, S., Whittaker, W.: Energy-based stability measures for reliable locomotion of statically stable walkers: Theory and application. The International Journal of Robotics Research 13(3), 272–287 (1994) doi:10.1177/027836499401300307
7. Gonzalez-de Santos, P., Garcia, E., Estremera, J.: Quadrupedal locomotion: an introduction to the control of four-legged robots. Springer, Berlin (2006)
8. Tsai, L.: Robot Analysis: The Mechanics of Serial and Parallel Manipulators. A Wiley-Interscience Publication. Wiley (1999)

ROS Methodology to Work with Non-ROS Mobile Robots: Experimental Uses in Mobile Robotics Teaching

Mariano J. Aznar, Fernando Gómez-Bravo,
Manuel Sánchez, J. Manuel Martín, and Raúl Jiménez

Dept. of Electrical Engineering, Computer Systems and Automation,
Higher Technical School of Engineering, University of Huelva,
Carretera Palos - Huelva, s/n, 21071 Palos de la Frontera (Huelva), Spain
marianojose.aznar@alu.uhu.es,
fernando.gomez@diesia.uhu.es,
{jmmartin,naharro}@uhu.es

Abstract. This paper presents a robotics platform for experimental and teaching purposes that allows control and remote access by using plain sockets (regular TCP/IP sockets) and websockets respectively. The system has been built from ROS middleware (Robot Operating System) and is capable of controlling multiple user operations. The robot hardware connected to sensors or/and actuators does not support ROS due to its features. This article focus attention on describing a work methodology to establish communication of the non-ROS robotic platforms and ROS.

Keywords: ROS, Sockets, Mobile Robot, Rosbridge.

1 Introduction

Nowadays, robots are not isolated systems, but interrelated complex systems composed of many different elements hardware and software. Generally, robots are included with more complex control architectures (e.g. industrial robots), or they have interaction with other systems (e.g. mobile robots). In a mobile robot several sensors (GPS, cameras, lasers, infrareds, etc.) and actuators (motors mostly) are included, and they allow both collecting and sending data, and at the same time receive orders from other systems.

Clearly, communications between systems is a key tool in a mobile robot. To perform this communication the robot needs set up a communication channel. In this context a middleware is a very powerful tool.

A middleware is a software that provides links between independent applications. This software helps an application to communicate with other software, network, hardware or even operating systems [8]. Using compatible interfaces, several components can exchange data through a communication channel provided by the middleware software.

Many development frameworks are available for robotics purposes, such as ROS, Player, CLARAty, Miro, MRDS, Marie, Orca, Orocos or RTT. However,

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_31,

most of them have been produced for a specific purpose, while the ROS middleware approach is more general and tries to integrate some of the above open source robotics libraries to be, nowadays, the more popular standard in robotics.

ROS is an initiative to assist the integration and cooperation between several robotics platforms, including software abstraction, messages between processes, and more [9] .

In this context, data availability and the capability of displaying them are very important matters. A mobile robot handles many data types received from several sensors (encoders, gyroscopes, GPS, laser, infrared, battery level), and accessing them in a simple way is a fundamental procedure. Also, because its nature, a mobile robot can be working far from the place where data are processed. ROS is an excellent tool to solve these problems. However, due to computational capabilities, or limitations of the operating system, all robots are not compatible with ROS. Then, an additional way to data accessing is needed. In this paper, it is supposed that the non-ROS system can support connections throw a TCP/IP architecture.

A web interface is a good solution in some systems because it allows access to a robotic system and the robot operations from everywhere with no dependency about the architecture. Therefore, a ROS system can be connected with a non-ROS system through websockets. Websockets are TCP/IP sockets that run from browsers connecting to an application server over http protocol. With this solution a multiuser and remote management of the robot can be built [7].

In order to perform this, in approach the rosbridge tool has been applied. Rosbridge is a Javascript based tool that provides a dynamic socket and websocket based access to the full capabilities of ROS [4].

On the other hand, the ROS and non-ROS connection problem can be solved in a different way by using plain sockets. Although there are tools based in TCP/IP sockets that can be useful for development or research purposes (ZeroMQ), regular TCP/IP sockets are used here because the fit better for the current approach due to their flexibility and easy implementation. So, a C/C++ methodology based in TCP/IP plain sockets to solve above problem is illustrated in this paper.

This article presents a study about connecting ROS and non-ROS systems, supporting connections by TCP/IP sockets, in the previous presented two ways, i.e., both rosbridge (websockets) and plain sockets. Also, a case of study in a teaching context is introduced and it illustrates how to apply these two techniques in a real situation. This paper is organized as follows: Section 2 is devoted to a formal presentation of ROS middleware, one of most famous middleware tools nowadays in robotics. Next, in Section 3 both methods to connect ROS and non-ROS systems are explained. First, in Subsection 3.1 plain sockets are exposed. Plain sockets are a more powerful and generic method in machine-machine interaction. Later, in Subsection 3.2 it will be show how websockets can be used to connect a ROS server with an web application through rosbridge. A websocket is an useful methodology to perform this human-machine interaction through browsers. Finally, section 5 presents some conclusions and future works.

2 ROS Features

Usually, the sensors reading and the actuators controlling procedure can be given by the following steps:

- Read data from encoders.
- Do the required calculations.
- Control the motors.

Hence, a first approach can be a whole application that performs all these steps. However, like it is known this is not the best way to develop applications.

A most efficient way to implement this could be cutting this program in small pieces with basic functions, and here, a middleware could play a very important role, improving code organization, maintenance and efficiency.

Robot Operating System (ROS) is an open source middleware for robotic software development, introduced primarily by Willow Garage company [10]. ROS tries to be an operating system for service robotics. It includes standard operating system services such as hardware abstraction, low-level device control, message-passing between processes, package management and even more [2].

ROS have two importants parts: the ROS system and the ROS supplementary packages that provides new funcionalities to ROS. These packages are supported by the ROS community and they implement functions supporting to robotic applications such as localization, perception, SLAM, etc. [11].

ROS works with a few of basic concepts, most importants of them are nodes, topics and services. A node in ROS is an executable file that is used to communicate with other nodes [3]. Then, ROS is compound of nodes because every single process in ROS is a node.

To exchange information, nodes can use topics (asynchronous communication) or services (synchronous communication). With topics, a node can publish or subscribe to a topic. The function of a topic is to store messages to be available in ROS. Topics are published (storing data) or subscribed (recovering data) by a node. The communication procedure between nodes is illustrated in Fig. 1.

When a node needs exchange information with another node, it publish a message in a topic. Then, this topic is available in ROS to be subscribed by another node. When a node needs recover the message, it subscribes the previous topic. This is ROS procedure to exchange information between nodes.

3 ROS and Non-ROS Communication

Although several ROS networked systems can be in different places, they can be fitted to work in coordination, acting like a whole single system.

Nevertheless if non-ROS systems (systems that are not compatible with ROS but whose architecture supports TCP/IP connections) are used, can be necessary another way to connect ROS systems with the non-ROS systems. In this approach, two ways of solving this problem are presented: plain sockets and websockets.

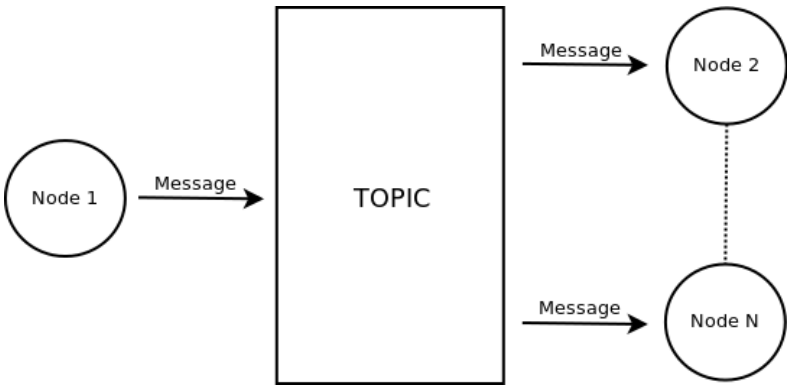


Fig. 1. Communication between nodes using a topic

3.1 Regular TCP/IP Sockets

Using regular sockets a system can exchange data with another one. For example, a producer system, building or collecting data needs to share them with an operator system. This procedure can be performed by TCP sockets.

A plain socket is a communication channel between applications that are running at the same time in different networked systems, or even in the same system. Plain sockets are a powerful and generic solution, because they do not require work under http protocol or browsers, they run over TCP/IP.

Sockets allow to build client-server architectures. First, a program starts the communication (client), and another program is awaiting this communication (server) [6].

The TCP socket implementation in UNIX environment is as follows. First a socket is necessary to be created and binded to an address, i.e., a transport address has to be assigned to the socket. These steps are the same in the server and client sockets. Calling the system function *listen()*, the server socket sets in listen mode to be ready to accept clients connection by the previous created socket. To perform this communication the system function *accept()* is used, so the socket will be capable of accepting incoming connections. Now, if a client connection is incoming, the server will create a new socket and will manage operations with it. Finally, all the created sockets have to close with *close()* call.

A possible pseudo-implementation resume for a simple client can be seen in Algorithm 2.

Algorithm 2. Server TCP/IP socket .

```
...
socket() // Create a server socket
bind() // Name the socket
listen() // Set the socket in listen mode
```

```
// Loop until end_condition is active
while(end_condition)
accept() // Wait for a client request connection
read() // Read data from client
write() // Send data back to client
close() // Close the client connection
endwhile

close() // Close server socket
...
```

In the other side, the client have a more easy coding. First, the client have to create a socket and, after that, the client only needs connect to the server with the connect() call.

Algorithm 3. Client TCP/IP socket.

```
...
socket(IP_Server) //Create a socket with the server IP address
connect() // Connect to server
write() // Send data to server
read() // Read data back from server
close() // Close the socket
 ...
```

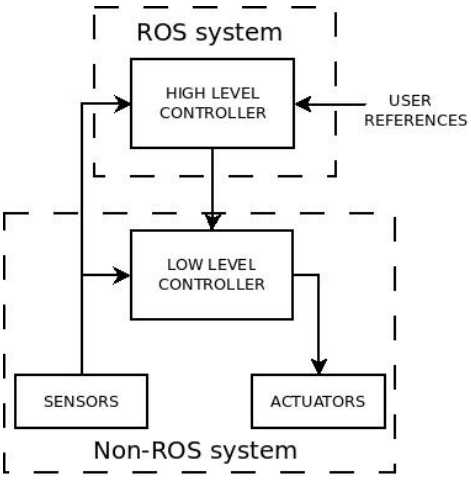


Fig. 2. Non ROS-ROS communication process

Often concurrency is needed in a bidirectional communication, such as listening encoders and controlling actuators. In Unix systems concurrency is performed with *fork()* function and allows multithreading, i.e., a process can generate another equal process. When a process forks, creates another instance of the parent

process. So, two instances of the same process are running, but doing different operations. The first process is named 'parent process' and the second will be the 'child process' or 'son process'.

The concurrency pseudo-implementation for the Server socket can be seen in Algorithm 4.

Algorithm 4. Concurrency in Server TCP/IP Socket.

```
function concurrent_Server()
   begin
      // Declare variables

      socket() // Create a server socket
      bind() // Name the socket
      listen() // Set the socket in listen mode

      // Loop until end_condition is active
      while (end_condition)
         accept() // Wait for a client request connections

         if (fork) // Parent process
        ...
            close() // Close the client connection
         else // Child process
        ...
        close() // Close the client connection
         endif

      endwhile

      close() // Close server socket
      ...
  end.
```

This paper presents an application with a non-ROS concurrent server sending data to a ROS client node. This node will publish the data in ROS, and other nodes will subscribe to operate it. When the operations are finished, the ROS node client will send orders back to the non-ROS concurrent server. Finally, the non-ROS system will give orders to the motors according to the received commands (Fig. 2).

3.2 Websockets (Rosbridge)

A websocket runs from a browser and establishes a simple bidirectional and full-duplex communication channel allowing connect to an application server through

http protocol running over a only TCP/IP socket. A websocket fits primarily in real time web applications. The websocket establishes a persistent connection between the browser and the server, and then the communication is open until client or server closes it.

In the client side, a websocket can be implemented in any of the supported web browsers. A minimal and simple implementation of a websocket and its funcions is showed in Algorithm 1.

Algorithm 1. Websocket client.

```
...
// Create a websocket variable
var ws = new WebSocket('/server_host');

// LISTENERS

// Open the communication with the server and send some data
ws.onopen = function () {
  ws.send('Client is connected');
};

// Log errors
ws.onerror = function (error) {
  console.log('WebSocket Error ' + error);
};

// Close the communication with the server
ws.onclose = function () {
  alert('Client is closed');
};

// Get messages from the server
ws.onmessage = function (e) {
  alert('Message from server: ' + e.data);
};
...
```

To send data available from a ROS server to a web application client through websockets, it is necessary to build a server application that provides the data to the client. In this way, rosbridge tool can be used.

Rosbridge was built by Brown University and allows to a remote client to publish messages, subscribe to topics and invoke services that are running in the ROS server [5].

Rosbridge is a multilingual and multiplatform protocol that allows connect non-ROS clients with a ROS server [1], providing a websocket for clients to communicate over http protocol. Due to the fact that it is a multilanguage platform,

rosbridge clients can be written in any language that supports websockets. Moreover, to establish the communication in a simple way, rosbridge provides a Javascript client named 'ros.js' that solve most of the communication tasks with ROS in the client side.

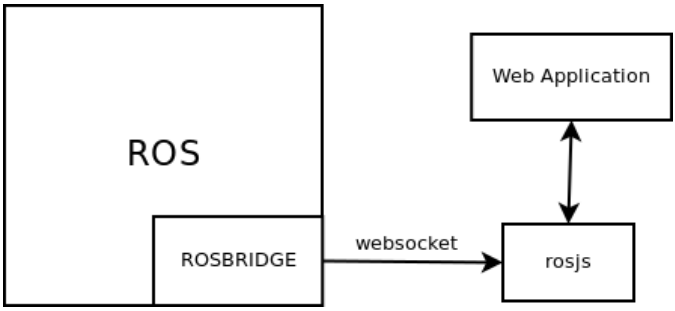


Fig. 3. Rosbridge communication

Rosbridge allows a human-machine interaction by web interface, doing easy the server-client communication. Also, because of the universality of browsers and the http protocol, a user can remotely connect to the server from everywhere. So, in the client side a browser can show images from a remote camera, path information, and several data collected remotely by the robot sensors.

4 Case of Study

This work illustrates a case of study based on controlling a mobile robot in a teaching context about robotics and, particullarly, in path planning and motion control. This robot represents a common resource for the students and it operates in a lab with several computers. In order to simplify, only motors and encoders will be taken into account. So, the robot has four motors and four encoders (one by wheel).

Students make experiments with the mobile robot, only one student at time. Experiments consist in generate a path, send it to the robot, and check if this path is suitable for the robot taking its position at each time.

The mobile robot has a network device and WIFI antenna, and is networked with the lab computers. Then students can connect with the robot from lab computers using SSH (Secure Shell protocol for secure data communication). To operate and take the results from the robot in a right way, students have to be sure that nobody is using the mobile robot.

When a student is over, she/he notifies their partners and so the mobile robot can be used by another student.

In order to perform this process in a more efficient way, a web application in which students can login, has been built. This application allows login into the system, make queues to use the mobile robot, trace the students activity, and

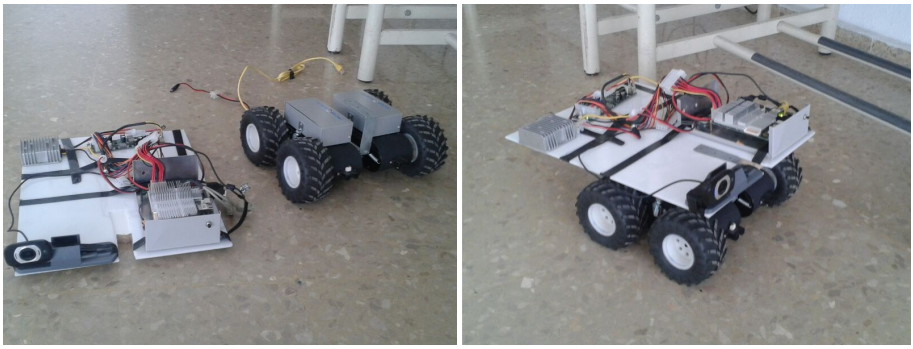


Fig. 4. a) ROS server and the mobile robot. b) The whole system working together.

share results. Also, this application works in a web environment, and students can do the above practice exercises from different terminals.

The whole system consists in a ROS system connected by WIFI to the robot. This connection is performed by regular sockets and forms the low level. This ROS system acts like a server where the students can connect to extract the current position of the robot and send paths. The connection between lab computers and the ROS system is performed by websockets.

Then, the whole application can be divided in two parts. The first one is the data extraction, from encoders, and send speeds to the motors, i.e., communication between the ROS system and the robot.

Second part is the communication between the ROS system and lab computers to, first, take the current position of the robot, calculated by the ROS server in the previous step, and send it to the web application, and second, send the planned path to the ROS system so that it calculates the speeds to send to the motors in the robot.

For a better understanding, a schema of the global communication is presented in Figure 5.

4.1 Communication between Mobile Robot and ROS

To accomplish this part a ROS middleware is used. However, although the operating system of the mobile robot is UNIX, it does not meet the requirements to install ROS. This problem can be solved using plain sockets as follows: on the one hand, the mobile robot becomes in a concurrent server and, on the other hand, the ROS computer is the client.

Using plain sockets the mobile robot can send encoders information to the ROS computer for processing it, and after that the ROS computer sends the wheels velocity commands to the mobile robot (Algorithm 1).

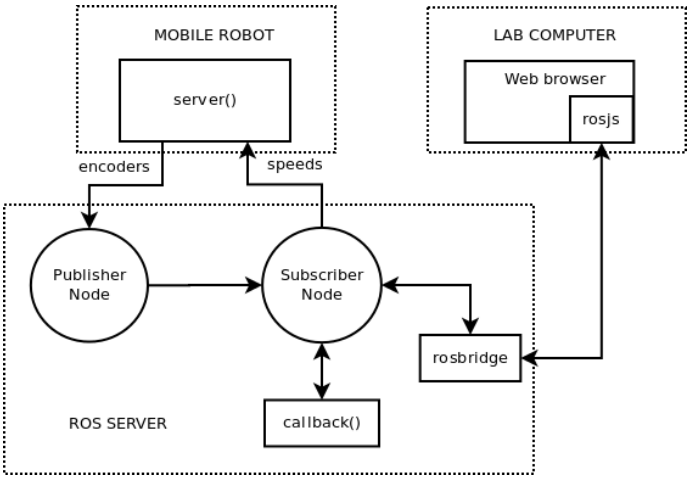


Fig. 5. Global communication schema

Algorithm 1. Concurrent Server.

```
function Server()
   begin
      // Declare variables
      // Init server socket
      ...
      while (1)
         // Wait for a client connection
         if (fork) // Parent process
            // Read data from encoders
            read(encoders);
            // Send sensor readings to socket
            send(encoders, socket);
            // Close the client connection
         else // Child process
            // Read speeds from socket
            read(speeds, socket);
            // Send speeds to the motors
            send(speeds);
            // Close the client connection
         endif
      endwhile
      ...
  end.
```

This algorithm can be explained as follows. First, the server socket is launched, so clients can connect to it. When a client has connected, if it forks then it means

that the client wants to receive the encoders information, then the server reads data from encoders and sends it to the client. Otherwise, the child process closes the socket to sends speeds to the server, so that, server reads speeds from the client and sends them to the motors.

The algorithm to connect with the concurrent server from a client is showed in Algorithm 2. This algorithm will be a ROS node and publishes in ROS the data received from the server.

Algorithm 2. Publisher Node.

```
function publisherNode()
   begin
      // Declare variables
      // Init ROS (ros::init)
      // Declare a NodeHandle (ros::NodeHandle)
      // Declare a Publisher element (ros::Publisher)
      ...
      while (ros::ok())
         // Open a connection with server socket
         // Read data from socket
         read(data, socket);
         // Prepare the message to publish it
         message = data;
         // Publish the message
         publish(message);
         // Close connection
      endwhile
   end.
```

So, first, this ROS node opens a connection and reads the encoders data from the server. Next, it publishes a message with this data in ROS through a ROS topic, so that other ROS node can connect to this message by the topic subscription to read it.

Then, a subscriber ROS node is necessary in order to process the data. This node is shown in Algorithm 3, and its function is taking the data published in ROS by Algorithm 2 and performs the speed calculation. So, the mobile robot can follow a path previously planned.

Algorithm 3. Subscriber Node.

```
function subscriberNode()
   begin
      // Declare variables
      // ...
      // Obtain current position
      (x, y) = position(...);
      // Read the path
      path = read(...);
```

```
      /* Calculate nearest point in the path
         from current position */
      p = nearest_point(path, [x, y]);
      ...
      // Init ROS (ros::init)
      // Declare a NodeHandle (ros::NodeHandle)
      /* Declare a Subscriber element
         (ros::Subscriber) using callback
         function */
      subscribe(topic, callback);
   end.
```

The above subscriber node needs a special function named 'callback function' (Algorithm 4). This special function is, indeed, which performs all tasks. The function of the subscriber node is used only for initializing variables and calling the callback function.

Algorithm 4. Callback function.

```
function callback(msg)
   begin
      // Declare variables
      // Open a connection with server socket
      ...
      if (final)
         send(STOP, socket)
      else
         // Calculate curvature
         c = curvature(...);
         // Obstain new speeds
         s = new_speeds(...);
         // Send speeds to socket
         send(s, socket);
         // Calculate odometry
         o = odometry(...);
      endif
      // Close connection
   end.
```

Here, all tasks are performed. This algorithm calculates curvature, obtains the new speeds to send to the server and performs the odometry. Finally, if the stop condition is accomplished, the client sends to server the STOP signal to stop the robot.

4.2 Communication between ROS and the Web Application

In order to use rosbridge in a ROS server, first it is necessary to set the port over the communication channel to get it established. The next step consist on launching it over roscore to open the communication with the remote browsers.

This communication is very simple. To use rosbridge it is necessary only a browser. Next, in Algorithm 5, a pseudo-HTML code that perform the communication with the rosbridge server is presented.

Algorithm 5. Example of HTML code using rosbridge.

```
<html>
   <head>
      ...
      <script type="text/javascript" src="ros.js"></script>
      <script type=‘‘text/javascript’’>
         // Declare variables
         // Establish a rosbridge connection
         var con=new ros.Bridge(ROS_IP, PORT)
         // Create a message compatible with
         //   the data that go to receive
         var setrosmessage = function(msg)
         {
            var divelement =
            document.getElementById("rosmessage");

            divelement.innerHTML = msg;
         };
         // When the page opens, register the
         // callback and associate it with the
         // topic
         con.onOpen = function()
         {
            var cback = function(msg)
            {
               // Draw the current position of the mobile robot in the map
            };

            // Subscribe to the position topic
            con.subscribe(cback, POSITION, ...);
         };
      </script>
      ...
   </head>
   <body>
      ...
      <!-- Draw the map -->
      <img src="map.png">

      <!-- Create the object to display topic messages >
      <div id = "rosmessage">
         ...
      </div>
      ...
   </body>
</html>
```

In this html code, first "rosjs" is included in order to prepare the communication with rosbridge in the ROS server. Next, it launches a rosbridge connection with the IP address and the port of the ROS server. It creates a message data type to receive the current position from websocket and it starts to receive the current position to draw the path followed by the robot in the browser.

Figure 6 shows a screen capture of the HTML page used to track the robot. This screen shows the path tracked by the robot over the map scenario. Also, in the right corner, it displays a webcam image taken by the robot camera and below this, it prints the encoders information, the position in x and y and the angle of the robot.

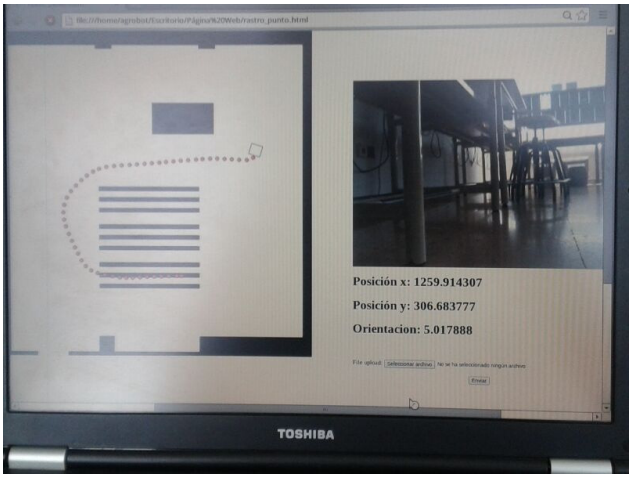


Fig. 6. Web application to the robot tracking

5 Conclusions

In this paper, a study about communication between ROS and non-ROS systems has been presented.

With plain sockets, ROS and non-ROS systems can be connected. This technique is usually applied to work with non-ROS robots when they do not support web enviroment. In this way, applications can receive data from robot, such as sensors data, battery level, etc. and give orders to the robot to move the wheels from a ROS system.

On the other hand, websockets allow communication between ROS systems and non-ROS remote clients. This is used to build web applications which allow to access the robot facilities, subscribing or publishing nodes in the ROS remote server.

Also, a case of study about a robotics platform for experimental and teaching purposes has been presented. It allows remote access by using a web interface that is connected to a ROS enviroment through rosbridge.

The ROS system did the high level control (odometry, localization, perception, etc.) and the non-ROS system did the low level control (communication with sensors and actuators). Finally, the ROS connection through websockets was used to establish the human-machine remote interface.

In future works, a more general solution and possibilities with other kind of robots will be research. Also, the system will be open to allow the remote access out of the laboratory.

References

1. Alexander, B., Hsiao, K., Jenkins, C., Suay, B., Toris, R.: Robot web tools. IEEE Robotics and Automation Magazine, 20–23 (2012)
2. Cousins, S.: Welcome to ROS Topics. IEEE Robotics and Automation Magazine 17, 13–14 (2010)
3. Cousins, S., Gerkey, B., Conley, K., Garage, W.: Sharing Software with ROS. IEEE Robotics and Automation Magazine 17, 12–14 (2010)
4. Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: ROS for non-ROS users. In: Proceedings of the 15th International Symposium on Robotics Research (2011)
5. Crick, C., Jay, G., Osentoski, S., Jenkins, O.C.: ROS and Rosbridge: Roboticists out of the loop. In: 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 493–494 (2012)
6. Kozierok, M.: `http://www.tcpipguide.com/`
7. Lee, J.: Web Applications for Robots using rosbridge. Brown University (2012)
8. Qilin, L., Mintian, Z.: The State of the Art in Middleware. In: 2010 International Forum on Information Technology and Applications (IFITA), vol. 1, pp. 83–85 (2010)
9. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Andrew, N.: ROS: an open-source robot operating system. In: Proceedings of the Open-Source Software Workshop of the International Conference on Robotics and Automation (2009)
10. Garage, W.: Robotic Operating System (ROS), `http://www.willowgarage.com/pages/software/ros-platform`
11. Robotic Operating System (ROS), `http://ros.org`

Automatic Control of a Large Articulated Vehicle

Hector Montes[1,2], Carlota Salinas[1], Javier Sarria[1],
Jesús Reviejo[1], and Manuel Armada[1]

[1] Centre for Automation and Robotics CSIC-UPM
Ctra. Campo Real Km 0.220, Arganda del Rey, 28020 Madrid, Spain
{hector.montes,carlota.salinas,javier.sarria,
jesus.reviejo,manuel.armada}@car.upm-csic.es
[2] Facultad de Ingeniería Eléctrica, Universidad Tecnológica de Panamá
Via UTP, Panama City, Panama
hector.montes1@utp.ac.pa

Abstract. This work presents the development of the automatic control of an articulated bus. It is describes the most relevant concepts of the architecture of perception and actuation implemented on a large wheeled vehicle, Volvo B10M articulated bus of 18.75m in length. In this work, some experimental results of strategies for lateral and longitudinal control of the bus are shown, when it travels on a guide rail through a guiding mechatronics box. Additionally, this article presents the concepts of the guiding rail detection and obstacle detection by means of two Lidar systems for the test lane detection in advance, and for the pedestrians' safety, respectively. Several experimental results are presented in this manuscript.

Keywords: Perception system, actuation system, Bus Rapid Transit, Intelligent Transportation System, lateral control, longitudinal control, obstacle detection, Lidar system.

1 Introduction

The development of autonomous vehicles has been subject of important research activities, where the control systems have a relevant role [1-5]. Considering a particular perspective of this concept, there is a growing interest to develop innovative mass transport systems, safer, more reliable and adequate comfort for users in the field of Intelligent Transportation Systems [2-3], [4], [7-9]. Meanwhile, for the development of this type of transport, there have been several studies of the dynamic control of various vehicles, which have been researched and documented over the past two decades [10-14]. Additionally, several designs and experimental results of lateral and longitudinal control strategies are documented in the literature [14-16]. On the other hand, Bus Rapid Transit system is a transport system used in some cities, which tend to be very promising, is being very popular, and in which, its automation is highly relevant [2], [17].

Firstly, in this manuscript is presented the description of the whole system, which includes the perception and actuation architectures for the automation of an

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_32,

articulated bus, Volvo B10M of 18.75m in length. In addition to some design considerations of the track in order to that this large vehicle moves properly on it. The main objective is to evaluate experimentally several control algorithms and control strategies designed for automatic driving of this articulated bus which emulates, in some manner, the train driving, but with better performance. For this purpose, the system has two on board computers, a guiding mechatronics box, and system for the lane early detection and the obstacles detection. One of the on board computers is responsible for the acquisition and processing of the data from sensors installed on the mechatronic box and the bus chassis, as well as control, in real time, the actuators installed on the bus. The second on board computer is in charge of the acquisition and processing of data from the Lidar systems, which performs the guide rail detection in advance, and the detection of objects to prevent collisions and pedestrians for safety.

Secondly, strategies for the lateral and longitudinal control are presented in this document, which have been applied in the articulated bus. Several experiments were carried out when the bus is moved on the test lane through an embedded guide rail. The main objectives are to maintain a constant speed and maintain comfortable driving, so that the bus oscillations are imperceptible during its travel by the test lane.

The guide rail detection in advance and the static and dynamic objects detection by means of Lidar systems are presented in thirdly. The first concept, guide rail detection, is used to send information of the localization of the guide rail, some meters ahead, toward the control system. These data are implemented in the external loop in the lateral and longitudinal control algorithms. The second one, objects detection, is used to prevent collisions with static objects on the track, besides to observe the movement of the pedestrians approaching the test lane to safeguard their security.

There is a dedicated area where it has built an experimental test circuit of 385 m in length, which contains an embedded guide rail, with specific dimensions, at the facilities of the Centre for Automation and Robotics - CAR (CSIC-UPM) in Arganda del Rey, Madrid, Spain [18]. This infrastructure was rebuilt to investigate new transport systems. The articulated bus used as experimental platform has a mass of ~15000 kg, which is variable depending on the number of its occupants, whose maximum technical capacity is ~29000 kg. This variability of the vehicle mass is taken into consideration to carry out the control strategies.

The transport system presented in this paper has high performance, lower cost and lower environmental impact than conventional trains. It is based on the tracking of a guide rail by means of a guiding mechatronics box (central guide) and a laser scanning system, with high performance, for the early detection of the guide rail. The central guide allows the convoy formation, which promotes high load capacity of this system. This system consists of pneumatic wheels which provide comfort to its occupants. It will be a versatile system that also can function without external guiding system as a normal bus. Therefore, this automatic articulated bus provides many of the principal advantages of either the conventional bus or the train, reducing many of their disadvantages.

2 Description of the System

The Centre for automation and Robotics (CAR-CSIC) has an experimental platform, fully instrumented, to provide a reliable starting point for developing various automatic control researches in articulated busses, among other automatic mobile systems [5], [18], [19]. Fig. 1 shows the most important subsystems that comprise the control architecture for the automation of the Volvo B10M articulated bus. Mainly, the system architecture comprises of two on board computers communicated by TCP/IP protocol; the guiding mechatronics box instrumented with several sensors; actuators to govern the throttle, brake and steering of the bus; Lidar systems; vision system; DGPS; and other sensors installed on the bus chassis.

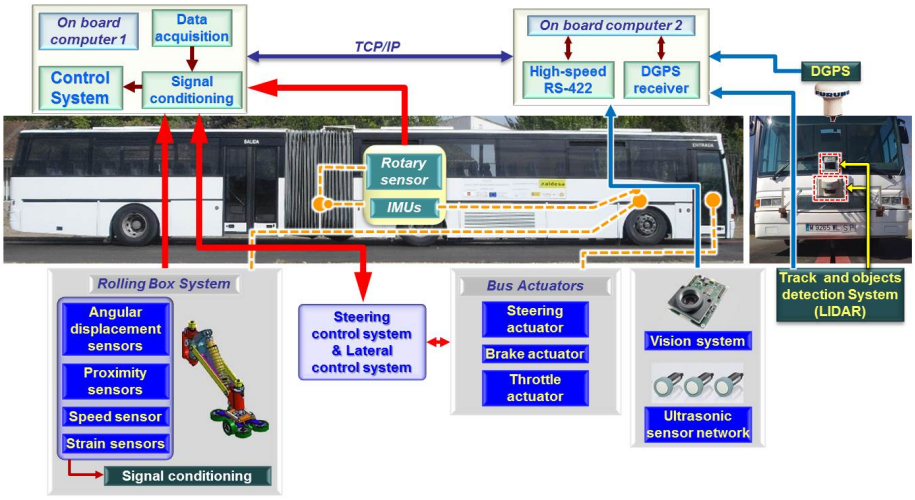


Fig. 1. System architecture for the automation of an articulated bus

The main idea is to control the bus movement on a dedicated test lane which contains a guide rail embedded in it. The first system that detects the guide rail is the guiding mechatronics box, which is an improved design of the original concept of "Rolling Box System - RBS" [20]. This system is installed on the front axle of the bus. The Lidar system is the second system used for the detection of the guide rail. This system detects the slot of the test lane in-advance to provide to the control system the necessary data to improve tracking of the circuit. The detection distance of the slot is variable, and it can be adjusted between 1-8 meters to have a high accuracy of measurement of the guide rail [21]. Fig. 2 shows the block diagram of the general control system for the articulated bus automation. The control system, as well as receive data from the aforementioned systems, has also inputs from the inertial measurements units and the DGPS data. Additionally, the dynamic model of the vehicle is

considered. As a result, the control system regulates the steering and the brake by means of electric actuators (24VDC servomotors), and the throttle driver with an analogue signal.

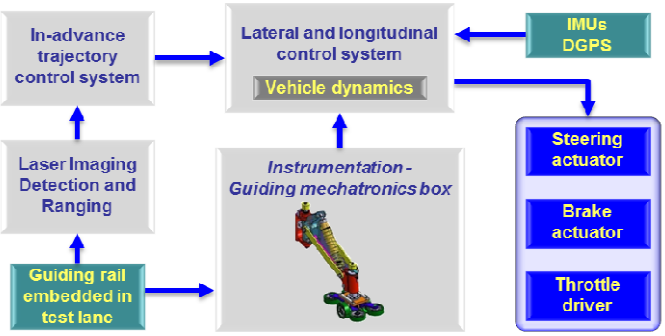


Fig. 2. General control system diagram

The mobile experimental platform is complemented with a special test lane built within CAR (CSIC-UPM) facility at Arganda del Rey, Madrid, Spain [18]. The test lane facility has been design according to the dimensions and the turn characteristics of the articulated bus. The idea is to avoid the rear outswing of the bus beyond the kerb when it moves by the track corners, and also to perform experiments in curves with the smallest radius of curvature possible of the circuit. Fig. 3 shows some details considered for the design of the test lane taking into account the rear outswing of the bus, the radii of curvature, the guide rail and the position of the front axle centre of the bus. The vehicle trajectories and the guide rail are compared in order to guarantee the suitable layout of the test lane. The test circuit has 385 m in length and seven curves with small radii of curvature. The smallest radius of curvature is 10 m, whose curve has a length of 41.58 m, approximately. The longest straight trajectory is about 112.25 m.

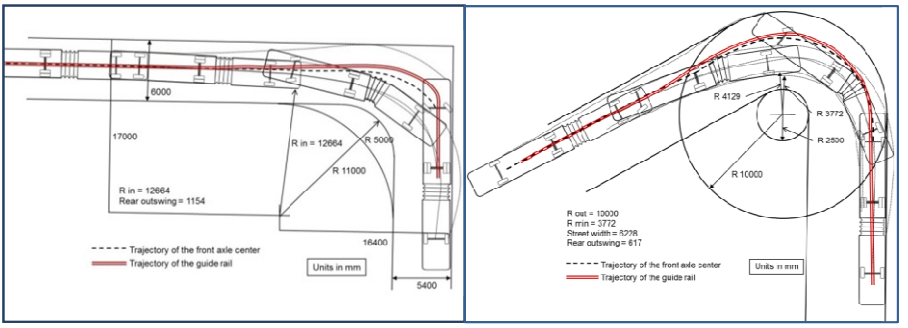


Fig. 3. Design considerations for building the track

The first sensing system used to provide feedback the bus position to the control system is the guiding mechatronics box. This system takes as reference the guide rail. The mechatronics box is fully instrumented for this purpose. In addition, this mechatronics box has been manufactured in stainless steel in order to provide high bus fixation to the guide rail to avoid derailment of it, in unforeseeable circumstances, which provides an added value of security to the system. The data which is sent from the guiding mechatronic box to the control system are obtained from the following sensors: two optical incremental encoders, inductive proximity sensors, strain gages and encoder for measuring the forward speed of the bus. The two encoders are high resolution and are placed in each degree of freedom of the mechatronics box on the azimuthal plane. These encoders provide the lateral displacement of the bus around the slot of the guide rail. The inductive proximity sensors are used for the process initialization, aligning the guiding central box with the advance axis of the bus. The strain gauges provide the signals related with the small deformation of the central box structure. This information can be used to determine the start of blocking in the guide rail or possible derailments. The velocity measurement is performed by means of an encoder installed in one wheel of the guiding mechatronics box. Fig. 4 shows two view of the guiding mechatronics box installed on the front axle of the bus. The top view is obtained through a hole on the bus floor, and the front view is obtained from a location for servicing the system.

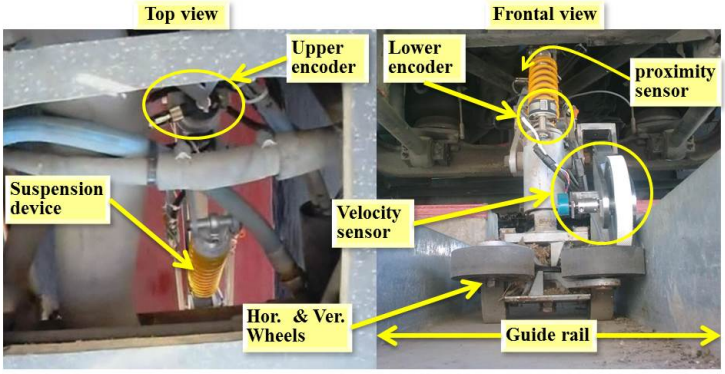


Fig. 4. Guiding mechatronics box installed on the front axle of the bus

The laser imaging detection and ranging is other sensory system used to detect in advance the slot of the guide rail in this project. With this sensing system the articulated bus can "observe" the guide rail with respect to the current vehicle position. During the experiments, the guide rail detection is from 3 to 8 meters ahead of the bus, to guarantee detection of the track slot, with sufficient accuracy, which is 60 mm. This optical sensor provides the necessary data in advance to the control system, so that the bus steering follows the trajectory, also considering the data sent by the instrumentation installed in the guiding mechatronics box. The laser scanning system

used in this work is a Laser Measurement System for outdoors LMS-221. On the other hand, another laser measurement system is used, the LMS291. Its function is to detect any objects, whether static or in moving, into the bus trajectory. The right side in Fig. 1 shows the two laser systems installed on the front of the bus. The one on the top is the LMS-291.

3 Control System

The control system for the automation of the articulated bus is divided into two control subsystems, that is, the longitudinal control system and the lateral control system of the vehicle.

The longitudinal control for the articulated bus Volvo B10M is carried out in order that the vehicle tracks the test lane with a speed set-point, previously adjusted in the control process. This velocity is relatively low due to the curvatures of the test circuit. The average speed for the circuit composed by seven curves with small curvature radii is 2.5 m/s, approximately. However, at longest rectilinear path (~112 m), the vehicle speed can reach about 6 m/s. They are not very high speeds, but for our research and the circuit layout is considered sufficient.

Fig. 5 shows the structure of longitudinal control implemented in the articulated bus to control its speed. The desired speed is relatively low so that the bus can follow the test lane properly. The control system acts on the accelerator driver, which has an action range of 0 to 5 VCC, instead of acting on the throttle. The acceleration range is limited in order to protect the accelerator driver, and at the same time to avoid exceeding maximum desired speed beyond the limits uncontrolled for this work.

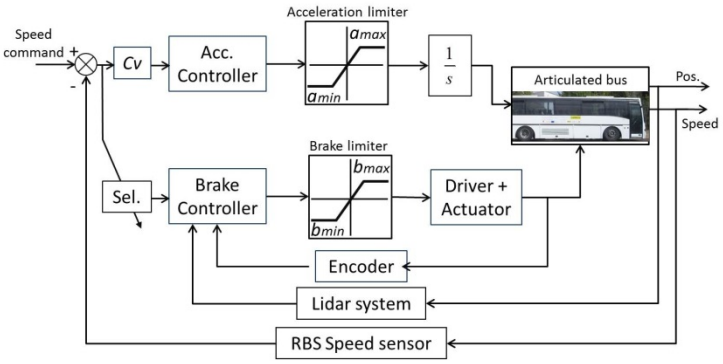


Fig. 5. Structure of longitudinal control

When the speed of the bus exceeds the desired speed, then the output of the accelerator controller is set to zero, and the brake controller starts to work. The brake controller has minimum and maximum limits, which determine its workspace. Within these limits, the controller generates a saw-tooth signal to command the position the

servo-actuator of the brake pedal. This causes the brake subsystem develops a gradual deceleration of the bus (depending on the initial conditions). When the speed decreases to a pre-set threshold, the brake subsystem stops, and starts the acceleration control subsystem. The speed used in the feedback loop is acquired from the speed sensor installed in the guiding mechatronic box, whose resolution is greater than that provided by the vehicle. The position measurement ahead of the bus is acquired by the Lidar system. The data from this system are observed by the brake controller to operate in the case of find objects in its field of view.

Fig. 6 shows the structure of lateral control implemented in the bus. In this control system must take into account the physical limits of the steering wheel, which is ±900 degrees, approximately. The rotation of the steering column corresponds to an angular displacement of the front wheels about of ±43°. The main feedback loop is realized by means of angular position sensors located on the central guide structure (see Fig. 4). This mechanical box has two degrees of freedom. One is located between the front axle of the bus and the clamping arms of the rolling box (upper dof). The other is on the bottom of the guide box (lower dof), located within the rail (lower dof). Both degrees of freedom are on the azimuthal plane, and they are instrumented with high accuracy optical encoders.

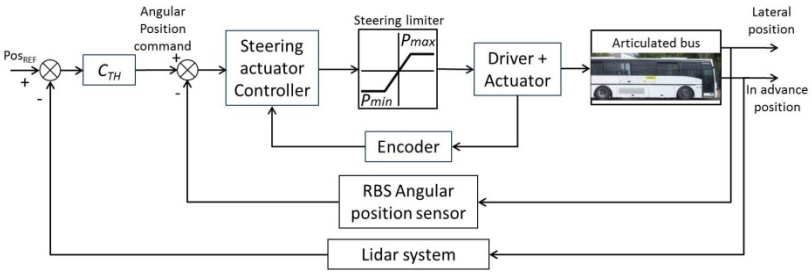


Fig. 6. Structure of lateral control

The control system continuously monitors the movement of the upper dof, which should not exceed ±45° with respect to the longitudinal axis of the bus, in order to avoid stressing the torsion springs that are installed there. Additionally, it controls the movement of upper dof to be within these boundaries, which causes sudden lateral movements are reduced, and thus prevents undesired blockages of the system and possible bus derailments. The maximum length of the guide mechatronic box (0.5 m) on the front axle of the bus causes the experimental tests are carried out at low speeds.

4 Laser Perception System

The laser perception system comprises two laser devices Lidar (SICK LMS221 and SICK LMS291), positioned on the vehicle front, and longitudinally aligned by their

centres (see Fig. 1). These devices perform the observation forward and downward, at an angle of - 6 ° for laser_objects (range 0 ° to 180 °, resolution 0.5°), and -26 ° for the laser_rail (range 40 ° to 140 °, resolution 0.25°) with respect to imaginary horizontal line. Both devices are connected by the communication interface Quatech High-Speed RS-422 to second on board computer. This interface allows reaching data transmission 500 KBaud an average sampling frequency of 75 Hz.

The high accuracy detection of the guide rail slot is considered for the development of reliable guiding of the vehicle. Some important considerations were taken into account in the design of the laser perception system. To avoid out of range measurements, the ahead observation distance L_h is a variable that depends on the dimensions of the test lane and the geometry of the guide rail. In the first experiments, the distance range used was $1900\ mm < L_h < 5000\ mm$. Another factor taken into consideration was the data distortion acquired from the rail, produced by the perspective of images and the circuit design.

This work presents a hybrid algorithm for the lines extraction, inspired by Split-and-Merge algorithm and Incremental algorithm [22]. With this hybrid algorithm, a method for detecting deformable lines edges from low resolution data has been proposed with the aim of detecting the central point of the guide rail slot, which is described by two control points (see Fig. 7). One of them is located in the centre of an imaginary line on the top slot (circle in Fig. 7), and the second one on the bottom centre of the guide rail (star in Fig. 7).

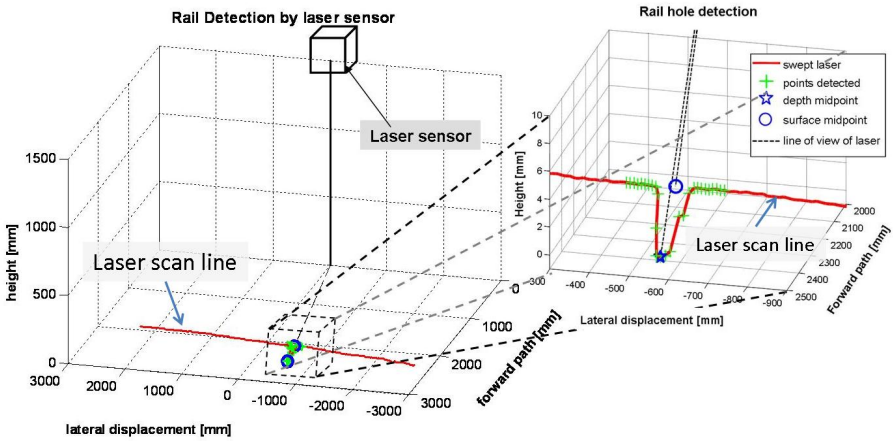


Fig. 7. Representation of the guide rail slot using the laser scanner data

Since the goal is only to detect the slot (top and bottom) trajectory of the guide rail and not the scene reconstruction, the input data can be simplified to two-dimensions. The procedure of the proposed hybrid algorithm is described in Table 1.

In this work, using the two control points (P_{top} and P_{bottom}) for each sample, collected due to the combination of the Lidar sensor and the guide rail geometry, the curvature of test circuit is computed. In order to improve the current active automatic control for vehicle guidance, a curve anticipation approach using a fuzzy logic classifier is introduced. This method allows imitating the human driving behaviour (notwithstanding the obvious differences) and it has been well-tested to low speeds. The goal is to identify whether an observed segment of the guide rail is curved or not. It is assumed that in every classifying stage, the observed segment dimensions are such that the steering angle and the vehicle velocities do not change within the segment boundaries. The segment at least must comprise 10 samples. The input and output variables of the fuzzy classifier are detailed in Fig. 8.

Table 1. Description of the hybrid algorithm for the line extraction

Pseudocode of the proposed algortihm	
1	Initial: Sliding window of N points. Compute maximum local point P_{max}
2	Split N from each extreme of P_{max} and start to construct lines from n points
3	If consecutive lines has the same line model merge collinear/overlapped segments
4	If both extremes fit to the same model (std and local mean) and the segment of P_{max} is bigger than threshold, possible object, else continue (go to 1).

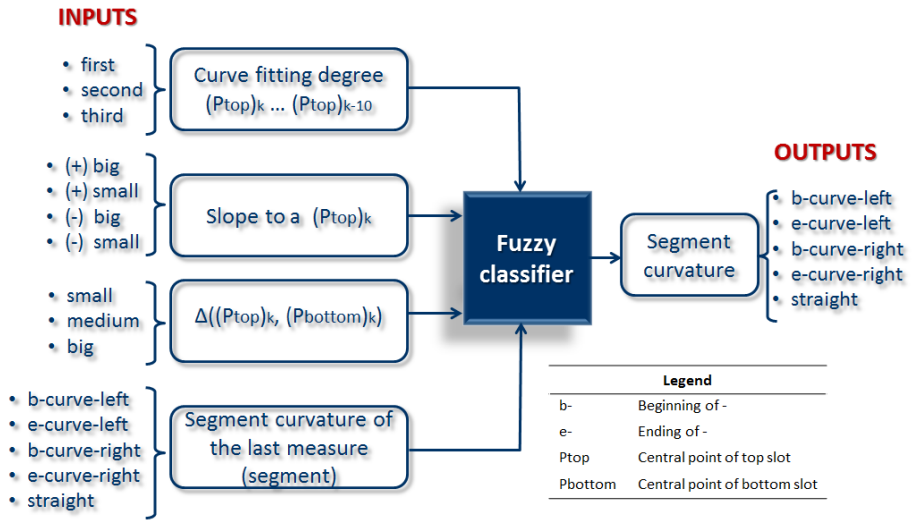


Fig. 8. Block diagram of the fuzzy classifier to detect curvature in advance

5 Experimental Results

There have been many experimental tests to validate the automatic control for articulated buses, yielding promising results. During these tests, several sensory systems, actuation systems and lateral and longitudinal control strategies were evaluated. For do this the mobile platform, as mentioned above, is complemented with the test lane built at CSIC facilities.

The experimental results associated with longitudinal control of the bus are shown in Fig. 9.The vehicle speed signals, the angular position of the brake pedal, and the acceleration in the axis of advance of the bus are compared. The longitudinal acceleration is obtained from an inertial measurement unit IMU440CA, installed on the front axle of the bus.

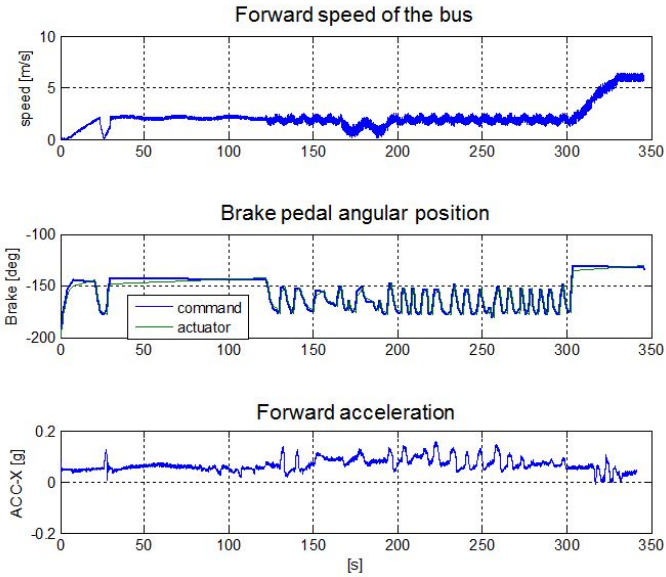


Fig. 9. Comparison among the bus speed signal, the angular position signal of the brake pedal and the forward acceleration signal of the bus

The speed used as set-point was *~2.5 m/s*, during most of the trajectory. This is because the tests lane contains seven curves with small radii of curvature, and slopes, which are considered. However, in the straight segment of the test circuit, the speed was increased up to *~6 m/s*. The braking system is used to maintain the output velocity close to the desired speed. Its function is most important when the bus travels the trajectory sections with slopes. This system acts between the times of 125 s and 300 s, approximately. The braking oscillatory signal produces a constant speed perception, although the bus is decelerating to decrease the increase of the vehicle speed due to the gravitational effects. Fig. 10 shows the experimental results obtained by applying the lateral control strategy for the bus during the trajectory tracking of the

guide rail. In this figure are compared the output steering column signal, the lateral acceleration, and the angular velocity on the azimuthal plane. The angular velocity sensor follows the curves of the test circuit commanded toward the bus steering column. The lateral acceleration signal is considered in a closed loop of the lateral control system to improve the trajectory tracking. Using this acceleration signal in the feedback loop of the lateral control, the small lateral movements are diminished, making it more comfortable the driving process.

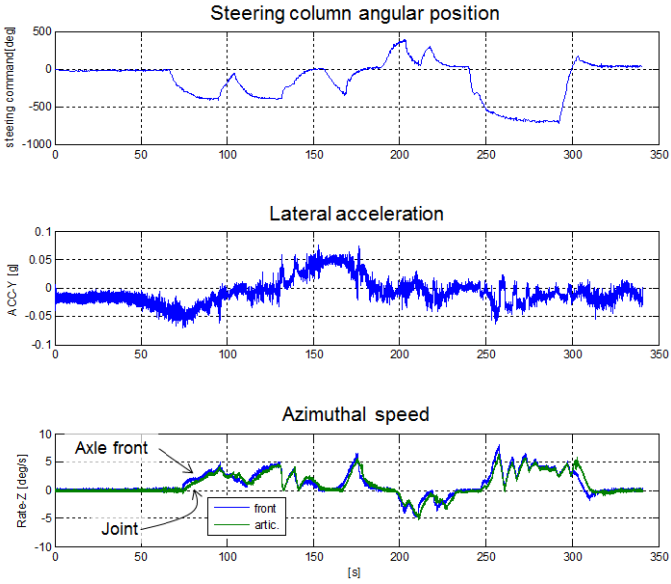


Fig. 10. Comparison among the steering signal, lateral acceleration signal, and angular velocity signal along the vertical axis during the bus travel around the test circuit

The lateral error during the test rail tracking is relatively small. The maximum lateral error was $\pm 0.35\ m$, approximately, on the circuit sections with smaller radii of curvature. This causes the angular displacement of the upper joint of the guiding mechatronics box, with regard to the longitudinal axis of the bus, is nearby to $\pm 44°$ (upper dof). This value is within the settled safety margins in the control strategy. In other tests the maximum lateral error was less than $|0.35|\ m$.

Fig. 11 shows a photographic sequence of bus tracking of the guide rail when it is entering to the curve up to exiting the curve with smaller radius of curvature. In this sequence can be appreciated the guide rail and the steering wheel in various positions. Note that when the bus is within the curve, the guide rail disappears to the vision systems.

Additionally, other experimental tests have been carried out in order to avoid the risk of bus collisions with any objects, whether they are static or dynamic. In this work the main sensory system to detect these objects has been the laser Sick LMS-291. The maximum distance used to perform the experiments was 12 m, although it is

possible to acquire acceptable measures until 30 m. The LMS-291was installed was installed at the bus front at a height of 1490 mm above the ground, and at an angle of -6° with respect to the imaginary horizontal line. The field of view of the Lidar system was established in its maximum range, $0º<fov<180º$, with resolution of 0.5º.

Fig. 11. Details of the steering wheel and the guide rail on a curve during an experimental test

Three warning levels were considered, the very high level, the high level, and the moderate level. These warning levels were delimited by zones in front of the bus. The first warning level, the very high, is bounded within an area of 4 m wide by 3 m long, centered in the longitudinal line of the bus. The second warning, the high level, is limited inside a zone of 8 m wide by 6 m long, and the last warning, the moderate level, is delimited within a region of 8 m wide by 12 m long.

The objects detection system is able to perceive with small time delay (in the order of tens milliseconds) the pedestrians in any of the three zones mentioned above, simultaneously. In Fig. 12 is possible to see the three warning zones and the pedestrians positioned in different levels in moment of capture de scenes.

When the system detects a pedestrian within a warning zone, one of the braking routine is initiated in order to avoid the risk of collision and, therefore, the pedestrian safety. The braking routine is directly related to the position of the pedestrian in the warning area, such that braking is imminent when the pedestrian is in the warning area very high.

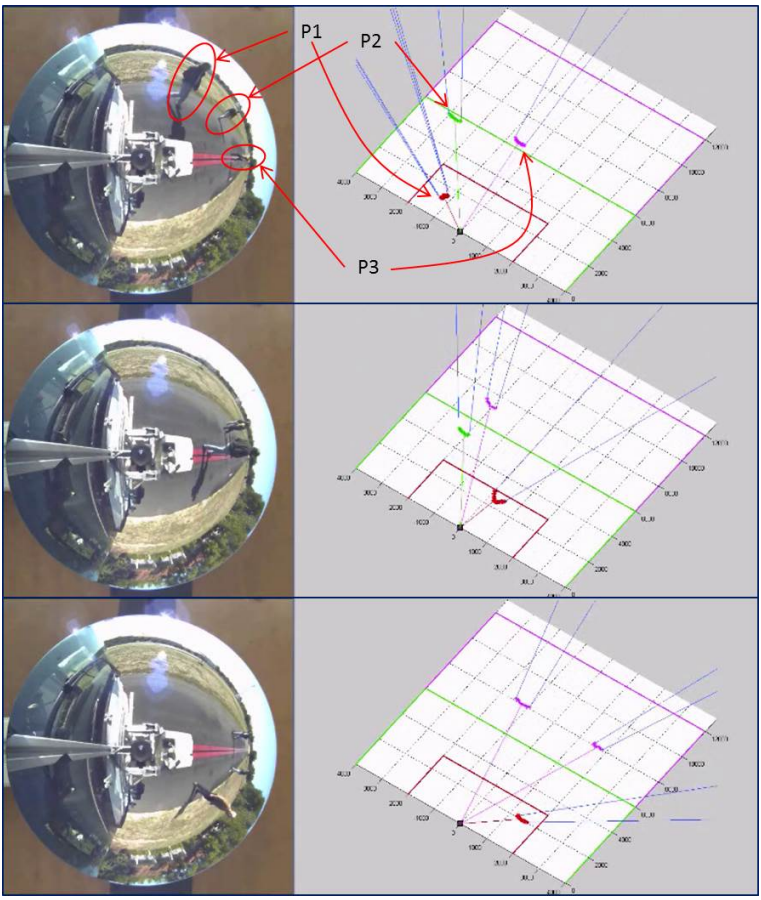


Fig. 12. Detection of pedestrians inside of three warning zones by means of Lidar system

6 Conclusions

In this work has been presented the architectures of perception and actuation of an experimental platform in order to research about of automatic control strategies for the automation of articulated buses, which consists of a bus Volvo B10M of 18.75 m in length. This automatic system principally consists of two measurement subsystems for detecting guide rail, a guiding mechatronics box fully instrumented, and the laser sensing system. The first perception system used, in addition to sending the position data of the guide rail with respect to the advance trajectory of the bus, also serves as fixation system of the bus for possible derailments. The second system is used to detect the guide rail in advance sending the position of the guide rail segments (straight or curved) to the control system. Thus, the automatic system attempts to mimic the human being behavior in driving a vehicle, because the system can see the

test circuit at a distance greater than the vehicle front. Excellent experimental results have been presented in order to the automation of the articulated bus.

Additionally, pedestrians' detection walking in front of bus, in different well-defined areas, has been properly performed, in real time, by means of the Lidar system. For the pedestrians safety diverse braking routines were executed depending on the area where the pedestrian was walking, obtaining good results. In future work will be implemented a catadioptric omnidirectional vision system to detect the pedestrians and other objects in front of the vehicle.

Acknowledgments. The authors acknowledge partial funding of this research under: Robocity2030 S-0505/DPI-0176, and IMADE PIE/62/2008 (Comunidad de Madrid, ALDESA CONSTRUCCIONES SA, MAXIMASDE). Dr. Héctor Montes acknowledges support form Universidad Tecnológica de Panamá and from CSIC under grant JAE-Doc.

References

1. Fenton, R., Melocik, G., Olson, K.: On the steering of automated vehicles: Theory and experiment. IEEE Trans. Automatic Control 21(3), 306–315 (1976)
2. Shladover, S., Desoer, C., Hedrick, J., Tomizuka, M., Walrand, J., Zhang, W., McMahon, D., Deng, H., Sheikholeslam, S., McKeown, N.: Automatic vehicle control developments in the path program. IEEE Trans. Vehicular Tech. 40(1), 114–130 (1991)
3. Broggi, A., Bertozzi, M., Fascioli, A., Bianco, C., Piazzi, A.: The argo autonomous vehicle's vision and control systems. International Journal of Intelligent Control and System 3(3), 409–441 (1999)
4. Urmson, C., Anhalt, J., Bae, H., Bagnell, J., Baker, C., Bittner, R., Brown, T., Clark, M., Darms, M., Demitrish, D., Dolan, J., Duggins, D., Ferguson, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T., Kolski, S., Likhachev, M., Litkouhi, B., Kelly, A., McNaughton, M., Miller, N., Nickolaou, J., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Sadekar, V., Salesky, B., Seo, Y., Singh, S., Snider, J., Struble, J., Stentz, A., Taylor, M., Whittaker, W., Wolkowicki, Z., Zhang, W., Ziglar, J.: Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Robotics 25(8), 425–466 (2008)
5. Naranjo, J.E., Gonzalez, C., de Pedro, T., Garcia, R., Alonso, J., Sotelo, M.A., Fernandez, D.: Autopia architecture for automatic driving and maneuvering. In: Proc. IEEE Intelligent Transportation Systems Conference, ITSC 2006, pp. 1220–1225 (2006)
6. Collier, W.C., Weiland, R.J.: Smart cars - smart highways. IEEE Spectrum 31(4), 27–33 (1994)
7. Parent, M.: Advanced urban transport: Automation is on the way. Intelligent Systems 22, 9–11 (2007)
8. DARPA. Urban challenge, `http://archive.darpa.mil/grandchallenge`
9. Henke, C., Tichy, M., Schneider, T., Böcker, J., Schäfer, W.: Organization and Control of Autonomous Railway Convoys. In: Proc. of AVEC 2008: 9th International Symposium on Advanced Vehicle Control, Kobe, Japan (2008)
10. Ackermann, J.: Robust control prevents car skidding. IEEE Control Systems Magazine 17(3), 23–31 (1997)

11. Kiencke, U., Daiß, A.: Observation of lateral vehicle dynamics. Control Eng. Practice 5(8), 1145–1150 (1997)
12. Gerdes, J.C., Rossetter, E.J.: A unified approach to driver assistance systems based on artificial potential fields. Journal Dyn. Sys., Meas., Control 123(3), 431–438 (2001)
13. Mammar, S., Glaser, S., Netto, M.: Vehicle Lateral Dynamics Estimation using Unknown Input Proportional-Integral Observers. In: Proc. 2006 American Control Conference, Minneapolis, Minnesota, pp. 14–16 (2006)
14. Byrne, R.H., Abdallah, C.T., Dorato, P.: Experimental results in robust lateral control of highway vehicles. IEEE Control Systems Magazine 18(2), 70–76 (1998)
15. Sotelo, M.: Lateral control strategy for autonomous steering of Ackerman-like vehicles. Robotics and Autonomous Systems 45, 223–233 (2003)
16. Falcone, P., Borrelli, F., Asgari, J., Tseng, H., Hrovat, D.: Predictive Active Steering Control for Autonomous Vehicle Systems. IEEE Trans. Contr. Syst. Tech. 15(3), 566–580 (2007)
17. Galicia, L.D., Cheu, R.L., Machemehl, R.B., Liu, H.: Bus Rapid Transit Features and Deployment Phases for U.S. Cities. Journal of Public Transportation 12(2), 23–38 (2009)
18. Montes, H., Salinas, C., Sarria, J., Armada, M.: Lateral and Longitudinal Control System for the Automation of an Articulated Bus. In: Proc. Emerging Trends in Mobile Robotics CLAWAR 2010, Nagoya, Japan, pp. 793–800 (2010)
19. Milanés, V., González, C., Llorca, D.F., Sotelo, M.A., Viangre, B.: Clavileño: Evolution of an Autonomous Car. In: Proc. of 13th International IEEE Conference on Intelligent Transportation Systems, pp. 1129–1134 (2010)
20. Blanco, R.: Integral train guidance system: “central rolling guide box”. Patent Nº US 7610860 B2 (2009)
21. Salinas, C., Montes, H., Armada, M.: A perception system for accurate automaticcontrol of an articulated bus. In: Proc. Emerging Trends in Mobile Robotics,CLAWAR 2010, Nagoya, Japan, pp. 1021–1028 (2010)
22. Nguyen, V., Gachter, S., Martinelli, A., Tomatis, N., Siegwart, R.: A comparison of line extraction algorithms using 2D range data for indoor mobile robotics. Autonomous Robots 23(2), 97–111 (2007)

Part XV

Field Robotics

How to Deal with Difficulty and Uncertainty in the Outdoor Trajectory Planning with Fast Marching

S. Garrido, Luis Moreno, and Javier V. Gomez

Carlos III University of Madrid, Spain
{sgarrido,moreno,jvgomez}@ing.uc3m.es
http://roboticslab.uc3m.es

Abstract. This paper presents an interesting technique for finding the trajectory of an outdoor robot. This technique applies Fast Marching to a 3D surface terrain represented by a triangular mesh in order to calculate a smooth trajectory between two points. The method uses a triangular mesh instead of a square one because this kind of grid adapts better to 3D surfaces. The novelty of this approach is that, in the first step of the method, the algorithm calculates a weight matrix W that can represents difficulty, viscosity, refraction index or incertitude based on the information extracted from the 3D surface characteristics and the sensor data of the robot. Within the bestowed experiments these features are the height, the spherical variance, the gradient of the surface and the incertitude in the position of other objects or robots and also the incertitude in the map because some portions of the map can't be measured directly by the robot. This matrix is used to limit the propagation speed of the Fast Marching wave in order to find the best path depending on the task requirements, e.g., the trajectory with least energy consumption, the fastest path, the most plain terrain or the safest path. The method also gives the robot's maximum admisible speed, which depends on the wave front propagation velocity. The results presented in this paper show that it is possible to model the path characteristics as desired, by varying this matrix W. Moreover, as it is shown in the experimental part, this method is also useful for calculating paths for climbing robots in complex purely 3D environments. At the end of the paper, it is shown that this method can also be used for robot avoidance when two robots with opposite trajectories approach each other, knowing each others position.

1 Introduction

The objective of a path planner for a mobile robot operating in environments with unknown obstacles (dynamic or not), is to calculate collision-free trajectories with the best possible characteristics such as smoothness. In outdoor environments, the safety, the gradient or the roughness of the trajectory are also important.

The main objective of this paper is the development of technologies for obtaining mobile robots capable of complex tasks that demand a high degree of

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_33,

autonomy and capacity of collaboration in the presence of human beings. One of the main applications of our work is the use of mobile robots in dangerous missions where the environment can be risky for humans (e.g., rescue missions, etc.). In this case, an unmanned autonomous vehicle (UAV) is sent in advance to obtain the images of the outdoor environment. Using the motion planning method proposed, the best trajectory to reach the goal is obtained over these images, and this trajectory is sent to the mobile robot. In the case that there are changes in the environment, the path can be recalculated using the same method. Another application would be, for example, that the images of the environment were obtained in real time using a laser scanner situated on the robot. Both situations will be considered in the experimental section.

If the Fast Marching (FM) techniques are used directly in Path Planning methods to calculate trajectories from one point to another. Nevertheless, following this strategy, the generated trajectories are not guaranteed to be safe and smooth. As explained in [1], the shortest geometrical path obtained can be unsafe, since it can touch the corners, walls, and obstacles, as shown in Figure 1. Because of these problems it is necessary to apply the FM method over a weight matrix that represents difficulty, nearness to obstacles, viscosity or refraction index.

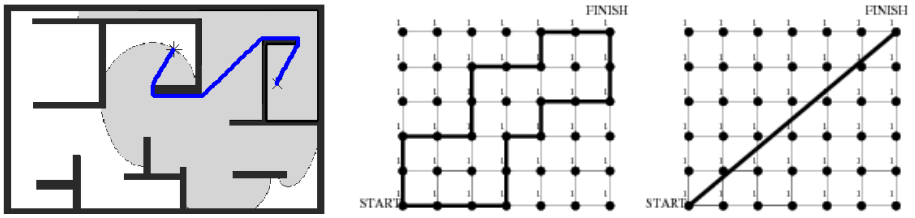


Fig. 1. Trajectory calculated by the Fast Marching method directly and comparison showing that Dijkstra's method gives multiple short paths (left image), whilst the Fast Marching method gives the optimal diagonal path (right image)

Moreover, in most works, these methods are used in $2D$ environments (indoor environments), whilst the method presented here adds another dimension to the problem (outdoor environments) and manipulates the 3D Fast Marching algorithm in order to modify the trajectories obtained to get a better and safer trajectory.

The new method proposed in this paper consists of several phases. First, as previously introduced, a $2\frac{1}{2}$ or 3D cloud of points of the surface is needed. This cloud can proceed from the robot laser, from a previous map or from a mixture of the two. Then, a triangular mesh is constructed over the image, which allows us to generate realistic surfaces due to the capabilities of the triangles to fit the characteristics of the map better than a square grid. Once the grid is constructed, the method extracts some information from the environment to obtain the height and to calculate the gradient and the spherical variance, which gives information

about the roughness of the surface. Then, it combines these data with the robot limitations to generate a weight matrix W. This matrix can be view as a difficulty or viscosity map which is situated on the 3D surface. Once the matrix is ready, the method applies the FM algorithm over this modified surface (the grid + matrix W) to generate the path.

If matrix W is not used, the trajectory obtained will be just the length of the shortest path between the two points, i.e., the geodesic distance. Applying matrix W, the proposed method gives a path which considers the features of the surface and the limitations of the robot. Moreover, it also gives us information about the speed of the robot based on the FM wave propagation speed [1,2].

The results presented correspond first to the application of this method over outdoor images (3D) taken in advance. The fast execution of this method allows the updating of the given path as the images of the environment are updated. Therefore, this could be considered as an on-line path planning method. Second, the method is also used over a 3D environment constructed (in real time) by the laser mounted on a mobile robot. In both approaches, we prove that, using the proposed method, it is possible to generate smooth and safe plans in outdoor environments.

The remainder of the paper is organised as follows. Section 2 introduces an explanation about the FM method and how this algorithm can be implemented on triangulated meshes. The following section, section 3, introduces the viscosity matrix W and how it is formed. Section 4 presents some results obtained by simulation. Finally, the main conclusions of this paper are summarised in section 5.

2 The Eikonal Equation and the Fast Marching Planning Method

In robotics, the path planner of the mobile robot must drive it in a smooth and safe way to the goal point. In nature, there are phenomena that work in the same way: the electromagnetic waves. If at the goal point there is an antenna that emits an electromagnetic wave, then the robot could drive himself to the destination following the waves to the source. The idea of the electromagnetic wave is especially interesting because the potential have all the good properties desired for the trajectory, such as smoothness (that is, C^∞) and the absence of local minima.

The first arrival of the wave front expansion can be described by the Eikonal equation. The Eikonal (from the Greek 'eikon', which means 'image') is the phase function in a situation for which the phaseand amplitude are slowly varying functions of the position. Constantvalues of the Eikonal (level surfaces or level sets) represent surfaces of constant phase, orwave fronts. The normals to these surfaces are rays (the paths ofenergy flux); thus, the Eikonal equation provides a method for 'raytracing' in a medium of slowly varying index of refraction.

One way to characterise the position of a front in expansion is to compute the arrival time T, in which the front reaches each point of the interface. It is evident

that, for one dimension, we can obtain the equation of the arrival function T in an easy way, simply considering the fact that the distance x is the product of the speed F and the time T.

$$x = F \cdot T \tag{1}$$

The spatial derivative of the solution function becomes the gradient

$$1 = F\frac{dT}{dx} \tag{2}$$

and therefore, the magnitude of the gradient of the arrival function $T(x)$ is inversely proportional to the speed:

$$\frac{1}{F} = |\nabla T| \tag{3}$$

For multiple dimensions, the same concept is valid because the gradient is orthogonal to the level sets of the arrival function $T(x)$.In this way, we can characterise the movement of the front as the solution of a boundary conditions problem. If speed F depends only on the position, then equation (3) can be reformulated as the Eikonal equation:

$$|\nabla T|\, F = 1. \tag{4}$$

The FM method is a numerical algorithm for solving the Eikonal equation, originally on a rectangular orthogonal mesh introduced by Sethian in 1996 [3]. The FM method is an $O(n)$ algorithm, as demonstrated in [4], where n is the total number of grid points. The algorithm relies on an upwind finite difference approximation to the gradient and a resulting causality relationship that lends itself to a Dijkstra-like programming approach.

The FM methods are used for problems in which the speed function never changes of sign, and so the wave front is always moving forward (there are no reflections, interferences, or diffractions). This allows us to transform the problem into a stationary formulation, because the wave front crosses each grid point only once time. This conversion into a stationary problem, offers a tremendous speed.

The wave propagation given by the Fast Marching Method gives us a distance that is the Geodesic distance measured with the metric given by the W matrix, which is completely different from the Euclidean Distance.

Since its introduction, the FM approach has been applied with success to a wide variety of problems that arise in geometry, computer vision, and manufacturing processes(see [5] for details). Varied advances have been made to the original technique, including the adaptive narrowband methodology [2]and the FM method for solving the static Eikonal equation [3].

2.1 Algorithm Implementation on an Triangular Mesh

The FM method applies to phenomena that can be described as a wave front propagating normal to itself with a speed function $F = F(x)$. The main idea is

to methodically construct the solution using only upwind values (the so called entropy condition). Let $T(x)$ be the solution surface at which the curve crosses the point x; then, it satisfies $|\nabla T|F = 1$, the Eikonal equation.

In order to understand how fast marching works, imagine an imprudent visitor that leaves unextinguished fire at some location in a natural reserve. The flame quickly becomes a forest fire, which expands outwards. Fire consumes the reached trees so the fire always propagates forward. We can record the fire front position at different points in time. It appears that the fire traverses the route having the smallest propagation time (and hence, the shortest length if the velocity is constant).In optics and acoustics this fact is known as Fermats principle or, in a more general form, the least action principle. In plain language, Fermats principle states that light traveling between two points always chooses the quickest path. Snells law of refraction follows directly from this principle.

It is necessary to know that the propagation happens from smaller to bigger values of T. The algorithm classifies the points of the triangular mesh into three sets: black, red and green, because our interface propagates like of a forest fire. Black points are points where the arrival time has been computed and is not going to change in the future. Green points are points that haven't been processed yet, for which the arrival time have not been computed up to now (corresponding with live trees). Red points are those belonging to the propagating wave front, which can be considered as an interface between the black and the green regions of the triangular mesh. In our forest fire example, red points correspond with trees that are currently in flames. Initially, only the source x_0 is marked as black and all points adjacent to it, are marked as red. The remaining points are marked as green. At each iteration, the red with the smallest value of T(x) is put into the black set. This T(x) value is calculated using the black points in triangles sharing it. The updated adjacent points are tagged as red. The process continues until all points become black.

The numerical basis of the fast marching method and its foremost difference with Dijkstras algorithm resides in the update procedure. While in Dijkstras algorithm the path is restricted to the graph edges, and a graph vertex was updated each time from an adjacent vertex, in fast marching, because the path can pass through the triangular faces of the mesh, a vertex has to be updated from a triangle, requiring two supporting vertices. We assume that the update step is applied to a triangle (x_1, x_2, x_3), where x_1 is the red point with the smallest arrival time $T_1 = T(x_1)$, x_2 is a point for which some arrival time approximation $T_2 = T(x_2)$ is available, and x_3 is the red or green point, whose arrival time approximation $T_3 = T(x_3)$ is that the triangle lies in the plane with $x_3 = 0$. In essence, given that the front reaches x_1 at time T_1 and x_2 at time T_2, the update step has to estimate the time when the front arrives to x_3.

3 Matrix W : The Difficulty or Viscosity Map

As expressed in section 1, the direct application of the FM method still has some problems. The foremost one that usually arises in mobile robotics is that

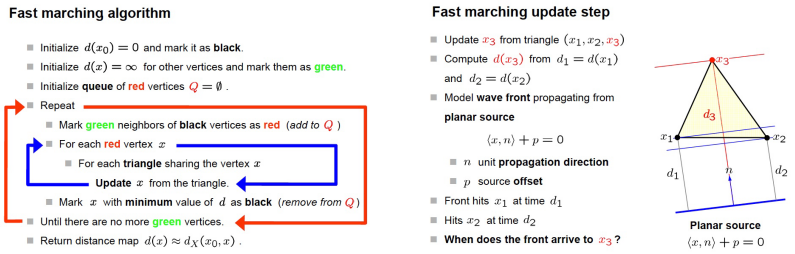


Fig. 2. Scheme of Fast Marching propagation

optimal motion plans may bring robots too close to obstacles, and this can be dangerous.

In our approach, this problem has been solved in the same way as nature does: the electromagnetic waves, as light, have a propagation speed that depends on the media, that is the slowness or refraction index of the front wave propagation of a medium. In our case, the refraction index is defined by the viscosity map. This map can also be interpreted as difficulty or uncertainty map.

The proposed technique is based on the FM method, changing the speed of the wave front using a potential surface generated from the3D environment characteristics and the robot limitations. By doing so, the method changes the time when the front reaches each point and when the generated trajectory is calculated. This trajectory is not going to be the simple geodesic, but it is going to be modified according to the robot and task needs. To be able to modify this speed, the proposed method creates a weight matrix W, which is currently built based on three main characteristics of the 3D surface: the *spherical variance*, the *saturated gradient*, and the *height*. Some other characteristics can be added to the method and it will build a different potential surface.

3.1 Spherical Variance

The spherical variance [6] finds the roughness of a surface to determine if it is crossable or not. In [7] a method to calculate the roughness degree is presented. This method is based on the normal vector deviation in each point of the surface. The spherical variance is obtained from the orientation variation of the normal vector in each point. The study uses the following reasoning:

- In a uniform terrain (low roughness), the normal vectors in a surface will be approximately parallel and, for this reason, they will present a low dispersion.
- On the other hand, in an uneven terrain (high roughness) the normal vectors will present great dispersion due to changes in their orientation.

The spherical variance is obtained as follows:

1. Given a set of n normal vectors to a surface, defined by their three components $\overrightarrow{N_i} = (x_i, y_i, z_i)$, the module of the sum vector R is calculated by:

$$R = \sqrt{\left(\sum_{i=0}^{n} x_i\right)^2 + \left(\sum_{i=0}^{n} y_i\right)^2 + \left(\sum_{i=0}^{n} z_i\right)^2} \tag{5}$$

2. Next, the mean value is normalized by dividing the module R between the number of data n, so the value of the result is within $[0, 1]$. In this way, we have

$$\frac{R}{n} \in [0, 1] \tag{6}$$

3. Finally, the spherical variance ω is defined as the complementary of the previous result.

$$\omega = 1 - \frac{R}{n} \tag{7}$$

When $\omega = 1$, there exists a maximum dispersion that can be considered as the maximum roughness degree, and when $\omega = 0$, a full alignment exists and the terrain will be completely flat.

3.2 Saturated Gradient

The gradient of a scalar field is a vector field which points in the direction of the greatest rate of increase of the scalar field, and whose magnitude is the greatest rate of change. The gradient of f is defined to be the vector field whose components are the partial derivatives of f. That is:

$$\nabla f = (\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n}) \tag{8}$$

The saturated gradient consists of giving a limit value to the gradient of each point over the 3D surface. It means that, if the gradient value exceeds that limit, the point will not be included in the list of accessible points determined by the robot limitations. The gradient depends on the robot capabilities; the maximum inclination that the robot is able to cross will be the limit value for the saturated gradient.

3.3 Construction of Matrix *W*

As previously explained, with this matrix the algorithm can modify the path that the robot is going to follow across the 3D surface. The way the matrix modifies the path is by giving a viscosity value for each point on the surface. It means that the propagation speed of the front end of the FM wave is modified.Hence, the time when the wave reaches each point will depend on that viscosity. We can add as many characteristics as we need to get different paths. These characteristics will modify the viscosity at each point.

The saturated gradient, the spherical variance, and the height are three matrices G, Sv, and H with the same size as the vertex matrix (the 3D mesh).The

value of each vertex of the 3D grid will be determined by the calculated gradient, spherical variance, and the height of each point.

To build matrix W we give a weight factor to each surface characteristic and we can determine which one is the most important depending on the task requirements.

The values of the three matrices vary from 0 to 1, so the values of matrix W are also within this range. The components of matrix W with a value of 0 will be points in the *vertex* matrix with maximum speed. Hence, these are points which the robot can cross without any problem and at its maximum speed. The components of W with a value of 1 will be points in the *vertex* with a minimum speed. This means that the robot will not be able to pass across them.

$$W = a_1 \cdot G + a_2 \cdot Sv + a_3 \cdot H \tag{9}$$

where:

$$\sum_i a_i = 1 \tag{10}$$

After matrix W is generated, the method runs the FM algorithm over the modified mesh (3D mesh + matrix W) to calculate the best trajectory. With the FM method the path found will be the less time path in the W metrics. In the normal FM evolution,this path will be the shortest because all the points in the surface will have the same 'speed' for the front propagation, i.e. it is the geodesic. With matrix W, the proposed method changes that 'speed', since this matrix gives information about the difficulty to pass through each point of the surface. The trajectory will be modified depending on the surface conditions and characteristics and according to the robot limitations. Since the method modifies the 'speed' of the Fast Marching wave, it gives not only the best trajectory, but also the speed to control the robot.

4 Algorithm Simulations

4.1 Test on Data Taken in Advance

As previously stated, in the proposed method we need terrain data that can be a previous image of the environment, an elevation map, laser data or a mixture of all. In relation to the outdoor environment reconstruction, there are many ways to build an environment and represent it as a $2\frac{1}{2}$ or 3D surface.

Since the method is working in 3D, there are 3 matrices, one for each coordinate X, Y, and Z. X and Y are the coordinates for a plain surface and Z is the height of each point. In order to create a triangular mesh, the algorithm reads the data from the bitmap file to create these three matrices X, Y, and Z and then, it builds a plain mesh based on X and Y coordinates. The algorithm generates a Delaunay triangulation in 3D[3].

After the mesh is created, the algorithm is able to extract the needed data, the vertices and the faces of the triangles, from the matrices. Using these values, the algorithm is able to model the 3D surface.

Next, in this section several paths over the surface already presented will be obtained between the same initial and final points. The initial and the final points have been situated over the top of the two sides of the mountain range. Those paths are obtained by varying the values of the weight factors a_i of matrix W.

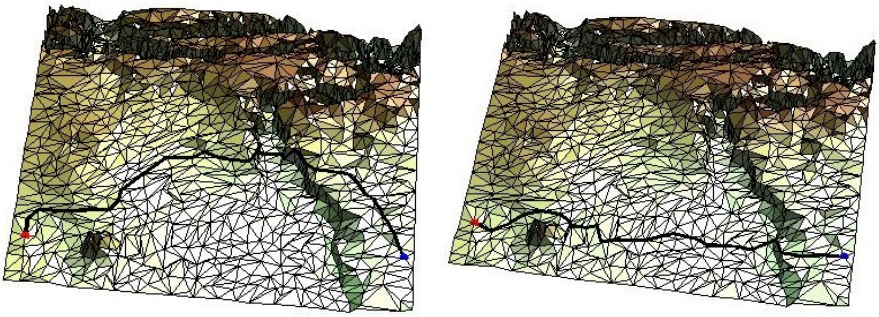


Fig. 3. Path calculated when $W = A$ and when $W = G$ in a mars map

In the case that $W = A$, this implies that the difficulty of the path will be determined by the height of every point of the mesh. In Figure 3, the path obtained when the height is penalised, without considering the roughness of the surface or its inclination, is presented. As can be observed, the calculated path will try to reach the final point passing trough the deepest part of the map.

On the other hand, if we decide to calculate the path penalising just the inclination of the surface, then the viscosity matrix is defined as $W = G$. In this case, as shown in Figure 3, the path will follow the parts with smallest slope.

Finally, the general idea proposed in this paper is the possibility of combining the different matrices in order to obtain a path that considers the height A, the roughness Sv, and the inclination G of the surface, among others. In the previous figures, it can be observed that, for the selected initial and final points, the height matrix favour that the path goes all the way trying to avoid the highest parts of it. On the other hand, the gradient matrix G favours the path with smallest slope. Therefore, we can select the values of each weight factor a_i in order to consider the limitations or features of the robot used.

Figure 4 shows a view of the path obtained when $W = Sv$. As can be observed the result is an intermediate path. Fig. 4 shows a view of the path obtained when $W = 0.20 * A + 0.40 * Sv + 0.40 * G$.

Moreover, the values of the weight factors a_i can be changed if the robot to be used is different or modified. It is also important to note that the trajectories calculated are a tentative path for the robot. The path can be modified online by modelling the environment with the robot sensors and recalculating the trajectory in a local area.

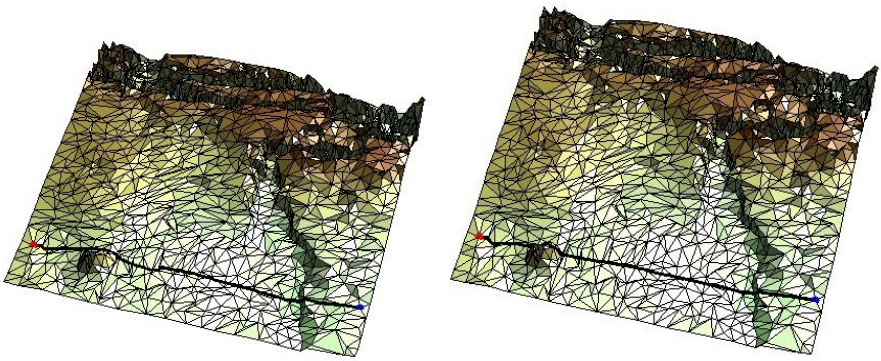


Fig. 4. Path calculated using $W = Sv$ and when $W = 0.20 * A + 0.40 * Sv + 0.40 * G$ in a mars map

4.2 Introduction of the Uncertainty in the Slowness Matrix *W*

When there is a certain uncertainty the robot has to modify the trajectory or the velocity. For example, in the case of robot in Mars, if the robot doesn't have enough information of part of the trajectory, because it hasn't visual data of that part, could be better to change the trajectory to zones the robot can visualise.

How can we introduce that uncertainty in the map in order to change the trajectory? Fortunately, the viscosity matrix W can also be understood as an uncertainty matrix, the grey degree can be understood as a measurement of the uncertainty.

For example, suppose that the robot has no data of the points lower than its altitude, in that case the shadow points are represented in the matrix W with values next to zero (velocity of the media). In Fig. 5 is shown the difference between the robot trajectories without and with unprecise data of the points lower than the robot's altitude. As can be seen in the figure on the right, the trajectory is modified to not go through the lower areas. In Fig. 6 is shown the difference between the difficulty-uncertainty W matrices when the robot has and hasn't data of the points lower than its altitude. As can be seen in the right figure the lower parts have a bluish colour due to a bigger uncertainty and lower values in the W matrix that correspond to lower media velocity. In Fig. 7 is shown the difference between the wave expansion D matrices when the robot has and hasn't data of the points lower than its altitude. As can be seen, in the figure on the right, the expansion of the wave is more directed to the zone with less incertitude.

In Fig. 8 is shown the difference in the paths when the gradient is not saturated and when it is.

In Fig. 9 are shown the saturated gradient and the spheric variance corresponding to the previous figures.

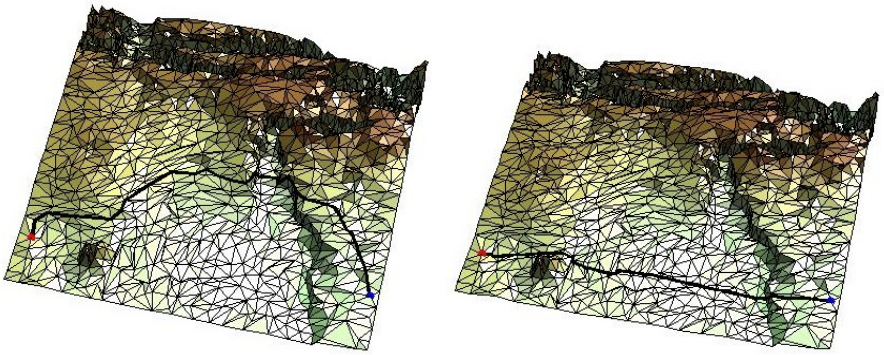


Fig. 5. Difference between the robot trajectories without and with unprecise data of the points lower than its altitude

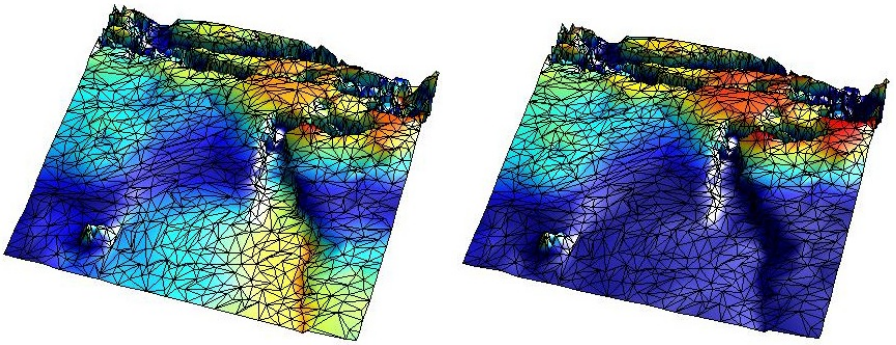


Fig. 6. Difference between the difficulty-uncertainty W matrices when the robot has and hasn't data of the points lower than its altitude

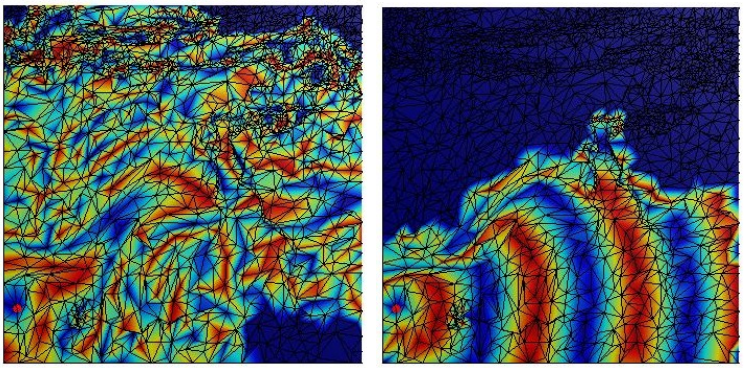


Fig. 7. Difference between the wave expansion D matrices when the robot has and hasn't data of the points lower than its altitude

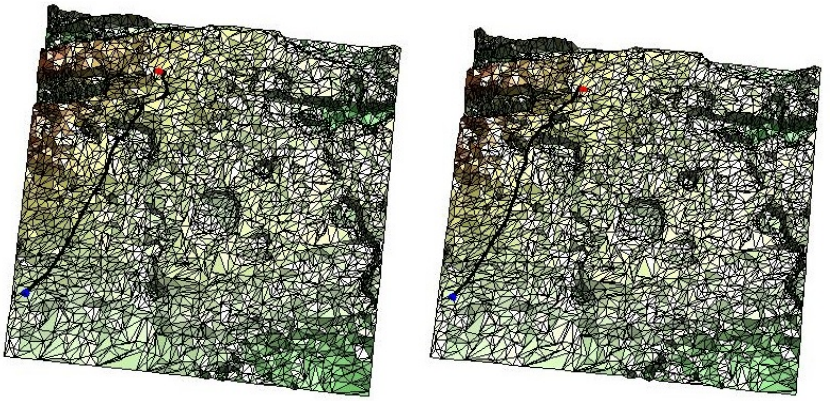


Fig. 8. Difference between the difference in the paths when the gradient is not saturated and when it is

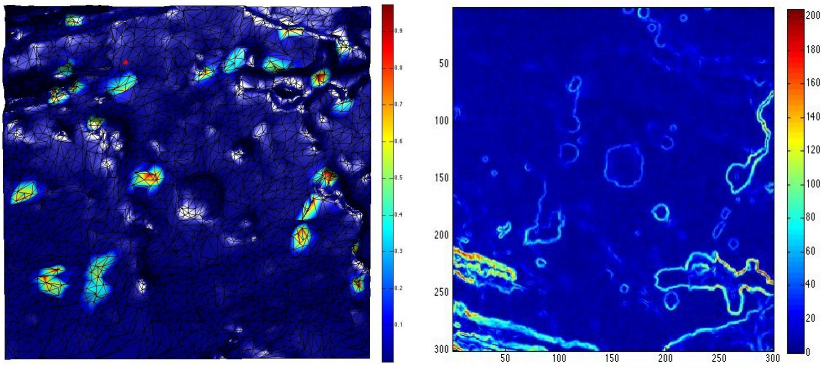


Fig. 9. Spheric variance and the saturated gradient corresponding to the previous figures

4.3 Dynamic Evolution of the Paths of Two Robots When They Sense Each Other's Position

Another interesting application of the proposed method is the obtaining of the dynamic trajectories of two robots which navigate approaching each other, sensing their mutual position.

In order to get these trajectories, each robot maintains a map with the matrix of difficulty of the terrain. In each of these maps, it is necessary to add a small Gaussian to the position of the other robot in order to avoid possible collisions.

This bivariable Gaussian distribution is centered at the sensed position of the other robot with an incertitude given by a standard deviation of 10 in the direction of the trajectory (longer vector), and of 3 in the second direction (shorter vector, orthogonal to the longer vector). This Gaussian has been rescaled with a factor of 10 so that it can have the necessary importance in the difficulty map.

In the initialization phase, the algorithm calculates the general difficulty matrix or viscosity map W. Then, each robot maintains a copy of this map and, at each iteration, a Gaussian $G_i(k)$ is added to the sensed position of the other robot i. Afterwards, at each iteration, the trajectory of the robot i is calculated using the described Fast Method on the map $W + G_i(k)$, until the goal point is reached.

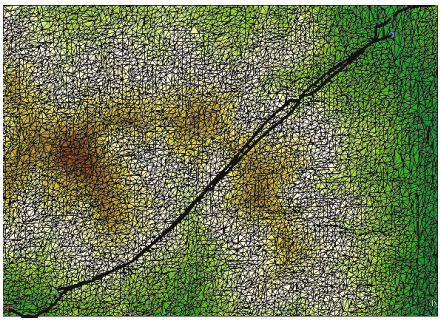


Fig. 10. Starting point. Both planned paths cross.

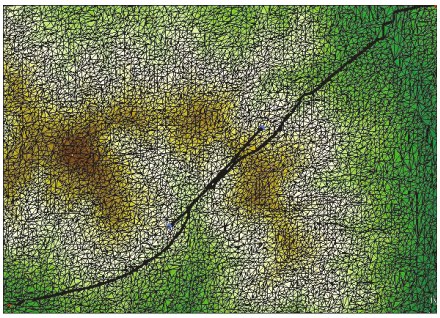


Fig. 11. Robots approaching with no view of each other

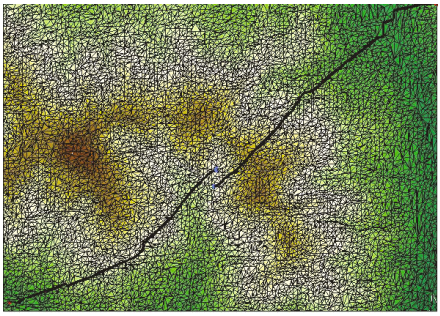


Fig. 12. Robots approaching sensing their mutual position

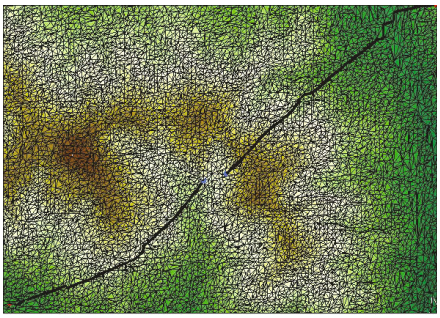


Fig. 13. Robots moving away from each other

In the following figures the dynamic evolution of the trajectories of two robots is shown. The goal of each robot is represented by two red dots (one of them situated in the upper-right corner and the other one in the bottom-left corner), while the current position is represented by two blue ones. The starting point of each robot, as shown in Figure 10, is almost the same goal point of the other one. Therefore, the first planned paths cross each other in several points, provoking potential collisions. As the robots move towards their goals, they approach each other. In Figure 11 we observe the situation where both robots are approaching but they are not aware of the position of the other. In this case, a collision of

both robots is expected. Later, when the robots are closer, they start sensing each other and, as can be seen in Figure 12, their paths run in parallel with no collision. Finally, in Figure 13 it is observed that each robot moves away from the other till they reach their goal points. As has just been explained, this robot avoidance is made thanks to the modification of the difficulty matrix W by adding a Gaussian to the sensed position of the robot.

5 Conclusions

The algorithm we have presented here is a new way to calculate trajectories for moving a robot over a 3D outdoor surface. One main point about the proposed method is that it can be used not only as a Path Planning method, but also to control the robot speed to keep it within a range given by the limit speed allowed over the 3D surface, taking into account environmental characteristics and task requirements reflected in matrix W, that can represent difficulty or incertitude. There are two values that can be attained from the results of the algorithm: the robot speed and the robot orientation. The speed is taken from the potential surface and the orientation can be taken from the next point in the trajectory that is going to be occupied by the robot. If the robot orientation and the next point where the robot is going to be are known, we can calculate the control law that has to be given to the robot in order to make it reach that next point. The most important thing about this algorithm is that it works in real time. It is really fast and give us the possibility to use it on-line to make decisions in order to avoid fixed or moving obstacles.

References

1. Garrido, S., Moreno, L., Blanco, D.: Exploration of a cluttered environment using voronoi transform and fast marching method. Robotics and Autonomous Systems 56(12), 1069–1081 (2008)
2. Garrido, S., Moreno, L., Abderrahim, M., Blanco, D.: Fm2: A real-time nsensor-based feedback controller for mobile robots. International Journal of Robotics and Automation 24(1), 3169–3192 (2009)
3. Sethian, J.A.: Theory, algorithms, and aplications of level set methods for propagating interfaces. Acta numerica, 309–395 (1996)
4. Yatziv, L., Bartesaghi, A., Sapiro, G.: A fast o(n) implementation of the fast marching algorithm. Journal of Computational Physics 212, 393–399 (2005)
5. Sethian, J.: Level set methods. Cambridge University Press (1996)
6. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. The National Academy of Sciences 95 (1998)
7. Castejon, C., Boada, B., Blanco, D., Moreno, L.: Traversable region modeling for outdoor navigation. Journal of Intelligent and Robotic Systems 43(2-4)

Energy Efficient Area Coverage for an Autonomous Demining Robot

José Prado and Lino Marques

Institute of Systems and Robotics,
University of Coimbra,
3030-290 Coimbra, Portugal
{jaugusto,lino}@isr.uc.pt

Abstract. This work studies different coverage approaches with a mobile robot equipped with a landmine detecting sensor attached to an actuator arm. Different coverage techniques were experiment in this work and the cost benefit was analyzed in terms of energy consumption. The problem of optimising the combined motion of a mobile platform with an arm was addressed. The feasibility and effectiveness of both algorithms are demonstrated by simulation results for coverage work of target area.

1 Introduction

The demining task is as important as dangerous, and humans are frequently put in risk during the search and de-activation of landmines. Therefore, significant efforts are being made by the scientific community, in order to develop systems able to automatically detect landmines while the users are kept safe. Blast antipersonnel mines include less than 100 g of explosive and very little metal. Due to the simple construction, the mines are mostly very cheap, consequently they were extremely spread. The most common type of installation is underground, therefore they are very difficult to see. Different approaches has been proposed for solving the mine detection problem, for example [1] and [2], which used a set of sensors that vary among metal-detectors, ground penetration radars, cameras, chemical sensors and others, the fusion of some of those sensors was also shown to be effective and to reduce the false positives [3].

Independently of the used set of sensors, the robot must carry this set across an area, while reading the ground in its search of land-mines. This task is called robotic coverage. There are several approaches for coverage in the literature and the selection of the proper approach is highly dependant of the application. For example: The line-sweep approach [4] and the spatial cell diffusion [5] are suitable for open rectangular areas due to the shape created by the trajectory of the robot, another important factor of these approaches is that the robot can starts from a corner or even from out-side the area. The smooth spiral path planning [6] and the squared spiral approach requires that the robot starts or ends at the middle of the area, a constraint that might be critical for some applications, as demmining. Multi-robot coverage algorithms [7] deals with the coordination of several robots to cover an area, to have multiple robots covering an area clearly leads to a faster coverage, however, due to collision avoidance processes the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_34,

number of maneuvers is higher than with a single robot, consequently also higher is the energy cost of the entire coverage. Target coordinated area coverage approaches [8] are meant to minimize the trajectory path in order to find a single target.

Hence, although the sensing and detection techniques can still be improved, in this paper we assume that once the sensor is swept over the mine, it detects it. Therefore, we focus on analysing the motion of our robot and its end-effector (which carries the sensor) and how this motion affects the energy-efficiency of different coverage techniques applied to our robot (Fig. 1). This work is a part of the ongoing project TIRAMISU (ISR, Coimbra, Portugal) aimed to develop demining technologies; among robots, sensors and devices that endows the exploration of fields and the location of landmines.

Fig. 1. Robot equipped with mine detecting sensors

2 Related Work

Covering large areas is a task that is obviously useful for a large number of applications such as cleaning large surfaces, mapping purposes [9], and autonomous mine clearance [10].

In indoor environments, the obstacles are usually walls, furniture and structure of the building, while in outdoor environments there can be trees, rocks, ravines, or none. More specifically, for the demining task, the mines by itself shall be considered a special type of obstacle, that after detection, shall be mapped and avoided.

Thus, the real coverage path is the motion of the robot when following a coverage plan combined with the effort to avoid stepping on the mines (obstacle avoidance). When applied to large areas, such effort requires that the robot is operational for long periods of time. Therefore, the interest on using algorithms that can maximize the covered area for a certain amount of energy extant on the robot arises.

3 Energy Consumption

3.1 Robot Base

A wheeled robot normally has several motors where each one rotates a wheel. The velocity of the i^{th} wheel is defined as $vi = \rho * \omega_{wi}$, where ρ is the wheel's radius and ω_{wi} is the respective motor's angular velocity. Similarly, the velocity of the arm sweep is given by $v5 = b * \omega_a$, where b is the length of the arm and ω_a is the angular velocity of the arm motor. If we consider that our robot exists in a 2D space, it is possible to represent its velocity by three variables: Vx, Vy, and Ω. The matrix $(Vx, Vy, \Omega)^T$ represents the velocities of the robot (linear and angular). Our robot has a skid-steer configuration, the wheels do not steer, the distance between wheels and the distance between axis are the same. From robot's top view, this configuration makes the center of the four wheels as the corners of a perfect square, inscribed in a circle, where the center of the circle is also the center of the robot. (see Fig. 2). Thus, the four wheels are mounted at distance r from the center of the robot.

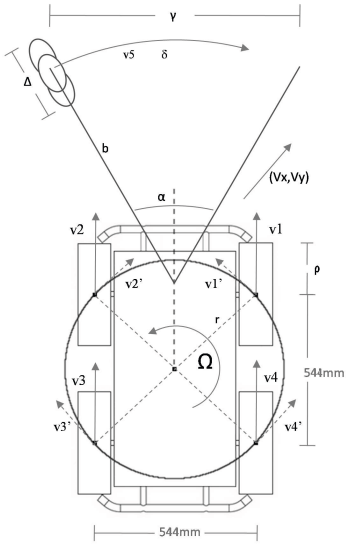


Fig. 2. Kinematics of our robotic platform

Let $v1$, $v2$, $v3$ and $v4$ be the velocities of the four wheels, and $v5$ the velocity of the arm sweep. Each of the wheels velocity, has a component that potentially performs rotation, this component can be seen as the projection if the velocity vector on the local tangent of the inner circle. Let $v1'$, $v2'$, $v3'$, $v4'$ be the respective potential-rotation components of each wheel velocity, where $vi^2 = vi'^2 + drift^2$. The potential-rotation components will null each other when exist together with the one from the other side of the robot, the same occurs with the drift components, therefore there is no rotation nor drift when the robot moves strait forward. The relationship between the potential-rotation components of the wheels and the velocity of the robot can be expressed by:

$$\begin{pmatrix} v1' \\ v2' \\ v3' \\ v4' \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & r \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & r \\ -\sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} & r \\ \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} & r \end{pmatrix} \cdot \begin{pmatrix} Vx \\ Vy \\ \Omega \end{pmatrix} \tag{1}$$

This relation, the manipulator Jacobian, allows to control the velocity of the robot $(Vx, Vy, \Omega)^T$ by changing the velocities $(v1, v2, v3, v4)^T$, consequently changing the components $(v1', v2', v3', v4')^T$. Let's consider that $V_t = (Vx_t, Vy_t, \Omega_t)^T$ is the velocity of the robot at time t. The potential-rotation component of the velocity of each wheel at time t is: $vi'(t) = J_{i,1} \cdot Vx(t) + J_{i,2} \cdot Vy(t) + J_{i,3} \cdot \Omega(t)$. Henceforth, the angular velocities of the respective motors is $\omega_{wi} = \frac{vi(t)}{\rho}$.

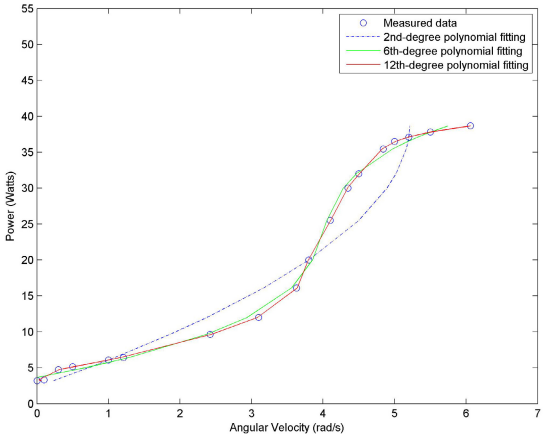


Fig. 3. Power model for different ω

In order to define the power model, lets consider that $Pm(\omega, \alpha)$ is the consumption of a motor moving at angular velocity ω, and with angular acceleration $\alpha = \frac{d\omega}{dt}$. According to [11], a polynomial fitting is a good approach to model the power of the motors since the behavior of those can be represented by an unbounded function. In [12], the power model of a basic motor was modeled as a second-degree polynomial of ω, in [13] it was shown that a sixth degree model would be more suitable. From our experimental data, see Fig. 3, we find that a 12th-degree polynomial fitting is better than both of the formers. For the 2nd, 6th and 12th degree polynomial fitting, the mean square error respectively decreased from 0.01668, through 0.002720, to 0.000068. The energy consumption per radian is presented in Fig. 4. Notice that for these motors, an specific interval of angular velocities that lies around the middle velocity, is more energy-efficient. Moreover, once considering that the model presented in Fig. 3 is the power model of the wheels' motors (P_{mw}), in a similar manner we found the power

model of the arm motor P_{ma}; thus the power consumption cost function of the robot can be defined as:

$$\sum_{i=1}^{4} P_{mw}(\omega_{wi}, \frac{d\omega_{wi}}{dt}) + P_{ma}(\omega_a, \frac{d\omega_a}{dt}) \tag{2}$$

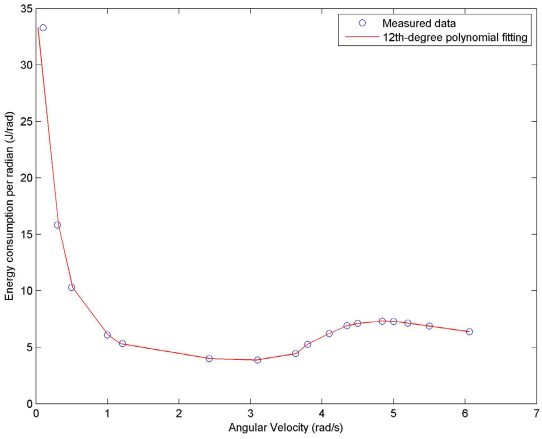


Fig. 4. Energy efficiency of each wheel

3.2 Robotic Arm Sweeping

Let α be the maximum angle of the arm sweep, $\frac{\alpha}{2}$ radians to the left and $\frac{\alpha}{2}$ radians to the right. The distance traversed during one arc sweep is $\delta = \alpha \cdot b$. The width of the covered strip is $\gamma = 2\sin(\frac{\alpha}{2})$.

As represented in Fig. 5, let Δ be the size of the sensor, consequently $\frac{\Delta}{2}$ is the distance that the robot shall move forward during one arc sweep. Thus, although setting $\alpha = \pi$ would maximise the coverage area in front of the robot, when moving forward $(0, Vy, 0)^T$, the arm angle of sweeping (α) also affects the maximum velocity of the robot (ω_w), by following relation:

$$\omega_w = \frac{\Delta \cdot \omega_a}{\alpha \cdot \rho} \tag{3}$$

Case Study In our robot, each coil of the metal detector is equipped with one chemical sensor, the arm swiping movement of the arm generates simultaneously 3 lines of dual data. These dual sensor data is fused and interpolated based on the method proposed at [3] and generates an arc strip of covered area in front of the robot with Δ length (see Fig. 5). The size of our dual sensor is $\Delta = 0.62\,m$.

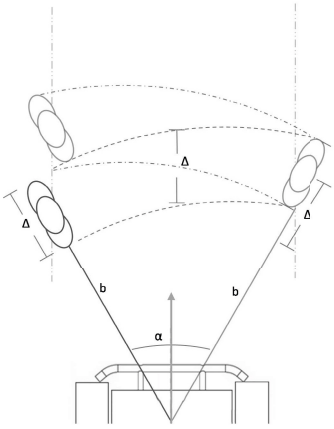


Fig. 5. Combined robotic movement and arm sweeping

This application bring some constraints to our problem:

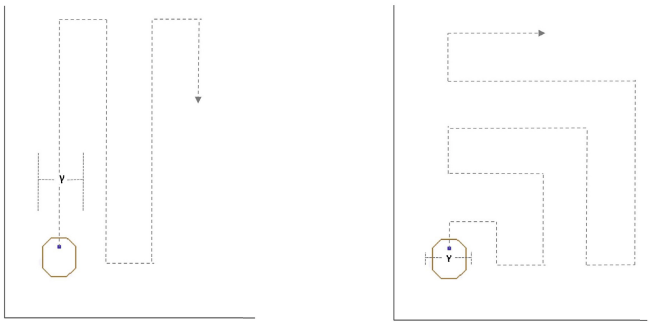
- $v5$ is limited to $\frac{1}{2}$ m/s, according to the manual of the metal detector and also with [14], this is the maximum sweep speed accepted for proper sensor functionality;
- according to [15], the size of the robot when rotating over it's own axis is $0.99\,m$;
- the covered strip while the robot moves forward ($(0, Vy, 0)^T$) has a detection width of at least $0.99\,m$, for the robot to safely pass through an area without step in mines.
- according to the power model defined on 3, the optimal speed of the robot would be around 3; to maximize the robot speed the arm angle of sweep need to be as small as possible.

Considering these constraints, and considering that our $b = 1| : m$, the angle of the arm sweep was set to 60 degrees, 30° to the left and 30° to the right. The distance of sweep became $\delta = \frac{\pi b}{3} m$. Thus, the angular velocity of the arm is $\omega_a = \frac{1}{2}$ rad/s. To match the desired scan presented on Fig. 5, without gaps, the robot needs to move $\frac{\Delta}{2} = 0.31\,m$ forward in $\frac{2\pi}{3}$ seconds.

Hence, according to the relation 3, these parameters allows us to apply a maximum angular velocity of the wheels' motors of: $\omega_w = 1.19\,rad/s$, which is inside an acceptable range of energy efficiency for the wheels' motors (see Fig. 4), and it is the optimal configuration that guaranty the arm scan to perform without gaps.

4 Area Coverage

Coverage is the problem of moving a sensor over all points in a given region. Since we already defined the relation between the motion of the sensor and the motion of the robot, we want a energy-efficient path for the robot. Thus, lets make some assumptions about important elements that describe a coverage problem:



(a) Line-sweep trajectory (b) SCD trajectory

Fig. 6. Target trajectories for both algorithms

- The coverage region: The region to be covered is continuous and smooth (or can be embedded in the plane), is connected, and is defined by an outer perimeter and some mines (obstacles) in its interior. In this paper we will assume that perimeter is polygonal, and the mines are spread randomly.
- The robot: It is a differential skid steer robot. The starting or ending position of the robot may be specified.
- The sensors: The sensor is considered to be two dimensional and it sweeps through an arc area as the robot moves. The scanned area is larger than the robot.
- The robot can navigate outside the target area in order to maneuver, thus the map is larger than the target area.
- A coverage algorithm must return the coverage path through a detailed sequence of velocities commands for the robot.

As referred in section 2, there are several algorithms for covering a planar area, the selection of the proper approach is highly dependant of the application. As our target application is mine clearance in open fields, some additional constraints take place: the environment is an open out-door area, the robot can not start from the middle/center of the area, the robot shall minimize the number of maneuvers, the area of mine fields are usually polygonal. Considering those constraints, we select two suitable approaches for our analysis, line-sweeping and spatial-cell-diffusion. The energy-efficiency (EE) of a covered rectangular $m \times n$ area, when the robot consumed e energy to cover this area, can be generally defined as: $EE = \frac{m \cdot n}{e}$.

4.1 Line-Sweeping

The line-sweeping (LS) algorithm works in two stages. In the first stage, the algorithm selects the longest edge of the field to determine the optimal direction of the sweep. In the second stage, it generates $\frac{n}{\gamma}$ tracks parallel to the field edges to be used as rows.

Although this algorithm can also be used for unknown sized areas, if the map is giver a priori enhanced decisions can be taken. Thus, let's consider a given rectangular area of size $m \times n$, where $m \geq n$ and γ is the width of the sensor detection area. The number of lines is defined as $\frac{n}{\gamma}$. The LS, described in Algorithm 1, than sends to the navigation cartesian coordinates of the points of rotation, defining the optimal trajectory as represented in Fig. 6(a). Since we are considering that the testing area is strongly connected, a mine field with sparse and randomly distributed mines, this trajectory is achieved without the need of further area segmentation.

Algorithm 1. LS Line-sweeping coverage

Input: Area size= $\{m \times n\}$ Scan strip size= γ
Output: Path=Sequence of navigation points on the map, CmdVel=Sequence of velocities commands
//function path gives the point x, y of the target // given the velocities and distance
if $m > n$ **then**
  $dir \leftarrow m$;
end
else
  $dir \leftarrow n$;
  and robot rotates 90°
end
$tracks \leftarrow \frac{dir}{\gamma}$
for $i \leftarrow 1$ ***to*** $tracks$ **do**
  //Robot moves forward dir distance
  $Path[i] \leftarrow path(f(0,Vy,0)^T, dir)$ $CmdVel[i] \leftarrow navigation(Path[i])$
  $\{v1, v2, v3, v4\} \leftarrow f(0,Vy,0)^T$
  //Robot rotates 90°
  $\{v1, v2, v3, v4\} \leftarrow f(0,0,\Omega)^T$
  //Robot moves forward γ distance
  $Path[i] \leftarrow path(f(0,Vy,0)^T, \gamma)$ $CmdVel[i] \leftarrow navigation(Path[i])$
  $\{v1, v2, v3, v4\} \leftarrow f(0,Vy,0)^T$
  //Robot rotates 90°
  $\{v1, v2, v3, v4\} \leftarrow f(0,0,\Omega)^T$
end
return $Path, CmdVel$

4.2 Spatial Cell Diffusion

The spatial cell diffusion (SCD), described in Algorithm 2, moves the robot in a spiral movement, although it differs from the square-spiral by alternating clock-wise and counter-clock-wise movements each time robot reaches to a border of the target area. This feature also allows it to be compatible with rectangular areas with $m \neq n$, since the motion starts another spiral when boundaries are reached.

Algorithm 2. SCD Spatial cell diffusion coverage

Input: Area size= $\{m \times n\}$ Scan strip size= γ
Output: Path=Sequence of navigation points on the map, CmdVel=Sequence of velocities commands
//function path gives the point x, y of the target // given the velocities and distance
$tracks \leftarrow \frac{dir}{\gamma}$ bool $side \leftarrow right : side\{right, left\}$ $dist \leftarrow gamma$
//Robot moves forward γ distance
$Path[i] \leftarrow path(f(0,Vy,0)^T, \gamma)$ $CmdVel[i] \leftarrow navigation(Path[i])$
$\{v1, v2, v3, v4\} \leftarrow f(0,Vy,0)^T$ //Robot rotates 90° to the $side$
$\{v1, v2, v3, v4\} \leftarrow f(0,0,\Omega)^T$ **for** $i \leftarrow 1$ ***to*** $tracks$ **do**
  for $j \leftarrow 1$ ***to*** 3 **do**
    if $j \neq 3$ **then**
      //Robot moves forward $dist$ distance
      $Path[i] \leftarrow path(f(0,Vy,0)^T, dist)$ $CmdVel[i] \leftarrow navigation(Path[i])$
      $\{v1, v2, v3, v4\} \leftarrow f(0,Vy,0)^T$
      if $j = 1$ **then**
        //Robot rotates 90°
        $\{v1, v2, v3, v4\} \leftarrow f(0,0,\Omega)^T$
      end
      if $j = 2$ **then**
        $side \leftarrow notside$
        //Invert rotation and rotates 90°
        $\{v1, v2, v3, v4\} \leftarrow f(0,0,\Omega)^T$
      end
    end
    else
      //Robot moves forward γ distance
      $Path[i] \leftarrow path(f(0,Vy,0)^T, \gamma)$ $CmdVel[i] \leftarrow navigation(Path[i])$
      $\{v1, v2, v3, v4\} \leftarrow f(0,Vy,0)^T$
      //Robot rotates 90°
      $\{v1, v2, v3, v4\} \leftarrow f(0,0,\Omega)^T$
    end
  end
end
return $Path, CmdVel$

5 Experimental Tests

5.1 Simulated Environment

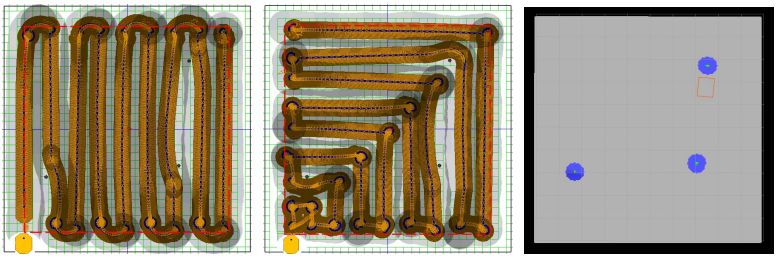
A two dimensional target area of size $m \times n$ was defined in the Stage Robot Simulator. It was also necessary to design a simple model of our robot so that the robot footprint had the dimensions of the Husky robot presented in figure 1. According to the Clearpath-robotics Husky tech-specs [15], the size of the robot is $0.99\,m \times 0.67\,m$, from a top point of view. The arm dimensions were also designed accordingly, $1\,m$ arm length, the arm sweeping angle was limited to 60° and the angular velocity of the arm was set to $\omega_a = \frac{1}{2}$ rad/s. The velocity commands $(v1, v2, v3, v4, v5)$ sent to this simulated robot are

the same as the ones issued for the real robot. The energy consumption was measured during the motion by our polynomial fitting generated models P_{mw} and P_{ma}. Since in the physical robot we use GPS RTK and a very accurate localization is achieved, in the simulator we are using perfect-odometry as the localization reference. In order to simulate the sensors, a bivariate Normal distribution was injected in the map, with the center on the location of each simulated mine, this is detected as the sensor is swept over the area of the mine.

To guaranty practical feasibility, navigation planning was configured to avoid stepping on the mines as the robot detects them. This important factor leads the robot's path to have slights deviations from the coverage algorithm initial plan; consequently also the robot's energy consumption.

5.2 Experiments

Each algorithm was tested several times in four different size squared areas ($m = n = \{2, 10, 20, 30\}\, m^2$), the mines were sparsely distributed keeping at least $\gamma + \frac{\gamma}{2}$ distance between them. The amount of mines ($\lfloor m/3 \rfloor$) varied according to the size of the area, the shape of the target area was kept as a square. The size of the mines was kept constant for all tests, this size was selected based on real tests [3] when scanning the M114 (also known as MAPS), an old Portuguese made anti-personal blast mine.



(a) Line Sweep (b) SCD (c) Mines Position

Fig. 7. Trajectory of the robot for both coverage algorithms

6 Results and Discussion

Our study is both theoretical and practical, we formulate a general problem in motion planning and use a skid-steer robot with a sweeping end-effector to experimentally validate our approach. The study presented on section 3, more specifically the relation between the arm sweeping speed and the maximum velocity of the robot, were validated on our robot, but can be easily adapted to any wheeled robot.

Figure 8 shows the estimated and measured energy efficiency of the tested algorithms paths. When the covered area is very small, both algorithms are less efficient because the number of maneuvers are high in relation with the distance traversed in straight lines. As the area increases, both algorithms become better, the Line-Sweeping algorithm

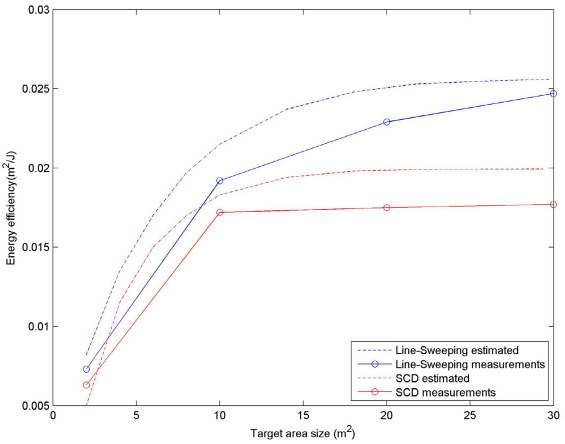


Fig. 8. Energy Efficiency comparison with different target area sizes

Table 1. Additional Comparison

	Line-Sweep	SCD
# of maneuvers	$(2 \cdot m - 1) + deviations$	$(3 \cdot m) + deviations$
Traversed distance	$m^2 - 1 + deviations$	$m^2 - 1 + deviations$
% of overlapping	15%	13%
% of covered area	100%	100%

was more efficient in all cases. Moreover, as expected, the measurements of the tested scenarios were worst than the estimated efficiency. The reason is that the estimated efficiency was computed for the perfect path of each algorithm, while the measured efficiency uses different accelerations as input, given the slights deviations that the robot has to do in order to avoid stepping on the mines.

Figures 7(b) and 7(a) present trajectory results, where three mines were sparsely positioned in a $10\,m \times 10\,m$ target area. Moreover, since the robot needs to be able to maneuver outside the target area, a $11\,m \times 11\,m$ maneuver space was defined. The robot trail is marked in orange, the sensor trail in light gray and the target area is contoured by a dashed line, the injected bivariate Normal distributions can be seen in Fig. 7(c).

Table 1 presents an additional comparison based on the average of the measurements collected along all the tests, notice that the relations presented in this table are valid for square fields. This study provides a foundation for future research in conserving the energy consumption of wheeled mobile robots. In future work we plan to investigate the behavior of these algorithms in different shapes of target area, and also with a denser set of mines.

7 Conclusions

In this paper we analyzed the kinematics of our robot and how it influences the energy consumption during movements. We also defined the power consumption model of the robot's motors, and a relation between the velocity of the arm sweep and the maximum velocity of the robot in order to guaranty a motion scan without any gaps. Moreover we analyzed, tested and compared different coverage algorithms in a simulated environment, also comparing the perfect planned path efficiency with feasible case scenarios for the demming application. In the considered setup, the end-effector motion results from the motion of the platform or the arm, or both. A cost function for these possible end-effector movements was defined and used to evaluate the global cost of covering a given area with different coverage algorithms. This paper describes ongoing works in this area. In the final version, a deeper study of the global cost influence of different coverage algorithms for different types of environments will be provided.

Acknowledgements. This work was partially carried-out in the framework of TIRAMISU project (www.fp7-tiramisu.eu). This project is funded by the European Community's Seventh Framework Program (FP7/SEC/284747).

References

1. Larionova, S., Marques, L., de Almeida, A.: Toward practical implementation of sensor fusion for a demining robot. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2004)
2. Rachkov, M.Y., Marques, L., de Almeida, A.T.: Multisensor demining robot. Autonomous Robots 18(3), 275–291 (2005)
3. Prado, J., Cabrita, G., Marques, L.: Bayesian sensor fusion for land-mine detection using a dual-sensor hand-held device. In: IECON 2013, the 39th Annual Conference of the IEEE Industrial Electronics Society (2013)
4. Huang, W.H.: Optimal line-sweep-based decompositions for coverage algorithms. In: IEEE International Conference on Robotics and Automation, Proceedings 2001 ICRA, pp. 27–32. IEEE (2001)
5. Ryu, S.-W., Lee, Y.-H., Kuc, T.-Y., Ji, S.-H., Moon, Y.-S.: A search and coverage algorithm for mobile robot. In: 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 815–821. IEEE (2011)
6. Lee, T.-K., Baek, S.-H., Oh, S.-Y., Choi, Y.-H.: Complete coverage algorithm based on linked smooth spiral paths for mobile robots. In: 2010 11th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 609–614. IEEE (2010)
7. Gonzalez, E., Gerlein, E.: BSA-CM: A multi-robot coverage algorithm. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, WI-IAT 2009., pp. 383–386. IET (2009)
8. Larionova, S., Almeida, N., Marques, L., de Almeida, A.: Olfactory coordinated area coverage. Autonomous Robots 20(3), 251–260 (2006)
9. Choset, H.: Coverage for robotics–a survey of recent results. Annals of Mathematics and Artificial Intelligence 31(1-4), 113–126 (2001)
10. Gonzalez de Santos, P., Cobano, J.A., Garcia, E., Estremera, J., Armada, M.: A six-legged robot-based system for humanitarian demining missions. Mechatronics 17(8), 417–430 (2007)

11. Irwin, J., Kazmierkowski, M.P., Krishnan, R., Blaabjerg, F.: Control in power electronics: selected problems. Elsevier (2002)
12. Tal, J.: Speed control by phase-locked servo systems–new possibilities and limitations. IEEE Transactions on Industrial Electronics and Control Instrumentation (1), 118–125 (1977)
13. Furlan, M., Cernigoj, A., Boltezar, M.: A coupled electromagnetic-mechanical-acoustic model of a dc electric motor. COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering 22(4), 1155–1165 (2003)
14. Lewis, A.M., Bloodworth, T.J., Guelle, D.M., Littmann, F.R., Logreco, A., Pike, M.A.: Systematic test & evaluation of metal detectors (STEMD) interim report laboratory tests italy. Institute for the Protection and Security of the Citizen, Tech. Rep. (2006)
15. Robotics, C.: Husky tech specs, `http://www.clearpathrobotics.com/husky/tech-specs/`

Integrating Internode Measurements in Sum of Gaussians *Range Only* SLAM

A. Torres-González, J.R. Martinez-de-Dios, and A. Ollero*

Robotics, Vision and Control Research Group, University of Seville, Spain
{arturotorres,jdedios,aollero}@us.es

Abstract. This paper proposes a *Range Only*-SLAM method based on Sums of Gaussians (SoG) that integrates direct robot-beacon measurements together with measurements between static beacons. It exploits the fact that most commercial off-the-shelf beacons can communicate and organize into sensor networks and can compute range measurements to other beacons. The proposed method adopts a scheme based on Sums of Gaussians, which allows integrating direct robot-beacon measurements in an undelayed way, which is a significant advantage particularly in environments where the robot has bad odometry. The method has been implemented and validated in real experiments performed in the *CONET Robot-WSN Integrated Testbed*. The proposed method achieves a reduction of 70% in map error and significant improvement in robot pose accuracy when compared to traditional schemes.

Keywords: SLAM, *Range Only* SLAM, sensor networks, field robots.

1 Introduction

SLAM (Simultaneous Localization and Mapping) is a fundamental problem in robotics that has been researched for many years. It consists of a robot placed at an unknown environment which generates a map of this environment and at the same time it localizes itself on this map. Researchers have given many solutions to this problem using different kind of sensors, such as laser [8,5], cameras [13,16] and range sensors [7,11].

This paper deals with *Range Only (RO)* SLAM, which relies on sensors that provide range measurements between the robot and a set of beacons, assumed static. A variety of *RO*-SLAM methods based on different Bayesian Filters have been developed. *RO*-SLAM is subject to the so-called partial observation problem, which needs to take measurements from different locations in order to give a disambiguate estimation of the beacon position. Two main approaches have been adopted for beacon initialization, usually delaying the initialization until many measurements have been taken. In delayed schemes the EKF is combined

* This work was supported by the EC-SAFEMOBIL (European Commission ICT-2011-288082), CLEAR (DPI2011-28937-C02-01) and the Ministerio de Educación y Deportes FPU Program.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_35,

with auxiliary tools that perform beacon initialization such as Probability Grids [4] or Particle Filters [9]. Measurements cannot be integrated in the EKF until these tools have converged. On the other hand, undelayed SLAM schemes use multi-hypothesis tools such as Sum of Gaussians (SoG) [2,1] to integrate measurements in the filter with no delay, which involve important advantages.

Most of the *RO*-SLAM methods use only direct measurements between the robot and static beacons disregarding the fact that in many cases beacons can organize into sensor networks, can communicate with other beacons and can compute range measurements to other beacons. This is the case of using sensor network nodes as beacons. Sensor networks such as Wireless Sensor Networks (WSN) are comprised of nodes with sensing, computing and communication capabilities. SLAM has high interest for cooperation between robots and sensor networks, see e.g. [3,17,10]. Integrating internode measurements in SLAM can involve a number of advantages. Having static beacons, internode measurements can improve the accuracy of the map and, indirectly, also that of the robot. In contrast, very few SLAM methods that integrate internode measurements have been reported.

This paper proposes a *RO*-SLAM scheme based on Sums of Gaussians that integrates direct robot-beacon measurements and also internode measurements. This method adopts an undelayed approach when integrating direct robot-beacon measurements, which has interesting advantages particularly in environments where the robot has poor odometry. Together with direct robot-beacon measurements, the method also exploits internode measurements. It integrates direct robot-beacon measurements in an undelayed way while internode measurements are integrated after at least one of the two SoG (one per beacon of the internode measurement) has converged to one hypothesis. As another novel contribution, it dynamically adapts the SoG representation to reduce the length of the state vector in order to reduce computer burden. Our method has been implemented and validated in real experiments performed in the *CONET Robot-WSN Integrated Testbed* (`https://conet.us.es`) [6]. Results show that it overtakes existing schemes in: convergence times of the Sums of Gaussians, reduction of 70% in map error and significant improvement in robot pose accuracy.

The paper is structured as follows. A review of the related work and a summary the proposed approach is presented in Section 2. The main SoG EKF *RO*-SLAM method is described in Section 3 as an introduction to Section 4, which presents how internode measurements integrated in SoG EKF *RO*-SLAM. Section 5 presents real experimental results that validate the method and confirm its expected advantages. Conclusions is the final section of the paper.

2 Description of the Problem

2.1 Related Work

Range measurements have the problem of partial observability, i.e. several range measurements are necessary to disambiguate a location. In *RO*-SLAM partial

observability is addressed by adopting mainly two different approaches: delayed and undelayed mapping initialization.

In delayed initialization, new beacon nodes are introduced in the SLAM state vector when an external tool has an initial estimation of its pose. Examples of tools for delayed initialization are Particle Filters [9], Probability Grids [4,12] and simple trilateration [10]. Trilateration methods for beacon initialization although simple and efficient are very (too) sensitive to measurement noise. Trilateration was the first approach proposed but was soon discarded since the performance of SLAM is significantly sensitive to the accuracy in the beacon initialization. Probability Grids provide better initializations but their accuracy is related to the size and resolution of the grid, which also increase the computer burden. Particle Filters are the most widely applied method for delayed mapping. They provide better initialization accuracies than the other methods and a good number of mechanisms have been developed to reduce their burden. In delayed methods measurements cannot be integrated in the SLAM filter until the initial location of the nodes has been computed.

In undelayed initialization all the measurements gathered by the robot can be directly integrated in the SLAM filter. Thus, the complete SLAM algorithm can be implemented without any delay. These methods use multi-hypothesis estimation tools to introduce a new beacon node directly in the state vector without having any previous information. EKF based on Sum of Gaussians [1] adds to the state vector the mean of M Gaussians, each representing one different hypothesis of the node location. When a new measurements arrive, the weight of each Gaussian is updated. Gaussians with low weights are removed -the hypotheses are considered bad and they are discarded- until only one survives the pruning (hopefully the good one).

Various SLAM filters have been developed in *RO*-SLAM. EKF and RBPF (FastSLAM) are maybe the most widely applied. EKF SLAM [4,9] is a well known method, it is optimal in presence of Gaussian noise and obtains good results in most implementations. FastSLAM [1] factorizes the state vector dividing it in the vehicle pose estimation and the map estimation. It is more flexible than EKF due to the use of a Particle Filter as the core of the algorithm, which allows having different noise distributions apart from only Gaussian as in the case of Kalman Filters. In FastSLAM each particle of the filter represents an hypothesis of the robot pose and the map.

Also, the above methods use only direct measurements between the robot and the beacons, disregarding the fact that most commercial off-the-shelf beacons have computing and networking capability and can measure range to other beacons. However, none of the mentioned methods except for [4] make use of internode measurements. Djugash et al. proposed different ways for incorporating internode measurements in *RO*-SLAM mainly by improving map estimation using virtual nodes and adopting off-line map improvement using multidimensional scaling (MDS). However, these off-line SLAM solutions are not suitable for the majority of applications, which require on-line map and robot location estimations.

2.2 Proposed Approach

Consider a sensor network which nodes have been deployed at random locations in an GPS-denied environment. For instance, this is the case of a set of sensor nodes that have been thrown by an Unmanned Aerial Vehicle (UAV) for real-time monitoring of the status of a disaster or accident in an urban or industry area, where the buildings prevent correct GPS reception. The nodes equipped with range sensors and also with suitable sensors for monitoring the event, e.g. toxic gas concentration sensors in a pollution episode. Assume the nodes can measure the distance to the robot or to other nodes. They also have communication capabilities and computing capacity sufficient for executing simple communication protocols and simple calculations such as filtering.

In this scenario a ground robot uses RO-SLAM for creating a local map with the static nodes and self-localizing in this local map. Undelayed schemes can integrate range measurements in the SLAM filter after beacon initialization, which can take a significant time since it requires the convergence of auxiliary tools such as Particle Filters. In delayed schemes, before using measurements the robot pose is estimated using only the robot odometry, which can have important noise in the aforementioned environments resulting in high errors in robot pose estimation. In contrast, undelayed SLAM schemes integrate all range measurements received by the robot and the robot pose is estimated using the odometry and also the range measurements since the beginning, which is a significant advantage in cases with bad odometry. The proposed method uses Sum of Gaussians to initialize beacons in the EKF in an undelayed way.

The proposed method intends to enable the use of *RO*-SLAM in applications where robots cooperate with sensor networks. Thus, it integrates not only direct robot-beacon but also internode measurements exploiting the fact that nodes belong to a larger sensor network. Using internode measurements significantly reduces map uncertainty and also improves robot localization accuracy.

3 SoG-EKF SLAM

A Gaussian Mixture Model is a probability distribution that results of the linear combination of a number of Gaussian distributions (Sum of Gaussians). They are capable of representing any non-Gaussian distribution. Thus, they can integrate any non-Gaussian distribution into Gaussian filters. Using Kalman Filters with Gaussian Mixture Models is particularly interesting in *Range Only* SLAM. Each Gaussian in the mixture is an hypothesis of the target distribution, and the weight of the Gaussian w_i is the probability assigned to that hypothesis.

$$f_X(x) = \sum_{i=1}^{K} w_i N(\mu_i, \sigma_i), \tag{1}$$

where w_i represents the weight of each hypothesis, $N(\mu_i, \sigma_i)$ is a normal distribution with mean μ_i and variance σ_i and K is the number of hypotheses in the mixture.

In SLAM Gaussian mixtures allow integrating all measurements since the beginning with no delay. Thus, when the first measurement z_t of one node is taken by the robot, a number K of Gaussians are deployed such that the sum of the Gaussians represent the probability distribution of the location of the node. This probability distribution is an uniform annular distribution centered at the current robot location, with radius z_t (the range measurement value) and width σ_ρ proportional to the measurement noise. An example of the deployment of the Sum of Gaussians in a real experiment can be found in Section 5.

When approximating the annular distribution with the sum of Gaussians, the number of Gaussians is selected as a tradeoff between accuracy and efficiency. Like in Particle Filters, high number of Gaussians involve higher accuracy but also higher computer burden. We selected between 6 and 12 as a good balance. For the experiments described in the paper we used 8.

The SLAM state vector $\boldsymbol{x}$ consists of the vehicle location and orientation (x, y, θ) and the locations of each node in the map:

$$\boldsymbol{x} = [x\ y\ \theta\ \boldsymbol{b}_1\ \boldsymbol{b}_2 \dots \boldsymbol{b}_N]^T, \tag{2}$$

where $\boldsymbol{b}_i$ is the position of each node i. $\boldsymbol{b}_i$ is expressed in polar coordinates with respect to the location of the robot when the first measurement was taken (x_{ci}, y_{ci}). The Sum of Gaussians that were deployed when the robot was at location (x_{ci}, y_{ci}) is represented in polar coordinates as follows:

$$\boldsymbol{b}_i = [x_{ci}\ y_{ci}\ \rho_i\ \theta_{i1}\ \theta_{i2}\ \dots\ \theta_{iK}]^T, \tag{3}$$

where ρ_i is the radio of the distance between (x_{ci}, y_{ci}) and the center of the Gaussians and θ_{ij} are the angular coordinates of Gaussian j with respect to (x_{ci}, y_{ci}).

This polar representation requires $L = 3 + K$ components for each node in the state vector $\boldsymbol{x}$, while the traditional cartesian representation, which requires $L = 2K$ components for each node. This representation is more efficient with $K > 3$ but it is worse with $K < 3$. Thus, our method adopts dynamically the SoG representation for each node considering the number of current hypotheses, decreasing the size of the state vector as in the following equation and, thus, reducing the computational burden of the SLAM algorithm.

$$\begin{matrix} x_{ci} \\ y_{ci} \\ \rho_i \\ \theta_i \end{matrix} \Rightarrow \begin{cases} x_i = x_{ci} + \rho_i \cos\theta_i \\ y_i = y_{ci} + \rho_i \sin\theta_i \end{cases} \tag{4}$$

The complexity of the EKF is $O(n^2)$ [14], being n the number of states in the state vector. Thus, reducing the size of the state vector will make the complexity lower. We compare the number of components with polar and cartesian representations. If one node has 4 hypotheses, polar representation is more efficient since the node requires $L = 7$ components in the state vector in the polar representation and $L = 8$ components in the cartesian representation. On the other

hand, if the node has only one hypothesis, the cartesian representation is more efficient since it involves $L = 2$ while the polar representation involves $L = 4$ components. Thus, the proposed method selects dynamically the representation with the shorter state vector, reducing its computer burden.

The change in representation requires adapting the state vector and also the variance. In the state vector it only affects the mean of the Gaussians but it is more complex in the variance. Thus, here we make the approximation of considering the resulting variables as Gaussians. The new variance is taken as the upper bound of the previous variances:

$$\begin{cases} \sigma_{xi}^2 \approx \sigma_{xc}^2 + \sigma_{\rho}^2 + \sigma_{\theta}^2 \\ \sigma_{yi}^2 \approx \sigma_{yc}^2 + \sigma_{\rho}^2 + \sigma_{\theta}^2 \end{cases} \tag{5}$$

When the first measurement of one node (e.g. i) is taken, the K Gaussians -hypotheses- are added to the state vector using the aforementioned representation. The values of the mean, which depend on the number of hypotheses K, are defined in (6). All hypotheses are assigned with the same weight ($w_{ij} = 1/K$). Also, they are assigned with the same standard deviations, as expressed in (7).

$$\theta_{ij} = \frac{2\pi j}{K} \quad j = 1, \ldots, K \tag{6}$$

$$\sigma_{ij} = \frac{2\pi}{1.35K} \quad \forall j \in [1, K] \tag{7}$$

3.1 Prediction Phase

The prediction phase of the algorithm is executed like in as standard EKF SLAM. Since nodes are static, only the states corresponding to the robot change in this phase. The mobile robot is assumed to have differential motion configuration, but the SLAM method can work with any other kind of vehicle changing only the kinematic model. The kinematic model of a differential vehicle, such as those used in the experiments is as follows:

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, u_t) = \begin{bmatrix} x_{t-1} + \Delta t v_t \cos(\theta_{t-1}) \\ y_{t-1} + \Delta t v_t \sin(\theta_{t-1}) \\ \theta_{t-1} + \Delta t \alpha_t \end{bmatrix}, \tag{8}$$

where $(x_{t-1}, y_{t-1}, \theta_{t-1})$ is the robot's pose at time $t-1$, Δt is the sample time. v_t and α_t are the linear and steering velocities of the robot at time t.

The linearization of the kinematic model f is the matrix A, Jacobian of the function f:

$$A = \frac{\partial f}{\partial \boldsymbol{x}_{t-1}} = \begin{bmatrix} 1 & 0 & -\Delta t v_t \sin(\theta_{t-1}) \\ 0 & 1 & \Delta t v_t \cos(\theta_{t-1}) \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

3.2 Correction Phase

The range measurements between the robot and static node are used in the correction stage of the EKF SLAM. All the hypotheses of the corresponding node are affected in the correction phase. We adopt the approach considered in [2] and [15], where the correction of the estimate of a random variable by a set of measurements pairs (z, R_j) is equivalent to the unique correction by (z, R) if:

$$R^{-1} = \sum_{j=0}^{K} R_j^{-1}, \tag{10}$$

where R is the covariance matrix of the measurement noise and R_j is the covariance matrix of the measurement noise associated to hypothesis j.

Thus, one single measurement is divided into K new measurements, one for each hypothesis. The variance of each new measurement R_j is calculated as $R_j = R/\lambda_{ij}$, where λ_{ij} is a weight proportional to the likelihood l_{ij} of each hypothesis. The likelihood is modeled as a Gaussian distribution of the difference between the estimated distance to the corresponding hypothesis and the measurement z. The likelihood l_{ij} is also used to update the weights of each hypothesis w_{ij} as follows: $w_{ij} \leftarrow w_{ij} l_{ij}$. This step increases the weights of the hypotheses are close to the actual node position and decrease the weights of the hypotheses far from the node location.

The observation model adopted for range measurements is as follows:

$$h_i(\boldsymbol{x}_t) = r_i = \sqrt{\delta_x^2 + \delta_y^2}, \tag{11}$$

where δ_x and δ_y represent the distance in x or y between the robot and the corresponding hypothesis of node i. For simplification subindex t was obviated. The above model is used for each of the hypotheses. For instance, with hypothesis j of node i, $\delta_x = x_{ij} - x$ and $\delta_y = y_{ij} - y$. Notice that the location of hypothesis j of node i is obtained as follows:

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \begin{bmatrix} x_{ci} + \rho_i \cos\theta_{ij} \\ y_{ci} + \rho_i \sin\theta_{ij} \end{bmatrix} \tag{12}$$

Again, h is nonlinear and should be linearized by its Jacobian H:

$$H_i = \frac{\partial h_i}{\partial \boldsymbol{x}_t} = \begin{bmatrix} \frac{\delta_x}{r^i} & \frac{\delta_y}{r_i} & 0 \cdots & \frac{-\delta_x}{r_i} & \frac{-\delta_y}{r_i} & \cdots \end{bmatrix} \tag{13}$$

3.3 Hypothesis Pruning

When the first range measurement of a node is received by the robot the Sum of Gaussians is initialized as described above. As new measurements are integrated, bad hypotheses are removed until only one hypothesis survives the pruning and becomes the estimate of that node in the state vector. Two main rules to remove bad hypotheses are used in this method:

- Hypothesis j of node i is removed if the weight w_{ij} is smaller than a defined threshold (e.g. $0.00001/K$). This removes bad hypotheses, which have less probability of being the good one.
- One hypothesis is removed if the distance to another hypothesis is below another threshold. This is useful to remove hypotheses that are very similar one another. This distance threshold was selected experimentally, finding a good value in the following equation:

$$d < \min\left(\frac{\pi\rho}{2K}, 1.5\right) \tag{14}$$

  When two similar hypotheses are found, the one with lower weight will be removed and the other will stay without changes. This is useful to improve the efficiency reducing the size of the state vector without loosing information, as the pruned hypothesis gave approximately the same information than the other.

Of course, when one or more hypotheses are removed, the weight of the rest of the hypotheses is normalized in order to keep the relation $\sum_{i=1}^{K} w_i = 1$.

4 SLAM with Internode Measurements

This section describes the integration of internode measurements in the SoG-EKF SLAM method described in Section 3. As described in Section 2 we assume that beacons are integrated in nodes that belong to larger sensor networks and are endowed with sensing, computing and communication capabilities. Thus, it is possible to make not only direct robot-beacon measurements but also internode measurements between every pair of nodes in their suitable sensing coverage. The proposed scheme integrates direct robot-node range measurements together with internode measurements. The integration of such measurements is expected to provide the following improvements:

- With a higher number of measurements between static nodes at different locations, convergence of the Gaussian Mixture will have higher convergence speed, i.e. higher speed of pruning bad hypotheses.
- Being static nodes, integrating internode measurements enables higher accuracy in the estimation obtained by Gaussian Mixtures, which directly improves the map estimation.
- Having a better map will indirectly result in a higher accuracy in the estimation of the robot's pose.

It should be noticed that it is straightforward to design protocols in sensor networks for gathering internode range measurements. Also, these protocols can be straightforward implemented in different technologies, such as for instance Wireless Sensor Networks. The robot periodically broadcasts a measurement request message. The nodes within the robot's sensing range, we call them nodes A, take a measurement and save it in a table. Then, each node A broadcasts a

measurement request message. The nodes receiving this request take a measurement and send it back to the requesting node A in a reply message. Then, nodes A receive all the reply message, make a new range measurement and save it in its table. Next, they compute the mean value of measurements in the table and transmit them to the robot in a reply message. Now, the robot can integrate these measurements in the SLAM algorithm.

Section 3 describes the steps to integrate direct robot-node measurements in the SoG-EKF SLAM method. When an internode measurement between two static nodes i and j is gathered, different steps should be carried out depending on the current stage of the nodes involved in the measurement. If the Sum of Gaussians that represent the node estimate has converged to only one hypothesis, we say that the node is at the "EKF stage". When the Sum of Gaussians of the node has more than one hypothesis we say the node is at the "SoG stage". The integration of the internode measurements in the SoG-EKF is as follows:

1. If at least one node involved in the internode measurement is at the "EKF stage", the measurement is used to update the EKF just as a direct robot-beacon measurement, but changing the observation model in the following equations in order to center the model on node i instead of on the robot.

$$h_{ij} = r_{ij} \tag{15}$$

$$H_{ij} = \left[0\ 0\ 0 \cdots \frac{\delta_x}{r_{ij}}\ \frac{\delta_y}{r_{ij}} \cdots \frac{-\delta_x}{r_{ij}}\ \frac{-\delta_y}{r_{ij}} \cdots\right] \tag{16}$$

2. If only one of the nodes (e.g. i) is at the "EKF stage" and this is the first measurement received of the other node (j), a new Sum of Gaussians representing the probability distribution of the location of node j is initialized in a ring centered at node i.
3. If none of the nodes is at the "EKF stage" the measurement is stored for future use. They will be used as described above when one of the nodes enters the "EKF stage".

The proposed method integrates direct robot-beacon measurements in an undelayed way but cannot integrate internode measurements until at least one of the two SoG has converges to only one hypothesis. It should be noticed that this delay is not large since, as demonstrated in real experiments in Section 5, using internode measurements reduces significantly the convergence times of the Sums of Gaussians.

5 Experimental Results

Experiments with real hardware have been carried out in the *CONET Robot-WSN Integrated Testbed*. The testbed (`http://conet.us.es`) is a remote open tool to assess and compare multi-robot and WSN methods and algorithms [6].

It is comprised of 5 Pioneer AT robots and 140 WSN nodes -static and mobile- of different models (TelosB, MicaZ, Iris and Mica2). Each robot is equipped with a PC for processing all the data and controlling the robot, an Hokuyo 2D laser range sensor, a Microsoft Kinect camera, GPS and Inertial Measurement Unit, among others. The robot can self-localize with low error ($< 0.01m$), which has been considered as ground-truth for the experiments. The testbed uses an open and modular architecture and is installed since 2010 at the basement of the building of the School of Engineering of Seville (Spain). Figure 1 shows a picture taken during the experiments. The static nodes used are marked with a red circle.

Fig. 1. Picture taken during the validation experiments carried out in the *CONET Robot-WSN Integrated Testbed*

Series of experiments with different nodes locations and robot's paths have been performed in the *CONET Robot-WSN Integrated Testbed* in order to obtain real datasets. The sensor nodes deployed were equipped with *Nanotron NanoPAN* sensors. They have an error variance $\sigma_m^2 = 1m$ for indoor scenarios, measured experimentally in preliminary tests. SLAM methods provide the generated map and robot location in a local coordinate frame. For comparison with the ground-truth data, the map and robot's path estimated by the SLAM method was submitted to an affine transform that re-aligns the local solution into the global coordinate frame. The figures presented in this section use blue color to represent the ground-truth map and robot's path and red color to represent the estimations of the SLAM filter.

Figure 2 shows the results of executing the proposed SoG-EKF SLAM method with and without internode measurements in one of the experiments. Both schemes perform satisfactorily and can estimate the nodes location (map) and the robot's path, validating the SoG-EKF SLAM approach. Figure 2-right shows

how the use of internode measurements can improve the map accuracy and also the robot's path estimation. Similar results were obtained in all the experiments carried out: 10 series with different nodes locations and robot trajectories, each series with 15 repetitions of the same experiment.

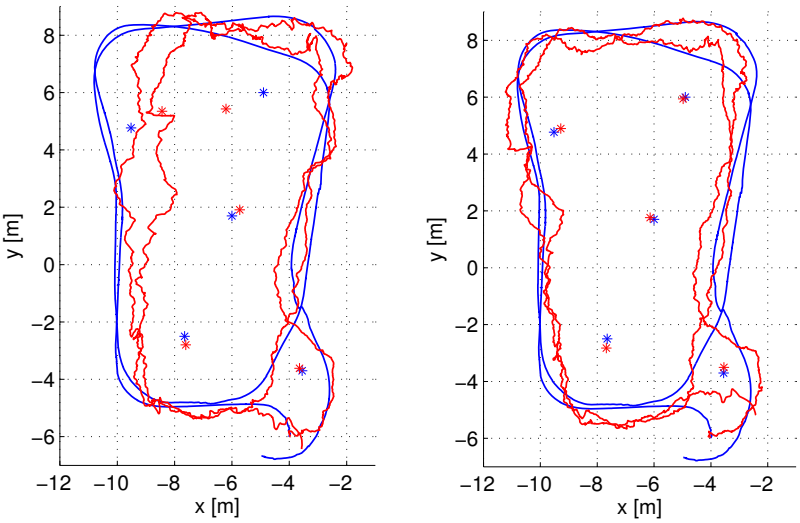


Fig. 2. Results with real experiments: Left) SoG-EKF without internode measurements; and Right) SoG-EKF with internode measurements

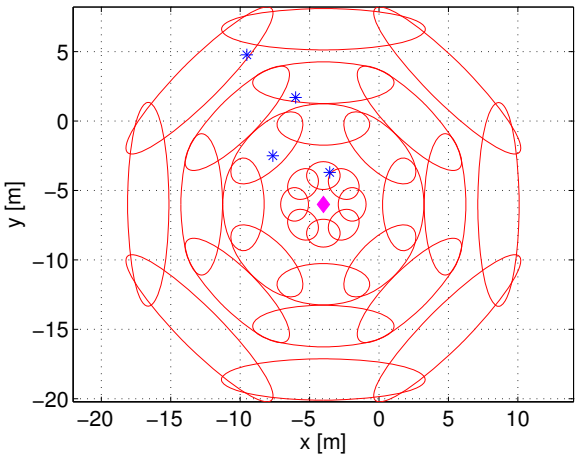


Fig. 3. Deployment of Gaussian mixtures in real experiments

Figure 3 shows the deployment of the Sum of Gaussians carried out by the robot (represented with a rhombus) after receiving measurements from different nodes (represented with asterisks). Each Gaussian represents an hypothesis on the node location. The interval confidence of each Gaussian is an ellipse.

Figure 4 represents the cumulate distribution of the robot localization error during the experiments using the SoG-EKF SLAM method with and without internode measurements. Each point in these curves is (e, p_e), meaning that a percentage p_e of the robot localization errors is lower than e. Without internode measurements mean error ($p_e = 0.5$) is 0.78 m and using internode measurements mean error becomes 0.66 m. Also, it can be noticed that 83.2% of the errors are lower than 1.5 m using traditional schemes without internode measurements whereas in the proposed method using internode measurements 99.9% are lower than 1.6 m. Also, the maximun robot localization error is reduced from 3.1 m (without internode mesurements) to 1.5 m (with them).

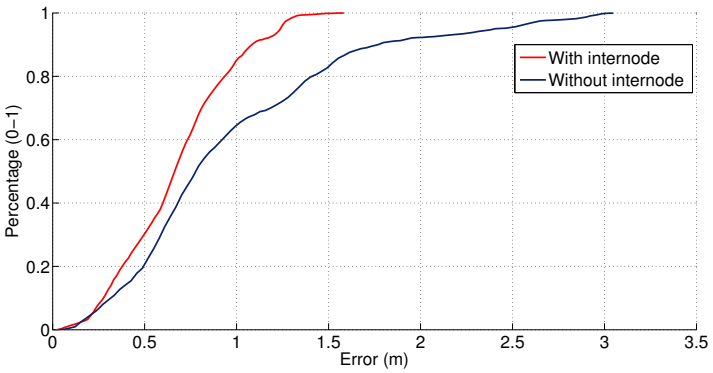


Fig. 4. Cumulate distribution of the errors on robot position estimation with the proposed method with and without internode measurements

Internode measurements are very useful to reduce the map error. Figure 5 shows the temporal evolution of the error in the estimation of the location of each beacon node since each Sum of Gaussian converges into a single hypothesis. Using internode measurements allows earlier convergence of the Sums of Gaussians, which is particularly evident for the case of Node 3. Internode measurements help the pruning of bad hypotheses making the Gaussian Mixture to converge faster into only one single hypothesis. Earlier convergence helps to reduce the computational burden since with low number of hypotheses, the proposed method adapts dynamically the SoG representation -recall Section 3- to a more efficient representation with shorter state vector.

In this example the mean of the node location estimated errors is $0.68m$ without internode measurements and $0.2m$ with them. Similar results were obtained in all the performed experiments: using internode measurements reduces map estimation error 70%. It can also be noticed in Fig. 5 that using internode

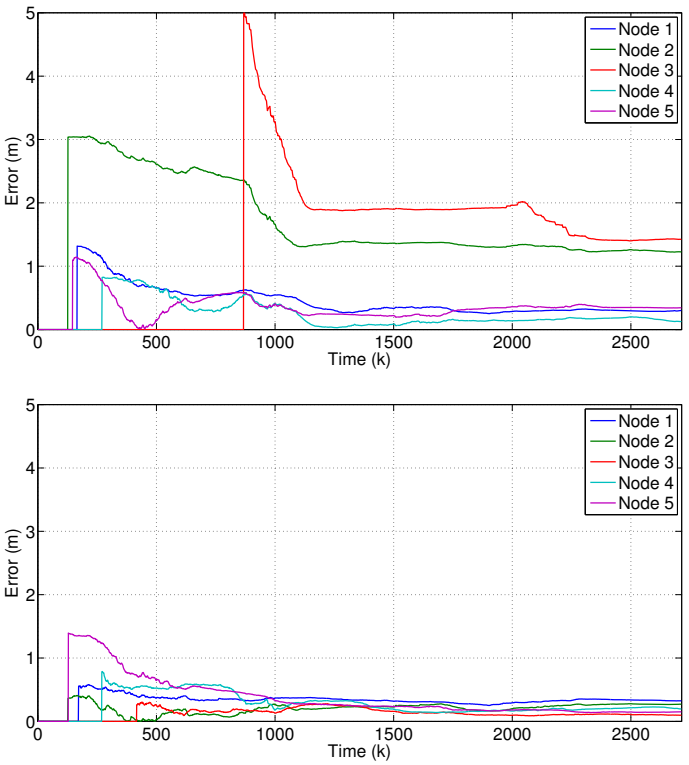


Fig. 5. Errors on map estimation of the SoG-EKF with (bottom) and without (top) internode measurements

measurements anticipate the convergence of the SoG. In the series of experiments carried out, internode measurements accelerated SoG convergence in 17%.

6 Conclusions

This paper proposes a *RO*-SLAM scheme based on Sums of Gaussians that integrates direct robot-beacon measurements and also internode measurements. Internode measurements can significantly reduce the uncertainty in the map estimation, which indirectly improves estimation also of the robot pose. Also, with higher number of measurements, convergence of the Gaussian Mixture will have higher convergence speed.

The proposed method has two main contributions. First, it dynamically adapts SoG representation in order to reduce computational burden. Second, it integrates direct robot-beacon and also internode measurements.

The method has been implemented and validated in real experiments performed in the *CONET Robot-WSN Integrated Testbed* (`https://conet.us.es`)

[6]. Extensive tests with different series of experiments with different robot paths and nodes locations have been performed. The results obtained confirm that the presented method overtakes existing SoG with no internode measurements in terms of earlier convergence of Sums of Gaussians, reduction of 70% in map error and significant improvement in robot pose accuracy.

References

1. Blanco, J.L., Fernandez-Madrigal, J.A., Gonzalez, J.: Efficient probabilistic range-only slam. In: Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, pp. 1017–1022 (September 2008)
2. Caballero, F., Merino, L., Ollero, A.: A general gaussian-mixture approach for range-only mapping using multiple hypotheses. In: Proc. IEEE Intl. Conf. on Robotics and Automation, pp. 4404–4409 (May 2010)
3. Challa, S., Leipold, F., Deshpande, S., Liu, M.: Simultaneous localization and mapping in wireless sensor networks. In: Intelligent Sensors, Sensor Networks and Information Processing Conference 2005, pp. 81–87 (December 2005)
4. Djugash, J., Singh, S., Kantor, G., Zhang, W.: Range-only slam for robots operating cooperatively with sensor networks. In: Proc. of the 2006 IEEE Intl. Conf. on Robotics and Automation, pp. 2078–2084 (May 2006)
5. Estrada, C., Neira, J., Tardos, J.: Hierarchical slam: Real-time accurate mapping of large environments. IEEE Transactions on Robotics 21(4), 588–596 (2005)
6. Jiménez-González, A., Martínez-de Dios, J.R., Ollero, A.: An integrated testbed for cooperative perception with heterogeneous mobile and static sensors. Sensors 11(12), 11516–11543 (2011)
7. Kantor, G., Singh, S.: Preliminary results in range-only localization and mapping. In: Proc. of the 2002 IEEE Intl. Conf. on Robotics and Automation., vol. 2, pp. 1818–1823 (2002)
8. Liu, Y., Thrun, S.: Results for outdoor-slam using sparse extended information filters. In: IEEE International Conference on Robotics and Automation, Proceedings. ICRA 2003, vol. 1, pp. 1227–1233 (2003)
9. Menegatti, E., Danieletto, M., Mina, M., Pretto, A., Bardella, A., Zanconato, S., Zanuttigh, P., Zanella, A.: Autonomous discovery, localization and recognition of smart objects through wsn and image features. In: 2010 IEEE GLOBECOM Workshops (GC Wkshps), pp. 1653–1657 (December 2010)
10. Menegatti, E., Zanella, A., Zilli, S., Zorzi, F., Pagello, E.: Range-only slam with a mobile robot and a wireless sensor networks. In: Proc. IEEE Intl. Conf. on Robotics and Automation, pp. 8–14 (May 2009)
11. Newman, P., Leonard, J.: Pure range-only sub-sea slam. In: Proc. of the 2003 IEEE Intl. Conf. on Robotics and Automation, vol. 2, pp. 1921–1926 (2003)
12. Olson, E., Leonard, J., Teller, S.: Robust range-only beacon localization. In: 2004 IEEE/OES Autonomous Underwater Vehicles, pp. 66–75 (June 2004)
13. Pinies, P., Tardos, J.: Large-scale slam building conditionally independent local maps: Application to monocular vision. IEEE Transactions on Robotics 24(5), 1094–1106 (2008)
14. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics, 3rd edn. The MIT Press, Cambridge (2005)

15. Sola, J., Monin, A., Devy, M., Lemaire, T.: Undelayed initialization in bearing only slam. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 2499–2504 (2005)
16. Steder, B., Grisetti, G., Stachniss, C., Burgard, W.: Visual slam for flying vehicles. IEEE Transactions on Robotics 24(5), 1088–1093 (2008)
17. Sun, D., Kleiner, A., Wendt, T.: Multi-robot range-only slam by active sensor nodes for urban search and rescue. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS, vol. 5399, pp. 318–330. Springer, Heidelberg (2009)

RiskRRT-Based Planning For Interception of Moving Objects in Complex Environments

Mario Garzón[1], Efstathios P. Fotiadis[1],
Antonio Barrientos[1], and Anne Spalanzani[2,3]

[1] Centro De Automática y Robótica, UPM-CSIC,
Calle José Gutiérrez Abascal, 2. 28006 Madrid, Spain
[2] Université Pierre-Mendès-France - Grenoble II
[3] INRIA Rhône-Alpes,
655 Avenue de l'Europe, 38334 Saint Ismier Cedex, France
{ma.garzon,efstathios.fotiadis,antonio.barrientos}@upm.es,
anne.spalanzani@inria.fr

Abstract. This paper presents a new approach for the interception of moving objects using UGVs in large complex environments. The planning for interception is based on the *Risk-RRT* algorithm. Several modifications have been made to the base algorithm to enhance its ability to move in uncertain environments. The planner is integrated with a navigation architecture. The full system is capable of parallel on-line planning and following of the path. It performs the interception and at the same time it avoids static and dynamic obstacles. Several tests, both in simulation and with real world robots, were carried out showing the effectiveness of the proposed system.

Keywords: Moving Objects Interception, RiskRRT, Planning for Interception, UGV Navigation.

1 Introduction

The interception of moving objects using unmanned ground vehicles (UGV) is a very important task. It can enhance the capabilities of a system for autonomous surveillance of critical infrastructures and can also be used in many other applications for robotic systems. Following and intercepting moving objects is a low-level function for humans, but when translating these capabilities to autonomous robots a lot of challenges appear. Those challenges can be divided in four main tasks: detecting the target, identifying or predicting its trajectory, generating a trajectory to intercept it and finally, executing the trajectory. The work presented here is focused on two last tasks from the previous list. Moreover, both the detection technique and the object to be intercepted are indifferent at this level.

Since this development is intended to be used in surveillance of large critical infrastructures, it should be able to generate and follow a trajectory for intercepting a moving target in a complex environment. This means that, the system

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_36,

must be able to handle the presence of both static and dynamic obstacles, as well as uncertainties and changes in the trajectory of the target.

The proposed schema is based on an algorithm originally used for planning in dynamic and uncertain environments[1]. After several modifications on the original algorithm, it was possible to take advantage of its probabilistic representation of the environment and the moving obstacles, and use it for intercepting moving objects instead of avoiding them. The main contribution of the proposed algorithm is the capability of finding a path that will likely intercept a given target, which is moving in a complex environment. Additionally, the algorithm is integrated into a well defined navigation architecture. The resulting system is able to perform on-line motion planning and navigation for an UGV and to intercept a moving target in a complex environment.

The paper is organized as follows. After this introduction, in the next section, previous work related is briefly discussed. In Section 3 The RiskRRT algorithm is described in detail. The software architecture is introduced in Section 4. A description of the robotic mobile platform used for the test is found in Sections 5. Then, Section 6 explains the experiments held in order to validate the method and the results obtained. Finally, Section 7 presents the conclusions of the work.

2 Related Work

In this section, a short review on the techniques used for following and intercepting moving objects using mobile robots is made. Several approaches have been proposed to solve this problem. Certain works are based on computer vision, such as visual-based behaviours[2], or visual servoing[3,4] and, more recently, on fusion combining image data with laser scanners[5]. Those developments are much more based on the detection of moving object and in general they use simple proportional control laws, or go-straight and turn behaviours, in order to follow and intercept the target. A more complex method, which also uses guidance-based visual servoing, overcomes the limitation of not being able to avoid obstacles by combining rendezvous-guidance and velocity-obstacle approaches[6], a later work from the same authors improves the proportional navigation technique by using an advanced predictive guidance law that integrates the non-linear pursuer and target equations forward in time[7].

A different approach is based in potential field methods. It generates motion planning for an UGV in an environment where the target and obstacles are moving[8]. Using a similar technique, a different work performs velocity planning for a robot in order to track a moving target[9]. In both works, the attractive and repulsive potential are generated by taking into account the relative position and velocity of the target and the obstacles with respect to the robot. Nevertheless, those methods suffer from the typical local minima problem characteristic of the potential fields approach.

Artificial Intelligence techniques have been also used to solve this problem. A combination of fuzzy target tracking and fuzzy wall following schemes is used in order to follow a target and avoid obstacles[10]. Also, a different system uses a

control law based on geometrical rules, combined with the kinematics of the UGV and the target, to intercept a moving object[11]. The interception of a group of mobile targets by a team of robotic pursuers have been also studied, one work integrates genetic algorithms and fuzzy logic controllers to solve it[12]. Another work uses centralized optimization techniques for pursuer-target assignments and time-optimal guidance for interception[13].

Most of the previous works, in order to intercept a moving object, rely on reactive techniques and use minimum to none path planning, without taking into account uncertainties. The performance of those techniques is not suitable for the application defined in this case. Therefore, a different approach should be used in order to obtain a system that fulfills all this requirements.

3 The Risk-RRT Algorithm

This section describes briefly the *Risk-RRT* algorithm, which is the base of the interception system proposed in this paper. *Risk-RRT* is a motion planning strategy proposed by *Fulgenzi et al.*[14]. It was designed to operate in dynamic and uncertain environments. Moreover, the risk of collision is calculated using probabilistic models that map both the static and dynamic models. Also, the search of the paths is based on an extension of the *Rapidly-exploring Random Tree* algorithm[15], that takes into account the likelihood of the future position of the obstacles and the before-mentioned probability of collision.

3.1 Probabilistic Risk of Collision

In the original implementation of the *Risk-RRT*, the objective is to safely reach a particular configuration q_N. In order to find a safe path towards the goal, it is necessary to avoid colliding with static or mobile objects. Therefore, the algorithm must compute the probability of collision P_c when the robot is in a given configuration $q(t_1)$ an action $u \in U$ is taken. This risk is calculated by taking into account the probability of occupation of the surface A that is swept when the robot moves from $q(t_1)$ under the control of u in the time interval $[t_1, t_2]$

$$q(t_2) = f(q(t_1), u, \tau) \tag{1}$$

$$A = \iint_{t_1}^{t_2} q(t)dt \tag{2}$$

Where $f(.)$ is the motion model of the robot and $\tau = t_2 - t_1$ is the time step. As mentioned before, the risk of collision must include both static and mobile objects. Also, it is assumed that the obstacles cannot overlap themselves. Therefore, the collision with static obstacles and the collision with the moving obstacles are mutually exclusive events, which yields:

$$P_c = P_{cs} + (1 - P_{cs}) \cdot P_{cd} \tag{3}$$

Where P_{cs} and P_{cd} are the probabilities of collision due to static and dynamic obstacles respectively. The static obstacles are represented as occupied cells in the occupancy grid M, which is obtained from the SLAM algorithm (see Section 4.2) and is assumed to be stationary. Additionally, a subset $\varsigma \subset M$ of cells, that correspond to the minimal approximation of the surface A, is defined. The risk of collision with static obstacles can be expressed as the maximum probability of collision over the subset ς:

$$\varsigma = \{(x, y) \in M, (x, y) \cap A \neq 0\} \tag{4}$$

$$P_{cs} = \max_{\varsigma}(P(Occ(x, y) = 1) \tag{5}$$

On the other hand, the probability of collision dynamic obstacles P_{cd} is the result of the combination of the probability of collision with each singe moving obstacle $P_{cd}(o_m)$:

$$P_{cd} = 1 - \prod_{m=1}^{M} 1 - P_{cd}(o_m) \tag{6}$$

And the risk of collision with a moving obstacle o_m can be approximated by the probability of an interception between two areas, one swept by the robot and the other by the moving obstacle, in a given time interval:

$$P_{cd}(o_m) = P(o_m(t) \cap A \neq 0, \forall t \in [t_1, t_2]) \tag{7}$$

The path performed by the moving obstacle $o_m(t)$ can be predicted using different approaches[14], but it is out of the scope of this work. Moreover, it is assumed to be an input of the proposed algorithm. The *Risk-RRT* algorithm uses the probabilities of collision in order to choose a safe path. Therefore, it needs to translate the risk of possible actions to the risk of possible paths:

$$\pi(q_N) = \{q_n\}_{n=0\ldots N} \mid q_{n+1} = f(q_n, u_n, \tau) \tag{8}$$

$$P_{c,\pi}(q_N) = 1 - \prod_{n=1}^{N}(1 - P_c(q_n)) \tag{9}$$

$$L_{\pi}(q_N) = 1 - P_{c,\pi}(q_N) = \prod_{n=1}^{N}(1 - P_c(q_n)) \tag{10}$$

Where q_0 is the current position of the robot and $\pi(q_N)$ represents the path from q_0 to q_N. Also, $L_{\pi}(q_N)$ represents the probability of success or *Likelihood* (i.e. probability of not encountering a collision) of the path. In conclusion, this computation allows the system to determine whether or not a determined trajectory will be safe. Also, it will allow performing comparisons between one or more paths in terms of which one will have higher risk of collision.

3.2 Determining the Best Path for Interception

For this work, the objective is to intercept one of those mobile objects. Therefore, assuming that the path followed by the moving object $o_m(t)$ is computed externally and given as an input to the algorithm, it is possible to use the probability of collision defined in equation 3 in order to determine whether or not a route that will likely intercept the moving object.

To find the possible paths that may intercept the target the configuration-time space has to be searched and every path should be evaluated. The search is performed using the well known Rapidly Exploring Random Tree[15]. This algorithm starts sampling random points in the configuration-time space. It then builds a tree T of reachable configurations. In order to build that tree, it tries to extended the tree towards a new configuration q_{i+1} from the closer point already in the tree. At the same time, a weight value $w(q)$ is computed for each node of the tree q (i.e. each configuration of the search space). The construction of the tree continues for a defined time. When the tree is completed, or the available time for planning is over, the node with the highest weight is selected as the next destination and the path to reach it is extracted out of the tree.

As mentioned before, the original implementation of the *Risk-RRT* algorithm has the objective of avoiding both the dynamic and the static obstacles. Therefore, the weight of the nodes, for that case, combines the risk of collision and the estimated length of the total path:

$$\tilde{w}(q_N) = \frac{\sqrt[N]{L_\pi(q_N)}}{dist(q_0, q_1 \dots q_N)} \tag{11}$$

$$w(q_N) = \frac{\tilde{w}(q_N)}{\Sigma_q \tilde{w}_q} \tag{12}$$

At the numerator of equation 11, the likelihood of the path $\pi(q_N)$ is normalized with respect to the length of the path N. At the denominator, $dist(.)$ is the sum of the length of the path from the initial node q_0 to the node q_N. In equation 12, the weight used to select the best node is obtained by normalizing the weights $\tilde{w}(q_N)$ over the set of nodes of the tree.

For the case of the interception, which is the main objective of this work, it is necessary to define a different weight for each node. It should involve the probability of collision, but as opposite to the previous case, a higher probability is desirable. Moreover, for this case only the probability of collision with a single mobile obstacle, defined in equation 7 should be considered, otherwise the robot may go towards a wall or any other fixed obstacle. A first approach to the definition of the weight for each node can be easily given by the probability of dynamic collision with respect to the path of the target $\chi(t)$:

$$w_I(q_N) = P_{cd}(\chi) \tag{13}$$

The selection of the node with the highest w_I will allow the robot to move towards the configuration where it is more likely to intercept the target. But

there is a drawback when this weight is used, which is the case where no node in the tree has a probable collision, or all the probabilities found are very low. To solve this, it is necessary to use the standard weight of the *Risk-RRT*, when there is no probability of collision:

$$w_I(q_N) = \begin{cases} P_{cd}(\chi),\ P_{cd}(\chi) > \varepsilon \\ w(q_N),\ P_{cd}(\chi) = \varepsilon \end{cases} \tag{14}$$

This new mixed definition of the weight will allow the system to achieve a mixed behavior. When the search finds a probable interception point, it will move in that direction, but if the search is unsuccessful it will move towards a temporary goal. This temporary target will allow the robot to be at a configuration where the *RiskRRT* will have more chances of finding a node with probable collision.

To select the temporary goal, the system takes advantage from the fact that the predicted path of the moving object is available. So, it is possible to set as temporary goal the farthest point of the predicted path.

3.3 On-line Decisions Update

The procedure for updating the tree when new information from the trajectory of the target is received is explained in this subsection. In order to be able to handle the variations of the movements of the target and the other components of the environment, it is necessary update the tree on-line, otherwise the planning will no longer be valid. Additionally, since the time available for the planning is limited, the path obtained in one planning cycle may be incomplete or a partial part of the trajectory.

The planning and execution of the trajectory are done in parallel, this means that while the robot moves along the planned path, the weights of each node of the tree are updated with new information received. Additionally, the tree is grown and after each time step a new evaluation is performed, if the algorithm finds a better path, it sends that new data to the local controller, otherwise the previous one is maintained.

4 Software Architecture

This section describes the software architecture used to controls all the components of the UGV as well as the high level algorithms for mapping and navigation, including the proposed schema to intercept moving targets. The entire software is supported by the software framework ROS (Robot Operating System), which comprises of libraries and tools that facilitate the development of new robot applications. A schematic diagram of the proposed software platform is found in Figure 1. All important modules, inputs and outputs, as well as their respective connections are shown. The dashed lines represent the modules and connections used only for the SLAM algorithm.

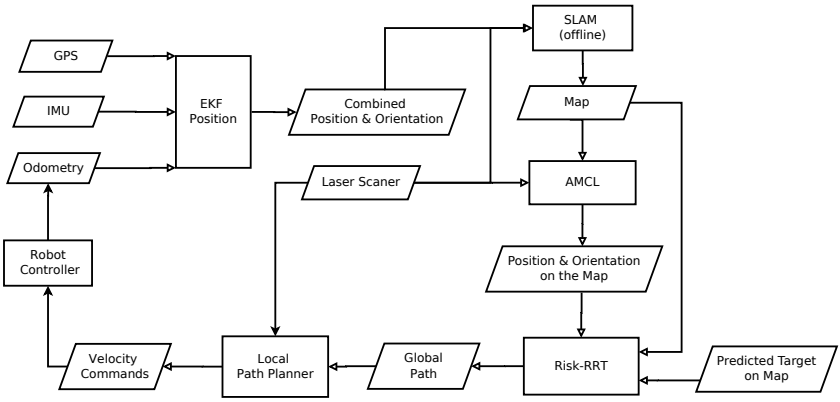


Fig. 1. Schematic diagram of the software architecture. All important modules, inputs and outputs, as well as their respective connections are shown. The dashed lines represent the modules and connections used only for the SLAM algorithm.

4.1 EKF Combined Odometry

A crucial component of our architecture consists of how the robot position and orientation are estimated. Readings from the wheel encoders are fused with GPS signal and Inertial Motion Unit data to produce the combined odometry output.

The fusion is done by an Extended Kalman Filter (EKF) approach, used to estimate the three-dimensional pose (position and orientation) of the robot. Once the robot odometry, GPS and IMU readings are received and their corresponding transformations are set, it is possible to establish a measurement model that links the measurements of each sensor with the global position of the mobile robot. Those models are first combined using probability fusion techniques and then expressed as another probability density function. A detailed explanation of this process can be found in [16].

The estimated pose can be obtained with the data available at each given moment, even if one sensor stops sending information. Also, if the information is received after a time-out, it is disregarded.

4.2 Navigation

The main navigation algorithms are a series of modules that, when combined, provide the UGV with the ability of performing autonomous navigation. Here each one of the modules will be briefly described. The first module is used for creating maps using readings from the laser and the combined odometry, which was described in Section 4.1, providing Simultaneous Localization and Mapping (SLAM). Another module is used for handling and publishing the maps, a third one is in charge of the localization within a given map and finally, the main component that performs path planning and following.

Fig. 2. The complete Summit XL robotic platform. The on-board computer and IMU are inside the chassis. The long range Hokuyo laser and the FireFly camera are mounted on a pedestal to facilitate the detection of the external objects. The shorter range laser rangefinder is below the robot and allows avoiding lower obstacles.

The navigation scheme proposed is based on navigating on a known map, thus building this map is a main part of the overall process. The algorithm uses raw laser range data and the combined odometry from the EKF in order to build grid maps using a highly efficient Rao-Blackwellized particle filter [17].

After the map is constructed it must be available to all the other elements of the system. This functionality if provided by the map server. The maps are based on gray scale images, where each pixel represents the occupancy state of each cell of the world; white pixels represent free space, black ones represent occupied cells and gray ones represent unknown spaces. The values are loaded from the images and published in one or more occupancy grids.

The localization of the robot is based on Adaptive Monte-Carlo Localization (AMCL) proposed in [18]. It works as a probabilistic localization system, using a particle filter to track the pose of the UGV in a known map. It relays on the information provided by the scans of laser rangefinder and then combined odometry data. The AMCL publishes the estimated pose of the robot in the map with its corresponding covariance and the set of pose estimates being maintained by the filter.

The path planning algorithms are crucial for the two-dimensional navigation in our case. They take a goal in the world and attempt to reach it with a mobile base. The planning is separated in two phases: global and local; each one of them

uses its own cost map. The global planner uses the *Risk-RRT* algorithm in order to scan the state space and it is explained in detail in Section 3.

The local planner provides a controller that helps the robot to follow the global path created by the global planner. It uses a smaller local occupancy map, which is constantly redefined around the current pose of the robot. This planner creates a kinematic trajectory for the robot, so it can get from a starting point to a local goal pose. Moreover, the algorithm provides the linear and angular velocities that will allow the platform to safely traverse the local occupancy grid. The local goal pose is selected as the farthest point of the global path that is covered by the local map. Furthermore it takes into account static and dynamic obstacles that may be detected by the on-board sensors. The velocity commands are sent directly to the controller of the robot.

5 Platform Description

This section describes the hardware components of platform used in the execution of the experiments. It is comprised of a mobile platform and various on-board sensors.

The platform is based on the Summit XL®robotic platform by Robotnik®. It has skid-steering kinematics based on four high efficiency motors. The robot can move autonomously or it can be teleoperated using video feed from an on-board camera. Furthermore, it is equipped with a small form factor PC which allows deploying all the data processing and navigation algorithms in a fully autonomous manner.

Various sensor modalities are also present. The odometry is provided by an encoder on each wheel and a high precision angular sensor assembled inside the chassis.

Two Hokuyo™laser rangefinders are mounted on the platform: the first one is an UTM-30LX-EW, it can scan a 270°semicircular field, with a guaranteed range that goes from 0.1 to 30 meters and a maximum output frequency of 40Hz. The second laser rangefinder is an URG-04LX, it has a semicircular scanning area of 240°; the guaranteed range for this device goes from 0.06 to 4.09 meters and it can operate at a maximum frequency of 10 Hz. They are placed at different heights, with the 4 meters rangefinder at 10 centimeters from the ground in the front of the robot and the 30 meters laser at 60 centimeters over the ground in the central part of the robot.

A Novatel OEM-4 GPS engine is also used; it can offer centimeter level positioning accuracy with a frequency of 2Hz. RS232 serial communication is used to read the incoming data and send correction commands. The engine is complemented with an ANT-A72GOLA-TW GPS antenna.

The MicroStrain 3DM-GX3 25 which is a high-performance, miniature Attitude Heading Reference System (AHRS) is mounted inside the robot. It combines accelerometers, gyroscopes and magnetometers in the three axes, with temperature sensors and an embedded processor to provide static and dynamic orientation, as well as inertial measurements.

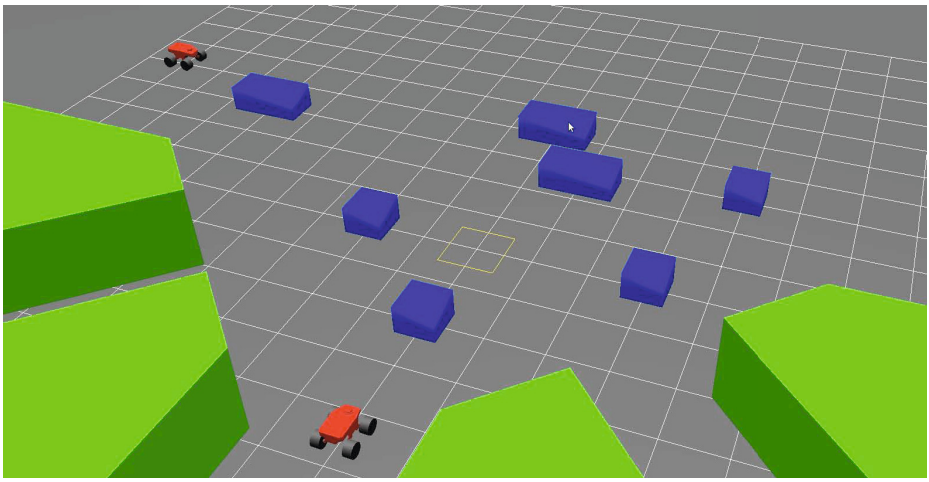


Fig. 3. Virtual simulation scenario. Starting position of the simulation experiments. The robot that is shown to the left of the figure is the target and the one at the bottom is the pursuer. Walls are depicted in green and obstacles with blue.

An image of the robot with all the aforementioned equipment mounted is shown in Figure 2.

6 Experiments and Results

This section describes the experiments that were held to assess the performance of the proposed method. These experiments are divided in two categories: simulations and real world experiments. As explained before, the detection of the target is not addressed in this work. In order to obtain the input list for the Risk-RRT algorithm, an object with a known position on the map is used. The prediction of future poses of the target is based on an EKF filter, that projects the velocity to a given future time-lapse.

6.1 Simulations

Before deploying our algorithms to the physical robot, simulations of the algorithms were run into a computer. This is very helpful for prototyping because it facilitates debugging and parametrization while at the same time prevents from hardware malfunction due to software misuse. For the simulations two identical robots are used, one as a target and one as interceptor. The robot model used was that of Summit XL described in detail in Section 5. Since both are equipped with a laser sensor and the can do self localization on the map and no initial calibration was needed. In order to realize the experiments a fictitious virtual scenario was use improvised. This scenario includes various obstacles between the two robots and the starting position can be seen in Figure 3. The target robot

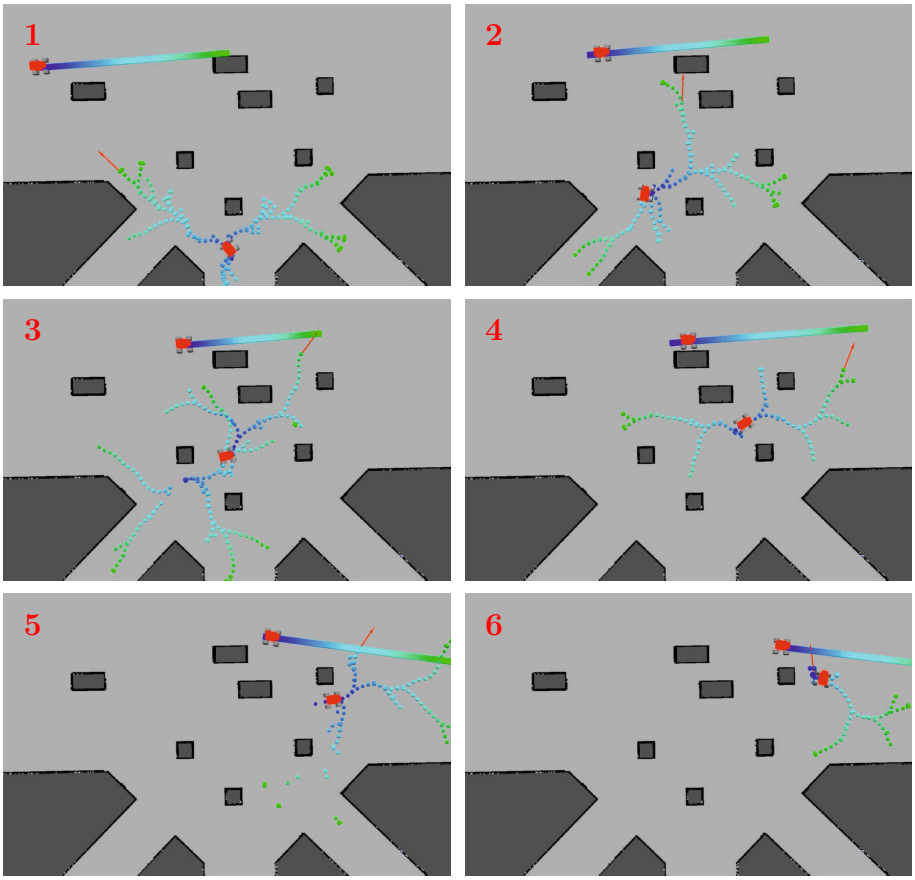


Fig. 4. Progress of the interception during the simulation

moves most of the time in straight line outside of the obstacle perimeter. The path which is predicted using odometry and EKF, is used by the pursuing robot in order to populate the Risk-RRT algorithm. At a first stage of the experiment, the pursuer is planning to move towards a point in the middle of the trajectory of the target. While the experiment evolves, the pursuer is delayed trying to evade the obstacles and the target robot continues its path. At some point the pursuer realizes that, it can no longer intercept the target in the initial section, so it changes the global plan in order to reach the target at a further section of its projected trajectory. Finally, while the target keeps moving, the pursuer manages to intercept it. The evolution of the experiment can be visualized in Figure 4.

6.2 Real World Experiments

Apart from the simulations, various experiments were also made using the actual robotic platform and the respective sensors described in Section 5. Those

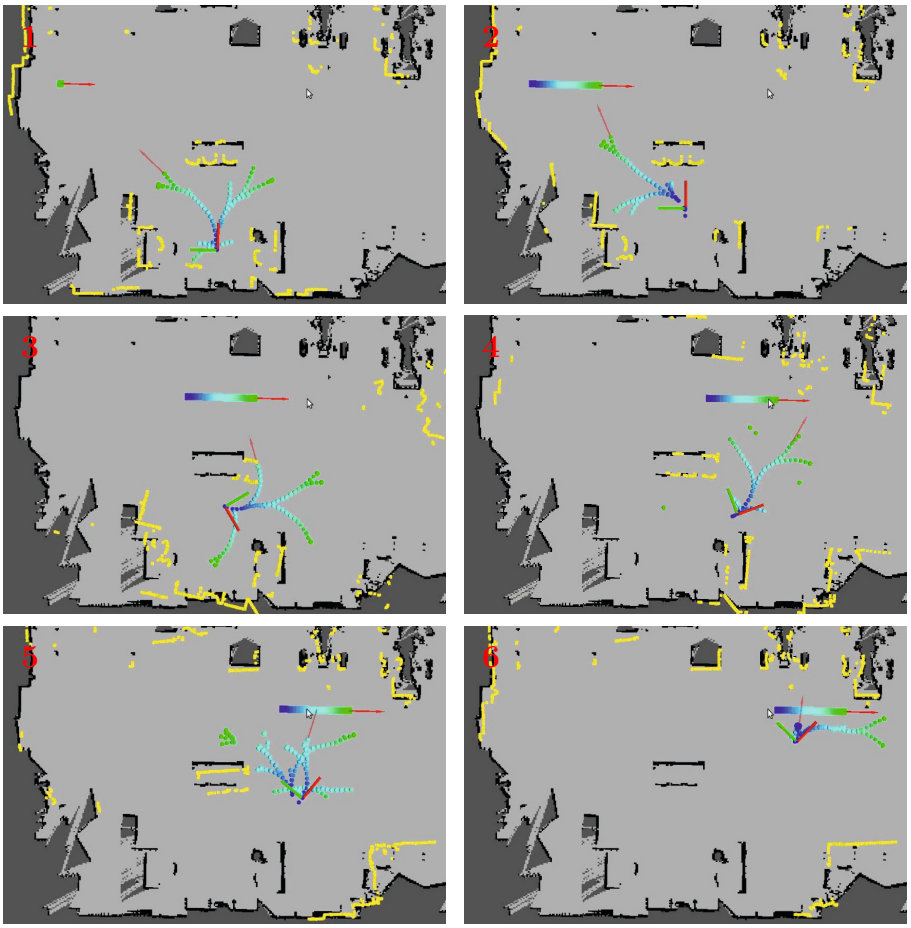


Fig. 5. Progress of the interception during the real world experiment

experiments took place at the facilities of the Center of Automation and Robotics in Arganda del Rey, Madrid. This time a Summit-XL robot is used as the pursuer and a Pioneer 3AT®as the target. Since the Pioneer robot is not equipped with laser, it is necessary to place it in a known position and orientation on the map and use only odometry for its localization. Furthermore the setting is simples with only an obstacle between the pursuer and the trajectory of the target. It should be noticed that, since the experiment was held inside a building, there was no GPS signal thus only the IMU was used to compute the combined odometry. For this experiment the target robot was manually teleoperated moving in a more or less straight line. When the experiment began the target was stopped so pursuing robot started moving towards it. After a brief period the target started moving and the EKF produced a projected trajectory. Using *Risk-RRT* the pursuer was constantly calculating a new path according to the movement

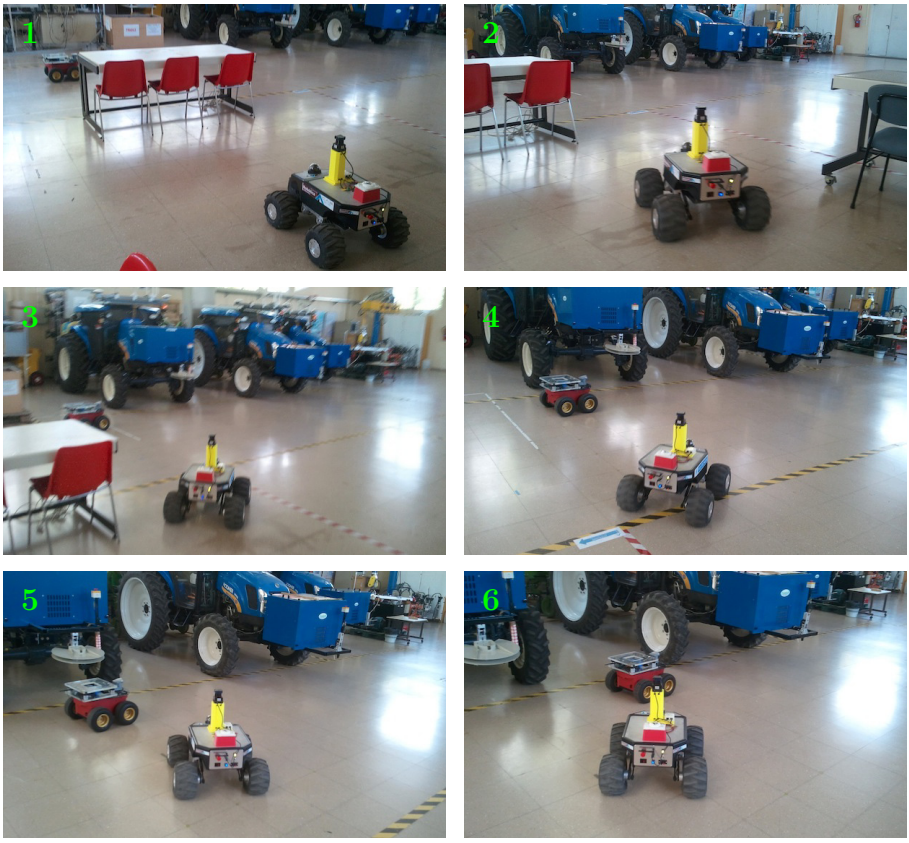


Fig. 6. Photo sequence of the interception during the real world experiment

of the target. Furthermore, at some point it became obvious to the algorithm that the obstacle is interfering with the targets trajectory. Then a new plan was created towards the projected position of the target and the obstacle was effectively overtaken allowing the interception of the target. It should be noticed that, in order to avoid damage on the equipment, there is no real collission and the interception is finished when the UGV is at a small distance from the target. The realization of the experiment is visualized in Figure 5 and a photo sequence of the interception is shown in Figure 6.

Both virtual and real world experiments have shown that our algorithm can effectively be used for the interception of a target even when the environment is full of obstacles that intervene between the pursuer and the trajectory of the target.

7 Conclusions

A system for interception of moving objects, based on techniques for path planning under uncertainties is presented. The planning algorithm is integrated in a

well established navigation schema and the resulting system is capable of performing the interception in complex environment. The system is designed in a modular manner and it is independent from the detection method. Moreover, the predicted path of the target is defined as an input to the algorithm. The planning algorithm is based on the *Risk-RRT* algorithm, which is a probabilistic extension of the *RRT*. The configuration used performs on-line updating of the tree and the corresponding weight of each node in it. By doing so, it allows the parallel execution of planning and following of the path. Moreover, the basic navigation schema allows the system to handle other undetected static or dynamic obstacles. The performance of the system was tested first in a simulated scenario and then using real robots. The results show that the proposed technique can successfully intercept the target even if it moves in complex scenarios.

Acknowledgments. This work was supported by the Robotics and Cybernetics Group at Universidad Politécnica de Madrid (Spain), and funded under the projects: ROTOS - Multi-robot system for outdoor infrastructures protection, sponsored by Spanish Ministry of Education and Science (DPI2010-17998), and ROBOCITY 2030, sponsored by the Community of Madrid (S-0505/DPI/000235).

References

1. Fulgenzi, C., Tay, C., Spalanzani, A., Laugier, C.: Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 1056–1062 (2008)
2. Santos-Victor, J., Sandini, G.: Visual behaviors for docking. Computer Vision and Image Understanding 67(3), 223–238 (1997)
3. Feyrer, S., Zell, A.: Detection, tracking, and pursuit of humans with an autonomous mobile robot. In: Proceedings. 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 1999, vol. 2, pp. 864–869 (1999)
4. Capparella, F., Freda, L., Malagnino, M., Oriolo, G.: Visual servoing of a wheeled mobile robot for intercepting a moving object. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 2737–2743 (2005)
5. Kobilarov, M., Sukhatme, G., Hyams, J., Batavia, P.: People tracking and following with mobile robot using an omnidirectional camera and a laser. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 557–562 (2006)
6. Kunwar, F., Benhabib, B.: Rendezvous-guidance trajectory planning for robotic dynamic obstacle avoidance and interception. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36(6), 1432–1441 (2006)
7. Kunwar, F., Sheridan, P.K., Benhabib, B.: Predictive guidance-based navigation for mobile robots: A novel strategy for target interception on realistic terrains. Journal of Intelligent and Robotic Systems 59(3-4), 367–398 (2010)
8. Ge, S., Cui, Y.: Dynamic motion planning for mobile robots using potential field method. Autonomous Robots 13(3), 207–222 (2002)
9. Huang, L.: Velocity planning for a mobile robot to track a moving target - a potential field approach. Robotics and Autonomous Systems 57(1), 55–63 (2009)

10. Li, T., Chang, S.J., Tong, W.: Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. IEEE Transactions on Fuzzy Systems 12(4), 491–501 (2004)
11. Belkhouche, F., Belkhouche, B., Rastgoufard, P.: Line of sight robot navigation toward a moving goal. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36(2), 255–267 (2006)
12. Jeong, I.K., Lee, J.J.: Evolving fuzzy logic controllers for multiple mobile robots solving a continuous pursuit problem. In: IEEE International Fuzzy Systems Conference Proceedings, FUZZ-IEEE 1999, vol. 2, pp. 685–690 (1999)
13. Sheridan, P., Kosicki, P., Liu, C., Nejat, G., Benhabib, B.: On-line task allocation for the robotic interception of multiple targets in dynamic settings. In: 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 291–296 (2010)
14. Fulgenzi, C., Spalanzani, A., Laugier, C., Tay, C.: Risk based motion planning and navigation in uncertain dynamic environment. Rapport de recherche (October 2010)
15. LaValle, S., Kuffner, J.J.: Randomized kinodynamic planning. In: Proceedings. 1999 IEEE International Conference on Robotics and Automation, vol. 1, pp. 473–479 (1999)
16. Garzón, M., Valente, J., Zapata, D., Barrientos, A.: An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas. Sensors 13(1), 1247–1267 (2013)
17. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Transactions on Robotics 23(1), 34–46 (2007)
18. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. In: AAAI/IAAI, pp. 343–349 (1999)

Analysis of Methods for Playing Human Robot Hide-and-Seek in a Simple Real World Urban Environment

Alex Goldhoorn, Alberto Sanfeliu, and René Alquézar

Institut de Robòtica i Informàtica Industrial, CSIC-UPC,
C/ Llorens i Artigas 4-6, 08028 Barcelona, Spain
{agoldhoorn,sanfeliu,ralqueza}@iri.upc.edu
http://www.iri.upc.edu

Abstract. The hide-and-seek game has many interesting aspects for studying cognitive functions in robots and the interactions between mobile robots and humans. Some MOMDP (Mixed Observable Markovian Decision Processes) models and a heuristic-based method are proposed and evaluated as an automated seeker. MOMDPs are used because the hider's position is not always known (partially observable), and the seeker's position is fully observable. The MOMDP model is used in an off-line method for which two reward functions are tried. Because the time complexity of this model grows exponentially with the number of (partially observable) states, an on-line hierarchical MOMDP model was proposed to handle bigger maps. To reduce the states in the on-line method a robot centered segmentation is used. In addition to extensive simulations, games with a human hider and a real mobile robot as a seeker have been done in a simple urban environment.

Keywords: Robotics, Human Robot Interaction, Hide-and-Seek, MOMDP.

1 Introduction

Small differences of the hide-and-seek game can be found in the interactions between humans, or robots and humans, in urban spaces, as for example looking for a person in a crowded urban environment. In 2005 Johansson and Balkenius [1] suggested that the game of hide-and-seek is an ideal domain for studying cognitive functions in robots and it is a basic mechanism for human robot interaction in mobile robotics, because hide-and-seek requires the robot to navigate, search, interact on and predict actions of the opponent. In this work we focus on the prediction of the opponent's actions. This could be used to find persons; furthermore hide-and-seek could be seen as a simplification of search-and-rescue.

The hide-and-seek game is an interactive game in which the two players interact indirectly, one trying to catch while the other is trying to flee (the seeker is a robot and the hider is a person in our case). Players of the game can follow several strategies to win the game, depending on their role. The robot's strategy could be simply pre-programmed, but a more intelligent approach would be

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_37,

to decide a strategy which can be applied in multiple situations. In this game, as in real life, there are uncertainties of the location of the other player. The later is a reason why we chose MOMDPs (Mixed Observability Markov Decision Processes) [2,3], a variant of Partially Observable Markov Decision Processes (POMDPs) [4,5]. POMDPs have been successfully applied to various robotic tasks [6,7], but unfortunately, computing an optimal policy exactly is generally intractable [8] because the size of the belief space grows exponentially with the number of states. POMDPs have been also used in [2,3,9] to play the game *tag*, a variant of hide-and-seek.

In our MOMDP approach of the hide-and-seek game the robot and the person move at the same time from one position to another (given by the (x, y) coordinates of the grid cells where they are located). The robot updates a belief (a probabilistic state estimate) and chooses an action (a robot movement) to maximize the expected future reward, meanwhile the person chooses also an action following its own strategy to win the game.

MOMDPs are POMDPs in which the partly and fully observable state variables are separated. In our game we assume that the robot's position is fully observable, and the hider position is not always visible (partially observable). This results in a compact lower-dimensional representation of its belief space.

In this work, we analyze and apply an off-line and on-line MOMDP model as proposed in [10]. This off-line model works very well for maps with a small number of grid cells, but it becomes intractable (PSPACE-hard, [8]) for a large number of grid cells. For this reason we proposed an on-line MOMDP model [10] that computes a locally near-optimal policy at every step, which in principle can be applied to large maps. The on-line method is a hierarchical model of two levels, where the top level MOMDP has a reduced number of states. The state reduction is obtained through a segmentation process of the map. The bottom level contains a fine resolution of the map in which the beliefs are computed. Finding a policy however is done with the top level MOMDP. The on-line method can also be applied to navigation problems, or other like problems where a high resolution map is used. Finally an automated heuristic seeker is tested for comparison.

All seeker methods have been tested in simulation and in real life experiments using a real robot (Tibi and Dabo [11,12]) against a human hider in a simple urban environment (see Figure 3).

2 Definition of the Hide-and-Seek Game

Our version of the hide-and-seek game is defined as follows. There are two players, a seeker and a hider, who play on a grid of $n \times m$ cells. The grid contains: a special free cell called the *base*, other *free cells* on which the players can move, and *obstacle cells* that are not accessible by the players and also limit their mutual visibility. In the initial state of the game, the seeker is placed on the base and the hider can be placed on any free cell not visible from the base.

The game is run for a maximum of H time steps. At each time step, both the seeker and the hider can stay in the same cell or move to a free neighbor cell in

an 8-connectivity neighborhood (i.e. a maximum of 9 actions for each player). The seeker wins if it approaches the hider sufficiently (we use a distance of 1 cell) and "catches" it. The hider wins if it reaches the base before being caught by the seeker. And the result is a *tie* when no player has won within the maximum predefined time H or if both reach the base at the same time. Orientation angles are not considered for the players, they are both supposed to have 360° visibility at each time step, only limited by the obstacles. Hence, the visibility for each player is calculated with a ray-tracing algorithm in simulation or with a range laser in the real world.

The MOMDPs presented in the next two sections model the game from the point of view of the seeker, this is, we want to learn a policy for the seeker assuming that the hider follows a certain unknown strategy. It is also assumed that the seeker's state is fully observable for itself (no local uncertainty), whereas the hider's state is partially observable, in the sense that the hider's position is identified if the hider is visible from the seeker's position and otherwise it is *unknown* for the seeker.

In our simulations, two virtual robots are involved: an automated seeker applying the MOMDP learnt policy or using a heuristic is confronted with a random or a "smart" (heuristically driven) hider. In our real-world experiments, a physical robot (Dabo) has the role of the seeker and plays against a human opponent in the role of the hider.

3 Off-line MOMDP Model for Hide-and-Seek

The hide-and-seek game can be cast as an off-line MOMDP model [2,3], where the state is composed by the grid cell positions of both players. This means that the number of states is the square of the number of grid cells of the 2D map where the game is going to be played. The number of grid cells depends on the resolution that we want to consider in the game (e.g., a grid cell of 1×1 m^2 in a 2D map of 10×10 m^2 implies 10 000 MOMPD states). Formally, the hide-and-seek game is modelled as:

$$\langle \mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{O}_\mathcal{X}, \mathcal{O}_\mathcal{Y}, T_\mathcal{X}, T_\mathcal{Y}, Z_\mathcal{X}, Z_\mathcal{Y}, R, \gamma \rangle \tag{1}$$

where:

- $\mathcal{X}$: the fully-observable state variable that contains the seeker's position $x = (x_{\text{seeker}}, y_{\text{seeker}})$;
- $\mathcal{Y}$: the partially-observable state variable containing the hider's position $y = (x_{\text{hider}}, y_{\text{hider}})$;
- $\mathcal{A}$: the 9 actions of the seeker: *north, northwest, west, ..., halt.* Each of the actions represents a movement of one grid cell at maximum per time step, except for the action *halt* which represents staying at the same state;
- $\mathcal{O}_\mathcal{X}$: $\mathcal{O}_\mathcal{X} = \mathcal{X}$, since $o_x = x$ for all states of the seeker, note that this is added for completeness and in reality x is used directly instead of o_x;

- $\mathcal{O}_\mathcal{Y}$: $\mathcal{O}_\mathcal{Y} = \mathcal{Y} \cup \{unknown\}$, which is the union of the set of hider positions and a special observation value *unknown*, which represents the cases when the hider is not visible to the seeker;
- $T_\mathcal{X}$, $T_\mathcal{X}(x, y, a, x') = p(x'|x, y, a)$: the transition probabilities of the seeker's state given an action. In our case the actions are deterministic for the seeker's position: given the current position x, action a, and the map it brings the seeker directly to a new state x', independently of the current position of the hider y. Therefore these probabilities will always be 1 or 0, taking into account that the result of an infeasible action is defined as staying on the same cell. For example, when the seeker has a wall in the north of it and it chooses the action *north*, the action cannot be done and therefore it will result in not modifying the seeker's state. Also reaching the final state will result in staying in the same state;
- $T_\mathcal{Y}$: the transition probabilities of the hider's state given a seeker's action and locations of the seeker and hider, $T_\mathcal{Y}(x, y, a, x', y') = p(y'|x, y, a, x')$. These probabilities are not as evident as the previous ones since the action of the hider is not known. There are two suggested solutions: the first is to spread the probabilities of the movement of the hider uniformly, the second option is to use historical data of human players. Both options are discussed in detail in [13], but since the results with historical data were not significantly better, we chose for the easier option of using uniform probabilities. Also here the probability will be 1 if a final state has been reached;
- $Z_\mathcal{X}$: the observation probabilities $Z_\mathcal{X}(x', y', a, o_x) = p(o_x|x', y', a)$ will be 1 if $o_x = x'$ and 0 otherwise, like $\mathcal{O}_\mathcal{X}$ this is put here for completeness;
- $Z_\mathcal{Y}$: the observation probabilities $Z_\mathcal{Y}(x', y', a, o_x, o_y) = p(o_y|x', y', a, o_x)$ depend on the locations of the seeker and hider and the map. The probability will be 1 if $o_y = y'$ and y' is visible from x', or if $o_y =$*unknown* and y' is not visible from x', otherwise the probability will be 0;
- R: two reward functions have been tested (see below);
- γ: the discount factor.

Two different reward functions R are described next; these give rise to two different off-line MOMDP models, from which a near-optimal policy can be learnt off-line [2,3]. These functions are:

- *Simple* reward: non-zero values only for final states (positive for $x = y$ and negative for $y =$*base*, $x \neq y$).
- *Triangle* reward: this reward makes use of the three important distances in the game: the distance between the seeker and the hider (d_{sh}), the distance between the hider and the base (d_{hb}) and between the seeker and the base and (d_{sb}). The reward is shown in Equation (2), where D is a maximum distance constant depending on the map size. In order to compute the three distances one may use the simple Euclidean distance or the length of the shortest path that depends on the map [13]. We have used the last one for our experiments.

$$R(s,h,b) = \begin{cases} D - d_{sh}, & \text{if } d_{hb} > d_{sb} \\ -d_{sh}, & \text{otherwise} \end{cases} \tag{2}$$

While the triangle reward is much more informative than the simple reward, its computational cost is also slightly higher. Note that the simple reward can be computed extremely fast at each step without the need of memorising its values for each state. On the other hand, for the triangle reward, either its values are precalculated for each state (higher memory cost) or the computation time is increased considerably if calculated at each step.

Finally we have to define the initial belief $b_{Y,0}$: if the hider is visible then the belief of that state is 1.0, otherwise the belief is uniformly distributed over the not visible states.

4 On-line MOMDP Model for the Hide-and-Seek Game

The issue with the off-line method is that it takes a relatively long time to generate a policy (from 2 hours for maps of 12×12 up to more than 40 hours for maps of 20×20). Furthermore the time and memory complexity grows with the number of states due to the curse of history and dimensionality [14].

We present a hierarchical model, as shown in Figure 1(a), in which the lower level is an MOMDP as defined in the previous section. The big difference is that this MOMDP is not used to calculate the policy, but instead the top level MOMDP with less states is used. The state reduction of the top level MOMDP is obtained by grouping a spatially adjacent group of positions in the bottom level map. In the top MOMDP, the transition and observation probabilities, and the initial belief will be calculated from those in the bottom MOMDP. The top MOMDP will be solved on-line and directly thereafter used to choose the best action to do. Furthermore the bottom level is used to keep track of the belief. The actions are common to both levels.

4.1 Bottom-Level MOMDP

The bottom level is a full MOMDP (as (1)) defined in the same way as described in Section 3, however no policy is computed at this level. Only the beliefs of all the states at this level, are computed before generating the top-level MOMDP. The belief is initialized as in the off-line version, and when an action a has been executed (whereby the seeker's position changed from x to x') and an observation (o_x and o_y) has been obtained, the bottom-level belief $b_{\mathcal{Y}}$ is updated:

$$\begin{aligned} b'_{\mathcal{Y}}(y') &= \eta p(o_x|x',y',a) p(o_y|x',y',a,o_x) \\ &\quad \times \sum_{y \in \mathcal{Y}} p(x'|x,y,a) p(y',x,y,a,x') b_{\mathcal{Y}}(y) \end{aligned} \tag{3}$$

where $\eta = 1/p(o|b,a)$ is a normalization factor.

4.2 Top-Level MOMDP

To reduce the number of states a segmentation function ψ is used that groups adjacent map cells. This segmentation is used to generate the new top state variables $\mathcal{Y}_T$. Where each value of y_T will be associated with a spatially adjacent set of values of y. Formally, the function $\psi(y_T)$ gives the set of bottom-level adjacent states which are covered by each of the top level state y_T. When reducing the number of partially observable states $\mathcal{Y}$, the belief space is reduced. Also the fully observable state variable $\mathcal{X}$ could be reduced in the same way, but this did not give significant better results nor did it increase the speed in finding the policy.

The problem of finding a proper function ψ can be posed as a segmentation based on the map itself, the location of the players, the reward obtained in each state and/or the belief of each state.

The segmentation can be done by applying some known image segmentation algorithm such as k-means [15] where a fixed set of k clusters is defined in the "intensity value" domain. In any case, the number of segmentation regions obtained should be limited to assure a small number of states.

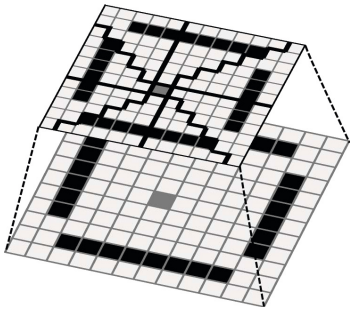
Robot Centered Segmentation. We propose a method that centers on the robot and divides the space based on the eight directions seen from the robot and the distance. Figure 1 shows the robot centered segmentation in which the robot is at location 0, and the segmentation is done from that point in the eight directions and based on a fixed distance to the center. The focus of this segmentation is the direction and not the exact location; which is sufficient because a new robot centered top MOMDP model is generated for each step. Since the hider and base positions are of vital importance for the game, they are added as a separate superstate if known; these superstates will represent only one cell in the bottom level.

Top MOMDP. The top-level MOMDP can be defined as follows:

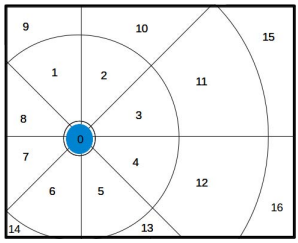
$$\langle \mathcal{X}_T, \mathcal{Y}_T, \mathcal{A}_T, \mathcal{O}_{\mathcal{X},T}, \mathcal{O}_{\mathcal{Y},T}, T_{\mathcal{X},T}, T_{\mathcal{Y},T}, Z_{\mathcal{X},T}, Z_{\mathcal{Y},T}, R_T, \gamma \rangle \tag{4}$$

where some of the top MOMDP components will be equal to those of the bottom level MOMDP:

- $\mathcal{X}_T = \mathcal{X}$;
- $A_T = A$, the actions keep referring to the lower level actions, in the top level however the transition probability is adapted to abstract top level states;
- $\mathcal{O}_{\mathcal{X},T} = \mathcal{O}_{\mathcal{X}}$;
- $\mathcal{O}_{\mathcal{Y},T} = \mathcal{O}_{\mathcal{Y}}$, these observations do not change, but their probabilities $Z_{\mathcal{Y},T}$ do change;
- $Z_{\mathcal{X},T} = Z_{\mathcal{X}}$.



(a) Layered segmentation



(b) Robot centered segmentation

Fig. 1. The hierarchical method with two layers is shown in (a), where the top layer has less states due to segmentation of the lower level map. The *robot centered* segmentation (b) centers on the robot's location (0 in the figure) and from there on creates segments based on the direction and distance.

Therefore, the Top MOMDP reduces to the following tuple:

$$\langle \mathcal{X}, \mathcal{Y}_T, \mathcal{A}, \mathcal{O}_\mathcal{X}, \mathcal{O}_\mathcal{Y}, T_{\mathcal{X},T}, T_{\mathcal{Y},T}, Z_\mathcal{X}, Z_{\mathcal{Y},T}, R_T, \gamma \rangle \tag{5}$$

where the transition and observation probabilities and rewards are averaged from the bottom level:

$$p(x'|x, y_T, a) = \frac{1}{|\psi(y_T)|} \sum_{y \in \psi(y_T)} p(x'|x, y, a) \tag{6}$$

$$p(y'_T \mid x, y_T, a, x') = \frac{1}{|\psi(y_T)|} \sum_{y' \in \psi(y'_T)} \sum_{y \in \psi(y_T)} p(y'|x, y, a, x') \tag{7}$$

Note that in our implementation we have not made the seeker's action (x) dependent on the hider's position (y), therefore equation (6) will not change in our case, but has been put here for completeness.

To speed up the process of finding a good policy the final state is defined to stay in the same state independent of the action a: $p(x_f|x_f, y_{T,f}, a) = 1.0$ and $p(y_{T,f}|x_f, y_{T,f}, a, x_f) = 1.0$ where $(x_f, y_{T,f})$ is a final state. The final state is defined as either $y_{T,f}$ being on the base, or if $x_f \in \psi(y_{T,f})$, i.e. the seeker is in the same superstate as the hider.

The observation probability is simply averaged:

$$p(o_y|x', y'_T, a) = \frac{1}{|\psi(y'_T)|} \sum_{y' \in \psi(y'_T)} p(o_y|x', y', a) \tag{8}$$

The top reward function $R_T(x, y_T, a)$ could have been defined averaging the rewards of the bottom level, however it was found that using only rewards in

the final states converged much quicker in a good policy. A reward of 1 is given when the seeker is in the hider's superstate ($x \in \psi(y_T)$), and -1 if the hider is at the base.

Before a policy is learned, the bottom level belief is compressed to the top level:

$$b_{\mathcal{Y},0,T}(y_T) = \sum_{y \in \psi(y_T)} b_{\mathcal{Y}}(y) \tag{9}$$

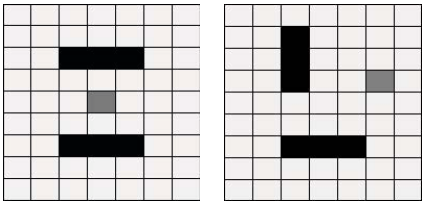
Note that besides the $\mathcal{Y}$ component, also the $\mathcal{X}$ and O components can be segmented; however experiments [10] showed that this did not give better results nor faster convergence.

4.3 The On-line Algorithm

The algorithm for solving the on-line two-level MOMDP is based on [16] and uses SARSOP to generate a policy [9,2]. SARSOP is a state-of-the-art off-line solver for POMDPs, but can be used on-line by simply alternating a planning and an execution phase [17]. Algorithm 1 shows how the on-line method is implemented. First the bottom belief is initialized based on the seeker position (x) and the belief of the hider position (b_Y), which will make the belief 1.0 on its visible position, otherwise it will be uniformly distributed over all non-visible cells. From now on the algorithm is run until a final condition is reached: some player wins or the time has passed. The segmented hider states are calculated in line 3; here we apply the robot centered segmentation (Figure 1). In line 4 the belief is compressed up using formula (9), where after the top level MOMDP M_T is generated from the bottom level MOMDP M and the segmented states S_T using the formulas presented in the previous subsection. In line 6 the policy P_T is learned and applied to get the best action. When the action is done an observation of the seeker's own position and the hider's position is done in line 9, which is used to update the bottom level belief.

Algorithm 1. On-line two-level MOMDP planner.

```
1: (b_Y, x) ← initBelief(M)
2: while not finished game do
3:     S_T ← segmentStates(M, x)
4:     b_{Y,T} ← compressBelief(M, S_T, b_Y, x)
5:     M_T ← generateTopLevel(M, S_T)
6:     P_T ← solveTopLevel(M_T, b_{Y,T}, x)
7:     a ← getBestAction(M_T, P_T, b_{Y,T}, x)
8:     doAction(a)
9:     (b_Y, x) ← doObservation()
10:    (b_Y, x) ← updateBelief(b_Y, x, o_y)
11: end while
```



(a) Map 1 (b) Map 2

Fig. 2. The maps used in the simulated and real experiments. Black cells are obstacles, the dark gray cell is the base.

5 Smart Seeker

Using the previously defined reward (2) an automated heuristic seeker has been made, called the *Smart Seeker*. This seeker calculates a score for each action it can take and then chooses the action with the maximum score. At maximum 9 actions are possible (one step or staying at the same position) for both the seeker and hider. One action gets the seeker to a position, which can be used to calculate R; but at the same time the hider can make a move, which we take into account by averaging the score over these moves:

$$w(s', h, b) = \sum_{h' \in \text{moves}(h)} R(s', h', b)/|\text{moves}(h)| \tag{10}$$

When the hider is not visible to the seeker the only thing we know is that the hider is at a not visible position; therefore the score w is calculated for every possible hider's position and then averaged.

6 Simulations and Experiments

In this section the set-up of the simulations and the real world experiments are explained.

6.1 Maps

In order to do the simulations and real-world experiments, we created a discretized 2D grid map of the environment, where the players can move in one of the 8 directions.

For both the simulations and the real-world experiments we built a small square in between two buildings with a size of 7 m × 9 m. The size of the grid cells is 1 m which implies a grid size of 7 × 9 cells. Two obstacles with a length of three cells (3 m) have been placed on different positions as well as the base. We used the two maps which can be seen in Figure 2, and for simulations we used the same maps but with a higher resolution: 9 × 12.

6.2 Simulations

We have first run a series of simulations playing against a *random hider*, which moves completely randomly, and a *smart hider*. The latter uses the triangle rule and uses the following equation to score each possible action: $w = D - d_{hb} + 0.4d_{sh} + noise$ where $D = \text{rows} \times \text{cols}$, d_{hb} the distance between the hider and the base, and d_{sh} the distance between the seeker and the hider. The noise has a maximum value of 2 and is reduced as soon as the distance is less than 3, because when a hider is either close to the seeker or to the base, it should respectively always flee or go to the base directly.

We have used the hierarchical and the off-line MOMDP models. To generate the policies we used the *Approximate POMDP Planning Software* [1] [2]. The simulations where done on a stand alone PC with 8 GB of RAM and an Intel Core^{TM}i5 CPU 760 @ 2.80 GHz with 4 cores and Ubuntu 12.04 as OS.

We set up a maximum time (measured in time steps) to play the game which is computed based on the size of the map: $2(rows + cols)$, that means 32 time steps for our simulations. The win condition for the robot has been adapted to its size. The robot wins if it is within one cell distance of the hider.

6.3 Real-World Experiments

The robot Dabo of the URUS project [11,12] has been used to play the hide-and-seek game. Dabo has two on-board computers (Intel Core 2 Quad CPU @ 2.66 and 3.00 GHz with 4 GB RAM) that manage all the running processes and sensor signals, and a laptop is used for external monitoring. The systems run Ubuntu 12.04 and use a middle ware called ROS, a software developmental environment for robot system integration that provides a useful and large set of libraries and tools.

A map was created by the robot by scanning the environment using the robot range-lasers. For the game, the Dabo robot used the front and back range-lasers to know its position and to detect persons and obstacles [11,12]. The robot movements were given by the seeker algorithm that uses the MOMDP models. To detect persons we used a classifier developed in the IRI based on the range-laser information.

Since the seeker, the Dabo robot, has been designed to work in a limited controlled environment, and because these experiments were a first step in the real-world, we had to impose some constraints in the hide-and-seek game. First of all, the robot and the person were only allowed to do one action at the same time step in one of the eight directions (or no motion) and they could move at most one grid cell. The grid cells were marked with tape on the floor such that it was clear to the person. The human hider started at any grid location. After this, the robot scanned the environment to detect any person in the neighborhood and to detect its own position using the localization algorithm. These two measurements were used as observations in the MOMDP model. Since the model only allowed discrete movements of one cell distance, the observations were checked before

[1] `http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/`

feeding them into the model. The people detector sometimes detected persons outside the limits of the field or on the obstacles, therefore these detections were filtered out. The observations were entered into the model which gave the action. Next the person is told to do its next movement and at the same time the robot was commanded to go to its next position. Then another scan was done, and this was continued until one player won or the time passed.

6.4 Results

Simulations. More than 5000 simulated games were done with the different models against the two automated hiders (*random* and *smart*). The results of the games are shown in Table 1 and Table 2. It shows that the off-line model with the *triangle* reward works best ($p < 0.001$; Fishers exact test, two-sided, this has been used to check all the win statistics). Only on the small map (7×9) the on-line method was found to work better than the off-line method ($p < 0.05$), and both the on-line method and off-line method were found to work better than the heuristic method ($p < 0.05$).

Comparing the automated hiders we see that more games were won against the *random* hider (97%) than against the *smart* hider (95.3%; $p < 0.001$). No significant difference was found in winning for the two map configurations (Figure 2), nor for their sizes.

In Table 2 win statistics are shown per map size and seeker type. It also shows the average number of actions and average duration per step for the won games. Note that passing 32 actions resulted in a *tie*. The off-line MOMDP model used

Table 1. The win percentages for the four seeker methods against the two automated hiders. The last column shows the total number of simulated games done.

Map	Hider	Seeker	Win	Lose	Tie	Total
1	random	off-line – simple	99.8%	0.2%	0.0%	483
		off-line – triangle	100.0%	0.0%	0.0%	481
		on-line	99.6%	0.0%	0.4%	245
		smart	92.8%	0.0%	7.2%	360
	smart	off-line – simple	90.9%	9.1%	0.0%	243
		off-line – triangle	100.0%	0.0%	0.0%	243
		on-line	93.5%	6.3%	0.3%	400
		smart	97.3%	0.0%	2.7%	366
2	random	off-line – simple	99.7%	0.3%	0.0%	380
		off-line – triangle	99.7%	0.0%	0.3%	380
		on-line	99.0%	0.5%	0.5%	194
		smart	89.2%	0.0%	10.8%	360
	smart	off-line – simple	91.1%	8.9%	0.0%	192
		off-line – triangle	99.5%	0.0%	0.5%	187
		on-line	95.5%	4.3%	0.5%	400
		smart	95.0%	0.0%	5.0%	361
Total			96.9%	0.9%	2.2%	4475

Table 2. The win percentages per map size and seeker type. The one before last column shows the average ± standard deviation number of actions for won games, and the last column shows the average ± standard deviation duration of one action for won games.

Map Size	Seeker	Win	Lose	Tie	Total	Win Actions	Win Dur.(s)/Act.
7 × 9	off-line – simple	96.6%	3.4%	0.0%	760	5.25 ± 2.49	0.19 ± 0.09
	off-line – triangle	99.7%	0.0%	0.3%	756	6.83 ± 4.3	0.17 ± 0.1
	on-line	97.6%	2.0%	0.5%	656	9.07 ± 6.09	2.39 ± 0.24
	smart	92.9%	0.0%	7.1%	1012	10.67 ± 7.31	0.13 ± 0.09
9 × 12	off-line – simple	97.2%	2.8%	0.0%	538	7.26 ± 3.61	0.17 ± 0.09
	off-line – triangle	100.0%	0.0%	0.0%	535	9.22 ± 5.57	0.15 ± 0.09
	on-line	94.5%	5.1%	0.3%	583	11.71 ± 7.42	6.70 ± 0.37
	smart	95.2%	0.0%	4.8%	435	12.77 ± 8.94	0.13 ± 0.09

least steps when winning ($p < 0.001$; Wilcoxon ranksum). When the off-line method used the *triangle* reward it required more steps to win than using the *simple* reward. It was found that the seeker needed more steps on map 1 than on map 2 ($p < 0.001$; Wilcoxon ranksum), which might be because map 1 is symmetric and map 2 is not. For the on-line method the average time per step is highest (see last column of Table 2), because the MOMDP model is calculated and a policy is learned at each time step; and lowest for the heuristic method ($p < 0.001$; Wilcoxon ranksum test, 2-sided).

The duration of the calculation of the off-line policies are shown in Table 3. Although the off-line method with triangle reward works better than the on-line method, we can also see from Table 3 that the calculation of an off-line policy with triangle reward takes relatively much more time. The on-line method requires us to calculate a policy at every time step, which for the 9 × 12 on average were 6.7 s, and on average it took 13 steps to win (see Table 2), which results in approximately 87 s to complete a game. This is quite less than the calculation of the off-line policy for the triangle reward.

Real-World Experiments. With the real robot a total of 44 games were played against 15 adults from which 12 games were won by the hider and 32 by the seeker, see Table 4 for the detailed results. From the 32 games won by the seeker, 9 games ended in a situation where the hider reached the base, but at the same time was caught by the seeker. No game ended due to reaching the maximum number of time steps, 32. The average number of actions for won and lost games is shown in the last columns of Table 4,

There is no significant difference in the game results for map 1 and map 2, but the games are won in significantly less steps on map 1 ($p < 0.001$; Mann-Whitney test).

Like in the simulations (Table 2) using the off-line MOMDP with the triangle reward resulted in significantly more won games than the on-line hierarchical method. The low win percentages shown for the on-line method could be explained by a special strategies used by the human players.

Table 3. The time it took to calculate the policies off-line for the different maps and with the triangle and simple reward

Map Size	Map	Reward	Time (s)
7 × 9	1	simple	4.5
		triangle	23.2
	2	simple	5.2
		triangle	71.5
9 × 12	1	simple	29.9
		triangle	480.0
	2	simple	36.8
		triangle	260.0

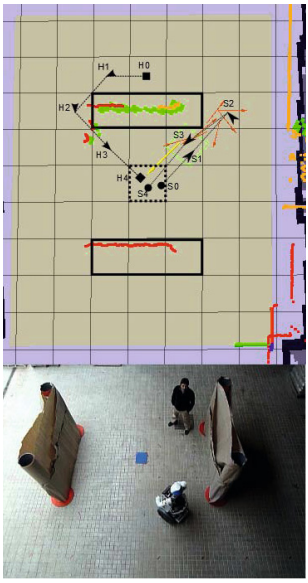
Table 4. The results of the real world experiments against de different seekers. The win column shows the percentage of games in which the seeker won even when the hider reached the base; **tie* shows the games in which the hider reached the base, but the seeker caught it. The last two columns show the average number of actions it took the seeker to win or lose the game respectively.

Map	Seeker	Win	(Tie*)	Lose	Total	Win Act.	Lose Act.
1	off-line – simple rew.	64.3%	(7.1%)	35.7%	14	4.4	7.2
	off-line – triangle rew.	100.0%	(25.0%)	0%	8	8.3	-
	on-line	40.0%	(0%)	60.0%	5	6.5	12.7
	smart	100.0%	(25.0%)	0%	4	14.5	-
2	off-line – simple rew.	100.0%	(50.0%)	0%	2	26	-
	off-line – triangle rew.	100.0%	(75.0%)	0%	4	10	-
	on-line	25.0%	(0%)	75.0%	4	19	17
	smart	66.7%	(33.3%)	33.3%	3	10	15

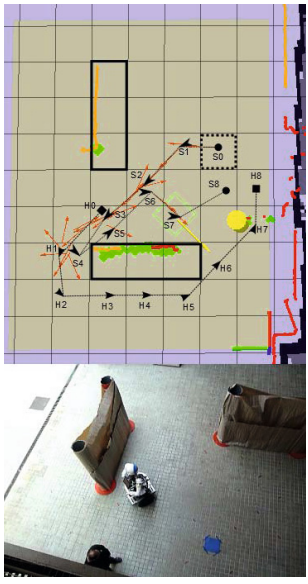
Three games are shown in Figure 3 which show the map and the laser collision detections which represent obstacles and walls. The light yellow area represents the game field. On top of the map the obstacles are drawn as black rectangles and the base as a dashed square. For both the seeker and hider, steps are shown with arrows and lines; S_0 is the seeker start place, which is always the base, and H_0 the hider's start place.

The first image (Figure 3(a)) shows a game played on map 1 in which the model used the simple reward. It can be seen that the robot followed the hider around the obstacle and with this the hider won. The second map (Figure 3(b)) shows map 1, but this time the robot used the triangle reward. It can be seen that the robot tried to find the hider at some hidden place, and when it could not see it (but it did have knowledge about the hider's position through the belief) it returned to protect the base. This game ended in a tie because both the hider and the seeker arrived at the base at the same time. The last map (Figure 3(c)) shows a game where the hider tries to get the seeker to follow him but when the distance to the base is too far, the seeker returns and finally catches the hider before reaching the base.

(a) Seeker at S_3 and hider at H_3.



(b) Players are at S_3 and H_3.



(c) Players are at S_7 and H_7.

Fig. 3. Fragments of three played games against a human hider. In (a) and (b) map 1 was used, in (c) map 2. The seeker used the simple reward in (a) and in (b) and (c) the triangle reward was used. The light yellow brown area shows the field on which the game was played. The dashed square is the base, the black rectangles are the obstacles, which were also detected by the robot's laser (red, orange and green). The yellow arrow shows the goal position and the red arrows the previous odometry.

7 Conclusions

In this paper we analyzed four methods to play the hide-and-seek game, from which three were based on an MOMDP model and one on a heuristic method. These have been extensively tested in simulation, and initial experiments have been performed with a mobile robot playing against a human hider in a simple real world urban environment. The simulated experiments showed that all methods performed really well, and the best method was the off-line MOMDP model with the triangle reward.

The on-line hierarchical method was previously proposed to reduce the number of partially observable states, and thereby tackling the *curse of dimensionality*. Even though the very good results of the on-line method in simulation, this was not reflected in experiments done in the real world. This can be explained by the strategy used by the hiders. In the simulations the *random* hider moved randomly and therefore was relatively easy to catch since it did not consider the game objectives. The *smart* hider on the other hand did take into account the rules of the game and therefore was more predictable. For the human players it was found that some of them did not take the optimal path, but used a strategy in which they "mislead" the robot by leading it around an obstacle and thereby won.

Although relatively few experiments were done, they gave us important insights in the functionality of the automated seeker methods used by a real mobile robot in the real world playing (*interacting* and *predicting*) against humans. In this first step towards the real world, limitations were set to have similar conditions as in the simulations, but our next steps are to overcome these limitations by incorporating sensing uncertainties and working in bigger real world environments.

Acknowledgments. This work has been partially funded by the Spanish Ministry of Economy and Competitiveness under projects TaskCoop DPI2010-17112.

References

1. Johansson, E., Balkenius, C.: It's a child's game: Investigating cognitive development with playing robots. In: International Conference on Development and Learning, vol. 164 (2005)
2. Ong, S.C.W., Png, S.W., Hsu, D., Lee, W.S.: Planning under Uncertainty for Robotic Tasks with Mixed Observability. The International Journal of Robotics Research 29(8), 1053–1068 (2010)
3. Araya-López, M., Thomas, V., Buffet, O., Charpillet, F.: A closer look at MOMDPs. In: 22nd International Conference on Tools with Artificial Intelligence - ICTAI (2010)
4. Braziunas, D.: Pomdp solution methods. Technical report, University of Toronto (2003)
5. Hauskrecht, M.: Value-function approximations for partially observable markov decision processes. Journal of Artificial Intelligence Research 13, 33–94 (2000)

6. Cassandra, A.R., Kaelbling, L.P., Kurien, J.A.: Acting under uncertainty: discrete bayesian models for mobile-robot navigation. In: Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems 1996, IROS 1996, vol. 2, pp. 963–972 (1996)
7. Spaan, M.T.J., Vlassis, N.: A point-based pomdp algorithm for robot planning. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, vol. 3, pp. 2399–2404 (2004)
8. Papadimitriou, C., Tsisiklis, J.N.: The complexity of markov decision processes. Mathematics of Operations Research 12(3), 441–450 (1987)
9. Kurniawati, H., Hsu, D., Lee, W.: Sarsop: efficient point-based pomdp planning by approximating optimally reachable belief spaces. In: Robotics: Science and Systems (2008)
10. Goldhoorn, A., Sanfeliu, A., Alquézar, R.: Comparison of momdp and heuristic methods to play hide-and-seek. Accepted for the Sixteenth International Conference of the Catalan Association of Artificial Intelligence (2013)
11. Trulls, E., Corominas Murtra, A., Pérez-Ibarz, J., Ferrer, G., Vasquez, D., Mirats-Tur, J., Sanfeliu, A.: Autonomous navigation for mobile service robots in urban pedestrian environments. Journal of Field Robotics (May 2010)
12. Sanfeliu, A., Andrade-Cetto, J., Barbosa, M., Bowden, R., Capitán, J., Corominas, A., Gilbert, A., Illingworth, J., Merino, L., Mirats, J.M., Moreno, P., Ollero, A., Sequeira, J.: Decentralized Sensor Fusion for Ubiquitous Networking Robotics in Urban Areas. Sensors 10(3), 2274–2314 (2010)
13. Georgaraki, C.: A POMDP approach to the hide and seek game. Master's thesis, Universitat Politècnica de Catalunya, Barcelona, Spain (2012)
14. Pineau, J., Gordon, G., Thrun, S.: Point-based value iteration: An anytime algorithm for pomdps. In: International Joint Conference on Artificial Intelligence, pp. 477–484 (2003)
15. Stockman, G., Shapiro, L.G.: Computer Vision, 1st edn. Prentice Hall, Upper Saddle River (2001)
16. Foka, A., Trahanias, P.: Real-time hierarchical POMDPs for autonomous robot navigation. Robotics and Autonomous Systems 55(7), 561–571 (2007)
17. Ross, S., Pineau, J., Paquet, S., Chaib-Draa, B.: Online Planning Algorithms for POMDPs. The Journal of Artificial Intelligence Research 32(2), 663–704 (2008)

Part XVI

Robot Networks and Telerobotics

Haptic Aids for Bilateral Teleoperators

Alexander Pérez[1,2,⋆] and Jan Rosell[1]

[1] Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain
jan.rosell@upc.edu
[2] Escuela Colombiana de Ingeniería Julio Garavito, Bogotá D.C., Colombia
alexander.perez@escuelaing.edu.co

Abstract. Teleoperation of robotic tasks is usually performed in the Cartesian space due to the kinematic differences between the master and the slave. This entails several requirements, like the definition of a proper mapping between workspaces, the need to avoid collisions of the teleoperated robot with the environment, and the use of the inverse kinematics and of a procedure to correctly manage the passing through singularities. Within a bilateral teleoperation framework to teleoperate an industrial robot with a desktop haptic device, the present work proposes a guiding system based on path planning techniques to cope with these issues. The proposed system also includes a reactive behavior to cope with the potential collisions with obstacles. Teleoperation tests on virtual and real scenarios are included to validate the approach.

Keywords: Teleoperation, motion planning, guidance system, haptic devices.

1 Introduction

Teleoperation has evolved in many aspects since its beginning in the late 40s [1], thus enlarging the kind of tasks that can be teleoperated. Recent advances, for instance, allow the design of bilateral teleoperation frameworks using communication channels with variable time delays, like the internet. These bilateral teleoperation frameworks require the use of force feedback devices, called haptic devices, as the master part in the teleoperation. Using these devices, teleoperation tasks are performed at Cartesian level, avoiding incompatibilities with the slave robot from the kinematic point of view. However, this entails some problems that need to be correctly handled, and that have often been overlooked.

The first problem is related to the fact that the workspace of the robot and that of the haptic device have quite different sizes, being the haptic device workspace much smaller than that of the robot. A proper mapping between workspaces is required, and several approaches have been proposed so far, like simple scale factors, drift control [2] or resynchronization [3].

⋆ This work was partially supported by the Spanish Government through the projects DPI2010-15446 and DPI2011-22471 and by the Generalitat de Catalunya through an AGAUR scholarship.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_38,

The second problem is due to the need to move the robot avoiding collisions with obstacles, which is difficult and non-intuitive because the robot is commanded in Cartesian space. When the task the user wants to teleoperate is known, path planning techniques can be used to guide him/her along a collision-free path, both visually and haptically.

The third problem arises when the robot is steered along paths that include some configuration changes, e.g. from elbow-up to elbow-down as shown in Fig. 1. Some teleoperation papers overlook this problem, others try to eliminate or minimize it by taking advantage of redundancy to assure large convex workspaces for singularity-free operation [4]. This problem can, however, be easily solved if a path is available and can be haptically suggested to the user. In this case, near configurations changes, the haptic suggestion can be strengthened, forcing the robot to cross the singularity in the desired way, i.e. as specified by the path.

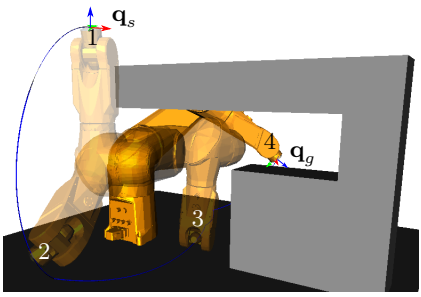


Fig. 1. Four snapshots of a task where where the arm configuration changes from elbow-down ($\mathbf{q}_s$) to elbow-up ($\mathbf{q}_g$)

Within a bilateral teleoperation framework to teleoperate an industrial robot with a desktop haptic device, the present work is focused on the proposal of a guiding system based on path planning techniques to cope with the above stated problems.

1.1 Related Works

The performance of teleoperation tasks can be significatively improved if some aids are provided to the operator. Several proposals in this line have already been published in the literature since the early nineties. A comprehensive overview of visual aids and cues to assist teleoperated tasks can be found in [5], which includes automatic end-effector tracking, computer assisted camera/lighting placement or the use of screen coordinates to solve the stimulus-response mismatch problem. The overview also presents display enhancements (later known as augmented reality) used: a) to highlight non-reachable or dangerous areas, or obstacles where an imminent collision can occur; b) to give visual cues to improve the orientation and positioning of the end effector; c) to present depth cues like virtual views and perspective grids. More recently, stereoscopic visual feedback

systems that combine images gathered from two remotely actuated video cameras have been proposed to improve depth perception [6]. In order to improve the performance in complex scenarios with obstacles (although constrained to three *dof* robots), some approaches also proposed to use not only the view of the workspace but also that of the configuration space, where the user manipulates a single point, overcoming some usual human spatial reasoning deficiencies [7].

Besides the visual aids proposed, other aids based on force feedback greatly help to enhance the teleoperation performance, provided that the local site is controlled with a haptic device. Virtual fixtures, for instance, are used to restrain the user motions within a given region or within a subspace of lower dimension, thus allowing him/her to concentrate on the commanding of the motions relevant to the task, resulting in faster and more precise task executions and with lesser operator workloads. Some predefined virtual fixtures can be defined for given actions like approaching motions, or insert or extract operations, as introduced in [8]. Also, virtual fixtures can be specified by geometric constraint solvers, that find a subspace of lower dimension as a parameterized manifold that satisfies a constraints set, that is easily defined by the user from the knowledge of the task to be executed [9].

Force feedback is also used: a) to avoid obstacles, i.e. a force field proportional to the proximity between the obstacles and the end-effector is generated to repel the approaching motions to the obstacles, e.g. by using a map of the environment [10] or an easy recognition of obstacles by means of predefined markers [11]; b) to avoid singularities, i.e. a force field proportional to the proximity of a singularity and to the magnitude and direction of the velocity command is generated to drive the user into the feasible direction defined by the manipulator Jacobian [12]; c) to avoid large contact forces in tasks where there is physical interaction, like assembly tasks, i.e. contact forces computed from a model of the task can be fed back to make the user react on time and avoid huge actual forces on the remote site [13].

Forces have also been used to guide the user motions towards a specific goal, in order to obtain an easier and faster task commanding. The approach in [14] proposes a simple way to guide the operator position during a teleoperated grasping task, by using an attractive potential field obtained from visual information, and the approach in [15] generates an attractive force from the position and orientation errors between the current configuration of the robot griper and the desired one. Other path planning techniques have been used to consider also the avoidance of obstacles, in order to generate forces to constrain the operator along a collision-free path towards a goal configuration, either for commanding 2 *dof* motions of a mobile robot [16], or the motions of an object being manipulated in 2 or 3 *dof* of translation [17] (i.e. without considering the kinematics and the geometry of the robot that manipulates the object), or in virtual 6 *dof* assembly/disassembly tasks [18].

A combination of several of the above mentioned aids are considered in the bilateral teleoperation framework proposed in [19]. However, no approaches include, at the authors' knowledge, with guiding aids to teleoperate robot manipulators.

1.2 Scope, Problem Statement and Solution Overview

Excluding those teleoperated tasks with an exploratory aim carried out in unknown environments, many others are performed in known environments, although this knowledge may be incomplete and/or subject to uncertainty, reason that justifies the need for teleoperation. This knowledge of the task and of the environment (that can be updated as new information is acquired) can be used to support the operator with different aids. It can, for instance, be used by path planning techniques to suggest collision-free paths, as already found in early works of supervisory teleoperation control [20], where the system suggested paths, graphically shown to the user, to avoid collisions with obstacles and also to avoid traversing visually occluded zones by defining them as virtual obstacles. Others approaches, as reviewed in the previous subsection, used path planning techniques to generate forces to attract the motion towards the goal while avoiding collisions of the manipulated object with the environment. We are interested in extending these methods to take also into account potential collisions of the whole robot with the environment, in order to alleviate the operator burden and let him/her concentrate on the task. The proposal is to be framed within a bilateral teleoperation framework to teleoperate an industrial robot with a desktop haptic device.

The use of a desktop haptic device to feed back force to guide the user poses some questions. The human perception of force and torque information is highly integrated, and humans cannot neatly perceive them in a separate way. To evaluate this, a recent work made a thorough study to determine whether force and torque cues interact in haptic discrimination of force, torque and stiffness [21]. These authors found a significant interference across force and torque dimensions, and their experiments proved that they cannot be processed separately. Moreover, the usefulness of torque feedback in haptic manipulation tasks is highly dependant on the task, and for simple tasks it has been demonstrated that force feedback alone approximates the performance obtained with force and torque feedback, thus justifying the popularity of desktop haptic devices that have 6 positioning degrees of freedom but only 3 in force feedback [22]. To evaluate the use of forces to haptically guide the user in teleoperation tasks, we have studied the response of several users when being randomly stimulated by either forces, torques or combinations of both, using a Phantom haptic device with force and torque feedback. Except for the torque around the axis of the stylus, users tended to mistake torques by forces. Taking all this into account, in the present proposal we have decided that no force and torque are to be simultaneously feed back to the user as a guide, because it is very difficult for the users to understand the suggestions felt, and often respond to torques with translations instead of rotations. They can, however, be separately fed back because if users have the chance to select which kind of feedback to receive, either forces or torques, they are prepared to respond accordingly.

Then, the problem to be solved is the implementation of aids to haptically guide an operator in the commanding of a teleoperated task, within a bilateral teleoperation framework involving an industrial robot and a desktop haptic de-

vice. The task and the environment are assumed known (although maybe incomplete and subject to uncertainty), and the forces and torques will be feed back separately, as required by the user. Four aids are proposed, which give the teleoperator framework the following features: a) *assisted re-synchronization:* the mapping is based on the re-synchronization method and during the clutching the system haptically guides the user towards the best configuration to relocate the haptic device; b) *collision-free guidance:* a sampling-based planner is used in Configuration Space to plan a collision-free path that, through the direct kinematics and the workspace mapping, is used to generate forces to suggest the user the motions required to teleoperate the task; c) *robust guidance:* the control module guarantees that the solution path is exactly followed at the configuration changes, guaranteeing a safe singularity crossing; d) *reactive behavior:* the damping values of the control algorithm at the remote site are increased when potential collisions with either fixed or mobile obstacles may occur, slowing down the teleoperation and preventing collisions.

The paper is organized as follows. After this introduction, Section 2 sketches the assisted bilateral teleoperation framework proposed and Section 3 describes in detail the haptic aids. The theoretical contribution is complemented with performance evaluation in Section 4. Finally, Section 5 summarizes and concludes the proposal.

2 Teleoperation Framework

The bilateral teleoperation framework used is composed of the local site, the remote site and the communication layer, as shown in Fig. 2. The basic blocs, drawn in blue, comprise at the local site the haptic device management (HDM) and the local control system (LCS), and at the remote site the robot (Rob) and the remote control system (RCS). The bilateral control scheme and the workspace mapping between the haptic device and the robot are briefly introduced in Sections 2.1 and 2.2, respectively. The aids proposed as assistance correspond to the blocs drawn in red and comprise the resynchronization and path following guiding aids at the local site and the reactive behaviour at the remote site. The assisted teleoperation procedure is sketched in Section 2.3, and the aids used are fully detailed in Section 3.

2.1 Control

The blue signals between blocks in Fig. 2 define the basic control loop, based on the impedance strategy, that uses the position input to provide force feedback. The controller is a P+d (proportional plus damping) that was proved to be stable and robust in front of time delays [24]. Moreover, this controller was proven to provide stiff force reflections of the remote environment, and for small time-delays a better transparency than other alternatives. For these advantages, this control scheme has been chosen in the present work.

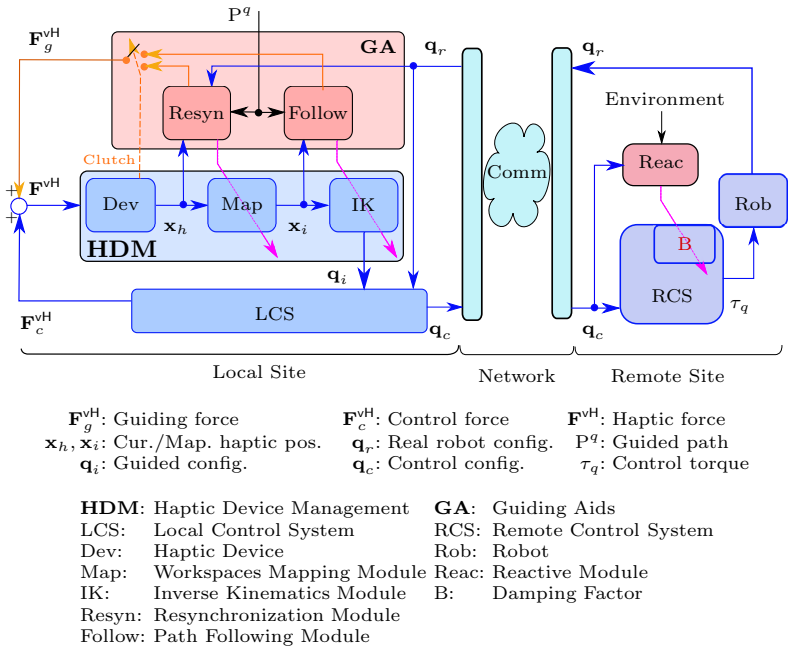


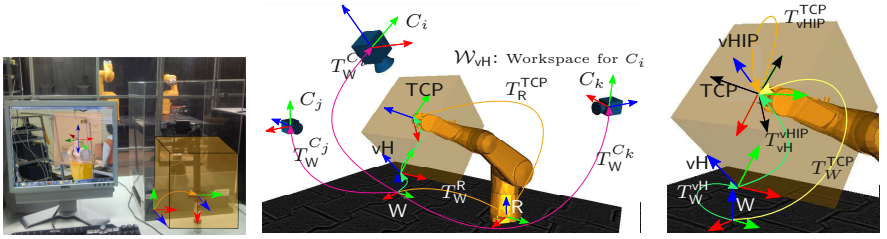
Fig. 2. General scheme of the teleoperation framework and its information flow

2.2 Workspace Mapping

The teleoperation framework requires the mapping between the workspace of the haptic device and that of the robot, i.e. the mapping between the movements of the Haptic Interface Point (HIP) and those of the robot Tool Center Point (TCP). The mapping used is based on the procedure proposed by the authors in [23] and summarize below, that uses the camera that feeds back the video from the scene as a common point between the local and the remote sites, as shown in Fig. 3a. This procedure maps the workspace of the haptic device ($\mathcal{W}_H$, with reference frame H) into that of the robot ($\mathcal{W}_W$, with reference frame W) to obtain the virtual haptic workspace ($\mathcal{W}_{\mathsf{vH}}$, with reference frame vH), following two guidelines: a) the orientation of vH is the same as that of the camera used to feed back the video (Fig. 3b), and b) the origin of the HIP within vH, called vHIP, is made coincident with that of the TCP (Fig. 3c).

The location of the HIP with respect to H, i.e. the transform $T_{\mathsf{H}}^{\mathsf{HIP}}$, is continually read from the haptic device, and the transform $T_{\mathsf{vH}}^{\mathsf{vHIP}}$ is made coincident with it or modified by a simple scale factor. Then, during the teleoperation activity the location of the TCP with respect to W is obtained as (Fig. 3c):

$$T_W^{\mathsf{TCP}} = T_{\mathsf{W}}^{\mathsf{vH}} \cdot T_{\mathsf{vH}}^{\mathsf{vHIP}} \cdot T_{\mathsf{vHIP}}^{\mathsf{TCP}}, \tag{1}$$



(a) Haptic workspace. (b) Virtual haptic workspace. (c) Frames and transformations.

Fig. 3. Bilateral teloperation framework equipped with an industrial Stäubli TX90 robot, a Phantom haptic device and three cameras: a) Physical devices and video feedback; b) Model of the remote site showing the virtual haptic workspace aligned with camera i; c) Virtual coupling attained by making the origins of frames vHIP and TCP coincident. Figures reproduced from [23].

where the transforms $T_{\mathsf{W}}^{\mathsf{vH}}$ and $T_{\mathsf{vHIP}}^{\mathsf{TCP}}$ are fixed and set at the synchronization instants as:

$$
\begin{aligned}
T_{\mathsf{vHIP}}^{\mathsf{TCP}} &= \left(\begin{array}{c|c} \mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} & \mathbf{0} \\ \hline 0 & 1 \end{array}\right) \\
\mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} &= (\mathcal{R}_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \cdot (\mathcal{R}_{\mathsf{W}}^{\mathsf{Ci}})^{-1} \cdot \mathcal{R}_{\mathsf{W}}^{\mathsf{TCP}} \\
T_{\mathsf{W}}^{\mathsf{vH}} &= T_W^{\mathsf{TCP}} \cdot (T_{\mathsf{vHIP}}^{\mathsf{TCP}})^{-1} \cdot (T_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1}
\end{aligned} \tag{2}
$$

Re-synchronization is done by pressing the clutch button on the haptic device stylus each time the user reaches the limits of the haptic workspace or when he/she wants to change the camera used.

2.3 Assisted Teleoperation Procedure

Teleoperation assistance is introduced in the teleoperation framework by the modules shown as red blocs in Fig. 2. The Path Following Module allows the generation of forces for the haptic device ($\mathbf{F}_g^{\mathsf{vH}}$) that suggest the following of a collision-free path, and its connection with the Inverse Kinematic Module permits a robust and smooth configuration change, when necessary. When the robot is detached from the haptic device, the Resynchronization Module is used to generate forces to assist the re-synchronization. The Reactive Module in the remote site is used to increase the damping gain if potential collisions are detected.

The proposed teleoperation assistance is based on the assumption that the task to be teleoperated is known (i.e. the initial and goal configurations, q_s and the q_g respectively, as well as the model of the robot and of the workspace). Then, the teleoperation assistance is designed based on the availability of a collision-free path computed by a sampling-based planner (Probabilistic RoadMap – PRM) on the configuration space of the robot ($\mathcal{C}$), connecting q_s and q_g:

– Let $\mathbf{P}^q$ be such a path: $\mathbf{P}^q = \{\mathbf{q}_s, \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_g\} \in \mathcal{C}$, where q_i are the configurations that have been collision-checked to validate the edges of the

Algorithm 1. Assisted teleoperation procedure

Require:
$\mathbf{q}_s, \mathbf{q}_g$: Start and goal configurations
δ: Step size of the sampling-based planner

$\mathbf{P}^q$ = Path_Planner($\mathbf{q}_s, \mathbf{q}_g, \delta$)
$\mathbf{P}^x$ = FK($\mathbf{P}^q$)
while teleoperation **do**
  $\mathbf{x}_i$ = MapHaptic()
  if clutch **then**
    $\mathbf{F}^{\mathsf{vH}}$ = Re-synchronization($\mathbf{x}_i$)
  else
    $\mathbf{q}_r$ = SubscribeRobotConfig()
    $\mathbf{F}_g^{\mathsf{W}}$ = ComputeForce($\mathbf{x}_i$)
    $\mathbf{F}_g^{\mathsf{vH}}$ = ChangeFrame($\mathbf{F}_g^{\mathsf{W}}$)
    $\mathbf{q}_i$ = IK_filtered($\mathbf{x}_i$)
    $[\mathbf{F}_c^{\mathsf{vH}}, \mathbf{q}_c]$ = Control($\mathbf{q}_i, \mathbf{q}_r$)
    PublishRobotCommand($\mathbf{q}_c$)
    $\mathbf{F}^{\mathsf{vH}} = \mathbf{F}_g^{\mathsf{vH}} + \mathbf{F}_c^{\mathsf{vH}}$
  end if
  Apply $\mathbf{F}^{\mathsf{vH}}$ to the haptic device
end while

path, i.e. $\mathbf{q}_j$ and $\mathbf{q}_{j+1}$ are close configurations separated a suficient small distance ϵ (i.e. no collision is assumed possible between two free configurations ϵ apart). The path $\mathbf{P}^q$ is piecewise linear, and when two consecutive configurations in $\mathbf{P}^q$ correspond to different kinematic configurations, then a linear interpolation is done and the resulting intermediate (singular) configuration, is added to $\mathbf{P}^q$.

– Let $\mathbf{P}^x = \{\mathbf{x}_s, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_g\} \in SE3$ be the path followed by the robot TCP corresponding to $\mathbf{P}^q$, computed by the forward kinematics of the robot, i.e. $\mathbf{x}_j$ is the position and orientation of the TCP when the robot is at $\mathbf{q}_j$. Since the configurations of the path are very close to each other, the path $\mathbf{P}^x$ can also be assumed to be piecewise linear, and each segment of the path $\overline{\mathbf{x}_j\mathbf{x}_{j+1}}$ is labeled with the kinematic configuration of $\mathbf{q}_j$.

Algorithm 1 sketches the cycle of the proposed assistance-based teleoperation procedure. It first computes the paths $\mathbf{P}^q$ and $\mathbf{P}^x$ and then the teleoperation loop starts. The algorithm uses the following functions:

– Path_Planner($\mathbf{q}_s, \mathbf{q}_g, \delta$): Returns a collision-free path $\mathbf{P}^q$ from $\mathbf{q}_s$ to $\mathbf{q}_g$ using a step size δ.
– FK($\mathbf{P}^q$): Uses the forward kinematics of the robot to obtain the path $\mathbf{P}^x$ from $\mathbf{P}^q$.

- MapHaptic: Uses Eq. (1) to compute the robot TCP, as detailed in Section 2.2.
- Re-sychronization($\mathbf{x}_i$): Uses Eq. (2) to set a new mapping between workspaces and returns the guiding forces generated to suggest the user the best place to re-synchronize, as introduced in Section 3.1.
- SubscribeRobotConfig: Receives the robot configuration ($\mathbf{q}_r$) from the remote site.
- ComputeForce($\mathbf{x}_i$): Computes the guiding force to be applied at $\mathbf{x}_i$ to follow a collision-free path, as introduced in Section 3.2.
- ChangeFrame($\mathbf{F}_g^{\mathsf{W}}$): Transforms $\mathbf{F}_g^{\mathsf{W}}$ from the robot frame to the vH frame.
- IK_filtered($\mathbf{x}_i$): Computes the robot command $\mathbf{q}_i$ from $\mathbf{x}_i$ using the inverse kinematics and a robust crossing of singular configurations, as introduced in Section 3.3.
- Control($\mathbf{x}_i, \mathbf{q}_r$): Runs the bilateral control algorithm.
- PublishRobotCommand($\mathbf{q}_c$): Sends the command to the robot at the remote site.

This teleoperation cycle can start again at user's will, i.e. a new path from the configuration where the robot may be can be queried at any time and used to assist the teleoperation form there.

3 Teleoperation Aids

This section describes the four aids proposed, that have been incorporated to the basic bilateral teleoperation framework. The focus is set at the generation of forces; the description of the planner is out of the scope of the present paper.

3.1 Re-Synchronization Aid

Since the haptic device workspace is usually much smaller than that of the robot, the mapping between workspaces needs to be continually updated. At user requirement (e.g. by pressing the button located at the stylus of the haptic device), the robot is detached from the haptic device. Let HIP_{curr} be the position of the HIP at this instant of time. Then, the **Re-synchronization** function used in Algorithm 1:

1. Computes the new position of the HIP (called HIP_{new}) from where the teleoperation can resume and proceed for as long as possible. This is done by computing the largest bounding box that satisfies that it contains the current position of the vHIP and as much points of the path $\mathbf{P}^x$ as possible, and that fits within the virtual haptic workspace. Then $\mathcal{W}_{\mathsf{vH}}$ is placed such that the bounding box is centered in it, and the corresponding new HIP position is computed (HIP_{new}).

2. Haptically guides the user from HIP_{curr} to HIP_{new}, i.e. generates forces along the direction defined by the position vector of T:

$$T = \left[T_{\mathsf{H}}^{\mathsf{HIP}_{\mathsf{curr}}}\right]^{-1} T_{\mathsf{H}}^{\mathsf{HIP}_{\mathsf{new}}} \tag{3}$$

The re-synchronization aid is implemented in the Re-synchronization Module shown in Fig. 2. Finer details of the procedure can be found in [23].

3.2 Guiding of Collision-Free Paths

The guiding aid has been conceived to fulfill the following requirements:

- No force and torque are to be simultaneously feed back to the user. The user must have the chance to select which kind of feedback to receive, either forces or torques, and hence be prepared to respond accordingly.
- The forces/torques fed back must guide the user toward the path, disappearing within a dead-zone around it (i.e. by giving up to the suggestions felt, the haptic device has to be correctly positioned/oriented on the path). Optionally, when the user is in the dead zone or near it, he/she must have the chance to select weather to receive or not a pushing force to indicate the sense of the path.

The following nomenclature needs to be defined. Let:

- $\mathbf{x} = \left(\begin{array}{c|c} \mathcal{R} & \mathbf{p} \\ \hline \mathbf{0} & 1 \end{array}\right)$ describe the configuration (position and orientation) of the TCP, $\mathbf{x}_i$ be the current one and $\mathbf{x}_d$ the configuration on the path closest to $\mathbf{x}_i$.
- $\mathbf{d}_t$ be the vector $\mathbf{d}_t = \mathbf{p}_d - \mathbf{p}_i$ and $(\mathbf{d}_r, \theta_r)$ the axis-angle representation of $\mathcal{R}_i^{-1}\mathcal{R}_d$.
- d_t be the translational distance between $\mathbf{x}_i$ and $\mathbf{x}_d$, i.e. $d_t = |\mathbf{d}_t|$, and d_r the rotational one defined as $d_r = \theta_r$.
- $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ be the nodes of $\mathbf{P}^x$ such that $\mathbf{x}_d$ lies between them, and $\mathbf{s}$ be the vector $\mathbf{s} = \mathbf{p}_{k+1} - \mathbf{p}_k$.
- ϵ_{t_n} and ϵ_{r_n} be, respectively, translational and rotational distance thresholds, with $n = \{1, 2, 3\}$.

Then, the **ComputeForce**$(\mathbf{x}_i)$ function used in Algorithm 1 computes a generalized guiding force $\mathbf{F}_g^{\mathsf{W}} = (\mathbf{f}_g^{\mathsf{W}}, \tau_g^{\mathsf{W}})^T$ from two components $\mathbf{F}_m$ and $\mathbf{F}_p$ such that:

- $\mathbf{F}_m = (\mathbf{f}_m, \tau_m)^T$ attracts the TCP to the path. Force and torque are separately fed back provided that the distances d_t and d_r lie above given thresholds.
- $\mathbf{F}_p = (\mathbf{f}_p, 0)^T$ pushes along $\mathbf{s}$ towards the next node in the path. This generalized force is generated if d_t lies below a given threshold and the user requests it.

The expressions of $\mathbf{f}_g^{\mathsf{W}}$ and τ_g^{W} are:

$$\mathbf{f}_g^{\mathsf{W}} = \begin{cases} \mathbf{f}_p \cdot \mathrm{A} & d_t < \epsilon_{t_1} \\ \mathbf{f}_m + \mathbf{f}_p \cdot \mathrm{A} & \epsilon_{t_1} < d_t < \epsilon_{t_2} \\ \mathbf{f}_m & d_t > \epsilon_{t_2} \end{cases}$$

$$\tau_g^{\mathsf{W}} = \begin{cases} 0 & d_r < \epsilon_{r_1} \\ \tau_m & d_r > \epsilon_{r_1} \end{cases}$$

$$\mathbf{f}_m = \frac{\mathbf{d}_t}{|\mathbf{d}_t|} \min\left(F_{max}, F_{max} \frac{d_t - \epsilon_{t_1}}{\epsilon_{t_3} - \epsilon_{t_1}}\right)$$

$$\tau_m = \frac{\mathbf{d}_r}{|\mathbf{d}_r|} \min\left(\tau_{max}, \tau_{max} \frac{d_r - \epsilon_{r_1}}{\epsilon_{r_3} - \epsilon_{r_1}}\right)$$

$\mathbf{f}_p =$ Constant force along $\mathbf{s}$

$\mathrm{A} = 1/0$ (Enable/Disable active force)

$F_{max} =$ Max. force exerted by the device

$\tau_{max} =$ Max. torque exerted by the device

In order to apply the guided force to the teleoperator using the haptic device, the generalized force $\mathbf{F}_g^{\mathsf{W}}$ is mapped to the frame vH, by applying the virtual work principle. This is done by the **ChangeFrame**($\mathbf{F}_g^{\mathsf{W}}$) function in Algorithm 1:

$$\mathbf{F}_g^{\mathsf{vH}} = \begin{bmatrix} \mathbf{f}_g^{\mathsf{vH}} \\ \tau_g^{\mathsf{vH}} \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{\mathsf{vH}}^{\mathsf{W}} & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_{\mathsf{vH}}^{\mathsf{W}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{f}_g^{\mathsf{W}} \\ \tau_g^{\mathsf{W}} \end{bmatrix} \tag{4}$$

The guiding aid is implemented in the Path Following Module shown in Fig. 2.

3.3 Aid to Change the Kinematic Configuration

The **IK_filtered** function in Algorithm 1 computes the set point $\mathbf{q}_i$ for the control algorithm, from the input $\mathbf{x}_i$ set by the user with the haptic device, using the inverse kinematics solution closer to the guided path. The function includes a robust and smooth crossing of points where the kinematic configuration changes. This is done by modifying the configuration commanded with a saturation that produces a funnel effect that enforces the crossing of the singular configuration in the desired way, i.e. following the solution path.

Let Fig. 4 illustrate the nomenclature required to describe the procedure:

$\mathbf{q}_0$: is a configuration where the kinematic configuration changes.

$\mathbf{q}_p$,$\mathbf{q}_n$: are, with respect to $\mathbf{q}_0$, the previous and the next configurations in the path, respectively.

$\mathbf{u}_{p/n}$: are the unitary vectors with origin at $\mathbf{q}_0$ and pointing towards $\mathbf{q}_p$ and $\mathbf{q}_n$, respectively.

$\mathbf{u}_j$: is the unitary vector of the jth coordinate of the configuration space (that which corresponds to the jth joint).

$\mathbf{q}_i$: is the robot configuration corresponding to the user position $\mathbf{x}_i$. It is computed from $\mathbf{x}_i$ using the inverse kinematics and choosing the kinematic configuration of the segment of the path to which $\mathbf{x}_d$ (the point on the path closest to $\mathbf{x}_i$) belongs to.

$\mathbf{v}_i$: is the vector from $\mathbf{q}_0$ to $\mathbf{q}_i$.

H_ρ: is the hyper-ball of radius ρ centered at $\mathbf{q}_0$, it bounds the region where the funnel effect is felt.

$\mathbf{u}_{p_j/n_j}$**:** are the unitary vectors defined as:

$$\begin{aligned} \mathbf{u}_{p_j} &= \text{sign}(\mathbf{u}_p \cdot \mathbf{u}_j)\mathbf{u}_j \quad \forall j \in 1, \ldots, d \\ \mathbf{u}_{n_j} &= \text{sign}(\mathbf{u}_n \cdot \mathbf{u}_j)\mathbf{u}_j \quad \forall j \in 1, \ldots, d \end{aligned} \tag{5}$$

$R_{P/N}$**:** are the regions inside H_ρ that are defined by the positive linear combination of the vectors $\mathbf{u}_{p_j} \forall j \in 1, \ldots, d$ and $\mathbf{u}_{n_j} \forall j \in 1, \ldots, d$, respectively.

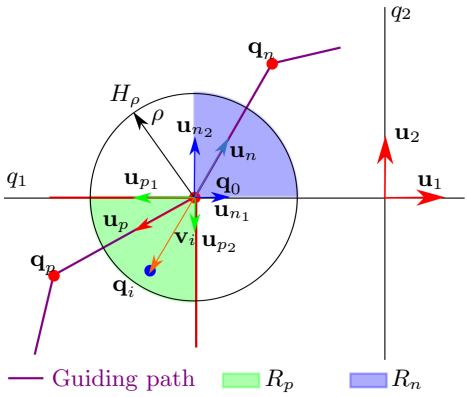


Fig. 4. Zones defined to correctly change the kinematic configuration in the 2 d.o.f. example of an RR robot shown

The aid is automatically activated when the configuration $\mathbf{q}_i$ lies within region H_ρ, modifying its value so as to make it lie within $R_P \cup R_N$, i.e. the coordinates of $\mathbf{q}_i$ are modified according to:

$$q_{i_j} = \begin{cases} q_{0_j} + v_{i_j} & \text{if } \mathbf{v}_i \cdot \mathbf{u}_{p_j} > 0 \text{ or } \mathbf{v}_i \cdot \mathbf{u}_{n_j} > 0 \\ q_{0_j} & \text{otherwise} \end{cases} \tag{6}$$

Fig. 5 shows an example where the guiding path crosses a point where the kinematic configuration changes, due to a change in the sign of q_2. The use of the proposed aid is as follows. Let the guiding path be $\mathbf{P}^q = \{\mathbf{q}_A, \mathbf{q}_B, \mathbf{q}_C, \mathbf{q}_D, \mathbf{q}_E\} \in \mathcal{C}$ and $\mathbf{P}^x$ be the corresponding path in $SE3$. Let also $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\} \in SE3$ be the path commanded by the user with the haptic device and $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5, \mathbf{q}_6\}$ the path in $\mathcal{C}$ computed using the same inverse kinematic solution as the corresponding closest point in $\mathbf{P}^x$. Configurations $\mathbf{q}_a$, $\mathbf{q}_b$, $\mathbf{q}_d$ and $\mathbf{q}_e$ already belong to either R_P or R_N and are therefore kept unchanged. However, $\mathbf{q}_3$ and $\mathbf{q}_4$ are first mapped, using Eq. (6), to $\mathbf{q}'_3$ and $\mathbf{q}'_4$ respectively, i.e. the path commanded to the robot is $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}'_3, \mathbf{q}'_4, \mathbf{q}_5, \mathbf{q}_6\}$. Therefore, the robot crosses the critical point as specified by the guiding path.

This aid is implemented in the Inverse Kinematic Module shown in Fig. 2.

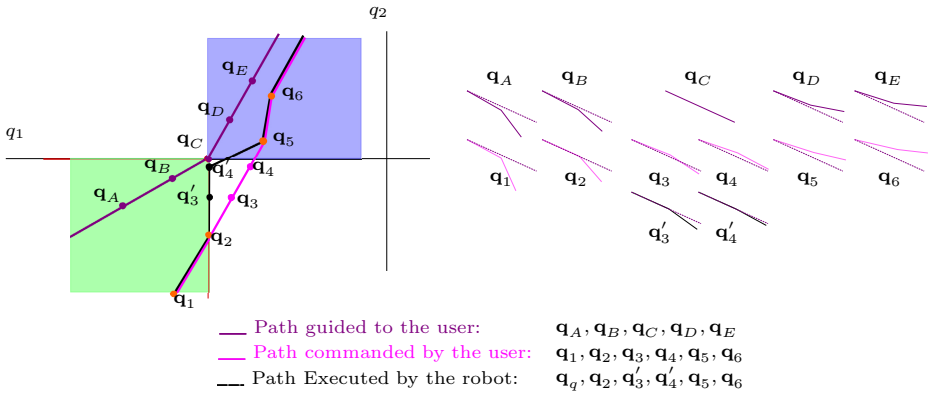


Fig. 5. Guiding, commanded and executed paths both in configuration space (left) and physical space (right)

3.4 Reactive System

The proposed assistance is completed with the Reactive Module in the remote site, as shown in Fig. 2. This module is designed to regulate the remote damping of the control system, as a function of the potential collisions with the environment.

As demonstrated in [24], the use of the P+d controller guarantees that the velocities and the position error are bounded, provided that the control gains (the proportional and the damping gains at the local and the remote site, $K_l, B_l, K_r, B_r > 0$), are set according to:

$$4B_l B_r > ({}^*T_l + {}^*T_r)^2 K_l K_r, \tag{7}$$

assuming passive the human operator and the environment, and variable time-delays with known upper bounds ($T_i(t) \leq {}^*T_i < \infty$) and with time derivatives that do not vary faster than time itself ($|\dot{T}_i(t)| < 1$).

According to Eq. (7), assuming the local damping B_l and the gains K_l and K_r have a fixed value, the control scheme will be stable if the damping B_r is above a lower bound, B_r^{min}:

$$B_r^{min} = \frac{({}^*T_l + {}^*T_r)^2 K_l K_r}{4B_l} \tag{8}$$

An increase of B_r above B_r^{min} will slow down the teleoperation, i.e. the user will feel an increased difficulty in moving the robot, and the system will remain stable.

(a) Without configuration change (b) With configuration change

Fig. 6. Two instances of a 2D problem where the task to be teleoperated requires or not a configuration change in order to avoid the black obstacle

The Reactive Module has information of the obstacles of the environment, and therefore can compute the distance d between them and the robot, and make the value of B_r to decrease with d:

$$B_r = \begin{cases} B_r^{max} & d < d_{th} \\ B_r^{max} + \frac{B_r^{min} - B_r^{max}}{d_{cov} - d_{th}}(d - d_{th}) & d_{th} < d < d_{cov} \\ B_r^{min} & d > d_{cov} \end{cases} \quad (9)$$

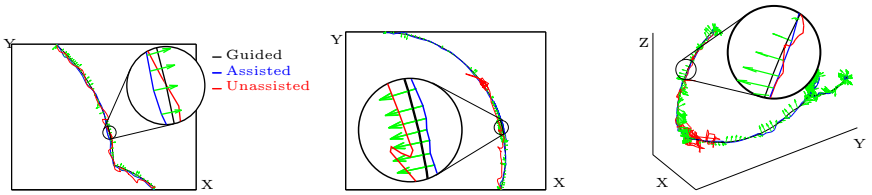
Where d_{th} is a fixed minimum allowable distance to the obstacles, $B_r^{max} > B_r^{min}$ the highest damping value set to slow down the motions as much as possible when the distance threshold d_{th} is exceeded, and d_{cov} the distance from where the effect of the obstacles disappears.

This proposal does not generate repulsive forces, and hence does not interfere with the guiding, while generating a reactive behaviour in front of potential collisions.

4 Evaluation

In order to test the performance of the guiding system, two scenarios have been defined with two tasks to be performed in each one, one requiring a configuration change and another not requiring it. The first scenario involves a RR 2 link planar robot (Fig. 6), and the second one the industrial robot Stäubli TX90 (Fig. 1), that has been used both in simulation and in real experiments. All the tasks have been teleoperated with and without using the guiding aid. To ease the visualization of results, the guiding aid has been restricted to force feedback (i.e. the torque feedback is disabled). The threshold values used for 2D task are $\epsilon_{t1} = 1$, $\epsilon_{t2} = 5$, $\epsilon_{t3} = 10$ and $\rho = 5$ and for the 3D task are $\epsilon_{t1} = 5$, $\epsilon_{t2} = 30$, $\epsilon_{t3} = 50$ and $\rho = 30$ (all values are in mm.).

Fig. 7a shows the data acquired while teleoperating the task in Fig. 6a, where it can be appreciated that the path followed with the guiding aid is smoother.



(a) 2D task without configuration change (b) 2D task with configuration change (c) 3D task with configuration change

Fig. 7. Guided and executed paths in Cartesian space, with and without aid, for the examples of Fig. 6 and 1. The guiding forces that drive the user to follow the guided path are shown with green arrows

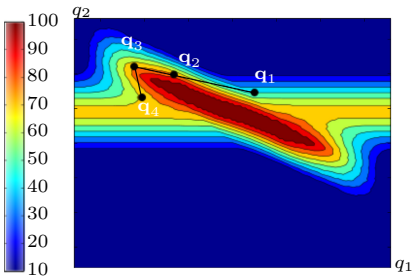


Fig. 8. Values of B_r computed by Eq. (9) on the configuration space corresponding to the task shown in Fig. 6a. The guided path is shown in black

Moreover, it could be executed a 27% faster (mean of 3 users). This figure also shows with green arrows how the generated forces drive the user to follow the guiding path. Fig. 8 illustrates for this example the values of B_r computed using $B_r^{max} = 100$, $B_r^{min} = 10$, $d_{th} = 2$ and $d_{cov} = 50$.

Fig. 7b shows the data acquired while teleoperating the task in Fig. 6b. In this case the task cannot be completed without guiding because it requires a change in the kinematic configuration. Using the guiding aid the task is easily completed, i.e. the user is able to change the configuration of the robot by simply passing near to the singular point, because the system aids to cross it correctly.

The 3D problem with the TX90 Staübli robot is more complex to command because the user has also to handle the orientation of the TCP, which is difficult and requires some training. Fig. 7c shows the data acquired while teleoperating the task in Fig. 1 with a real robot. In this case the task cannot be completed without guiding because a change in the kinematic configuration is also required.

5 Conclusions

The execution of teleoperated task with robots can be improved with guiding systems in order to aid the user to increase not only the velocity but also the safety. Within a bilateral teleoperation framework where an industrial robot is

teleoperated by a desktop haptic device, a guiding system has been proposed based on a previously planned path and on the generation of forces to lead the user towards the path and to move along it. The guiding forces fed back are soft, i.e. the user is by no means restricted to move exactly along the path, and moreover any new path from the configuration where the robot may be can be queried at any time.

Also, the assistance proposed helps in the crossing of singularities often required during the execution of the task, which could not be done otherwise because the teleoperation is carried out at the cartesian level.

The bilateral teleoperation used is based on a proportional plus damping control approach that guarantees stability besides variable time delays, provided the coefficients satisfy a given condition. The assisted teleoperation framework proposed uses the damping factor at the remote site as a safety factor in front of possible collisions, i.e. teleoperation is slowed down by increasing this factor (above the minimum value that satisfies stability) when potential collisions with static or dynamic obstacles can occur. This reactive behavior does not interfere with the guiding.

The proposal has been validated with several users, showing that the assistance provided effectively increases the velocity and safety of the task executions, and eases the teleoperation burden.

References

1. Basañez, L., Suárez, R.: Teleoperation. In: Nof, S. (ed.) Springer Handbook of Automation, pp. 449–4685. Springer (2009)
2. Conti, F., Khatib, O.: Spanning large workspaces using small haptic devices. In: Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 183–188 (2005)
3. Azorin, J.M., Sabater, J.M., Sabater, J.M., Paya, L., Garcia, N.: Kinematics correspondence & scaling issues in virtual telerobotics systems. In: Proc. of the World Automation Congress, pp. 383–388 (2004)
4. Peer, A., Stanczyk, B., Buss, M.: Haptic Telemanipulation with Dissimilar Kinematics. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3493–3498 (2005)
5. Brooks, T., Ince, I.: Operator vision aids for telerobotic assembly and servicing in space. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, vol. 1, pp. 886–891 (1992)
6. Portilla, H., Basañez, L.: Augmented reality tools for enhanced robotics teleoperation systems. In: 3DTV Conference, pp. 1–4 (May 2007)
7. Ivanisevic, I., Lumelsky, V.: Configuration space as a means for augmenting human performance in teleoperation tasks. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 30(3), 471–484 (2000)
8. Payandeh, S., Stanisic, Z.: On application of virtual fixtures as an aid for telemanipulation and training. In: Proc. of the 10th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 18–23 (2002)
9. Rodríguez, A., Nuño, E., Palomo, L., Basañez, L.: Nonlinear control and geometric constraint enforcement for teleoperated task execution. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 5251–5257 (2010)

10. Diolaiti, N., Melchiorri, C.: Obstacle avoidance for teleoperated mobile robots by means of haptic feedback. In: Proc. of the IEEE 1st Int. Workshop on Advances in Service Robotics (2003)
11. Park, H., Lim, Y.A., Pervez, A., Lee, B.C., Lee, S.G., Ryu, J.: Teleoperation of a multi-purpose robot over the internet using augmented reality. In: Proc. of the Int. Conf. on Control, Automation and Systems, pp. 2456–2461 (2007)
12. Maneewarn, T., Hannaford, B.: Augmented haptics of manipulator kinematic condition. In: Proc. SPIE, vol. 3840 (1999)
13. Rosell, J., Vázquez, I.: Haptic rendering of compliant motions using contact tracking in C-space. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 4223–4228 (2005)
14. Hua, J., Gu, D.: Haptically Guided Teleoperation System Based on Virtual Attractive Force. In: Int. Conf. on Mechatronics and Automation, ICMA 2009, pp. 2777–2782 (August 2009)
15. Hua, J., Wang, Y., Xi, N.: Attractive Force Guided Grasping for a Teleoperated Manipulator System Based on Monocular Vision. In: IEEE Int. Conf. on Control and Automation, ICCA 2007, May 30-June 1, pp. 2453–2457 (2007)
16. Jarvis, R.: Terrain-aware path guided mobile robot teleoperation in virtual and real space. In: Proc. of the 3rd Int. Conf. on Advances in Computer-Human Interactions, pp. 56–65 (2010)
17. Rosell, J., Vázquez, C., Pérez, A., Iñiguez, P.: Motion planning for haptic guidance. Journal of Intelligent and Robotic Systems 53(3), 223–245 (2008)
18. Ladeveze, N., Fourquet, J., Puel, B., Taix, M.: Haptic Assembly and Disassembly Task Assistance using Interactive Path Planning. In: IEEE Virtual Reality Conference, VR 2009, pp. 19–25 (March 2009)
19. Basañez, L., Rosell, J., Palomo, L., Nuño, E., Portilla, H.: A framework for robotized teleoperated tasks. In: Proc. of ROBOT 2011 Robótica Experimental, pp. 573–580 (2011)
20. Park, J., Sheridan, T.: Supervisory teleoperation control using computer graphics 1, 493–498 (1991)
21. Wu, B., Klatzky, R., Hollis, R.: Force, Torque, and Stiffness: Interactions in Perceptual Discrimination. IEEE Trans. on Haptics 4(3), 221–228 (2011)
22. Verner, L., Okamura, A.: Force & Torque Feedback vs Force Only Feedback. In: Third Joint EuroHaptics Conf. and Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, World Haptics 2009, pp. 406–410 (March 2009)
23. Pérez, A., Rosell, J.: An assisted re-synchronization method for robotic teleoperated tasks. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 886–891 (2011)
24. Nuño, E., Basañez, L., Prada, M.: Asymptotic stability of teleoperators with variable time-delays. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 4332–4337 (2009)

Adaptive Self-triggered Control for Remote Operation of Wifi Linked Robots

Carlos Santos, Manuel Mazo, Enrique Santiso,
Felipe Espinosa, and Miguel Martínez

Electronics Department, University of Alcala, Spain*
{carlos.santos,mazo,santiso,espinosa,miguel.martinez}@depeca.uah.es

Abstract. In this paper the authors deal with the problem of variable channel delays in a tele-robotics context. Considering the effect of time-varying network delays, a novel self-triggered strategy has been designed to adjust the triggering condition to the observed tracking error. This adaptive self-triggered approach has been implemented on the trajectory tracking of two P3-DX mobile robots remotely controlled through the widespread WLAN IEEE 802.11g standard. Taking into account the maximum channel delay, results show that this solution achieves a good tracking performance with a high reduction of the network occupancy.

Keywords: Adaptive self-triggered control, Network control systems, Variable network delays, Tele-robotics, Wifi Network (IEEE 802.11g).

1 Introduction

Reducing the amount of communication in Network Control Systems (NCS) with a good level of performance is an increasingly interesting research field due to its various benefits. This has motivated the appearance of many techniques that abandon the periodic implementations for strategies based on the idea of sampling only when necessary [1–6]. These techniques use feedback from the state of the plant to decide when the control signal needs to be recomputed with recent measurements. Two of the most important techniques are event-triggered and self-triggered.

In event-triggered control implementations [1], [4], the current state of the plant is measured constantly in order to decide when the control execution must be triggered. On the other hand, the self-triggered control abandons the continuous monitoring of the state of the plant. The strategy of self-triggered control emulates the event-triggered implementation, instead of continuous measurement the control predicts when the system needs to be updated [2, 3]. In this way the control determines the next update time from the last measurement.

In the robotics context, NCS are especially interesting for remote control and cooperative guidance. Several control solutions for the remote control of a

* The authors thank Manuel Mazo Jr, researcher at the Delft Center for Systems and Control, Delft University of Technology (The Netherlands), for its collaboration in this work.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_39,

robotic unit [7] and the guidance of robots platooning [8–13], were developed with periodic sampling and with the assumption of constant network delay. However, in the context of tele-robotics (robotics formation or cooperation), this strategy represents a conservative solution.

On the other hand, the use of the widespread 802.11g IEEE standard presents a challenge due to its variable behaviour. In this technology, the access to the channel is on demand, without collision free periods. For this reason, adding more units to NCS application increases the number and magnitude of the variable delays [14].

In [15], the authors showed in a real P3DX robot how the implementation of a velocity tracking controller in self-triggered form can greatly reduce the amount of changes in the control signals without significant loss of performance. Moreover, they compared different implementation approachs: a periodic, two convencional self triggerd and an adaptive self-triggered one. In this study the benefits of the adaptive implementation were proved. However, constant channel delays was assumed, which can be considered a reasonable assumption only for dedicated communication channels.

In this paper, the authors go on the research line described in [15] dealing with new challenges: trajectory tracking of two P3DX units simultaneously and remotely controlled by the same remote center through a Wifi channel with variable channel delays, as shown in Fig.1 .

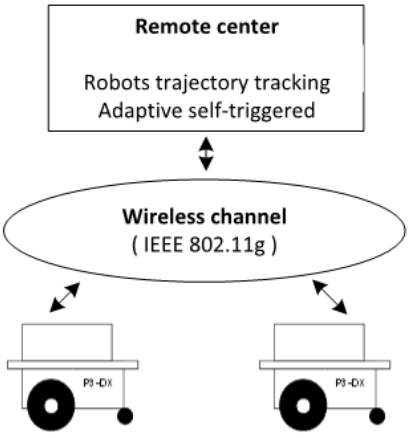


Fig. 1. Structure of the self-triggered control

The rest of the paper is arranged as follows: Sect.2 illustrates the theoretical support. Sect.3 points out the set-up as well as the simulated and experimental results of a remotely controlled P3-DX. Finally the fourth section includes conclusions and future works.

2 Theory

This section introduces the notions of self-triggered control for time-invariant linear systems on the continuous time domain, the specific problem statement and the interest of applying an adaptive triggering condition to solve it.

2.1 Preliminaries

Positive real numbers are denote by $\mathbb{R}^+$, by $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$ and natural numbers by $\mathbb{N}$. The usual Euclidean (l_2) vector norm is represented by $|\cdot|$. A classic result on stability theory for linear systems is presented in the following theorem, see e.g. [16].

Theorem 1. *A linear system $\dot{x} = Ax$ is exponentially stable, i.e. $\exists\ M, \lambda \in \mathbb{R}^+$ such that $|x(t)| \leq Me^{-\lambda t}|x(0)|$, if and only if there exists positive definite matrices P, Q such that:*

$$A^T P + PA \leq -Q. \tag{1}$$

If the linear system is stable then, the function $V(t) = x(t)^T Px(t)$ is said to be a Lyapunov function for the system.

2.2 Self-triggered Control Review

This section briefly reviews the self-triggering policy presented in [2]. The system under consideration is:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{2}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{m\times n}$ are the characteristic matrices and $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^r$ and $y(t) \in \mathbb{R}^m$ the state, input and output vectors respectively. If the pair (A, B) is stabilizable, we find a linear feedback controller rendering the closed loop asymptotically stable:

$$u(t) = Kx(t) \tag{3}$$

The resulting closed loop system is thus described by the equation:

$$\dot{x}(t) = (A + BK)x(t) \tag{4}$$

If the closed loop system is asymptotically stable, there exists a Lyapunov function of the form:

$$V(t) = x(t)^T Px(t) \tag{5}$$

where function $V(t)$ is considered to be positive-definite. Given a symmetric and positive-definite matrix P, Q is the solution to the Lyapunov equation (1).

Now we define the state measurement error as:

$$e(t) = x(t_k) - x(t),\ t \in [t_k, t_{k+1}[\tag{6}$$

where t_k is the latest actuation update instant. At the sampling instant t_k, the state variable vector $x(t_k)$ is available through measurement and provides the current plant information. That is, at times t_k the controller is recomputed with fresh measurements and the input is kept constant until a new measurement is received, *i.e.:* $u(t) = Kx(t_k),\ t \in [t_k, t_{k+1}[$. The closed loop dynamics under this zero-order-hold implementation of the controller is:

$$\begin{bmatrix}\dot{x}(t)\\ \dot{e}(t)\end{bmatrix} = \begin{bmatrix} A+BK & BK \\ -A-BK & -BK\end{bmatrix}\begin{bmatrix}x(t)\\ e(t)\end{bmatrix}\ ; t \in [t_k, t_{k+1}[\tag{7}$$

The objective of the self-triggered control strategy is finding a sequence of update times $\{t_k\}$ such that the number of updates is minimized while stability is preserved. This sequence will be implicitly defined as the times when some triggering condition is violated. To guarantee the stability of the closed loop implementation we introduce a performance function $S : \mathbb{R}_0^+ \times \mathbb{R}^n \to \mathbb{R}_0^+$. This function is forced to upper-bound the evolution of V, thus determining the desired performance of the implementation. Hence, the update times $\{t_k\}$ are determined by the time instants at which next equation is violated:

$$V(t, x_{t_0}) \leq S(t, x_{t_0}), t \geq t_0 \tag{8}$$

The inter-executions times $(t_{k+1} - t_k)$ should be lower bounded by some positive quantity t_{min} in order to avoid Zeno executions[1] [17] of the hybrid system (7).

In order to guarantee inter-execution times greater than zero, it is sufficient to design S satisfying $\dot{V}(t_k) < \dot{S}(t_k)$ at the execution times t_k. This study [2] suggests the use of a function $S(t)$ given by:

$$S(t) := x_s(t)^T P x_s(t) \tag{9}$$

$$\dot{x}_s(t) = A_s x_s(t),\ t \in [t_k, t_{k+1}[\tag{10}$$

$$x_s(t_k) = x(t_k) \tag{11}$$

where A_s is a Hurwitz matrix satisfying the Lyapunov equation:

$$A_s^T P + P A_s = -R \tag{12}$$

with $0 < R < Q$, which guarantees that $\dot{V}(t_k) < \dot{S}(t_k)$. The matrix R describes the stability requirements for the implementation as it defines A_s, which in turn determines $S(t)$.

Finally, how can one compute the next time the controller needs to be refreshed with new measurements? Following the technique in [3], one can predict from a measurement at time t_k and the dynamics of the system, the evolution of the state $x(t_k + t),\ t \in \mathbb{R}^+$. Thus, one can compute ahead of time V and S at times separated Δ units of time, and check if:

$$V(t_k + p\Delta, x_{t_k}) \leq S(t_k + p\Delta, x_{t_k}),\ p \in [1, 2, \ldots, N], \tag{13}$$

[1] A Zeno execution, in our context, is an execution of the system in which an infinite number of discrete events (transmissions) happen within a finite time interval.

for $N \in \mathbb{N}$ some pre-specified horizon. Then, one can compute $t_{k+1} = t_k + p\Delta$ such that: either $p = N$ or:

$$V(t_k + (p+1)\Delta, x_{t_k}) > S(t_k + (p+1)\Delta, x_{t_k}) \tag{14}$$

This paper aims at tracking of piecewise constant reference signals. This means that for non-zero references, the equilibrium of the system x_{eq} is different than the origin, while all the above discussion is performed assuming the equilibrium of the system to be the origin. Nonetheless, this poses no problem, as the new equilibrium point is easily computed from the reference, and a simple change of coordinates: $\tilde{x} = x - x_{eq}$, brings the equilibrium back to the origin.

2.3 Adaptive Self-triggering Condition

To assure the condition $\dot{V}(t_k) < \dot{S}(t_k)$, R is set equal to σQ where $0 < \sigma < 1$. The choice of σ values provides a trade-off between the number of updates and the stability requirements. In a qualitative way, if $\sigma \to 0$, a significant reduction on the controller updates and a corresponding degradation of the performance can be achieved. On the other hand, when $\sigma \to 1$, a better performance was obtained at the cost of increasing the number of updates. To evaluate the performance of the control system, the Integral of the Squared Error (ISE) [18] Index is applied to the output tracking. In this work the ISE has been calculated from sampled measurements:

$$ISE = \sum_{k=0}^{\infty} |y(k\Delta) - y_{ref}(k\Delta)|^2 \Delta. \tag{15}$$

The main idea of the adaptive triggering condition is to take advantage of the benefits of executions reduction without losing performance. Consequently, the σ value is selected depending on the deviation of the output vector of its equilibrium point ($|y(t) - y_{eq}|$).

When $|y(t) - y_{eq}|$ is small enough, the lowest range of the triggering condition ($\sigma \to 0$) is chosen to obtain a reduction of the wireless channel accesses and thus attenuating the channel congestion. On the other hand, when $|y(t) - y_{eq}|$ is significant, the highest range ($\sigma \to 1$) is selected to achieve a fast response of the feedback control system [15]. In this way, a fast response when the system is far of its equilibrium point is achieved and a reduction in the controller updates is obtained when it is near. The threshold values delimiting the mentioned ranges of $|y(t) - y_{eq}|$ are customized by the designer depending on the specific performance requirements and the available resources.

3 Experimental Set-Up and Results

To highlight the contribution of this work on the tele-robotic context, the described strategy has been applied to two P3-DX units using the same Wifi channel (IEEE 802.11g) to communicate with the remote center.

The control strategy presented in this paper allows to split the problem into two parts. The first part is a linear problem, where the designer identifies the plant model of the P3-DX and designs the velocity servocontroller without including the channel delay (Sect 3.1). The second part is a nonlinear problem, in this part a solution, presented in [19], is applied in order to compensate the variable channel delays (Sect 3.2). In this way, the linear model of the P3-DX can be used, even if, the channel delays are variable because this delays will be compensated.

3.1 Plant Model and Servocontroller Design

The first stage for the experimental set-up was modelling the P3-DX robot locally, i.e. without including the wireless channel. The resulting continuous state-space model of the P3-DX robotic unit is:

$$\dot{r}(t) = A_d r(t) + B_d u(t) \tag{16}$$

$$= \begin{bmatrix} -4.094 & -0.015 \\ -0.008 & -5.042 \end{bmatrix} r(t) + \begin{bmatrix} -4.159 & -0.002 \\ -0.001 & -5.057 \end{bmatrix} u(t) \tag{17}$$

$$y(t) = C_d r(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} r(t) \tag{18}$$

$$\tag{19}$$

where:

- $r(t) \in \mathbb{R}^2$ is the plant state vector;(current linear and angular velocities of the robot);
- $u(t) \in \mathbb{R}^2$ is the input state vector (velocities sent to the robot);
- $y(t) \in \mathbb{R}^2$ is the measurement vector obtained from the odometric system (matching the state vector).

A servo-system has been designed to properly track the gradual variation of velocities references (linear and angular). The structure of the servo-system shown in Fig.2 assures null tracking error in steady state. We apply an LQR design technique [20] to the original dynamics extended with the integrator state n. Therefore the controller constants K_I and K_R have been designed for the following system:

$$\begin{bmatrix} \dot{r}(t) \\ \dot{n}(t) \end{bmatrix} = \begin{bmatrix} A_d & 0 \\ -C_d & 0 \end{bmatrix} \begin{bmatrix} r(t) \\ n(t) \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \end{bmatrix} \begin{bmatrix} K_R & K_I \end{bmatrix} \begin{bmatrix} r(t) \\ n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} y_{ref}(t). \tag{20}$$

$$x(t) = [r(t)^T n(t)^T]^T \tag{21}$$

where:

- $n(t) \in \mathbb{R}^2$ is the servo-system integrator state vector;

The weighting matrices used in the LQR design are:

$$Q_{LQR} = \begin{bmatrix} 0.1 I_{2x2} & 0_{2x2} \\ 0_{2x2} & I_{2x2} \end{bmatrix}; R_{LQR} = I_{2x2} \tag{22}$$

and the resulting constants of the controller K_R and K_I are:

$$K_R = \begin{bmatrix} -0.2605 & 0.0004 \\ 0.0005 & -0.2234 \end{bmatrix}; K_I = \begin{bmatrix} 1 & 0.0007 \\ -0.0007 & 1 \end{bmatrix} \tag{23}$$

Figure 2 shows the global structure of the self-triggered control remotely implemented for each robot. Each robotic unit only incorporates the lowest control level related to the active wheels. The remote center, a PC in the same wireless network as the robots, deals with three main tasks: generation of the velocity reference vector to track the desired trajectory, implementation of the servosytem providing the robotic motion actuator, and execution of the self-triggered scheduler. The last one is responsible for deciding when the plant state vector estimation has to be updated and when the control action has to be applied. It is clear that the highest the interval inter-executions the lowest the load of the wireless channel.

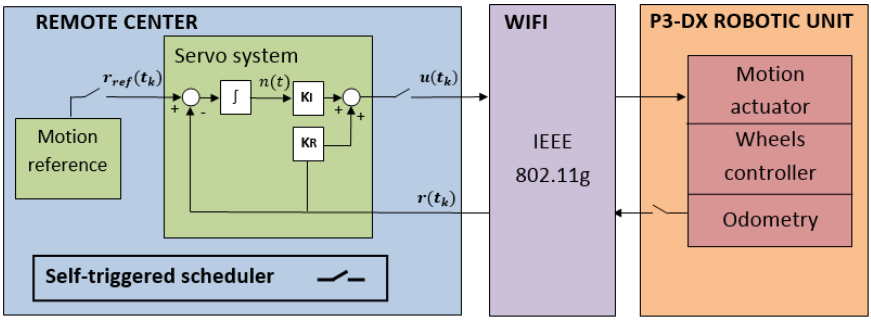


Fig. 2. Structure of the self-triggered control

3.2 Communication Procedure

In [15], authors assumed a constant channel delay and included it as a part of the plant model through a Padé(1,1) approximation. However in this paper, the delay is considered time variant and is not part of the plant model [19, 21, 22]. To compensate the variable channel delay, the robots send their measurements at the time $t_{s_k} = t_k - \tau_{MAX}$ (see Fig.3), where τ_{MAX} is the assumed known maximum wireless channel delay [19]. In this way, the robot is certainly going to receive the information before the time t_k when the control action is updated. This approach has a slight predictive flavor, and requires the remote center to estimate the value of the state vector at the time t_k, in order to compute the control signal $u(t_k)$ (see Fig.4).

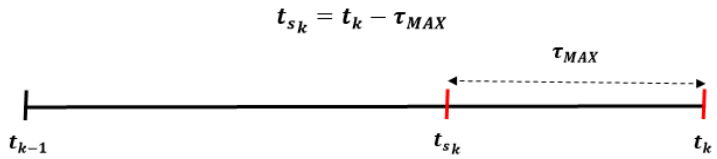


Fig. 3. Transmission instant t_{s_k}

A sketch of the information exchange protocol between a robot and the remote center is depicted in Fig.4.

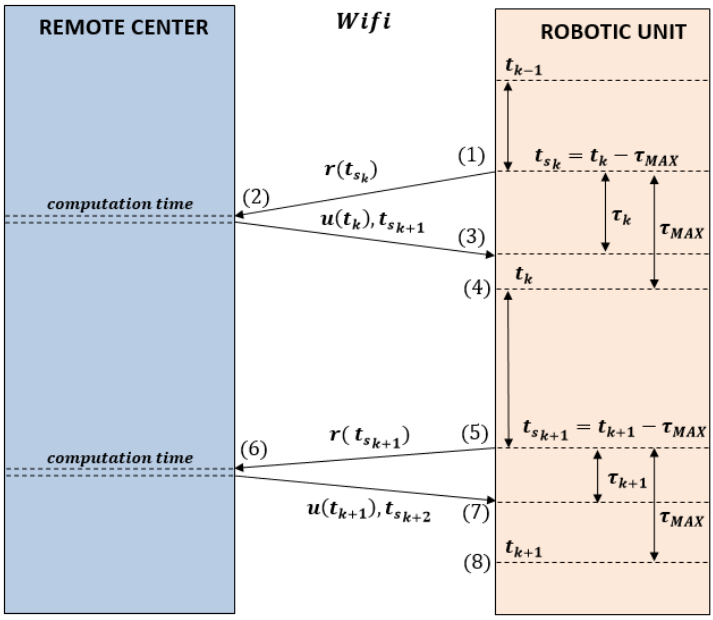


Fig. 4. Control communication protocol

The following steps should be considered:

Step 1: At the time t_{s_k} the robot state $r(t_{s_k})$ is sent to the remote center.
Step 2: The remote center receives the state vector $r(t_{s_k})$ and predicts $\hat{x}(t_k)$ the value of the extended state $x(t_k)$. In this way, the servo controller computes the control signal $u(t_k)$ and the Self-triggered scheduler calculates the next sampling time $t_{(k+1)}$. Finally, the remote center sends to the robot the time $t_{s_{k+1}} = t_{k+1} - \tau_{MAX}$ and the control action $u(t_k)$.
Step 3: The robot receives the mentioned information with a delay τ, where $\tau \leq \tau_{MAX}$, and it waits until time t_k.

Step 4: At time t_k the robot applies the control signal $u(t_k)$.
Step 5: At time $t_{s_{k+1}}$ it begins the next iteration.

In the Step 2, the remote center has to predict the global state $\hat{x}(t_k)$ from $x(t_{s_k})$ as follows:

$$A_\tau = e^{A\tau} \tag{24}$$

$$B_\tau = \int_0^\tau e^{A(\tau - r)} B dr \tag{25}$$

$$\hat{x}(t_{s_k} + \tau_{MAX}) = A_{\tau_{MAX}} x(t_{s_k}) + B_{\tau_{MAX}} u(t_{k-1}) \tag{26}$$

It is important to remark that synchronization is not necessary between the remote center and the robot to measure the total network delay. Only an upper bound on this delay is required.

3.3 Adaptive Control Parameters

The triggering condition is chosen depending on the deviation from the equilibrium state $|y(t) - y_{eq}|$. From all the possibilities for the triggering condition, $\sigma]0; 1[$, only two values are considered. Each one is applied to the case of: lowest magnitude (σ_1), and highest magnitude (σ_2) of the tracking error.

1. $\sigma_1 = 0.2$, for values of $|y(t) - y_{eq}| \leq 0.01$, that means state variables near equilibrium point;
2. $\sigma_2 = 0.8$, for values of $|y(t) - y_{eq}| > 0.01$, it takes place when a change in the reference is applied and a quick answer of the servo-control is required.

The minimum inter-transmission time has to be kept higher than the maximum delay. In the case under study, the maximum delay τ_{MAX} is 25ms. For this reason a $\sigma_2 = 0.8$ is chosen, due to its minimum inter-transmission time is 25.9 ms. In this way, t_{s_k} is always greater than t_{k-1}, see Fig.3.

3.4 Simulation Results

In order to evaluate the theoretical proposal, three simulations on the remote guidance of P3-DX robotic units have been carried out. A combination of linear and angular velocities to follow a desired trajectory during 90 seconds have been chosen as references and the servo-control is implemented in the remote center according to the block diagram shown in Fig.2. We considered for comparison purposes three different implementations of the servo controller:

1. A periodic implementation with constant sampling period equal to the discretization step $\triangle = 10ms$. In this case, due to channel limitations, only the remote control of a robotic unit was simulated.

2. A conservative self-triggered implementation with two fixed triggered conditions: one close to 0 ($\sigma = 0.2$) and other close to 1 ($\sigma = 0.8$).
3. An adaptive self-triggered implementation applying the triggering condition described in Sect. 3.3.

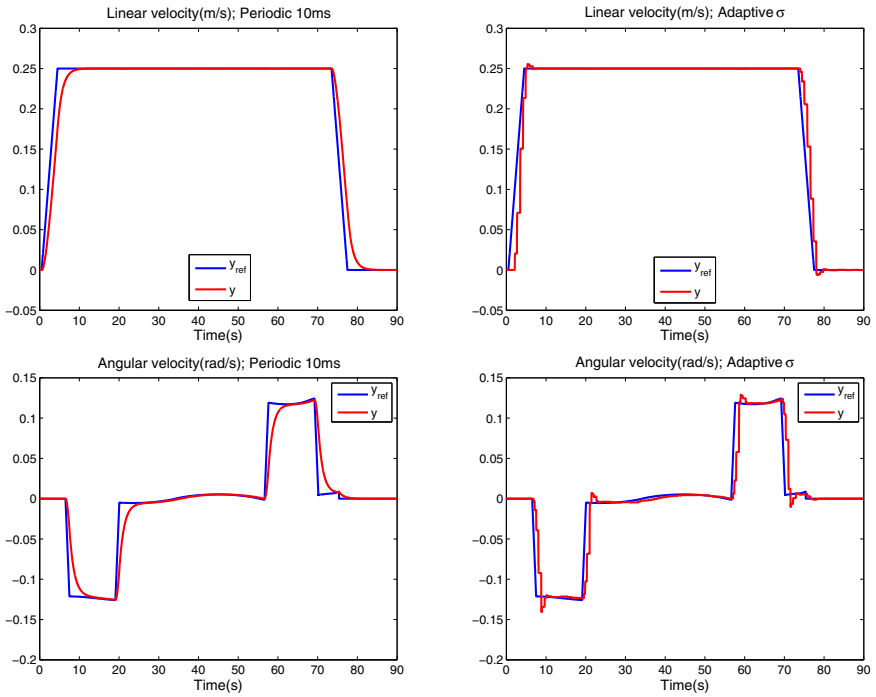


Fig. 5. Registered (red line) linear and angular velocities when a reference (blue line) is applied to the robot. Simulation results from different implementations are shown: periodic sampling (left) and adaptive solution proposed by the authors (right).

Table 1. Number of updates and ISE index obtained comparing different solutions

	Periodic	$\sigma = 0.2$	$\sigma = 0.8$	Adaptive
Updates (Wifi TX)	9000	93	200	135
ISE	0.0616	0.2582	0.06725	0.0705

Figure 5 illustrates the linear velocity (first component of the output vector $y(t)$) and the angular velocity (second component of $y(t)$) from the P3-DX model with different implementations. Pictures on the left side correspond to a fixed sampling time (10ms). Pictures on the right side show our adaptive self-triggered solution presenting a meaningful lower number of channel accesses and

an appropriate control performance. Graphical (see Fig.5) and quantitative (see Table1) results highlight the benefits of the authors' s proposal. The adaptive solution presents, without a significant system performance degradation, a number of updates (wireless transmissions) significantly lower than the periodic case. In addition, the adaptive solution joins the benefits of executions reduction near to $\sigma = 0.2$ result with a level of performance close to $\sigma = 0.8$ solution.

3.5 Experimental Results

Once the authors' proposal is validated by simulation, an experiment is carried out with two real P3-DX robots working over an IEEE 802.11g standard wireless network. These tests were implemented in the laboratories area of the Polytechnics School at the University of Alcala. As it is indicated at Fig. 6, both robots have the opposite origin and destination points. One of the robots departs from Lab 3, moves along a corridor and arrives to Lab 6. The other departs from Lab 6 to Lab 3 (video available at [23]).

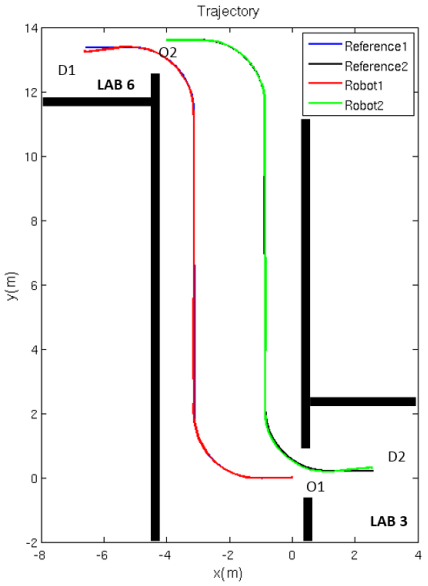


Fig. 6. Trajectory tracking of both robots from origin(O) to destination(D)

In Fig.7, the linear and the angular velocities from robot2 with the adaptive implementation, are shown. Three types of signals are drawn together: reference one (green line) applied to the servo system, real one (blue line) obtained from odometric system, and filtered one (red line) for feedback purposes. It can be appreciated that the adaptive self-triggering solution presents a good control performance.

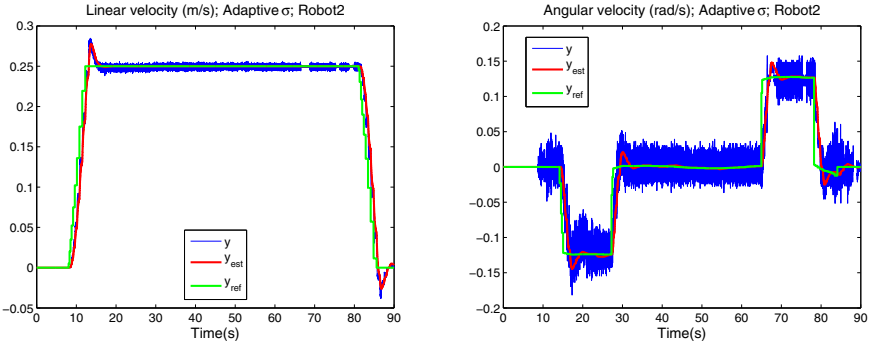


Fig. 7. Linear and angular velocities registered from robot 2

The observed behaviour is confirmed employing a performance measure as the ISE to quantify the tracking quality and the number of transmissions for the network load. Table 2 summarizes these performance measurements for the two robotic units and Fig. 6 presents the registered trajectory of the robots.

Table 2. Key parameters of the experimental results of the adaptive self-triggered strategy

	Robot 1	Robot 2
Updates(Wifi TX)	172	175
ISE	0.0653	0.0680

4 Conclusions

The paper confirms via simulation and real demonstrator results, how an adaptive self-triggered controller can greatly reduce the amount of changes in the control signals applied to remotely controlled robots achieving an appropriate performance and considering variable wireless channel delay. The less the number of channel accesses the less the probability of communication problems (delays, dropouts, congestion, etc).

This solution in turn helps reducing the network traffic is of special significance when several robots are simultaneously tele-operated by the same remote center. However the more units sharing the Wifi channel, the more highly variable delays the channel presents, increasing the maximum delay and restricting the posible performance requirements. For this reason, authors are currently working on a less conservative solution trying to adapt the self-triggering mechanism to the actual state of the channel. This new research step will contribute to optimize the number of units sharing the WLAN with a satisfactory performance.

References

1. Cogill, R.: Event-based control using quadratic approximate value functions. In: Proceedings of the 48th IEEE Conference on Decision and Control, pp. 5883–5888 (December 2009)
2. Mazo Jr., M., Anta, A., Tabuada, P.: On self-triggered control for linear systems: Guarantees and complexity. In: European Control Conference (2009)
3. Mazo Jr., M., Anta, A., Tabuada, P.: An ISS self-triggered implementation of linear controller. Automatica 46(8), 1310–1314 (2010)
4. Tabuada, P.: Event-triggered real-time scheduling of stabilizing control tasks. IEEE Transactions on Automatic Control 52(9), 1680–1685 (2007)
5. Wang, X., Lemmon, M.: Event design in event-triggered feedback control systems. In: 47th IEEE Conference on Decision and Control, pp. 2105–2110 (December 2008)
6. Xue, Y., Liu, K.: Controller design for variable-sampling networked control systems with dynamic output feedback. In: 7th World Congress on Intelligent Control and Automation, WCICA, pp. 6391–6396 (June 2008)
7. Espinosa, F., Salazar, M., Pizarro, D., Valdés, F.: Electronics proposal for telerobotics operation of P3-DX units. In: Remote and Telerobotics, pp. 1–16. Intech (2010)
8. Bocos, A., Espinosa, F., Salazar, M., Valdés, F.: Compensation of channel packet dropout based on TVKF optimal estimator for robotics teleoperation. In: International Conference on Robotics and Automation, ICRA (2008)
9. Espinosa, F., Salazar, M., Valdés, F., Bocos, A.: Communication architecture based on player/stage and sockets for cooperative guidance of robotic units. In: 16th Mediterranean Conference on Control and Automation, pp. 1423–1428 (June 2008)
10. Espinosa, F., Santos, C., Marrón-Romera, M., Pizarro, D., Valdés, F., Dongil, J.: Odometry and laser scanner fusion based on a discrete extended kalman filter for robotic platooning guidance. Sensors 11(9), 8339–8357 (2011)
11. Gǿmez, J.V., Lumbier, A., Garrido, S., Moreno, L.: Planning robot formations with fast marching square including uncertainty conditions. Robotics and Autonomous Systems 61(2), 137–152 (2013)
12. Santos, C., Espinosa, F., Pizarro, D., Valdés, F., Santiso, E., Díaz, I.: Fuzzy decentralized control for guidance of a convoy of robots in non-linear trajectories. In: IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–8 (September 2010)
13. Valdés, F., Iglesias, R., Espinosa, F., RodríGuez, M.A., QuintíA, P., Santos, C.: Implementation of robot routing approaches for convoy merging manoeuvres. Robot. Auton. Syst. 60(11), 1389–1399 (2012)
14. Lee, J.-S., Su, Y.-W., Shen, C.-C.: A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi. In: 33rd Annual Conference of the IEEE Industrial Electronics Society, IECON 2007, pp. 46–51 (2007)
15. Santos, C., Mazo, M., Espinosa, F.: Adaptive self-triggered control of a remotely operated robot. Advances in Autonomous Robotics, 61–72 (2012)
16. Antsaklis, P., Michel, A.N.: Linear Systems. McGraw-Hill (1997)
17. Zhang, J., Johansson, K.H., Lygeros, J., Sastry, S.S.: Dynamical systems revisited: Hybrid systems with Zeno executions. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, pp. 451–464. Springer, Heidelberg (2000)
18. Duton, K., Thompson, S., Barraclough, B.: The art of control engineering. Addison-Wesley (1997)

19. Araujo, J., Mazo Jr., M., Anta, A., Tabuada, P., Johansson, K.: System architectures, protocols and algorithms for aperiodic wireless control systems. IEEE Transactions on Industrial Informatics (2012)
20. Levine, W.: The control handbook. IEEE-Press (1996)
21. Garcia, E., Antsaklis, P.: Model-based event-triggered control with time-varying network delays. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 1650–1655 (December 2011)
22. Meng, X., Chen, T.: Event-based stabilization over networks with transmission delays. J. Control Sci. Eng. 2012, 2:2–2:2 (2012)
23. Video (July 2013), http://www.geintra-uah.org/idi/demostraciones/demostraciones#remote-self-triggered

Effect of Video Quality and Buffering Delay on Telemanipulation Performance

Alexander Owen-Hill, Francisco Suárez-Ruiz, Manuel Ferre, and Rafael Aracil

Centre for Automation and Robotics (UPM-CSIC), 28006 Madrid, Spain
{alex.owenhill,fa.suarez,m.ferre}@upm.es
http://www.car.upm-csic.es/

Abstract. There is much current research into telemanipulation which deals with time delays introduced by the communication channel. However, surprisingly little research has been carried out into how varying delay and quality in solely the video stream affects teleoperation task performance. In this paper the parameters of a) buffering delay and b) video quality (encoding bitrate) are tested to examine their effect on time to task completion and accuracy of a mock welding task with an industrial telemanipulator. From these relationships, several key parameter values arise which can aid engineers when designing video streaming systems for telemanipulation.

Keywords: Teleoperation, communication delay, video quality, video streaming, networking.

1 Introduction

Telemanipulation is widely used in environments into which a human worker cannot enter, such as subsea manipulation [20], military bomb disposal [6], maintenance within radioactive environments [5] and robotic surgery [2]. It involves the control of a robot (often an arm) which is situated in the remote environment for manipulation tasks. As the remote environment is often in a distant location from the operator, delays are introduced in the transmission of information between locations.

There is much current research into telemanipulation which deals with time delays introduced by the communication channel. This research almost entirely focuses on methods to stabilize bilateral control systems in the presence of time delay, both with and without jitter, and several very effective techniques have been developed [13,19,8,14,7]. Telemanipulation over a long distance, such as intercontinental teleoperation over the internet [17], has also been researched thoroughly, again with focus on controller stabilization. While the stability of the control system is a highly important consideration in the development of a telemanipulation system, it is only half of the story, as force reflection is only one of three forms of fundamental feedback in teleoperation, the other two being visual and audio information from the remote environment [21].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_40,

Given the huge importance of visual feedback in telemanipulation, surprisingly little research has been carried out into how varying delay and quality in solely the video stream affects teleoperation task performance. Thompson and Ottensmeyer [10] note that the bandwidth required for teleoperator control signals is very low when compared with that required to send real-time video. Thus, it is likely that bandwidth issues will arise here, even when the communication channel is sufficient for stable telemanipulation.

As video streaming by nature introduces delays and degradation of video quality [9] the effect of this on telemanipulation performance becomes a pertinent consideration for the design of any teleoperation system. Much research has been carried out into techniques for the high-quality transmission of real-time video streams over IP networks [11,23,4]. This research is relevant to teleoperation, where high-quality video is beneficial to the task but differs in that low-latency is not such a high priority. For example, in the broadcasting of live events, high quality at the cost of up to several seconds of latency is usually acceptable, whereas in teleoperation it could mean complete loss of usability of the system.

Experience suggests that the ideal video stream for telemanipulation tasks would be high-bandwidth, low-latency. However, Zhou et al [25] discuss that there exists a trade-off between video bandwidth and delay (latency) in video streaming. Thus, it is not possible to achieve both of these properties simultaneously and it follows to develop a model which allows designers to assess whether it is more advantageous, in their particular setup, to degrade the quality of the encoded video to achieve a more real-time feed or whether it is preferable to introduce some delay to achieve a better picture quality. Of course, the exact values (bitrate and buffer size) are likely to vary depending on the task and specific setup but regardless of this a knowledge of the relationship between buffering delay, video quality and task performance will be useful to system designers.

2 Scope of This Study

In this paper the effect of parameters a) buffering delay and b) video quality (encoding bitrate) on the task completion time and accuracy of task is tested to examine the following hypotheses:

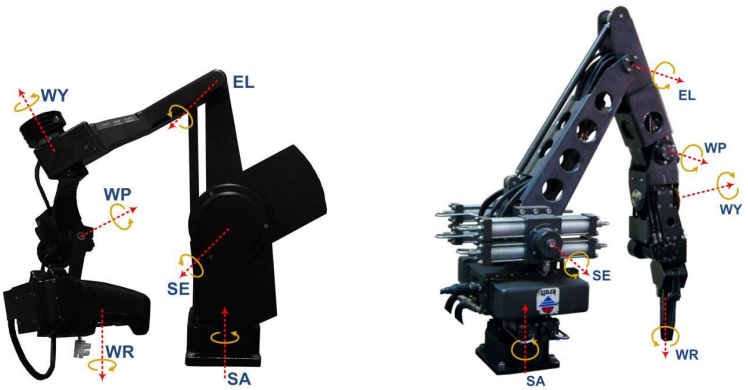
1. Increased buffering delay will increase task completion time.
2. Increased buffering delay will decrease the accuracy of task completion.
3. Reduction in video quality will increase task completion time.
4. Reduction in video quality will decrease accuracy of task completion.

While these hypotheses may be seen as being common-sense, their veracity cannot be taken as certain without such proofs as detailed in this paper. Notably, Thompson and Ottensmeyer [10] have proven that in some surgical telemanipulation tasks, a delay of up to 0.6 seconds can in fact improve task time to completion over control where no such delay exists.

3 Experimental Setup

3.1 Laboratory Setup

The robot used in this study is the Grips Telemanipulator master and slave from Kraft Telerobotics, Fig. 1. The slave is hydraulically actuated, with 6 DoF and a gripper, and uses pressure difference in the hydraulic cylinders to calculate forces. The master, with 6 DoF, is electrically driven in the first 5 joints to allow force reflection and is kinematically similar to the slave.



(a) Master (b) Grips Slave

Fig. 1. Industrial Grips Telemanipulator and master from Kraft Telerobotics

The manufacturer's controller for the Grips Telemanipulator has been replaced by a bilateral controller running on a National Instruments PXI running Real-Time Labview, Figure 2. This control architecture does not include a communication time delay.

The Grips Telemanipulator is largely used for heavy duty applications, such as sub-sea lifting operations. It has a maximum lifting capacity of 82kg with a robust but imprecise control. As such its use in tasks involving detailed manipulation is less than ideal. However, there are situations in which the ability for a telemanipulator which can handle large loads and also carry out precise manipulation are necessary [16].

3.2 Line Following Task – Line Welding

The general performance of any task is not straightforward to quantify, as currently no standard metrics exist to compare different telemanipulators. The most common metric, as used in many studies, is time to task completion. While this is certainly important, it by no means gives the whole story as, for instance, a

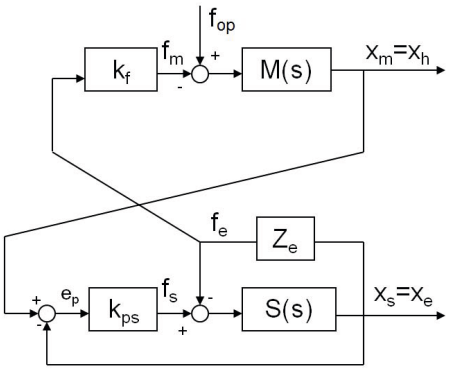


Fig. 2. The Force-Position bilateral control architecture used in the system

person can complete the task very quickly but inaccurately. Some possible metrics, such as the building of lego bricks proposed by Yokokohji [24], are suitable only for very precise manipulators, not the heavy duty manipulator used in this study. In this study a path-following task was chosen, to simulate a welding task as a common maintenance task in many of the application areas. This allowed for the distance to the path to be used as a measure of the task accuracy [22], as force minimization used in many studies was not possible due to a scaling of 10 in the force in the manipulator. However, we predict that the patterns derived in this study will be applicable to any system and task, as they involve the relationships between task performance, video quality and buffering delay.

The task under experiment in this paper is that of welding in a line, a task which requires both accuracy and speed and has been used in telemanipulation maintenance of fusion reactors [12,15]. Figure 3 shows a mock-up of the task as used in this experiment.

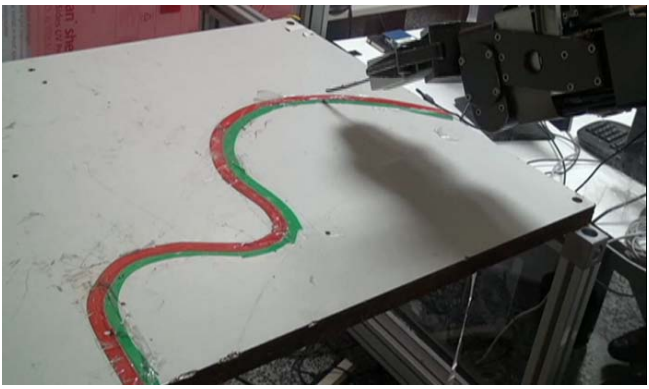


Fig. 3. Mock-up of a welding line with tool

The line following task was to trace the tip of the tool in the gripper manipulator along the center line of this path. As the Kraft manipulator is not designed for very detailed tasks such as this, if the tip was kept within the area marking the line, 2cm each side, it was counted as a successful weld for that section of the path.

3.3 Video Streaming

Video was displayed to the operator using the web page interface shown in Figure 4. The video was streamed from two cameras using Adobe Flash Media Encoder and Server, with the two encoders running on one machine and the server on a second machine, to minimize the possibility of high CPU load causing the video to lag.

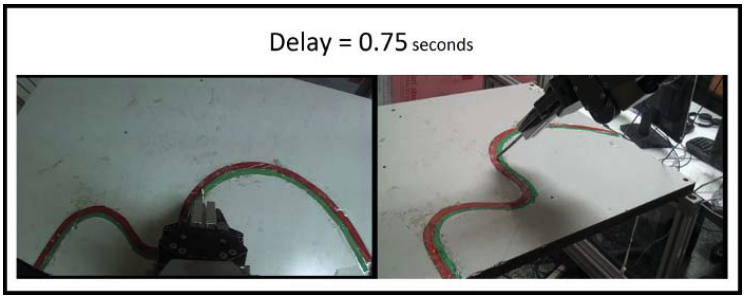


Fig. 4. Web page interface showing simultaneous images from one arm mounted and one "head" mounted camera

The camera views were fixed, with *Camera 1* (a Logitech 1080p webcam) mounted on the end of the arm, to allow a perfect view of the gripper, and *Camera 2* (a Sony 1080p Handycam) in "human's eye" view. Table 1 shows parameters of the 6 streams of these views as entered into Adobe Media Encoder, using VP6 encoding.

Table 1. Encoding properties for 6 streams of video from 2 cameras

Camera	Resolution	Frame Rate	Bitrates (kbps)		
1	864 x 480	30	650	200	50
2	720 x 424	30	650	200	50

Each participant carried out 6 tests, with the following conditions being tested in a random order to avoid learning effects:

1. No delay with bitrates of:
   - 650 kbps
   - 200 kbps
   - 50 kbps

2. 650 kbps and a buffering delay of:
   - 0.5 seconds
   - 0.75 seconds
   - 1 second
   - 1.25 seconds
   - 1.75 seconds

Each participant was tested under all 3 levels of video quality and 3 tests with different delays were chosen randomly, with an almost equally distributed number of tests for each delay value over all the participants.

Due to the limitations of Adobe Flash Media Encoder only 3 different bitrates were able to be encoded simultaneously for each camera. These values were chosen for the following reasons: 650 kbps provided the best quality with smallest visible delay, determined empirically, using the hardware available and is higher (by 44%) than the maximum bitrate (450 kbps) used by Skype VoIP calls; 200 kbps is just less than the average bitrate (300 kbps) used by VoIP calls and produced an approximate halving of the general video quality; and 50 kbps is around the minimum required bandwidth (40 kbps) for VoIP calls [3].

4 Experimental Procedure

Each of 9 participants carried out 6 tests each under different conditions of delay and video quality. This variation was implemented at the encoding of the streams, in the case of bitrate variations, and at the buffering stage, in the case of imposed delay. The network properties were kept constant throughout all tests, with encoding performed on one machine and data transmitted across a 1 Gbit/s LAN to the streaming server, which also displayed the stream to the user. This setup ensured that unknown delays produced by network transmission and CPU load were kept to a minimum. The bilateral master and slave control loops, Figure 2, were computed within the same device, ensuring that the highest possible bandwidth was available for the transmission of video and that encoding bitrate was the only factor affecting the video quality.

The test consisted of tracing a path, shown in Figure 5, from right to left. Participants were encouraged to carry out the task as accurately as possible, not focusing on the time to task completion. They were asked to keep the tip of the tool on the line between the two colored strips of tape, without bending the tool - which had a compliant tip to reduce damage to the tool itself. Before each test began the participant was informed of the properties (delay, bitrate) of the video and this information was also present on the display. This made sure that the task performance results were not affected by the participants' surprise at the presence of delay or jerkiness in the video, with the importance being the level of performance which they could achieve with these properties.

Audio feedback was provided in real-time due to the fact that the manipulator was in the same room as the operator, although behind a shield to block visual view. That there was no delay or degradation in the audio was a reasonable provision as the bandwidth required for real-time audio is generally much less than

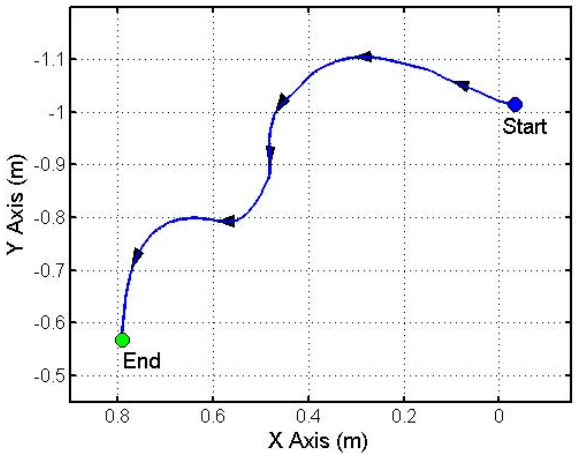


Fig. 5. A plot of the reference path with start and end points.

the video bandwidths investigated in this experiment [1]. This allowed the participants to perceive when the tool made contact with the surface, avoiding huge damages to the tool when a collision with the table occurred. However, as this extra feedback affected the Z-coordinate of the tool tip in a way unattributable to video feedback, this coordinate was disregarded in analysis of the results.

Force reflection in the Kraft system was not a factor in this experiment as it is scaled by a factor of 1:10 between master and slave. Thus, the compliant tool tip lessened the forces applied on the tool by the surface to such an extent as to be imperceptible at the master.

The data collected for each test were:

1. Time to task completion
2. X,Y coordinates of the tool

5 Results

To derive a measurement of task accuracy the X,Y coordinates of the tool tip were analysed in the following way.

A simple mean square error (MSE) cannot be used as a straight-forward metric since any slight difference in velocity can grow into a large error between the compared paths. In order to avoid this undesirable property, a time scaling procedure was performed prior to analysis of the data [18].

5.1 Time Scaling

In order to obtain a metric that is invariant to the speed of movement in any segment of the trajectory, and take into account all the data, the sorter trajectory, meaning the faster test, was taken as reference to establish the amount of

points that will be used to scale all the trajectories. To do this, a continuous function interpolating the trajectories is needed.

In the collected data, the sampling rate was a constant 10 ms and the trajectories were in a range of $25 - 250$ seconds ($2500 - 25000$ points). Given the amount of points, spline interpolation was an adequate interpolation method. Due to the knowledge of the starting and ending points and based on the interpolation, the samples in a trajectory α can be arranged to match the number of their counterparts in the trajectory β:

$$\begin{aligned} \alpha(i) = [x_i, y_i] \quad \forall \quad i = 1, \ldots, I, \quad t = 0, \ldots, T_\alpha \\ \beta(j) = [x_j, y_j] \quad \forall \quad j = 1, \ldots, J, \quad t = 0, \ldots, T_\beta \\ T_\alpha = T_\beta \end{aligned}$$

After time scaling, the metrics shown below were used to determine the accuracy of the tests.

5.2 Accuracy of Test

Percentage of Track Complete. The percentage of the reference line which was "successfully welded" was taken as a measurement of task accuracy. As the telemanipulation system most suited for low precision tasks, a "successfully welded" point was taken as one where the tool tip fell within the two strips of tape, shown in Figure 3, meaning that it fell within 2cm on either side of the reference line. Figure 6 shows one test with the unsuccessful sections shown in black.

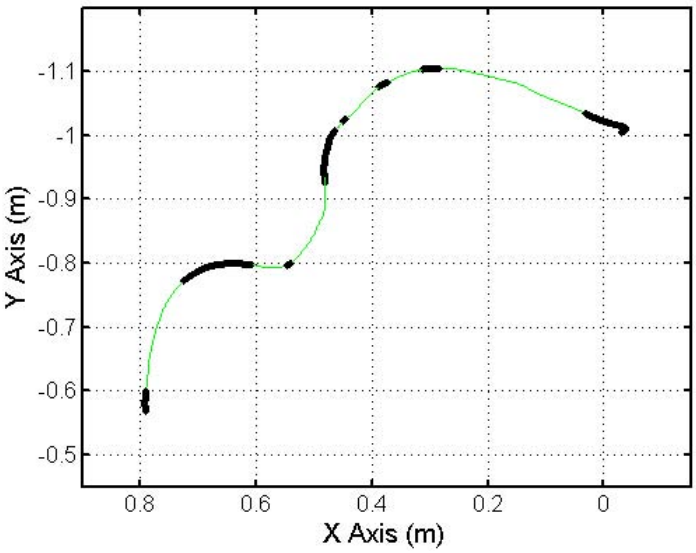


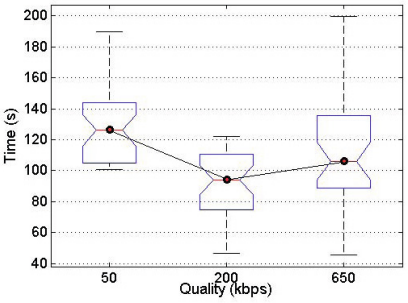
Fig. 6. A test with unsuccessful sections shown. *% track complete = 61.7%.*

Error. As *% track complete* only determines effectiveness at the task, and not accuracy in the task, distance of the tool tip from the reference path was calculated point-by-point in the time-scaled path. From this, the mean error, μ over the whole path was calculated.

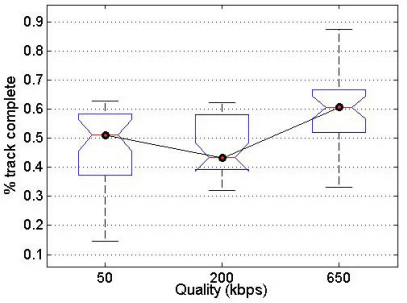
5.3 ANOVA Test

To determine whether each of the test conditions had a significant effect on the task, a one-way ANOVA was carried out on each of the following pairings. A significance value of $p = 0.1$ was taken. A p-value of less than 0.1 indicates a high likelihood that the test condition has had an effect on the result, when compared with random chance.

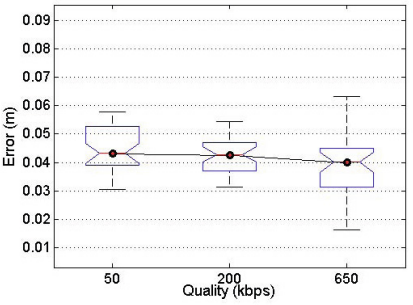
Effect of Video Quality. The results of the ANOVA tests of the video quality on the three factors are shown in Figure 7. Of these it is clear that the video quality had a significant effect on the percentage of the task which was able to be completed successfully by the participants and the time in which it took them.



(a) Time $p = 0.0597$

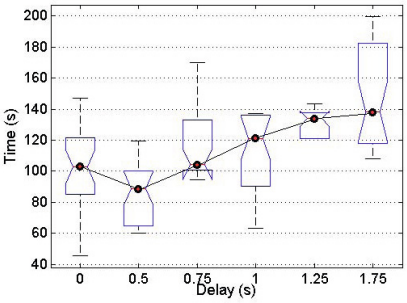


(b) *% track completed* $p = 0.0501$

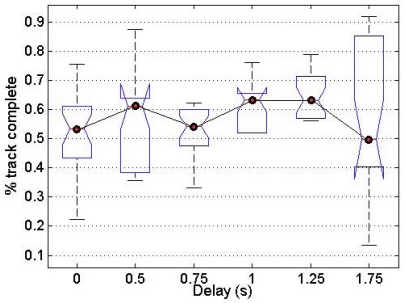


(c) Mean Error $p = 0.365$

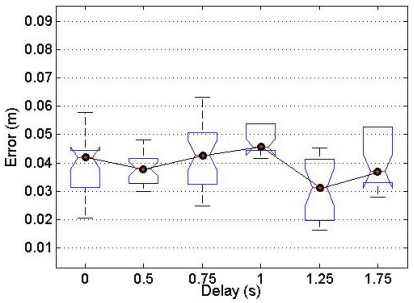
Fig. 7. The effect of Video Quality on the three factors with mean values plotted.



(a) Time $p = 0.024$



(b) *% track completed* $p = 0.6685$



(c) Mean Error $p = 0.3528$

Fig. 8. The effect of Buffering Delay on the three factors with mean values plotted

Large variance in the ANOVA results indicate the wide variability across participants.

With increased quality of video more of the task was completed successfully, with a bitrate of 50 kbps producing the lowest percentage. However, the mean error did not vary significantly as a result of this change. It is also clear that, although there was a very wide variance in the time to completion for the highest quality video, 50 kbps caused the task to take longer time than with higher bitrates. At a quality of 200 kbps tests were generally quicker, but less accurate than those at full quality.

Effect of Buffering Delay. The results of the ANOVA tests of the buffering delay on the three factors are shown in Figure 8. Of these the time to task completion is the only one in which the delay has a statistically significant effect, and is the case even for a significance value of 5%. It is clear that increased buffering delays above 0.5 seconds cause longer times. However, the mean time is in fact reduced in delays of 0.5 seconds when compared to no buffering delay.

The p-value of $p = 66.85\%$ for delay against *% track completed* suggests a high likelihood that delay does not affect this factor. However, for delays between 0.75 seconds and 1.25 seconds the distributions steadily increased for this factor.

While it is not statistically significant that delay affects the mean error of tests there is a definite shift in average mean error at a delay of between 1 second and 1.25 seconds, suggesting a change in the task difficulty at this value of delay. Similarly, the variance in *% task completed* changes drastically between 1.25 seconds and 1.75 seconds.

6 Discussion

6.1 Increased Buffering Delay will Increase Task Completion Time

It is true that task completion times are increased with more delay in the video. This makes sense as the operator must wait longer between movements to see what effect this has had in the remote environment. However, interestingly a delay of 0.5 seconds in the video actually reduces the task completion time. This result, agrees with the result by Thompson and Ottensmeyer [10] who found a similar result with 0.6 seconds of video delay for some tasks.

The reason for this could be that below a certain delay value (somewhere between 0.5 and 0.75 seconds) the operator doesn't perceive the delay as a delay. Instead they may feel that the slave arm is moving slower than they wish it to and speed up their movement to compensate, resulting in a quicker time.

While it cannot be asserted for sure that the accuracy of the task is also improved within this range of delays, there is a suggestion in the data of both *% track completed* and error that this may be the case, but would require further experimentation to verify.

6.2 Increased Buffering Delay will Decrease the Accuracy of Task Completion

From the data it cannot be said for certain that this hypothesis is true and in fact there is evidence that it may not be. For increased delay values there seems to be a trend to complete higher percentage of the track, which could be attributable to their increased concentration to compensate for the delay, up until 1.75 seconds where the participants' success at the task diverges, with a wide variance of *% track completed*.

This divergence concurs with observations made during the tests. All participants agreed that such a high level of delay made the task very difficult; however, between 1.25 and 1.75 seconds delay there seems to be a diffuse border, beyond which some participants were not able to compensate for the delay despite taking a longer time.

6.3 Reduction in Video Quality will Increase Task Completion Time

While it can be said that varying video quality has an significant effect on the time to task completion, the results do not show that lower bitrate necessarily produces longer times. What can be said is that between 200 and 50 kbps the task took a longer time, and this translated to a slightly higher accuracy at the task.

When comparing lower bitrates with 650 kbps, increased times may have resulted in higher accuracy, while for lower bitrates this increase in time did not result in much of an improvement in the task.

6.4 Reduction in Video Quality will Decrease Accuracy of Task Completion

This hypothesis has been shown to be true. While there is a slight improvement in *% track completed* from 200 kbps to 50 kbps, it is likely, as mentioned above, that this is due to increased time spent on the task. In general the trend is that the task is performed more successfully at higher bitrates. It could be that participants were unable to distinguish whether they were touching the line or not due to the lower quality video.

References

1. Brunelli, D., Maggiorotti, M., Benini, L., Bellifemine, F.L.: Analysis of Audio Streaming Capability of Zigbee Networks. In: Verdone, R. (ed.) EWSN 2008. LNCS, vol. 4913, pp. 189–204. Springer, Heidelberg (2008)
2. Casals, A.: Robots in surgery. In: de Almeida, A., Khatib, O. (eds.) Autonomous Robotic Systems. LNCIS, pp. 222–234. Springer, Heidelberg (1998)
3. De Cicco, L., Mascolo, S., Palmisano, V.: Skype Video congestion control: An experimental investigation. Computer Networks 55(3), 558–571 (2011)
4. Deshpande, H., Bawa, M., Garcia-Molina, H.: Streaming Live Media over Peers. Tech. rep., Stanford InfoLab (2001)
5. Draper, J.V.: Human factors in telemanipulation: perspectives from the Oak Ridge National Laboratory experience. In: Kim, W.S. (ed.) Telemanipulator Technology and Space Telerobotics, vol. 2057, pp. 162–174. SPIE (1993)
6. Drascic, D., Milgram, P., Grodski, J.: Learning effects in telemanipulation with monoscopic versus stereoscopic remote viewing. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Cambridge, MA, pp. 1244–1249 (1989)
7. Franken, M., Stramigioli, S., Misra, S., Secchi, C., Macchelli, A.: Bilateral Telemanipulation With Time Delays: A Two-Layer Approach Combining Passivity and Transparency. IEEE Transactions on Robotics 27(4), 741–756 (2011)
8. Hirche, S., Buss, M.: Human perceived transparency with time delay. In: Ferre, M., Buss, M., Aracil, R., Melchiorri, C., Balaguer, C. (eds.) Advances in Telerobotics. STAR, vol. 31, pp. 191–209. Springer, Heidelberg (2007)

9. Hou, Y., Zhu, W., Zhang, Y.Q., Peha, J.: Streaming Video over the Internet: Approaches and Directions. IEEE Transactions on Circuits and Systems for Video Technology 11(3), 282–300 (2001)
10. Thompson, J.M., Ottensmeyer, M.P., Sheridan, T.B.: Human factors in telesurgery: effects of time delay and asynchrony in video and control feedback with local manipulative assistance. Telemedicine 5(2), 129–137 (1999)
11. Liu, Y., Guo, Y., Liang, C.: A survey on peer-to-peer video streaming systems. Peer-to-Peer Networking and Applications 1(1), 18–28 (2008)
12. Ming, H., Huat, Y., Lin, W., Bin, Z.: On teleoperation of an arc welding robotic system. In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN, vol. 2, pp. 1275–1280 (April 1996)
13. Niemeyer, G., Slotine, J.-J.E.: Telemanipulation with Time Delays. The International Journal of Robotics Research 23(9), 873–890 (2004)
14. Nuño, E., Basañez, L., Ortega, R.: Control of Teleoperators with Time-Delay: A Lyapunov Approach. In: Loiseau, J.J., Michiels, W., Niculescu, S.-I., Sipahi, R. (eds.) Topics in Time Delay Systems. LNCIS, vol. 388, pp. 371–381. Springer, Heidelberg (2009)
15. Onozuka, M., Alfile, J.P., Aubert, P., Dagenais, J.F., Grebennikov, D., Ioki, K., Jones, L., Koizumi, K., Krylov, V., Maslakowski, J., Nakahira, M., Nelson, B., Punshon, C., Roy, O., Schreck, G.: Manufacturing and maintenance technologies developed for a thick-wall structure of the ITER vacuum vessel. Fusion Engineering and Design 55(4), 397–410 (2001)
16. Palmer, J., Irving, M., Järvenpää, J., Mäkinen, H., Saarinen, H., Siuko, M., Timperi, A., Verho, S.: The design and development of divertor remote handling equipment for ITER. Fusion Engineering and Design 82(15-24), 1977–1982 (2007)
17. Peer, A., Hirche, S., Weber, C., Krause, I., Buss, M., Miossec, S., Evrard, P., Stasse, O., Neo, E., Kheddar, A.: Others: Intercontinental multimodal tele-cooperation using a humanoid robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 405–411. IEEE (2008)
18. Pomplun, M., Matarić, M.: Evaluation Metrics and Results of Human Arm Movement Imitation. In: Proceedings of the 1st IEEE-RAS International Conference on Humanoid Robotics, pp. 7–8 (2000)
19. Prokopiou, P.A., Tzafestas, S.G., Harwin, W.S.: Towards Variable-Time Delays-Robust Telemanipulation Through Master State Prediction. In: Proceedings of the IEEE/ASME International Conference on Advance Intelligent Mechatronics, pp. 305–310 (1999)
20. Ridao, P., Carreras, M., Hernandez, E., Palomeras, N.: Underwater telerobotics for collaborative research. In: Ferre, M., Buss, M., Aracil, R., Melchiorri, C., Balaguer, C. (eds.) Advances in Telerobotics. STAR, vol. 31, pp. 347–359. Springer, Heidelberg (2007)
21. Sheridan, T.B.: Teleoperation, telerobotics and telepresence: A progress report. Control Engineering Practice 3(2), 205–214 (1995)
22. Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., Goodrich, M.: Common metrics for human-robot interaction. In: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction, HRI 2006, pp. 33–40. ACM, New York (2006)

23. Wang, Z., Banerjee, S., Jamin, S.: Studying Streaming Video Quality: From An Application Point of View. In: Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA, pp. 327–330 (2003)
24. Yokokohji, Y., Imaida, T., Iida, Y., Doi, T., Oda, M., Yoshikawa, T.: Bilateral Teleoperation: Towards Fine Manipulation with Large Time Delay. In: Rus, D., Singh, S. (eds.) Experimental Robotics VII. LNCS, vol. 271, pp. 11–20. Springer, Heidelberg (2001)
25. Zhou, L., Zhang, Y., Rodrigues, J., Geller, B., Cui, J., Zheng, B.: Quality-Delay Tradeoff for Video Streaming over Mobile Ad Hoc Networks. In: IEEE Internation Conference on Communications, pp. 228–232 (2012)

Pose Estimation of a Mobile Robot Based on Network Sensors Adaptive Sampling

Miguel Martínez, Felipe Espinosa, Alfredo Gardel,
Carlos Santos, and Jorge García

Electronics Department, University of Alcalá, Spain
{miguel.martinez,espinosa,alfredo,carlos.santos,
jorge.garcia}@depeca.uah.es
http://www.geintra-uah.org

Abstract. This paper covers the use of an adaptive sampling strategy on a sensor network for the estimation of the pose of a remotely controlled mobile robot. The more measurements can be taken, the better the estimation becomes, but if some level of uncertainty is tolerable, the amount of measurements can be significantly reduced. The proposed adaptive sampling technique is triggered by the estimation error covariance matrix, computed by Kalman filter algorithms. This sampling strategy leads to the minimum amount of measurements needed to achieve a certain estimation accuracy, reducing the load on important resources such as computational power and network communications. An example of use is given as the pose estimation of a wheeled robot remotely controlled. A network of camera sensors provides pose measurements allowing the application of an Unscented Kalman Filter estimator in the remote centre.

Keywords: Pose Estimation, Event Based Estimation, Adaptive Sampling, Unscented Kalman Filter.

1 Introduction

The recent popularity of Network Control Systems (NCS) has motivated the study of state estimators with asynchronous measurements in the last years [1,2] especially when they are combined with Wireless Sensor Networks (WSN)

Communication between NCS elements such as sensors to estimator, estimator to control and control to plant are driven through band-limited links, often shared with other nodes and systems. As a result, messages can experiment different delay times and can even be lost. Thus, the communication channels become a valuable resource so control and estimation strategies are being developed in order to make an efficient use of it.

Event-trigger control [3] reduces the use of computation and network resources triggering the execution of the controller according to the state of the plant. The main drawback is that the plant needs to be permanently monitored. Self-trigger control [4] takes this idea further and plans ahead the time of the next control update.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_41,

Many event based estimation strategies have been proposed [5,6,7]. A common approach, known as send-on-Delta or Lebesgue sampling, is to trigger the sampling when the signal deviates from the previous sample by some quantization level [8]. The sampling events do not depend on time but on the value of the measured signal.

This sampling strategy requires the sensor to be running and periodically monitoring the signal, waiting for the triggering condition to happen in order to send the sample. It also requires some intelligence in the the sensor side. If the triggering condition is set on the state associated with a signal and not in the signal itself, some processing or estimation has to be performed.

As an example, a camera measuring the position of a robot with an object recognition algorithm takes pictures from a scene. The sampling event should be triggered by the robot moving a certain distance, and not by some variation of the image, which can be caused by changes in illumination or movement of nearby objects.

The present work studies an adaptive sampling strategy based on the uncertainty of the estimation instead of the measured signal. The decision of triggering a measurement is made by the estimator, located in a remote centre, and the sensor can be kept off in between measurements.

This adaptive measuring estimator scheme and how the estimation error covariance (calculated by the Unscented Kalman Filter algorithm) plays an important role on triggering the measurements is covered on Sect. 2. On Sect. 3 a simulation demonstrating the authors' method is presented. Results from the simulation are shown and discussed on Sect. 4 and finally on Sect. 5 conclusions are drawn.

2 Adaptive Measuring Estimator

A linear state-space system is defined by a matrix differential equation like

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the vector of inputs and $\mathbf{w} \in \mathbb{R}^n$ the system noise. $\mathbf{A}$ and $\mathbf{B}$ are matrices with appropriate dimensions.

Although the plant model can be a continuous-time or discrete-time model, the measurements of the outputs of the plant are usually taken at certain time instants t_k.

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

$\mathbf{y}_k \in \mathbb{R}^l$ is the output vector at instant t_k and $\mathbf{v}_k \in \mathbb{R}^l$ is the measurement noise.

These instants, although labelled by suffix k may not be periodical, so time between measurements may not be constant. Then, in general

$$t_k - t_{k-1} \neq T \quad \forall k \ . \tag{3}$$

The problem of estimation is to approximate $\mathbf{x}$ knowing $\mathbf{u}$ and $\mathbf{y}$. A secondary problem in estimation is how many information from the output is needed to have an appropriate estimation.

2.1 Kalman Filter Estimators for Linear and Non-linear Systems

The Kalman Filter is an optimal estimator if the system noise and measurement noise are zero-mean gaussian uncorrelated noises, but even when these conditions are not satisfied, the Kalman filter often yields reasonably good results [9,10].

Noises are described by their covariance matrices.

$$\mathbf{Q} = E\left[\mathbf{w}\mathbf{w}^T\right] \tag{4}$$

$$\mathbf{R} = E\left[\mathbf{v}\mathbf{v}^T\right] \tag{5}$$

Kalman estimators take into account not only the mean value of the estimation $\hat{\mathbf{x}}$ but also the uncertainty of the estimation with a covariance matrix $\mathbf{P}$. It is defined as the covariance of the estimation error.

$$\mathbf{P} = E\left[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\right] \tag{6}$$

So the estimation is really a gaussian probability density function (PDF) with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$. As a covariance matrix, $\mathbf{P}$ is symmetric and positive-definite. The elements of the diagonal of this matrix are the variances of the estimation of each state, and the rest of the values are the cross-covariances of each pair of states.

$$p_{i,j} = E\left[(x_i - \hat{x}_i)(x_j - \hat{x}_j)\right] \tag{7}$$

$$p_{i,i} = \sigma^2_{\hat{x}_i} \tag{8}$$

The Kalman Filter is divided into two different stages: prediction and correction. The prediction part is done continuously. The estimation is updated by applying the known model of the plant, such as (1). $\mathbf{P}$ is updated accordingly, as a function of the plant dynamics and the system noise covariance $\mathbf{Q}$. This usually makes $\mathbf{P}$ rise.

The correction part can be performed whenever some measurement from a sensor is available. The correction of the prediction is proportional to the difference between the measurement and the predicted output.

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{L}_k\left(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-\right) \tag{9}$$

The symbols $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{x}}_k^+$ stand for the estimated states before the correction (*a priori* estimation) and after the correction (*a posteriori* estimation) respectively.

The Kalman gain $\mathbf{L}_k$ is a function of the measurement quality – given by $\mathbf{R}$ – and the *a priori* estimation covariance $\mathbf{P}$, which in turn depends on $\mathbf{Q}$. After the correction is made, $\mathbf{P}$ is updated. The information provided by the measurement lowers the estimation uncertainty. Hence, $\mathbf{P}$ is reduced every time a correction is performed [10,11].

Non-linear Filters. The Unscented Kalman Filter (UKF) is an estimation algorithm for non-linear systems based on the Kalman Filter that outperforms the Extended Kalman Filter (EKF) when the non-linearities are severe [12].

The EKF simply linearizes the plant around the working point and proceeds as with the linear Kalman Filter. The UKF propagates the mean and covariance of the estimation with a statistical approach, giving more accurate results and better estimations of the error covariance matrix, without a significant increase in computation complexity.

2.2 Conditional Measuring

The diagonal values of $\mathbf{P}$ as seen in (8) are a good indicator of the quality of the estimation. The square root of these values are the standard deviation of the estimation errors. These errors can be lowered by getting more measurements from the sensors.

Although getting measurements always improves the estimation of the states, when the estimation is already good enough, it would be interesting to keep from using the sensors. It can reduce the effort taken in getting the measurement, communicating, or processing it, which in turn reduce power consumption, traffic load in the communication networks and computational load.

The use of the sensors can be managed by the needs of the estimator to keep $\sigma_{\hat{x}_i}$ below some bounds. The sensors can remain turned off until the diagonal values of $\mathbf{P}$ exceed some thresholds. When any of the diagonal values is greater than its respective threshold, a measurement is triggered which lowers these values again. The trigger condition for a sensor can be expressed as

$$\bigcup_{i=1}^{n} \sqrt{p_{i,i}} > \sigma_{\mathrm{thr}\,i} \tag{10}$$

where $\sigma_{\mathrm{thr}\,i}$ is the threshold value for the i-th state.

In a linear time-invariant plant where the system and measurement noises have constant covariance matrices, setting up such thresholds to trigger the sensors leads to periodical sampling, and in turn, periodical sampling leads to a steady state of both $\mathbf{P}$ and $\mathbf{L}$ [10]. This is because every measurement carries the same amount of information, and $\mathbf{P}$ always grows in the same way. When the growth of $\mathbf{P}$ in between measurements and the information provided by each measurement are equal and cancel each other out, the steady state of the estimation is reached.

However, if the plant is not time-invariant, non-linear or the noise covariances are not constant, this steady-state may not exist. In this case, some thresholds for $\sigma_{\hat{x}_i}$ can reduce the use of the sensors and still keep the uncertainty under the desired level.

Figure 1 describes the sampling strategy with an example. Suppose we have a single sensor for two states x_1 and x_2 we want to estimate with a certain accuracy. This accuracy imposes higher bounds to each state estimation error standard deviation $\sigma_{\hat{x}_i}$, which in turn results in a threshold condition for $\mathbf{P}$, as in (10).

For simplicity, a single threshold level was set for both states. Every time any of the $\sigma_{\hat{x}_i}$ curves touch the threshold line, a measurement is triggered, which makes them decrease.

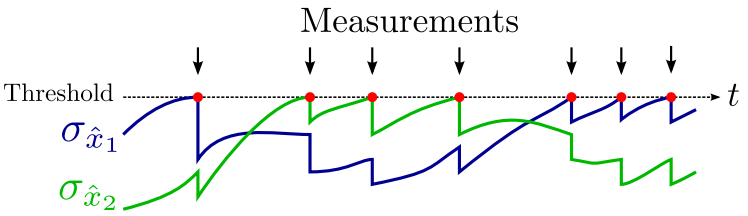


Fig. 1. Example of the adaptive sampling strategy with a single threshold for two states

Nevertheless, not every possible value for $\sigma_{\hat{x}_i}$ can be achieved. The limit is imposed by the fastest pace the sensor can take the measurements and the noise related to them. If the thresholds are too low, the sensors will work at the fastest rate and the estimation will be as better as it can be but its variance would be worse than the (unachievable) target variance.

3 Problem Description

The following section illustrates the benefits of adaptive sampling of the pose estimation of a wheeled robotic unit. While there is a non-linear relationship among the state variables, an UKF is applied to correct the odometric prediction through external camera sensors.

3.1 System Description

The proposed system is composed by a mobile robotic unit with encoders attached to each active wheel, an external sensor network, and a remote control centre, as pictured in Fig. 2. It is based on the system described in [13] for the remote control of a P3DX unit, but with the addition of an external sensor network that is able to measure the robot pose.

The robot accepts commands of linear and angular speed, sent by the remote centre through a wireless communication channel. The robot dynamics are modelled with a continuous-time state-space plant model where the channel delay is approximated through a *Padé*$(1, 1)$ approximation [11,14]. The state vector and input vectors of the plant are as follows:

$$\mathbf{x} = \begin{bmatrix} v & \omega & v_{\mathrm{d}} & \omega_{\mathrm{d}} \end{bmatrix}^T \tag{11}$$

$$\mathbf{u} = \begin{bmatrix} c_v & c_\omega \end{bmatrix}^T \tag{12}$$

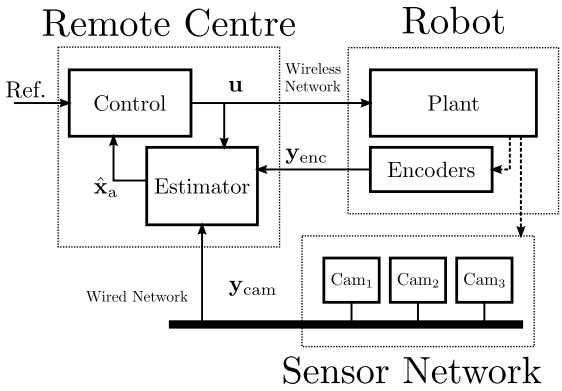


Fig. 2. Block diagram showing the different elements of the simulated system and their interactions

where v and ω are the linear and angular speed respectively, v_{d} and ω_{d} the states associated with the delays, and c_v and c_ω represent the commands of linear and angular speed.

The identified state-space model of the plant is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w} \tag{13}$$

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \\ \dot{v}_{\text{d}} \\ \dot{\omega}_{\text{d}} \end{bmatrix} = \begin{bmatrix} -4.094 & -0.015 & 1664 & 0.7227 \\ -0.008 & -5.042 & 0.326 & 2023 \\ 0 & 0 & -200 & 0 \\ 0 & 0 & 0 & -200 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ v_{\text{d}} \\ \omega_{\text{d}} \end{bmatrix} + \begin{bmatrix} -4.159 & -0.002 \\ -0.001 & -5.057 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_v \\ c_\omega \end{bmatrix} + \mathbf{w} \tag{14}$$

The system noise vector $\mathbf{w}$ affects the states in the form of disturbances that may appear and uncertainties on the system model and parameters.

Rotary encoders on the wheels allow to measure the actual linear and angular speeds, but these measurements are affected by noise, caused by wheel slippage, uncertainties on the wheel diameter (because of surface wear and/or air pressure), and the quantification of the measurements. The equation that describes the output of the plant is

$$\mathbf{y}_{\text{enc}} = \mathbf{H}_{\text{enc}}\mathbf{x} + \mathbf{v}_{\text{enc}} \tag{15}$$

$$\mathbf{y}_{\text{enc}} = \begin{bmatrix} v \ \omega \end{bmatrix}^T + \mathbf{v}_{\text{enc}} \tag{16}$$

where $\mathbf{v}_{\text{enc}}$ is the measurement noise of the encoders. Hence, $\mathbf{H}_{\text{enc}}$ is

$$\mathbf{H}_{\text{enc}} = \begin{bmatrix} \mathbf{I}_{2\times 2} \ \mathbf{0}_{2\times 2} \end{bmatrix} \ . \tag{17}$$

Additionally, a sensor network is distributed along the working area and their data is sent to the remote centre to estimate the robot position.

3.2 Servo-System

The control strategy is designed as a servo-system as depicted in Fig. 3. The integrator assures null tracking error in steady-state but introduces two more states to the closed-loop system:

$$\dot{\mathbf{n}} = \mathbf{y}_{\text{ref}} - \mathbf{y}_{\text{enc}} \tag{18}$$

where $\mathbf{n} \in \mathbb{R}^2$ is the vector containing the two additional states and $\mathbf{y}_{ref}$ is the vector of reference speeds.

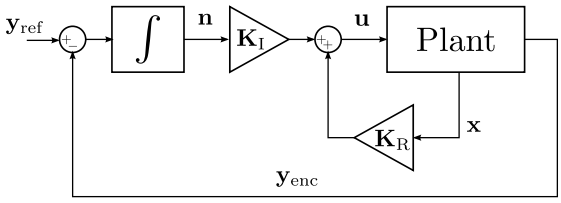


Fig. 3. Block diagram of the servo-system

The value of the $\mathbf{K}_{\text{R}}$ and $\mathbf{K}_{\text{I}}$ gains are computed applying the LQR design technique [11].

The control blocks are located in the remote centre. The speed commands are computed and transmitted to the robot at some discrete time instants. This requires that the full state vector, as well as the outputs (the encoders measurements) is known in the remote centre, which implies the need of an estimator.

In a previous work [13] this control structure has been used in an adaptive self-triggered control scheme in order to reduce and optimize the use of the communication channel between the remote centre and the robot, keeping the performance of the system at an acceptable level.

For simplicity, in this work the control block is executed periodically, every 200 ms. Because the speed states are modelled with a time-invariant linear system and the encoder noise is constant, the measurements needed for the estimation will be periodical. In order to reduce the communication between the robot and the remote centre to only the instants of control events, the information of the encoders is only sent every 200 ms.

3.3 Pose Estimation

In addition to the estimation of the plant states, a remaining problem is the estimation of the robot pose. The state vector described in (11) can be augmented to include the position and orientation.

$$\dot{x} = v \cos\theta \tag{19}$$

$$\dot{y} = v \sin\theta \tag{20}$$

$$\dot{\theta} = \omega \tag{21}$$

$$\mathbf{x}_{\mathrm{a}} = \left[x\ y\ \theta\ v\ \omega\ v_{\mathrm{d}}\ \omega_{\mathrm{d}}\right]^T \tag{22}$$

The variables x and y are scalars that denote the position of the robot in each axes and should not be confused with the state vector and output vector $\mathbf{x}$ and $\mathbf{y}$. θ denotes the orientation of the robot.

For the augmented state vector, the state equation becomes an

$$\dot{\mathbf{x}}_{\mathrm{a}} = f\left(\mathbf{x}_{\mathrm{a}}, \mathbf{u}\right) + \mathbf{w}_{\mathrm{a}} \tag{23}$$

where f is the combination of (14) and (19)-(21). $\mathbf{w}_{\mathrm{a}}$ is the augmented system noise vector:

$$\mathbf{w}_{\mathrm{a}} = \left[w_x\ w_y\ w_\theta\ \mathbf{w}^T\right]^T \tag{24}$$

The noises affecting the pose states (w_x, w_y, and w_θ) are due to vibrations of the robot. These noises are negligible.

Equations (19) and (20) are not linear, so a non-linear estimator such as the UKF is needed to estimate these states.

With the use of the rotary encoders, odometry techniques allow for a basic position estimation as long as the start position is known, but because of the aforementioned noise of (15) it has the drawback of cumulative error. This is why external sensors are needed in order to accurately determine the robot position at any given time. In an intelligent space, the pose is usually obtained with cameras and computer vision techniques [15].

$$\mathbf{y}_{\mathrm{cam}} = \mathbf{H}_{\mathrm{cam}}\mathbf{x}_{\mathrm{a}} + \mathbf{v}_{\mathrm{cam}} \tag{25}$$

$$\mathbf{y}_{\mathrm{cam}} = \left[x\ y\ \theta\right]^T + \mathbf{v}_{\mathrm{cam}} \tag{26}$$

$\mathbf{H}_{\mathrm{cam}}$ looks like this:

$$\mathbf{H}_{\mathrm{cam}} = \left[\mathbf{I}_{3\times 3}\ \mathbf{0}_{3\times 4}\right] \tag{27}$$

Capture times and processing times limit the rate of pose measurements cameras can provide. In this work, a tentative rate of 12.5 measurements per second (12.5 FPS, 80 ms per image) was assumed. The noise related with these sensors represent the errors in the object recognition algorithms used and the quantification imposed by the pixel resolution.

Another limitation of cameras is the working area they can monitor. For a wide area, several cameras are needed to cover every section of it with good accuracy. The farther the robot is, the worse the localization error becomes.

In this paper, 4 cameras were simulated, which covered any spot of the working area with good accuracy. These cameras are modelled after the Unibrain Fire-i cameras, using the pin-hole geometric model [16]. These cameras have an image resolution of 640×480 pixels and a focal length of 4.3 mm. The cameras were located 3 metres above the floor with a downwards 30° tilt angle.

To locate the robot, an image is captured by the camera which provides better accuracy for the current estimated position. An image recognition algorithm would yield a pixel of the image that is related to a point in the floor ($z = 0$ plane)

using the geometric model equations. To simulate the errors and disturbances on this image recognition algorithm, gaussian random numbers were added to each coordinate of the point in the image captured by the camera. This numbers had zero mean and a standard deviations of 25 pixels on both image axes.

To obtain the noise covariance matrix $\mathbf{R}_{\text{cam}}$ of each pose measurement, the covariance of these random numbers have to be transformed through the non-linear equations of the camera model. The Unscented Transformation [12] was the method used to address this problem. The orientation of the robot is supposed to be also obtained by the image recognition algorithm. A gaussian noise is added to it with a standard deviation of $\pi/4$ radians. These noise values are exaggeratedly big but serve the purpose of proving how good the estimation can become even with poor measuring conditions.

4 Simulation Results

The simulation was performed with Matlab in a discrete manner, with a simulation step time of $T_s = 10\,\text{ms}$. The system described in (14) and (19-21) had to be converted to a discrete system of the form

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_{k-1} \tag{28}$$

where $\mathbf{F}$ and $\mathbf{G}$ are computed with the state-space discretization method [11]:

$$\mathbf{F} = e^{\mathbf{A}T_s} \tag{29}$$

$$\mathbf{G} = \mathbf{A}^{-1}\left(\mathbf{F} - \mathbf{I}\right)\mathbf{B} \tag{30}$$

4.1 Case A. Periodic Sampling

A combination of linear and angular speeds were chosen as the reference signals. Figures 4 and 5 show the speed references as well as the actual speeds developed by the robot and the estimated speeds.

The circles represent the measurements of speed read from the encoders. The estimator follows the real state accurately, with an estimation error below $0.01\,\text{m/s}$, which is an order of magnitude smaller than the reference values.

As an indicator of the goodness of the estimation, $2\sigma_{\hat{v}}$ and $2\sigma_{\hat{\omega}}$, as well as the actual estimation errors, are shown.

Because the estimation is a probability density function, the real state value must lie close to the estimated one and the difference between the two is likely to be close to the standard deviation σ of the PDF. For a gaussian distribution, about 95.45% of the samples are found within the region $(\mu \pm 2\sigma)$. In the lower plots it can be seen that the estimation error is most of the time below the 2σ curve.

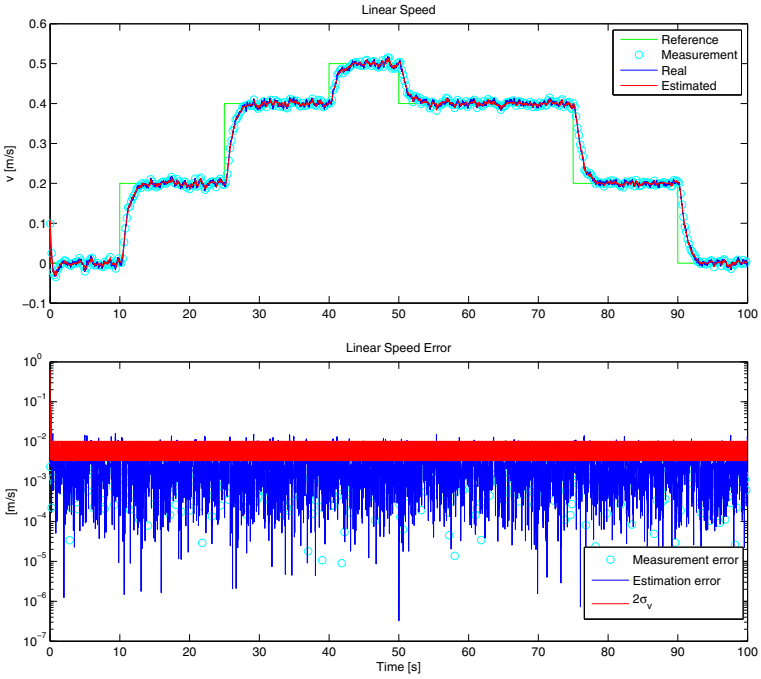


Fig. 4. Comparison of estimated and actual linear speeds of the robot. Measuring and estimation error compared with the standard deviation of the estimation.

This standard deviation follows a periodic pattern. With each encoder measurement it falls to a certain value, and then it starts growing until the next measurement is available. For the pose estimation, a periodic sampling strategy was tried, using a sample time of 80 ms. The trajectory followed by the robot is shown in Fig. 6 as well as the estimated position. It is also shown the cameras positions and their Field-Of-View (FOV). The measurements taken by the cameras are clearly influenced by the distance between the point and the camera.

The initial position of the robot was the point (1, 2) but this initial state is unknown to the estimator. The initial estimation is the point (0, 0) with an initial standard deviation of 3 metres on each axis.

At the beginning of the simulation the estimator quickly corrects the vague initial position and tracks the position pretty well, even though the measurements have a considerable noise level. As can be seen in Fig. 7 the estimation error on x and y is reduced down to a few centimetres.

Like in Fig. 4, this error is upper bounded by twice the standard deviation of the state estimation. However, in this plots the variance does not reach a steady state and varies along the path because these states are not linear.

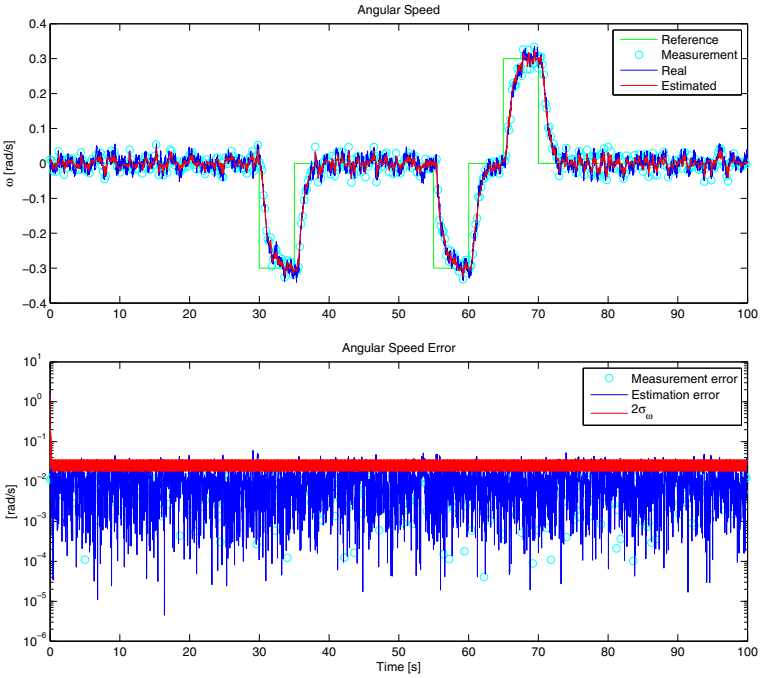


Fig. 5. Comparison of estimated and actual angular speeds of the robot. Measuring and estimation error compared with the standard deviation of the estimation.

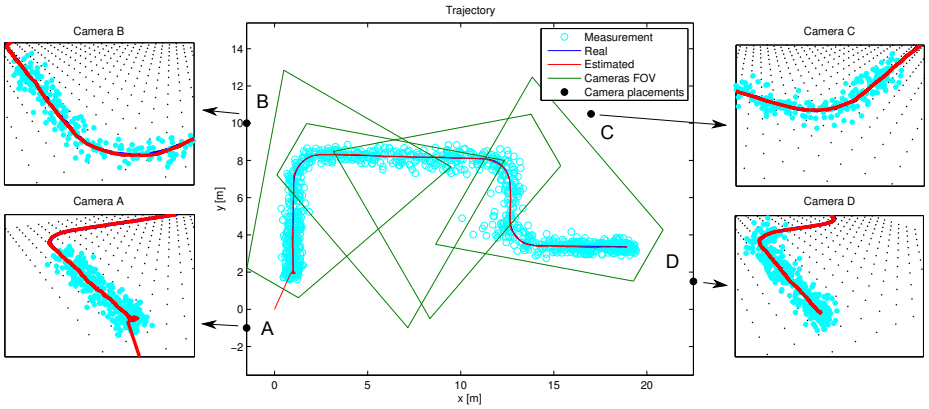


Fig. 6. Trajectory developed by the robot and estimated position with periodic sampling

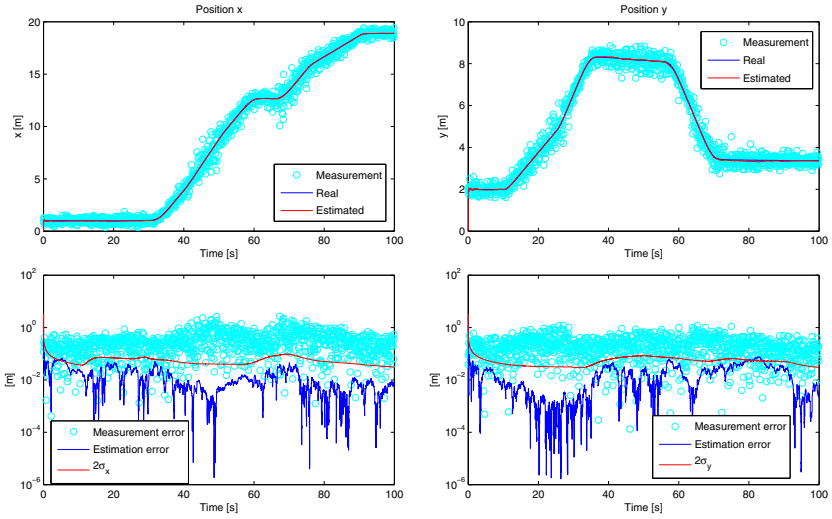


Fig. 7. Estimated position and error of estimation in the x and y axes with periodic sampling of pose

The orientation estimation results have been left out because they do not provide additional understanding to the authors' proposal.

In order to achieve this estimation performance, a total of 1250 measurements were taken from the cameras. Gathering and processing this amount of data would mean a big load on computation and network communications.

4.2 Case B. Adaptive Sampling

Figure 7 shows the best possible estimation with the given measuring system. In these conditions, the maximum value of $2\sigma_{\hat{x}}$ is 95 mm. Most of the time, the estimation error covariance is much better than that, but if a maximum error of 100 mm on both x and y is considered acceptable, then some measurements are not needed and the sensors can notably reduce their working load.

This estimation performance would be accomplished if $2\sigma_{\hat{x}}$ and $2\sigma_{\hat{y}}$ are kept lower than 100 mm. A threshold value of 0.05 m s is assigned to the standard deviation of the position states, to trigger the captures of the camera as explained in (10) (see Sect. 2.2). This way the number of measurements can be drastically reduced to 383 samples. Figure 8 shows the estimated trajectory on this new simulation.

With the adaptive measuring strategy the estimation gets slightly worse but it still accomplishes the the maximum accepted error. Figure 9 shows the estimation errors, the standard deviation and also the thresholds for position states

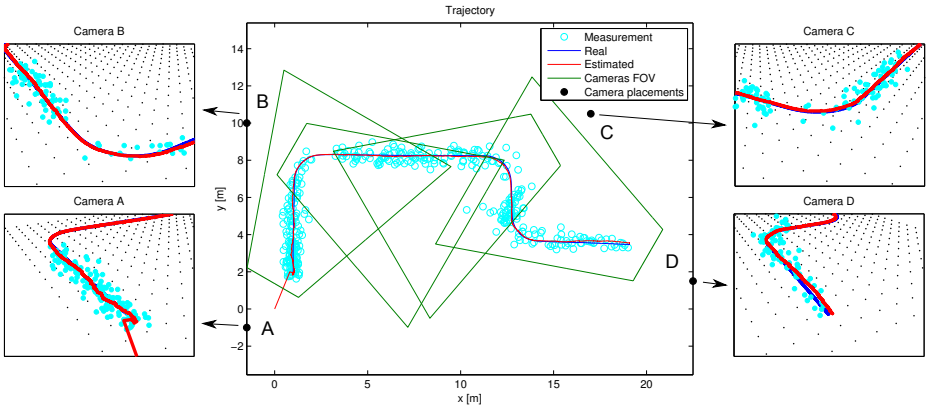


Fig. 8. Trajectory developed by the robot and estimated position with adaptive sampling.

x and y. Every time the uncertainty in x or y approaches the threshold, a measurement is produced and the uncertainty is kept within the limits.

Sometimes the uncertainty of x triggers the capture of a new measurement while the uncertainty of y is far from the threshold. Sometimes it is the other way round. The time between camera measurements is not constant and depends on the rate the covariance grows, which in turn depends on the working point because the plant model is non-linear.

For example, at the beginning of the simulation some measurements are needed to locate the robot for the first time. Once the position is known with enough precision, no further measurement is needed because the robot is not moving. When it starts moving at time $t = 10\,\mathrm{s}$, the measurements are triggered again.

Table 1 shows the number of measurements taken and the Root Mean Squared Error (RMSE) of estimation as a distance to the actual point for several simulations, for different theshold values. The results shown are computed beginning on second 2 of the simulation, when the filter has converged and the initial uncertainty has disappeared. By allowing more estimation error, the RMSE is worsen but in turn the number of samples is extremely reduced.

Table 1. Comparison of tracking errors for different threshold values

	Number of measurements	RMS distance error [mm]
Periodic	1250	37.8
$\sigma_{\mathrm{thr}x,y} = 50\,\mathrm{mm}$	383	61.2
$\sigma_{\mathrm{thr}x,y} = 70\,\mathrm{mm}$	166	84.8
$\sigma_{\mathrm{thr}x,y} = 100\,\mathrm{mm}$	72	118.7

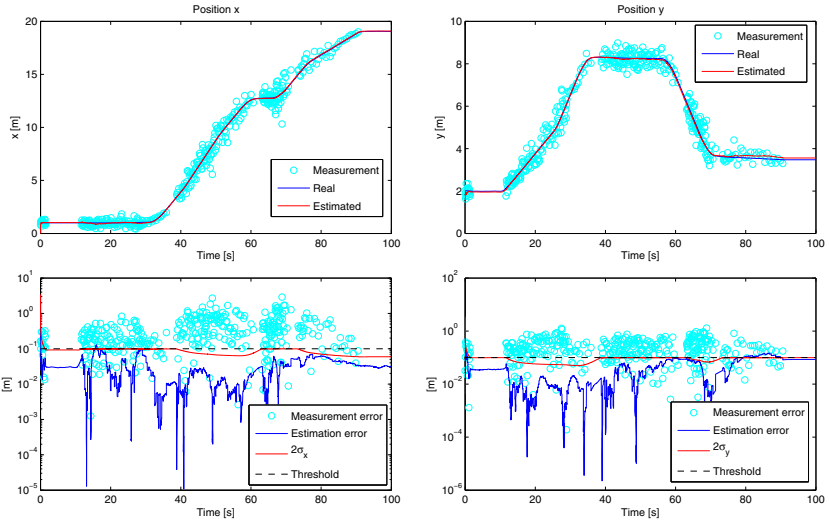


Fig. 9. Estimated position and error of estimation in the x and y axes with adaptive sampling based on thresholds

5 Conclusion

When the computational load needed to process sensor data or the communication network load to transmit them are an important concern in a NCS, a measuring strategy is needed to allow network sensors to work in an efficient way.

The use of sensor information for state estimation can be reduced in a way where the desired estimation accuracy is still achieved. The proposed adaptive sampling estimation takes the minimum amount of data from the sensors that keep the diagonal values of the estimation covariance matrix below some bounds.

This proposal is specially interesting in the case of non-linear systems and/or time-variant systems. The following sampling instant can even be computed in advance, which allows to keep the sensor switched off in the meantime. A linear time-invariant system would end up with periodic sampling, so the only problem would be to compute the sampling time interval.

A clear example of application is the pose estimation of robotic units using external information of a camera network. As a future work, the authors are working on a real implementation of the proposal in order to validate experimentally the results obtained by simulation.

References

1. Mallick, M., Coraluppi, S., Carthel, C.: Advances in asynchronous and decentralized estimation. In: IEEE Proceedings of the Aerospace Conference, vol. 4, pp. 4/1873–4/1888 (2001)

2. Hespanha, J., Naghshtabrizi, P., Xu, Y.: A survey of recent results in networked control systems. Proceedings of the IEEE 95(1), 138–162 (2007)
3. Tabuada, P.: Event-triggered real-time scheduling of stabilizing control tasks. IEEE Transactions on Automatic Control 52(9), 1680–1685 (2007)
4. Mazo Jr., M., Anta, A., Tabuada, P.: On self-triggered control for linear systems: Guarantees and complexity. In: European Control Conference (2009)
5. Battistelli, G., Benavoli, A., Chisci, L.: Data-driven communication for state estimation with sensor networks. Automatica 48(5), 926–935 (2012)
6. Sánchez, J., Guarnes, M.A., Dormido, S.: On the application of different event-based sampling strategies to the control of a simple industrial process. Sensors 9(9), 6795–6818 (2009)
7. Sijs, J., Lazar, M.: Event based state estimation with time synchronous updates. IEEE Transactions on Automatic Control 57(10), 2650–2655 (2012)
8. Miskowicz, M.: Send-on-delta concept: An event-based data reporting strategy. Sensors 6(1), 49–63 (2006)
9. Kamen, E., Su, J.: Introduction to Optimal Estimation. Advanced Textbooks in Control and Signal Processing. Springer, London (1999)
10. Simon, D.: Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches. Wiley (2006)
11. Duton, K., Thompson, S., Barraclough, B.: The art of control engineering. Addison-Wesley (1997)
12. Julier, S., Uhlmann, J.: Unscented filtering and nonlinear estimation. Proceedings of the IEEE 92(3), 401–422 (2004)
13. Santos, C., Mazo Jr., M., Espinosa, F.: Adaptive self-triggered control of a remotely operated robot. In: Herrmann, G., Studley, M., Pearson, M., Conn, A., Melhuish, C., Witkowski, M., Kim, J.-H., Vadakkepat, P. (eds.) TAROS-FIRA 2012. LNCS, vol. 7429, pp. 61–72. Springer, Heidelberg (2012)
14. Levine, W.: The control handbook. IEEE-Press (1996)
15. Lee, J.H., Hashimoto, H.: Intelligent space. In: Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), vol. 2, pp. 1358–1363 (2000)
16. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)

Operational Space Consensus in Networks of Robots: The Leader-Follower Case

Carlos I. Aldana[1], Eduardo Romero[2], Emmanuel Nuño[2], and Luis Basañez[1]

[1] Institute of Industrial and Control Engineering (IOC),
Technical University of Catalonia (UPC). Barcelona 08028, Spain
{carlos.aldana,luis.basanez}@upc.edu
[2] Department of Computer Science (DCC)
University of Guadalajara (UdG). Guadalajara 44430, Mexico
eduardo.romerof@red.cucei.udg.mx, emmanuel.nuno@cucei.udg.mx

Abstract. This paper provides a solution to the leader-follower consensus problem in networks composed of multiple heterogeneous robots, modeled in the operational space. Using a simple distributed proportional controller and without using velocity measurements, the paper shows that all the robots in the network asymptotically reach a given leader pose (position and orientation) provided that, at least, one follower robot has access to the leader pose. The singularity-free unit quaternions are used to describe the orientation of the robots manipulators. The effectiveness of the theoretical results is validated through numerical simulations of a network composed of ten robots with 6-Degrees-of-Freedom (DoF).

Keywords: Position and orientation control, leader-follower consensus, Euler-Lagrange systems, Lyapunov stability.

1 Introduction

This work studies the consensus in the operational space of multiple heterogeneous robot manipulators. In particular, it deals with the solution of the leader–follower problem, where the objective is to ensure that all robot manipulators reach a consensus at a given leader pose[1]. In general, the operational space is a subspace of the Special Euclidean space of dimension three, denoted SE(3). Cooperative and collaborative tasks are performed and described in the operational space. Moreover, when multiple robot manipulators, which can be kinematically dissimilar (heterogeneous), cooperate to perform a common task it becomes evident that the solutions to the consensus problems have to be within the SE(3) [1,2]. Cooperative control has several areas of applications, some of them are: synchronization [3,4,5], flocking [6,7], coordination[8,9] and consensus[10,11,12]. The objective of the consensus is that all agents in a cooperative system can reach an agreement on their states. In the leader-follower consensus, an agent of the system specifies the the desired state for the whole system.

[1] It is considered that a pose is composed by a position term and an orientation (attitude) term.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_42,

The leader-follower consensus problem has been studied for first order and second order linear systems in [13] and [10], respectively. The remarkable works [14,15,16], for the attitude-only control, and [17], also aim at solving the consensus problem reported here. Using unit quaternions and full-state feedback controllers, [14] solves the consensus problem, provided that the leader orientation is available to all the followers. Assuming complete knowledge of the system dynamics and employing the, possibly singular, modified Rodriguez parameters, [15] presents a full-state feedback controller for the solution to the leader-follower consensus problem. The work in [16] solves the same problem by using a *virtual* system for each agent in the network without velocity measurements, provided that the agents physical parameters are all exactly known.

This paper reports a leader-follower consensus algorithm, that is model independent and distributed, for networks of robots modeled in the operational space. The control implementation does not require of the general costly and noisy velocity sensors and the singularity-free representation, unit quaternions, is used to describe the orientation of the robot manipulators. The controller ensures the asymptotic convergence of the robots' pose to a constant leader pose, provided that only a subset of followers (at least one) has a direct access to the leader constant pose. Furthermore, the linear and the angular velocities also asymptotically converge to zero. Simulations, using a network with ten robots, evidence the performance of the proposed scheme.

2 Preliminaries

2.1 Notation

- $\mathbb{R} := (-\infty, \infty)$, $\mathbb{R}_{>0} := (0, \infty)$, $\mathbb{R}_{\geq 0} := [0, \infty)$.
- The spectrum of a square matrix $\mathbf{A}$ is denoted by $\sigma(\mathbf{A})$. The minimum and the maximum eigenvalues are denoted by $\sigma_{\min}(\mathbf{A})$ and $\sigma_{\max}(\mathbf{A})$, respectively.
- $|\mathbf{x}|$ stands for the standard Euclidean norm of vector $\mathbf{x}$.
- $\mathbf{I}_k$ and $\mathbf{\emptyset}_k$ represent the identity and all-zeros matrices of size k. Similarly, $\mathbf{1}_k$ and $\mathbf{0}_k$ represent the column vectors of size k with all entries equal to one and to zero, respectively.
- For a given matrix $\mathbf{A} \in \mathbb{R}^{a \times b}$, where $b \geq a$, $\mathbf{A}^\dagger$ is its pseudo-inverse and it is given by $\mathbf{A}^\dagger := \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1}$.
- For any function $\mathbf{f} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, the $\mathcal{L}_\infty$-norm is defined as $\|\mathbf{f}\|_\infty := \sup_{t \geq 0} |\mathbf{f}(t)|$, $\mathcal{L}_2$-norm as $\|\mathbf{f}\|_2 := (\int_0^\infty |\mathbf{f}(t)|^2 dt)^{1/2}$. The $\mathcal{L}_\infty$ and $\mathcal{L}_2$ spaces are defined as the sets $\{\mathbf{f} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n : \|\mathbf{f}\|_\infty < \infty\}$ and $\{\mathbf{f} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n : \|\mathbf{f}\|_2 < \infty\}$, respectively.

2.2 Unit Quaternions

A unit quaternion[2] $\boldsymbol{\xi}_i \in S^3$ can be split in two elements: one scalar term $\eta_i \in \mathbb{R}$ and one vectorial term $\boldsymbol{\beta}_i \in \mathbb{R}^3$. Thus $\boldsymbol{\xi}_i := (\eta_i, \boldsymbol{\beta}_i^\top)^\top$ and, from the unit norm

[2] The set $S^3 \subset \mathbb{R}^4$ represents an unitary sphere of dimension three and it is defined as $S^3 := \{\boldsymbol{\xi} \in \mathbb{R}^4 : |\boldsymbol{\xi}|^2 = 1\}$.

constraint, $\eta_i^2 + \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i = 1$. The unit quaternion $\boldsymbol{\xi}_i$ can be easily obtained from the direct kinematics function of each robot manipulator, via the rotation matrix $\mathbf{R}_i \in SO(3) := \{\mathbf{R}_i \in \mathbb{R}^{3\times 3} : \mathbf{R}_i^\top \mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) = 1\}$ [18]. Compared to other orientation representations, e.g., yaw-pitch-roll parameters, Euler angles, etc., the unit quaternions are known to be free of singularities [19]. Refer to [20,21] for a detailed list of properties and operations involving unit quaternions.

The orientation error, relative to the world frame, between two different frames, $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_j$, can be described by the rotation matrix $\tilde{\mathbf{R}}_{ij} := \mathbf{R}_i \mathbf{R}_j^\top \in SO(3)$. The unit quaternion describing such orientation error is given by

$$\tilde{\boldsymbol{\xi}}_{ij} = \boldsymbol{\xi}_i \odot \boldsymbol{\xi}_j^* = \begin{bmatrix} \tilde{\eta}_{ij} \\ \tilde{\boldsymbol{\beta}}_{ij} \end{bmatrix} = \begin{bmatrix} \eta_i \eta_j + \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_j \\ \eta_j \boldsymbol{\beta}_i - \eta_i \boldsymbol{\beta}_j - \mathbf{S}(\boldsymbol{\beta}_i)\boldsymbol{\beta}_j \end{bmatrix} \tag{1}$$

where $\odot$ denotes the quaternion product, $\boldsymbol{\xi}_{(\cdot)}^* = (\eta_{(\cdot)}, -\boldsymbol{\beta}_{(\cdot)}^\top)^\top$ is the quaternion conjugate and $\mathbf{S}(\cdot)$ is the skew-symmetric matrix operator[3] such that, for any $\mathbf{a} \in \mathbb{R}^3$,

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

The relation between the time-derivative of the unit quaternion and the angular velocity, relative to the world reference frame, is given by

$$\dot{\boldsymbol{\xi}}_i = \frac{1}{2} \mathbf{U}(\boldsymbol{\xi}_i) \boldsymbol{\omega}_i, \tag{2}$$

where

$$\mathbf{U}(\boldsymbol{\xi}_i) := \begin{bmatrix} -\boldsymbol{\beta}_i^\top \\ \eta_i \mathbf{I}_3 - \mathbf{S}(\boldsymbol{\beta}_i) \end{bmatrix}. \tag{3}$$

Using the normality condition and the skew-symmetric matrix properties it holds that $\mathbf{U}^\top(\boldsymbol{\xi}_i)\mathbf{U}(\boldsymbol{\xi}_i) = \mathbf{I}_3$, hence $\text{rank}(\mathbf{U}(\boldsymbol{\xi}_i)) = 3$ and $\ker(\mathbf{U}(\boldsymbol{\xi}_i))$=span$(\boldsymbol{\xi}_i)$.

Finally, for $\tilde{\boldsymbol{\omega}}_{ij} = \boldsymbol{\omega}_i - \boldsymbol{\omega}_j$, it also holds that [22,23,24]

$$\dot{\tilde{\boldsymbol{\xi}}}_{ij} = \frac{1}{2} \begin{bmatrix} -\tilde{\boldsymbol{\beta}}_{ij}^\top \\ \tilde{\eta}_{ij} \mathbf{I}_3 + \mathbf{S}(\tilde{\boldsymbol{\beta}}_{ij}) \end{bmatrix} \tilde{\boldsymbol{\omega}}_{ij} - \begin{bmatrix} 0 \\ \mathbf{S}(\tilde{\boldsymbol{\beta}}_{ij}) \end{bmatrix} \boldsymbol{\omega}_i. \tag{4}$$

3 Network Model

Figure 1 shows an example of a network with three followers, the leader and their respective reference frames. In this work is assumed that each follower node contains a fully-actuated, revolute joints, robot manipulator and that the interconnection of the network can be analyzed using graph theory, besides the information exchange is assumed without time-delays.

[3] The following, are some interesting properties of the skew-symmetric matrix operator $\mathbf{S}(\cdot)$ that will be used throughout the paper: For any $\mathbf{a} \in \mathbb{R}^3$, it holds that $\mathbf{S}(\mathbf{a})^\top = \mathbf{S}(-\mathbf{a}) = -\mathbf{S}(\mathbf{a})$ and $\mathbf{S}(\mathbf{a})\mathbf{a} = \mathbf{0}_3$.

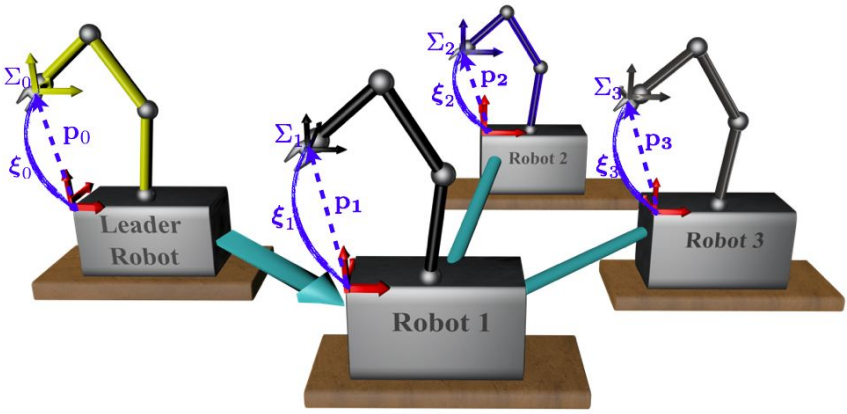


Fig. 1. Leader-follower network

3.1 Network Interconnection

The interconnection of the N robot manipulators is modeled using the Laplacian matrix $\mathbf{L} := [\ell_{ij}] \in \mathbb{R}^{N\times N}$, whose elements are defined as

$$\ell_{ij} = \begin{cases} \sum\limits_{j\in\mathcal{N}_i} a_{ij} & i = j \\ -a_{ij} & i \neq j \end{cases} \tag{5}$$

where a_{ij} are the interconnection weights defined as $a_{ij} > 0$ if $j \in \mathcal{N}_i$ and $a_{ij} = 0$ otherwise. $\mathcal{N}_i$ is the set of robot manipulators transmitting information to the ith robot, recall that $i \in \bar{N}$ and $\bar{N} := \{1, \ldots, N\}$.

In order to ensure that the interconnection forces are generated by a gradient of a potential function, the following assumption is used in this paper:

(**A1**) The robot interconnection graph is *static, undirected and connected.*

Note that, by construction, $\mathbf{L}$ has a zero row sum, i.e., $\mathbf{L}\mathbf{1}_N = \mathbf{0}_N$. Moreover, Assumption (**A1**), ensures that $\mathbf{L}$ is symmetric, has a single zero-eigenvalue and that the rest of the spectrum of $\mathbf{L}$ has positive real parts. Thus, $\text{rank}(\mathbf{L}) = N-1$. Using these facts, it is straightforward to show that, for any $\mathbf{z} \in \mathbb{R}^N$,

$$\mathbf{z}^\top \mathbf{L}\mathbf{z} = \frac{1}{2}\sum_{i\in\bar{N}}\sum_{j\in\mathcal{N}_i} a_{ij}(z_i - z_j)^2 \geq 0$$

Furthermore,

$$\frac{d}{dt}\mathbf{z}^\top\mathbf{L}\mathbf{z} = 2\dot{\mathbf{z}}^\top\mathbf{L}\mathbf{z} = 2\sum_{i\in\bar{N}}\dot{z}_i\sum_{j\in\mathcal{N}_i} a_{ij}(z_i - z_j). \tag{6}$$

Fact 1. *Consider a diagonal matrix* $\mathbf{A}_0 := diag(a_{10}, \ldots, a_{N0}) \in \mathbb{R}^{N\times N}$ *with all* $a_{i0} \geq 0$ *and, at least, one* $a_{i0} > 0$. *Assume that* (**A1**) *holds, then the matrix* $\mathbf{L}_\ell := \mathbf{L} + \mathbf{A}_0$ *is symmetric, positive definite and of full rank.* ◇

Proof. Assumption (**A1**), ensures that $\sigma_{\min}(\mathbf{L}) = 0$ is a single eigenvalue and the rest of the spectrum is strictly positive. Moreover, $\mathbf{L}$ is symmetric and since $\mathbf{A}_0$ is diagonal, then $\mathbf{L}_0$ is symmetric. Hence, $\sigma(\mathbf{L}_0)$ has only real numbers. Inspired by [25], let us now invoke the Gershgorin Disk Theorem to locate the spectrum of $\mathbf{L}_0$. There exist N disks, denoted $\mathcal{D}_i$, centered in the complex plane at $\ell_{ii}+a_{i0}$ with radius $\sum_{j\in\mathcal{N}_i} a_{ij}$, such that

$$\mathcal{D}_i := \left\{ \lambda : |\lambda - \ell_{ii} - a_{i0}| \leq \sum_{j\in\mathcal{N}_i} a_{ij} \right\},$$

since $\ell_{ii} = \sum_{j\in\mathcal{N}_i} a_{ij}$ then $\lambda \geq 0$ and $\mathcal{D}_i$ may tangentially intersect the origin. Moreover, from Assumption (**A2**), there exists at least one $a_{i0} > 0$, thus not all the disks intersect the origin. Further, since the undirected graph is connected then $\mathbf{L}$ is irreducible and, by Taussky's Theorem, if 0 is an eigenvalue of $\mathbf{L}_0$ then it has to lie in the boundary of *all* the disks [26], we conclude that $0 \notin \sigma(\mathbf{L}_0)$, thus $\sigma(\mathbf{L}_0)$ is strictly positive and real. Which, together with symmetry, ensures that $\mathbf{L}_0$ is positive definite. The full rank property follows directly.

3.2 Dynamics of the Nodes

The ith-node is modeled as n_i-DoF robot manipulator. Its EL-equation of motion, in joint space, is given by

$$\bar{\mathbf{M}}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \bar{\mathbf{C}}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + \bar{\mathbf{g}}_i(\mathbf{q}_i) = \boldsymbol{\tau}_i \tag{7}$$

where $\mathbf{q}_i, \dot{\mathbf{q}}_i, \ddot{\mathbf{q}}_i \in \mathbb{R}^{n_i}$, are the joint positions, velocities and accelerations, respectively; $\bar{\mathbf{M}}_i(\mathbf{q}_i) \in \mathbb{R}^{n_i\times n_i}$ is the symmetric and positive definite inertia matrix; $\bar{\mathbf{C}}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) \in \mathbb{R}^{n_i\times n_i}$ is the Coriolis and centrifugal effects matrix, defined via the Christoffel symbols of the fist kind; $\bar{\mathbf{g}}_i(\mathbf{q}_i) \in \mathbb{R}^{n_i}$ is the gravitational torques vector and $\boldsymbol{\tau}_i \in \mathbb{R}^{n_i}$ is the torque exerted by the actuators (motors).

The pose of the ith-end-effector, relative to a common reference frame, can be denoted by the vector $\mathbf{x_i} \subset \mathbb{R}^7$ and it is composed by the position vector $\mathbf{p}_i \in \mathbb{R}^3$ and the orientation unit quaternion $\boldsymbol{\xi}_i \in S^3$, such that $\mathbf{x}_i := (\mathbf{p}_i^\top, \boldsymbol{\xi}_i^\top)^\top$.

The relation between the joint velocities and the linear $\dot{\mathbf{p}}_i$ and angular $\boldsymbol{\omega}_i$ velocities of the ith-end-effector, expressed also relative to a common reference frame, is given by

$$\boldsymbol{\upsilon}_i = \begin{bmatrix} \dot{\mathbf{p}}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \mathbf{J}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i \tag{8}$$

where $\boldsymbol{\upsilon}_i \in \mathbb{R}^6$ and $\mathbf{J}_i(\mathbf{q}_i) \in \mathbb{R}^{6\times n_i}$ is the *geometric Jacobian* matrix. Using the principle of the virtual work, the following relation between joint torque $\boldsymbol{\tau}_i$ and the Cartesian forces $\mathbf{f}_i$ is obtained

$$\boldsymbol{\tau}_i = \mathbf{J}_i^\top(\mathbf{q}_i)\mathbf{f}_i, \tag{9}$$

where $\mathbf{f}_i \in \mathbb{R}^6$, $\mathbf{f}_i := (\mathbf{h}_i^\top, \mathbf{m}_i^\top)^\top$ and $\mathbf{h}_i, \mathbf{m}_i \in \mathbb{R}^3$ represent the Cartesian linear forces and moments, respectively. Pre-multiplying (8) by the Jacobian pseudo-inverse $\mathbf{J}_i^\dagger(\mathbf{q}_i)$ and differentiating, yields

$$\ddot{\mathbf{q}}_i = \mathbf{J}_i^\dagger(\mathbf{q}_i)\dot{\boldsymbol{\upsilon}}_i + \dot{\mathbf{J}}_i^\dagger(\mathbf{q}_i)\boldsymbol{\upsilon}_i. \tag{10}$$

Using (9), (8) and (10) allows to transform (7) to its operational space model, which is given by

$$\mathbf{M}_i(\mathbf{q}_i)\dot{\boldsymbol{\upsilon}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\boldsymbol{\upsilon}_i + \mathbf{g}_i(\mathbf{q}_i) = \mathbf{f}_i \tag{11}$$

where

$$\mathbf{M}_i(\mathbf{q}_i) := \left(\mathbf{J}_i^\dagger\right)^\top \bar{\mathbf{M}}_i(\mathbf{q}_i)\mathbf{J}_i^\dagger; \quad \mathbf{g}_i(\mathbf{q}_i) := \left(\mathbf{J}_i^\dagger\right)^\top \bar{\mathbf{g}}_i(\mathbf{q}_i);$$
$$\mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) := \left(\mathbf{J}_i^\dagger\right)^\top \left(\bar{\mathbf{M}}_i(\mathbf{q}_i)\dot{\mathbf{J}}_i^\dagger + \bar{\mathbf{C}}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\mathbf{J}_i^\dagger\right).$$

The operational space model (11) has the following well-known properties [18,27]:

- (**P1**) $\mathbf{M}_i(\mathbf{q}_i)$ is symmetric and there exist $\lambda_{m_i}, \lambda_{M_i} > 0$ such that $0 < \lambda_{m_i}\mathbf{I}_6 \leq \mathbf{M}_i(\mathbf{q}_i) \leq \lambda_{M_i}\mathbf{I}_6$ holds for all $\mathbf{q}_i \in \mathbb{R}^{n_i}$.
- (**P2**) The matrix $\dot{\mathbf{M}}_i(\mathbf{q}_i) - 2\mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ is skew-symmetric.

4 Consensus Algorithm

Consider the case when the network of N robot manipulators (followers) has to be regulated at a leader's constant pose, denoted $\mathbf{x}_0 := (\mathbf{p}_0^\top, \boldsymbol{\xi}_0^\top)^\top$. The leader pose is regarded as *node 0* of the $N+1$ agent network.

The control objective is to ensure that all robot manipulators pose asymptotically converge to the constant leader pose, i.e., for all $i \in \bar{N}$,

$$\lim_{t\to\infty} |\boldsymbol{\upsilon}_i(t)| = 0, \quad \lim_{t\to\infty} \mathbf{x}_i(t) = \mathbf{x}_0,$$

provided that the leader pose (node) is only known by a certain set of robot manipulators and that joint and operational space velocities are not available for measurements.

The proposed controller is given by

$$\left.\begin{aligned} \dot{\mathbf{y}}_i &= \mathbf{x}_i - k_{yi}\mathbf{y}_i \\ \mathbf{f}_i &= -k_i a_{i0}\mathbf{e}_{i0} - k_i \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{e}_{ij} + \mathbf{g}_i(\mathbf{q}_i) - d_i\boldsymbol{\Psi}_i(\mathbf{x}_i - k_{yi}\mathbf{y}_i) \end{aligned}\right\} \tag{12}$$

where, for any pair (i, j),

$$\mathbf{e}_{ij} = \begin{bmatrix} \tilde{\mathbf{p}}_{ij} \\ \tilde{\boldsymbol{\beta}}_{ij} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_i - \mathbf{p}_j \\ \eta_j\boldsymbol{\beta}_i - \eta_i\boldsymbol{\beta}_j - \mathbf{S}(\boldsymbol{\beta}_i)\boldsymbol{\beta}_j \end{bmatrix}, \tag{13}$$

$\mathbf{y}_i \in \mathbb{R}^7$ is the controller state, $k_i, d_i, k_{yi} \in \mathbb{R}_{>0}$ are the controller gains,

$$\boldsymbol{\Psi}_i = \begin{bmatrix} \mathbf{I}_3 & \boldsymbol{\emptyset}_{3\times 4} \\ \boldsymbol{\emptyset}_3 & \frac{1}{2}\mathbf{U}^\top(\boldsymbol{\xi}_i) \end{bmatrix}, \tag{14}$$

and $a_{i0} > 0$ if the leader's pose $\mathbf{x}_0$ is available to the ith robot manipulator and $a_{i0} = 0$, otherwise.

The closed-loop system (11) and (12) is

$$\left.\begin{aligned} \dot{\mathbf{y}}_i &= \mathbf{x}_i - k_{yi}\mathbf{y}_i \\ \dot{\boldsymbol{\upsilon}}_i &= -\mathbf{M}_i^{-1}(\mathbf{q}_i)\left[\mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\boldsymbol{\upsilon}_i + k_i \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{e}_{ij} + k_i a_{i0}\mathbf{e}_{i0} + d_i\boldsymbol{\Psi}_i(\mathbf{x}_i - k_{yi}\mathbf{y}_i)\right] \end{aligned}\right\} \tag{15}$$

With regards to the leader-followers interconnection, this paper has the following assumption,

(**A2**) At least one of the robot manipulators has access to the leader's pose $\mathbf{x}_0$, i.e., in the graph of $N+1$ nodes, being node 0 the leader node, there exists at least one directed edge from the leader to any of the N followers.

Assumptions (**A1**) and (**A2**) ensure that the leader pose (node 0 of the interconnection graph) is *globally reachable* from any ith node, i.e., from the leader's node there exist a path to any ith robot-manipulator for $i \in \bar{N}$.

The following remark shows that the closed-loop system (15) has only two possible equilibrium points.

Remark 1. Suppose that Assumptions (**A1**) and (**A2**) hold, then the equilibrium points of (15) satisfy

$$a_{i0}\mathbf{e}_{i0} + \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{e}_{ij}(t) = \mathbf{0}_6,$$

which in turn implies that

$$a_{i0}\tilde{\mathbf{p}}_{i0} + \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{p}}_{ij} = \mathbf{0}_3 \tag{16}$$

$$a_{i0}\tilde{\boldsymbol{\beta}}_{i0} + \sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\boldsymbol{\beta}}_{ij} = \mathbf{0}_3. \tag{17}$$

Defining $\mathbf{p} := (\mathbf{p}_1^\top, \ldots, \mathbf{p}_N^\top)^\top$, (16) can be written as

$$(\mathbf{A}_0 \otimes \mathbf{I}_3)(\mathbf{p} - (\mathbf{1}_N \otimes \mathbf{p}_0)) + (\mathbf{L} \otimes \mathbf{I}_3)\mathbf{p} = \mathbf{0}_{3N}$$

and thus, since $\mathbf{L}\mathbf{1}_N = \mathbf{0}_N$,

$$(\mathbf{L}_0 \otimes \mathbf{I}_3)(\mathbf{p} - (\mathbf{1}_N \otimes \mathbf{p}_0)) = \mathbf{0}_{3N},$$

where $\mathbf{A}_0$ and $\mathbf{L}_0$ are defined in Fact 1 and $\otimes$ is the standard Kronecker product. Further, Fact 1 and the Kronecker product properties ensure that $\operatorname{rank}(\mathbf{L}_0 \otimes \mathbf{I}_3) = 3N$; thus, for all $i \in \bar{N}$, $\mathbf{p}_i = \mathbf{p}_0$ is the only solution to (16).

Now, using (13) and (3) it holds that $\tilde{\boldsymbol{\beta}}_{ij} = -\mathbf{U}^\top(\boldsymbol{\xi}_i)\boldsymbol{\xi}_j = \mathbf{U}^\top(\boldsymbol{\xi}_j)\boldsymbol{\xi}_i$, and thus (17) can be written as

$$-a_{i0}\mathbf{U}^\top(\boldsymbol{\xi}_i)\boldsymbol{\xi}_0 - \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{U}^\top(\boldsymbol{\xi}_i)\boldsymbol{\xi}_j = \mathbf{0}_3.$$

Adding the term $a_{i0}\mathbf{U}^\top(\boldsymbol{\xi}_i)\boldsymbol{\xi}_i + \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{U}^\top(\boldsymbol{\xi}_i)\boldsymbol{\xi}_i$, to the previous equation, with the fact that $\mathbf{U}^\top(\boldsymbol{\xi}_i)\boldsymbol{\xi}_i = \mathbf{0}_3$, yields

$$a_{i0}\mathbf{U}^\top(\boldsymbol{\xi}_i)(\boldsymbol{\xi}_i - \boldsymbol{\xi}_0) + \sum_{j\in\mathcal{N}_i} a_{ij}\mathbf{U}^\top(\boldsymbol{\xi}_i)(\boldsymbol{\xi}_i - \boldsymbol{\xi}_j) = \mathbf{0}_3. \tag{18}$$

Defining $\boldsymbol{\xi} := (\boldsymbol{\xi}_1^\top, \ldots, \boldsymbol{\xi}_N^\top)^\top$ and $\bar{\mathbf{U}} := \operatorname{diag}(\mathbf{U}(\boldsymbol{\xi}_1), \ldots, \mathbf{U}(\boldsymbol{\xi}_N)) \in \mathbb{R}^{4N\times 3N}$, (18) can be written as

$$\bar{\mathbf{U}}^\top\left[(\mathbf{A}_0 \otimes \mathbf{I}_4)(\boldsymbol{\xi} - (\mathbf{1}_N \otimes \boldsymbol{\xi}_0)) + (\mathbf{L} \otimes \mathbf{I}_4)\boldsymbol{\xi}\right] = \mathbf{0}_{3N}$$

using $\mathbf{L}\mathbf{1}_N = \mathbf{0}_N$, the last expression is equivalent to

$$\bar{\mathbf{U}}^\top(\mathbf{L}_0 \otimes \mathbf{I}_4)(\boldsymbol{\xi} - (\mathbf{1}_N \otimes \boldsymbol{\xi}_0)) = \mathbf{0}_{3N}$$

since $(\mathbf{L}_0 \otimes \mathbf{I}_4)$ is of full rank then the trivial solution $\boldsymbol{\xi} = \mathbf{1}_N \otimes \boldsymbol{\xi}_0$ satisfies this equation. However, the fact that $\operatorname{rank}(\bar{\mathbf{U}}) = 3N$ and the properties of (3) ensure that $\boldsymbol{\xi}_i = \pm\boldsymbol{\xi}_0$, for all $i \in \bar{N}$. Even though $\boldsymbol{\xi}_i = \boldsymbol{\xi}_0$ and $\boldsymbol{\xi}_i = -\boldsymbol{\xi}_0$ represent the same orientation, in the proof of Proposition 1, it is shown that $\boldsymbol{\xi}_i = -\boldsymbol{\xi}_0$ corresponds to an unstable equilibrium point.

At this point, the main result of this paper can be stated.

Proposition 1. *Consider a network of N robot manipulators, in the operational space, whose dynamics fulfill (11) and in closed-loop with the controller (12). Suppose that Assumptions (**A1**) and (**A2**) hold. Under these conditions, the velocities and the pose error, between each robot manipulator and the leader, asymptotically converge to zero, i.e.,*

$$\lim_{t\to\infty} |\dot{\mathbf{x}}_i(t)| = \lim_{t\to\infty} |\mathbf{e}_{i0}(t)| = 0, \qquad \forall i \in \bar{N}.$$

Moreover, $\boldsymbol{\xi}_i = \boldsymbol{\xi}_0$ is a part of the asymptotically stable equilibria.

Proof. Consider the following energy-like function $\mathcal{E}_i := \mathcal{K}_i(\boldsymbol{\upsilon}_i) + a_{i0}\mathcal{U}_i(\mathbf{x}_i, \mathbf{x}_0) + \mathcal{U}_{ci}(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathcal{K}_i$ is the scaled kinetic energy of the ith robot manipulator, given by

$$\mathcal{K}_i(\boldsymbol{\upsilon}_i) = \frac{1}{2k_i}\boldsymbol{\upsilon}_i^\top \mathbf{M}_i(\mathbf{q}_i)\boldsymbol{\upsilon}_i, \tag{19}$$

$\mathcal{U}_i(\mathbf{x}_i, \mathbf{x}_0)$ is the potential energy stored in a virtual leader-follower spring, given for the following expression for $j = 0$

$$\mathcal{U}_i(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}|\tilde{\mathbf{p}}_{ij}|^2 + (1 - \tilde{\eta}_{ij})^2 + |\tilde{\boldsymbol{\beta}}_{ij}|^2 = \frac{1}{2}|\tilde{\mathbf{p}}_{ij}|^2 + 2(1 - \tilde{\eta}_{ij}) \geq 0, \tag{20}$$

where the fact that $|\tilde{\boldsymbol{\xi}}_{ij}|^2 = 1$ has been used to obtain the second equation, and $\mathcal{U}_{ci}(\mathbf{x}_i, \mathbf{y}_i)$ is the scaled ith controller potential energy defined as

$$\mathcal{U}_{ci}(\mathbf{x}_i, \mathbf{y}_i) := \frac{d_i}{2k_i}|\mathbf{x}_i - k_{yi}\mathbf{y}_i|^2. \tag{21}$$

Now, $\dot{\mathcal{E}}_i = \dot{\mathcal{K}}_i(\boldsymbol{\upsilon}_i) + a_{i0}\dot{\mathcal{U}}_i(\mathbf{x}_i, \mathbf{x}_0) + \dot{\mathcal{U}}_{ci}(\mathbf{x}_i, \mathbf{y}_i)$ and, moreover, using Property (**P2**), the fact that $\dot{\mathbf{x}}_i = \boldsymbol{\Psi}_i^\top \boldsymbol{\upsilon}_i$, relation (4) and $\dot{\mathbf{x}}_0 = \mathbf{0}_7$, $\dot{\mathcal{E}}_i$ evaluated along (15) is given by

$$\dot{\mathcal{E}}_i = -\frac{d_i k_{yi}}{k_i}|\mathbf{x}_i - k_{yi}\mathbf{y}_i|^2 - \sum_{j\in\mathcal{N}_i} a_{ij}\boldsymbol{\upsilon}_i^\top \mathbf{e}_{ij}.$$

Consider now the total energy-like function given by

$$\mathcal{E} := \sum_{i\in\bar{N}} \left[\mathcal{E}_i + \frac{1}{2}\sum_{j\in\mathcal{N}_i} a_{ij}\mathcal{U}_i(\mathbf{x}_i, \mathbf{x}_j)\right]$$

where the term $\mathcal{U}_i(\mathbf{x}_i, \mathbf{x}_j)$, defined in (20), is the potential energy stored in the virtual spring of every connected pair of robot manipulators (i, j). $\dot{\mathcal{E}}$ is given by

$$\begin{aligned}\dot{\mathcal{E}} &= \sum_{i\in\bar{N}} \left[\dot{\mathcal{E}}_i + \frac{1}{2}\sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\boldsymbol{\upsilon}}_{ij}^\top \mathbf{e}_{ij}\right] \\ &= \sum_{i\in\bar{N}} \left[\sum_{j\in\mathcal{N}_i} a_{ij}\left(\frac{1}{2}\tilde{\boldsymbol{\upsilon}}_{ij}^\top \mathbf{e}_{ij} - \boldsymbol{\upsilon}_i^\top \mathbf{e}_{ij}\right) - \frac{d_i k_{yi}}{k_i}|\mathbf{x}_i - k_{yi}\mathbf{y}_i|^2\right]\end{aligned}$$

According to Lemma 1 (see the Appendix) the term inside the parenthesis in the above expression is equal to zero and thus

$$\dot{\mathcal{E}} = -\sum_{i\in\bar{N}} \frac{d_i k_{yi}}{k_i}|\mathbf{x}_i - k_{yi}\mathbf{y}_i|^2 \leq 0.$$

On one hand, from (15), $\mathbf{x}_i - k_{yi}\mathbf{y}_i = \mathbf{0}_7$ implies that $\dot{\mathbf{y}}_i = \mathbf{0}_7$ and hence $\ddot{\mathbf{y}}_i = \mathbf{0}_7$. Noting that $\ddot{\mathbf{y}}_i = \dot{\mathbf{x}}_i - k_{yi}\dot{\mathbf{y}}_i$ it also holds that $\dot{\mathbf{x}}_i = \mathbf{0}_7$. Now, since $\mathbf{U}_i(\boldsymbol{\xi}_i)$ is a full column rank matrix and using (2), $\dot{\mathbf{x}}_i = \mathbf{0}_7$ implies that $\boldsymbol{\upsilon}_i = \mathbf{0}_6$ and hence $\dot{\boldsymbol{\upsilon}}_i = \mathbf{0}_6$.

On the other hand, $\mathcal{E}$ is positive definite and radially unbounded with regards to $\boldsymbol{\upsilon}_i$, $\tilde{\mathbf{p}}_{i0}$, $\tilde{\mathbf{p}}_{ij}$ and $\mathbf{x}_i - k_{yi}\mathbf{y}_i$. Moreover, for $\tilde{\eta}_{ij} \in [1, -1]$, $1 - \tilde{\eta}_{ij} \geq 0$ and it vanishes only at $\tilde{\eta}_{ij} = 1$. Furthermore, $\mathcal{E} \geq 0$ and $\dot{\mathcal{E}} \leq 0$ ensures that $\mathcal{E}(t) \leq \mathcal{E}(0)$ for all t. Recalling Remark 1, which ensures that there are two possible equilibrium points of the closed-loop system (15), namely

$$(\boldsymbol{\upsilon}_i, \mathbf{p}_i, \boldsymbol{\xi}_i, \mathbf{x}_i - k_{yi}\mathbf{y}_i) = (\mathbf{0}_6, \mathbf{p}_0, \boldsymbol{\xi}_0, \mathbf{0}_7) \tag{22}$$

and

$$(\boldsymbol{\upsilon}_i, \mathbf{p}_i, \boldsymbol{\xi}_i, \mathbf{x}_i - k_{yi}\mathbf{y}_i) = (\mathbf{0}_6, \mathbf{p}_0, -\boldsymbol{\xi}_0, \mathbf{0}_7), \tag{23}$$

and using the fact that $\boldsymbol{\xi}_i = \boldsymbol{\xi}_0$ implies that $\tilde{\eta}_{i0} = 1$ and $\boldsymbol{\xi}_i = -\boldsymbol{\xi}_0$ implies that $\tilde{\eta}_{i0} = -1$, it is proved that $\tilde{\eta}_{i0} = -1$ corresponds to a maximum energy point (see (20)). Since $\mathcal{E}$ is a decreasing function, any perturbation from $\tilde{\eta}_{i0} = -1$ will drive the system to $\tilde{\eta}_{i0} = 1$, which corresponds to a minimum energy point. Hence, the equilibrium point (23) is unstable.

This last and the fact that $\mathbf{x}_i - k_{yi}\mathbf{y}_i = \mathbf{0}_7$ is inside the largest invariant set at which $\dot{\mathcal{E}}$ vanishes, LaSalle's Invariant Theorem ensures that (22) is asymptotically stable for all the state space except at the unstable equilibrium point (23). This completes the proof.

5 Simulations

This section presents some numeric simulations to validate the theoretical results derived in the previous sections, a network of ten robots manipulators of 6-DoF is used for this purpose. The interconnection graph is shown in Fig. 2 and the weights have been set such as the resulting Laplacian matrix is the following

$$\mathbf{L} = \begin{bmatrix} 1.1 & 0 & 0 & -0.4 & -0.4 & 0 & 0 & -0.3 & 0 & 0 \\ 0 & 2.4 & -0.8 & 0 & -0.8 & 0 & 0 & 0 & -0.8 & 0 \\ 0 & -0.8 & 2.1 & -0.65 & 0 & 0 & -0.65 & 0 & 0 & 0 \\ -0.4 & 0 & -0.65 & 1.85 & 0 & 0 & 0 & 0 & 0 & -0.8 \\ -0.4 & -0.8 & 0 & 0 & 1.7 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.5 & 1.65 & -0.55 & 0 & 0 & -0.6 \\ 0 & 0 & -0.65 & 0 & 0 & -0.55 & 2.05 & -0.85 & 0 & 0 \\ -0.3 & 0 & 0 & 0 & 0 & 0 & -0.85 & 1.7 & -0.55 & 0 \\ 0 & -0.8 & 0 & 0 & 0 & 0 & 0 & -0.55 & 1.9 & -0.55 \\ 0 & 0 & 0 & -0.8 & 0 & -0.6 & 0 & 0 & -0.55 & 1.95 \end{bmatrix} \tag{24}$$

The simulations have been carried out using Matlab version 7.14, Simulink version 7.9 and the Robotics Toolbox version 9.6.0 [28]. The algorithm proposed

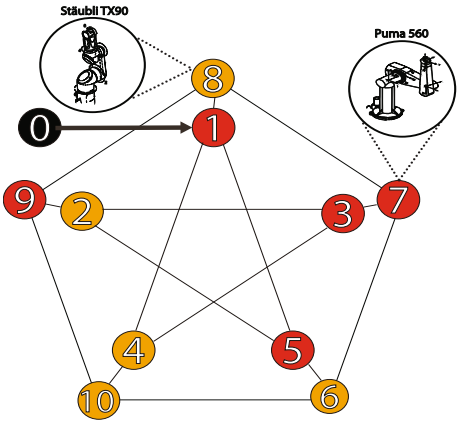
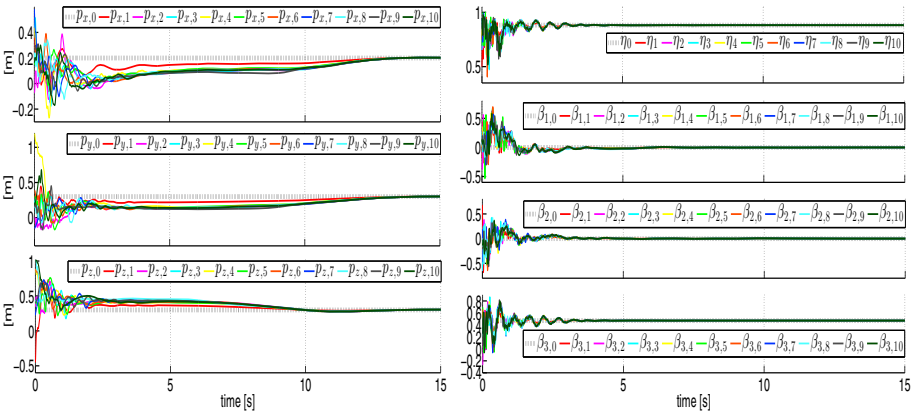
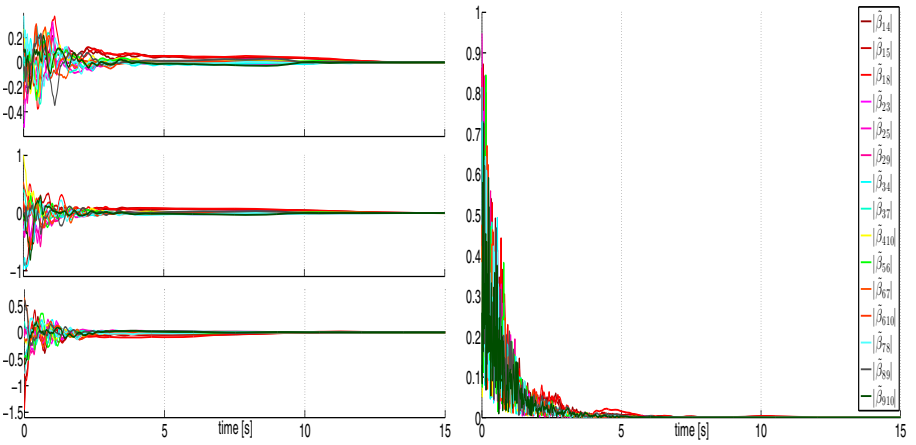


Fig. 2. Graph of the simulated network



(a) Positions (b) Orientations

Fig. 3. Pose behaviors of the robot manipulators

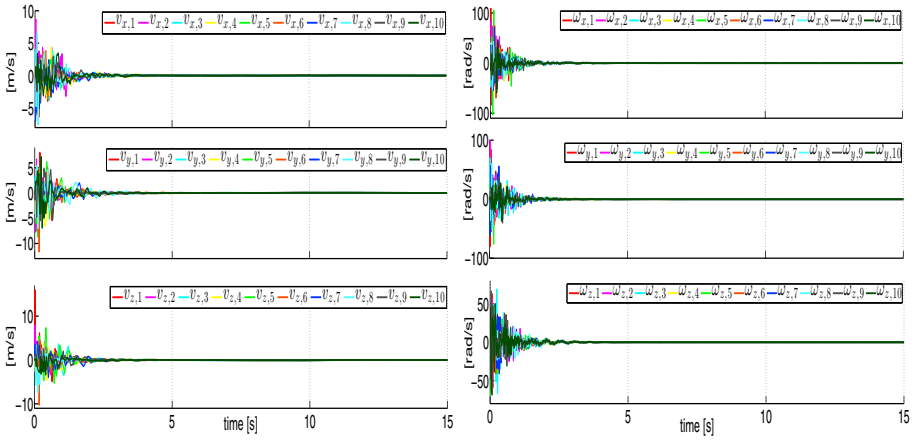


(a) Positions errors (b) Orientations errors

Fig. 4. Pose errors of the robot manipulators

in [29] has been used to obtain the unit quaternions from the rotation matrices. The dynamic parameters of the PUMA 560 robots can be found in the toolbox documentation and those of the Stäubli TX90 can be found in [30].

In the simulated case a robot leader (node zero in Fig. 2) sends a constant pose only to the robot represented by the node 1 of the graph and the initial conditions of the followers robots are set to different poses. The Fig. 3 shows the different pose initial conditions of the followers robot and how they converge to the constant leader pose (black dashed line). In the Fig. 4 and Fig. 5 are depicted the convergence to zero of the pose errors and velocities, respectively. The controller gains employed in these simulations are $k_i = 600$, $d_i = 50$, and $k_{yi} = 350$.



(a) Linear velocities (b) Angular velocities

Fig. 5. Linear and angular velocities of the robot manipulators

6 Conclusions

A simple proportional controller that does not requires velocity measurements is proposed to solve the leader-follower problem for networks of robots modeled in the operational space. It is theoretically proved that the followers pose asymptotically converges to a constant given leader pose with the sole condition that the leader has to be connected to a subset of the followers network. The unit quaternions are used to describe the robots orientation due to its well-known quality of being a singularity free orientation representation. Simulation results, using a network composed of ten robots with 6-DoF, are presented to validate the proposed consensus algorithm.

Acknowledgements. This work has been partially supported by the Mexican CONACyT grants 168998 and 261475, the Mexican CONACyT project CB-129079 and the Spanish CICYT projects: DPI2010-15446 and DPI2011-22471.

Appendix A

Lemma 1 *Consider a network satisfying Assumption (***A1***). Then,*

$$\sum_{i\in\bar{N}}\sum_{j\in\mathcal{N}_i} a_{ij}\left(\frac{1}{2}\tilde{\boldsymbol{\upsilon}}_{ij}^{\top}\mathbf{e}_{ij} - \boldsymbol{\upsilon}_i^{\top}\mathbf{e}_{ij}\right) = 0 \tag{25}$$

Proof. Let us split the linear and the angular elements, of the term inside the parenthesis, as

$$\frac{1}{2}\tilde{\boldsymbol{\upsilon}}_{ij}^{\top}\mathbf{e}_{ij} - \boldsymbol{\upsilon}_i^{\top}\mathbf{e}_{ij} = \frac{1}{2}\dot{\tilde{\mathbf{p}}}_{ij}^{\top}\tilde{\mathbf{p}}_{ij} - \dot{\mathbf{p}}_i^{\top}\tilde{\mathbf{p}}_{ij} + \frac{1}{2}\tilde{\boldsymbol{\omega}}_{ij}^{\top}\tilde{\boldsymbol{\beta}}_{ij} - \boldsymbol{\omega}_i^{\top}\tilde{\boldsymbol{\beta}}_{ij}.$$

On one hand, using the error definitions in (13) and doing some algebra, it is shown that

$$\frac{1}{2}\dot{\tilde{\mathbf{p}}}_{ij}^{\top}\tilde{\mathbf{p}}_{ij} - \dot{\mathbf{p}}_i^{\top}\tilde{\mathbf{p}}_{ij} = -\frac{1}{2}(\dot{\mathbf{p}}_i + \dot{\mathbf{p}}_j)^{\top}\tilde{\mathbf{p}}_{ij} = \frac{1}{2}(\mathbf{p}_i^{\top}\dot{\tilde{\mathbf{p}}}_{ij} - \dot{\mathbf{p}}_i^{\top}\tilde{\mathbf{p}}_{ij} - \mathbf{p}_i^{\top}\dot{\mathbf{p}}_i + \mathbf{p}_j^{\top}\dot{\mathbf{p}}_j)$$
$$= \frac{1}{2}(\mathbf{p}_i^{\top}\dot{\tilde{\mathbf{p}}}_{ij} - \dot{\mathbf{p}}_i^{\top}\tilde{\mathbf{p}}_{ij} - \rho_i + \rho_j) \tag{26}$$

where, to obtain the second equation, we added and subtracted the term $\frac{1}{2}\mathbf{p}_i^{\top}\dot{\mathbf{p}}_i$ to the first equation, in the third equation, the scalar $\rho_{(\cdot)}$ is defined as $\rho_{(\cdot)} := \mathbf{p}_{(\cdot)}^{\top}\dot{\mathbf{p}}_{(\cdot)}$. On the other hand,

$$\frac{1}{2}\tilde{\boldsymbol{\omega}}_{ij}^{\top}\tilde{\boldsymbol{\beta}}_{ij} - \boldsymbol{\omega}_i^{\top}\tilde{\boldsymbol{\beta}}_{ij} = -\frac{1}{2}(\boldsymbol{\omega}_i + \boldsymbol{\omega}_i)^{\top}\tilde{\boldsymbol{\beta}}_{ij} = \frac{1}{2}\boldsymbol{\omega}_i^{\top}\mathbf{U}^{\top}(\boldsymbol{\xi}_i)\boldsymbol{\xi}_j - \frac{1}{2}\boldsymbol{\omega}_j^{\top}\mathbf{U}^{\top}(\boldsymbol{\xi}_j)\boldsymbol{\xi}_i$$
$$= \boldsymbol{\xi}_i^{\top}(\dot{\boldsymbol{\xi}}_i - \dot{\boldsymbol{\xi}}_j) - \dot{\boldsymbol{\xi}}_i^{\top}(\boldsymbol{\xi}_i - \boldsymbol{\xi}_j) \tag{27}$$

where we have used, to obtain the second equation, the property that $\tilde{\boldsymbol{\beta}}_{ij} = -\mathbf{U}^{\top}(\boldsymbol{\xi}_i)\boldsymbol{\xi}_j = \mathbf{U}^{\top}(\boldsymbol{\xi}_j)\boldsymbol{\xi}_i$ and, for the third equation, the relation given in (2) plus adding and subtracting the term $\boldsymbol{\xi}_i^{\top}\dot{\boldsymbol{\xi}}_i$.

Now, with the Laplacian matrix defined in (5) and since $\mathbf{1}_N^{\top}\mathbf{L} = \mathbf{0}_N$, it holds that

$$-\frac{1}{2}\sum_{i\in\bar{N}}\sum_{j\in\mathcal{N}_i} a_{ij}(\rho_i - \rho_j) = -\frac{1}{2}\mathbf{1}_N\mathbf{L}\boldsymbol{\rho} = 0 \tag{28}$$

where $\boldsymbol{\rho} := (\rho_1, \ldots, \rho_N)^{\top} \in \mathbb{R}^N$. Using (26), (27) and (28) yields

$$\sum_{j\in\mathcal{N}_i} a_{ij}\left(\frac{1}{2}\tilde{\boldsymbol{\upsilon}}_{ij}^{\top}\mathbf{e}_{ij} - \boldsymbol{\upsilon}_i^{\top}\mathbf{e}_{ij}\right) = \frac{1}{2}\,\mathbf{p}_i^{\top}\sum_{j\in\mathcal{N}_i} a_{ij}\dot{\tilde{\mathbf{p}}}_{ij} - \frac{1}{2}\dot{\mathbf{p}}_i^{\top}\sum_{j\in\mathcal{N}_i} a_{ij}\tilde{\mathbf{p}}_{ij}$$
$$+ \boldsymbol{\xi}_i^{\top}\sum_{j\in\mathcal{N}_i} a_{ij}(\dot{\boldsymbol{\xi}}_i - \dot{\boldsymbol{\xi}}_j) - \dot{\boldsymbol{\xi}}_i^{\top}\sum_{j\in\mathcal{N}_i} a_{ij}(\boldsymbol{\xi}_i - \boldsymbol{\xi}_j)$$

Further, defining $\mathbf{p} := (\mathbf{p}_1^{\top}, \ldots, \mathbf{p}_N^{\top})^{\top}, \dot{\mathbf{p}} := (\dot{\mathbf{p}}_1^{\top}, \ldots, \dot{\mathbf{p}}_1^{\top})^{\top} \in \mathbb{R}^{3N}$ and $\boldsymbol{\xi} := (\boldsymbol{\xi}_1^{\top}, \ldots, \boldsymbol{\xi}_N^{\top})^{\top} \subset \mathbb{R}^{4N}, \dot{\boldsymbol{\xi}} := (\dot{\boldsymbol{\xi}}_1^{\top}, \ldots, \dot{\boldsymbol{\xi}}_N^{\top})^{\top} \in \mathbb{R}^{4N}$, and (6) we have that

$$\sum_{i\in\bar{N}}\sum_{j\in\mathcal{N}_i} a_{ij}\left(\frac{1}{2}\tilde{\boldsymbol{\upsilon}}_{ij}^{\top}\mathbf{e}_{ij} - \boldsymbol{\upsilon}_i^{\top}\mathbf{e}_{ij}\right) = \frac{1}{2}\,\mathbf{p}^{\top}(\mathbf{L}\otimes\mathbf{I}_3)\dot{\mathbf{p}} - \frac{1}{2}\dot{\mathbf{p}}^{\top}(\mathbf{L}\otimes\mathbf{I}_3)\mathbf{p}$$
$$+ \boldsymbol{\xi}^{\top}(\mathbf{L}\otimes\mathbf{I}_4)\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}^{\top}(\mathbf{L}\otimes\mathbf{I}_4)\boldsymbol{\xi}$$

The proof is finished invoking the fact that the Laplacian matrix is symmetric.

References

1. Liu, Y., Chopra, N.: Controlled synchronization of heterogeneous robotic manipulators in the task space. IEEE Trans. Robot. 28(1), 268–275 (2012)
2. Aldana, C., Nuño, E., Basañez, L.: Bilateral teleoperation of cooperative manipulators. In: IEEE Int. Conf. Robot. Autom, pp. 4274–4279 (May 2012)

3. Rodriguez-Angeles, A., Nijmeijer, H.: Mutual synchronization of robots via estimated state feedback: A cooperative approach. IEEE Trans. Control Syst. Technol. 12(4), 542–554 (2004)
4. Nuño, E., Ortega, R., Basañez, L., Hill, D.: Synchronization of networks of nonidentical Euler-Lagrange systems with uncertain parameters and communication delays. IEEE Trans. Autom. Control 56(4), 935–941 (2011)
5. Hatanaka, T., Fujita, Y.I.M., Spong, M.: Passivity-based pose synchronization in three dimensions. IEEE Trans. Autom. Control 57(2), 360–375 (2012)
6. Lee, D., Spong, M.: Stable Flocking of Multiple Inertial Agents on Balanced Graphs. IEEE Trans. Autom. Control 52(8), 1469–1475 (2007)
7. Gu, D.-B., Wang, Z.: Leader-follower flocking: Algorithms and experiments. IEEE Trans. Control Syst. Technol. 17(5), 1211–1219 (2009)
8. Wang, X., Yu, C., Lin, Z.: A Dual Quaternion Solution to Attitude and Position Control for Rigid-Body Coordination. IEEE Trans. Robot. 28(5), 1162–1170 (2012)
9. Qin, J., Zheng, W., Gao, H.: Coordination of Multiple Agents With Double-Integrator Dynamics Under Generalized Interaction Topologies. IEEE Trans. Syst., Man, Cybern. B 42(1), 44–57 (2012)
10. Zhua, W., Cheng, D.: Leader-following consensus of second-order agents with multiple time-varying delays. Automatica 46(12), 1994–1999 (2010)
11. Nuño, E., Sarras, I., Panteley, E., Basañez, L.: Consensus in networks of nonidentical Euler-Lagrange systems with variable time-delays. In: 51st IEEE Conf. on Decision and Control, Maui, Hawaii, USA, pp. 4721–4726 (December 2012)
12. Abdessameud, A., Tayebi, A.: On consensus algorithms design for double integrator dynamics. Automatica 49(1), 253–260 (2013)
13. Li, Z., Duan, Z., Huang, L.: Leader-follower consensus of multi-agent systems. In: Proc. Amer. Contr. Conf, pp. 3256–3261 (2009)
14. Ren, W.: Distributed attitude alignment in spacecraft formation flying. Int. J. Adapt. Contr. Signal Process. 21(2-3), 95–113 (2007)
15. Ren: Distributed cooperative attitude synchronization and tracking for multiple rigid bodies. IEEE Trans. Control Syst. Technol. 18(2), 383–392 (2010)
16. Abdessameud, A., Tayebi, A., Polushin, I.-G.: Attitude synchronization of multiple rigid bodies with communication delays. IEEE Trans. Autom. Control 57(9), 2405–2411 (2012)
17. Abdessameud, A., Tayebi, A.: Formation control of VTOL unmanned aerial vehicles with communication delays. Automatica 47(11), 2383–2394 (2011)
18. Spong, M., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control. Wiley (2005)
19. Caccavale, F., Siciliano, B., Villani, L.: The role of Euler parameters in robot control. Asian J. Control 1(1), 25–34 (1999)
20. Chou, J.: Quaternion kinematic and dynamic differential equations. IEEE Trans. Robot. Autom. 8(1), 53–64 (1992)
21. Kuipers, J.B.: Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality. Princeton University Press (2002)
22. Wen, J.-Y., Kreutz-Delgado, K.: The attitude control problem. IEEE Trans. Autom. Control 36(10), 1148–1162 (1991)
23. Fjellstad, O.: Control of unmanned underwater vehicles in six degrees of freedom: a quaternion feedback approach. Ph.D. dissertation, Norwegian Institute of Technology, University of Trondheim (1994)
24. Campa, R., Camarillo, K.: Unit quaternions: A mathematical tool for modeling, path planning and control of robot manipulators. In: Robot manipulators, pp. 21–48. InTech (2008)

25. Hu, J., Hong, Y.: Leader-following coordination of multi-agent systems with coupling time delays. Phys. A Stat. Mech. Appl. 374(2), 853–863 (2007)
26. Serre, D.: Matrices: Theory and Applications. Springer (2010)
27. Kelly, R., Santibáñez, V., Loria, A.: Control of robot manipulators in joint space. Advanced textbooks in control and signal processing. Springer (2005)
28. Corke, P.I.: Robotics, Vision & Control: Fundamental Algorithms in Matlab. Springer (2011)
29. Spurrier, R.A.: Comment on singularity-free extraction of a quaternion from a direction-cosine matrix. J. Spacecraft and Rockets 15(4), 255 (1978)
30. Mazo, F.: Modelado dinámico y simulación del robot industrial stäubli TX90. Masters thesis (2011), http://upcommons.upc.edu/e-prints/handle/2117/520

Teleoperation of Mobile Robots Considering Human's Commands

Franco Penizzotto, Emanuel Slawiñski, and Vicente Mut

Instituto de Automática, Universidad Nacional de San Juan, San Juan, Argentina
{fpenizzotto,slawinski,vmut}@inaut.unsj.edu.ar

Abstract. Performance in teleoperation of robot strongly depends on user's accomplishment. But most of these systems do not consider human factors in the control design. This paper proposes a teleoperation scheme applied to mobile robots considering operator's dynamic command execution. The controller is based on an impedance law to avoid crashes against obstacles. A novel on-line metric is proposed, which offers a quantitative level about performance of the human operator commands in front of risk. This metric is based on evaluating the risk induced by the operator decisions and execution. The obtained signal is applied in the feedback control loop, modifying the proportional gain of the impedance law. Finally, experimental results are shown in order to test the proposal scheme.

Keywords: Human-robot interaction, Impedance control law, Decision Making, Human Factors, Teleoperation of mobile robots.

1 Introduccion

Despite advances in autonomy, there will always be a need for human involvement in vehicle teleoperation [1].

Thus, several control schemes and strategies for the teleoperation of mobile robots have been developed in order to solve tasks such as land surveying in inaccessible or remote sites, transportation and storage of hazardous material, inspection of high-voltage power lines, de-activation of explosive devices, high-risk fire control, pesticide and fertilizer crop spraying and dusting, mining exploration and various other tasks [2], [3], [4], [5] , [6], [7], [8]. However, none of them take into account the human factors (HF), in spite of they may lead to increase the overall performance of the whole system. [9] emphasizes that HF potential is underexploited.

In [10] it is expressed that in systems with human operators, human−automation interaction associated with both normative and erroneous human behavior can contribute failures such as breakdowns in complex systems. In addition, [11] states that many serious accidents in which human error has been involved, can be attributed to faulty operator decision making. In this context, an operator can correctly select the actions but execute the action poorly by overcorrecting, losing control or flipping the commands. Thus, *good decisions* making are related with

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_43,

decisions that produce "good" outcomes, and bad decisions conversely cause bad outcomes [11]. On the other hand, in [12] a model of situation awareness in dynamic decision making is presented.

This work is focused in the *inclusion into a teleoperation system of the performance of the users actions.* A known control scheme for teleoperation of mobile robots based on impedance [13], is used to avoid obstacles, but the impedance gain is changed according to the user's command execution performance. This is different of the current state-of-art because our proposal includes on-line human factors in the design in order to improve the performance of the teleoperation system. Besides, a metric to quantify the user's action in front of risk is proposed. It focused over the *effects of the human commands* considering the operator's skill required by the situation. Figure 1 shows the control scheme implemented, where no time delay in the communication channel is considered.

This work is organized as follows: First, statement of the problem and the control proposal are presented, while the robot model and the control law are also described. Next, a definition and method for a dynamic command execution metric is exposed. Then, an explanation about the implementation of the method developed is given. After that, some experiments are shown. Finally, conclusions are exposed.

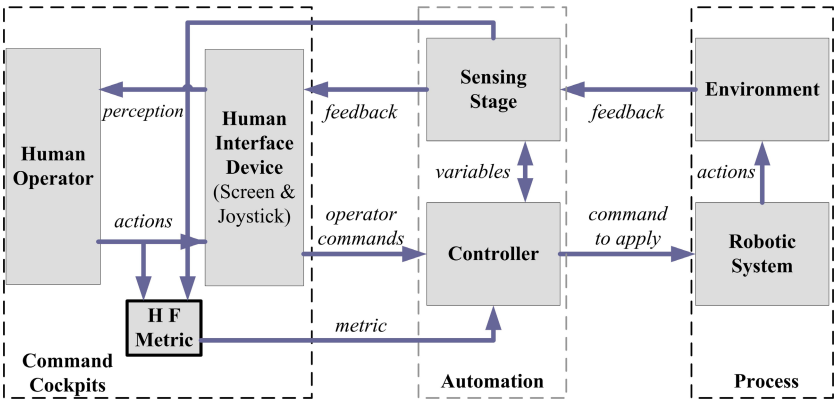


Fig. 1. General block diagram of a human-robot operated system

In order to understand how humans work while driving a complex machine as a robot, figure 2 shows a model of the human information processing stages [11]. In the model presented, the flow of information can be initiated at any point. The blocks that form such model are described in [11].

2 Proposed Controller

The controller of the robot gets the *operator commands* ($[u_h\ ;\ w_h]$) and modifies them delivering the *velocity reference* ($[u_r\ ;\ w_r]$) to the low-level robotics system.

When the controller does not change the input signal, the *command applied* remains equal to *human command*. This work condition is called *control in direct mode*. We define a signal called Pc_i, which represents the collision probability that would arise after the direct application of the user's commands, normalized in the range $[0 - 1]$. Pc_i represents the level of *potential* collision probability due to the *operator commands*.

2.1 Control Law and Stability

Figure 2 represents the subsystems that form the complete control system. Two PIDs outline the low level control loop that receives velocity references ($[u_r \ ; \ w_r]$) and sets voltage signals for the servomotors of the robot. References to the mentioned speed close loop ($[u_r \ ; \ w_r]$) are generated joining the human command and the velocity impedance signal, as it is described in equation 1.

$$\begin{bmatrix} u_r(t) \\ w_r(t) \end{bmatrix} = \begin{bmatrix} u_h(t) \\ w_h(t) \end{bmatrix} - \begin{bmatrix} u_k(t) \\ w_k(t) \end{bmatrix} \tag{1}$$

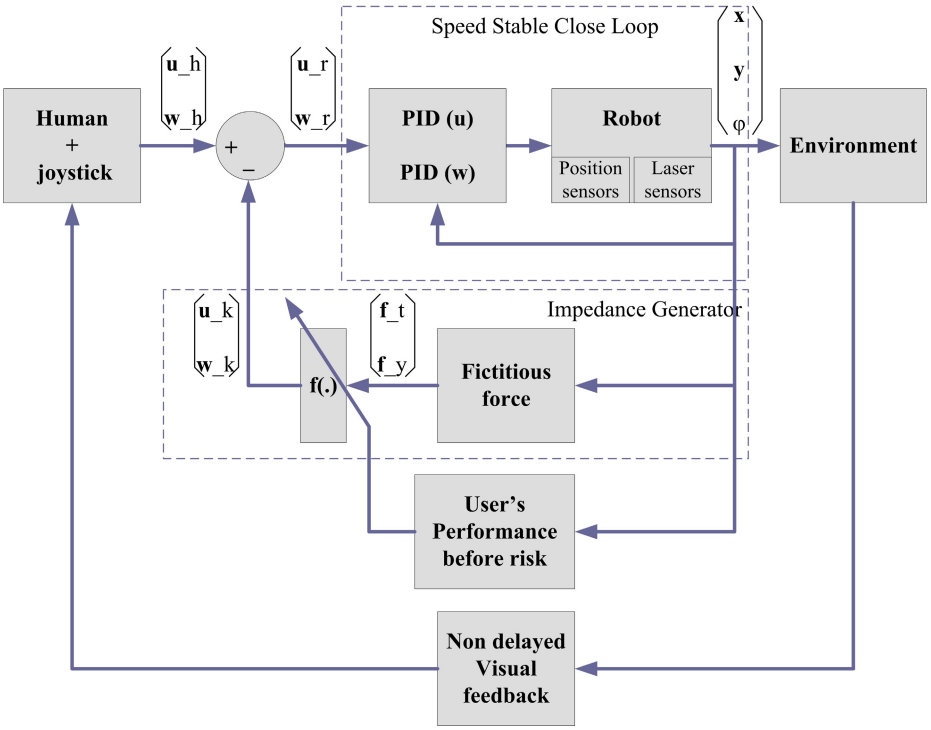


Fig. 2. Control system scheme

Where ($[u_h \ ; \ w_h]$) is the linear and angular speed human command and ($[u_k \ ; \ w_k]$) is the change in the linear and angular speed, generated by the Impedance Generator block described below.

As it is well known, if the PIDs are well adjusted then the velocities of reference are properly followed by the real system speed (stable motion loop). Equation of the robot kinematic for a unicycle mobile robot is described in equation 2. The state $[x \ ; \ y \ ; \ \varphi]$ represents the position of the robot in the horizontal plane and its orientation.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} u_r \cos(\varphi) \\ u_r \sin(\varphi) \\ w_r \end{bmatrix} \tag{2}$$

The impedance Generator block as well as the human commands only change the references for the speed closed loop (stable). Therefore, the whole system is stable too.

2.2 Impedance Generator Block

In the control loop of the teleoperated robot, an impedance control law is applied to a fictitious force, which depends on the obstacles around the robot in order to prevent collisions [14]. Despite operator commands, the robot automatically reduces its mobility when some object (detected by the laser) is near it. Therefore, generally the reference commands sent by the human differ from the real actions executed by the controller.

Figure 3 shows a graphics representation of the fictitious force vector, generated from the robot-obstacle interaction. $d(t)$ is the distance between the robot and the obstacle, and the designer sets a minimum d_{min} and a maximum d_{max} distance, in order to confine the values of the fictitious force vector. In this work, $d(t)$ is measured from the laser information ($d_{min} < d(t) < d_{max}$). The fictitious force vector is decomposed in two components that affect to the linear and angular speed impedance signal. Equation for computing the vector mentioned is described in [15]; where $f(t) = (f_t(t), f_y(t))$, being $f(t) = a - b \ d(t)$, with a and b as positive constant such that $a - bd_{max} = 0$ and $a - bd_{min} = 1$. Then, $f_t = f(t) \cos \beta(t)$ and $f_y = f(t) \sin \beta(t)$, where $\beta(t)$ is the angle between $f(t)$ and $f_t(t)$.

The impedance signal in calculated like equation 3, where speeds references will keep the sense and direction of the user commands.

$$\begin{bmatrix} u_k \\ w_k \end{bmatrix} = f(.) = \begin{bmatrix} u_h(1 - e^{(-\frac{k_u \ f_t}{2})}) \\ w_h(1 - e^{(-\frac{k_w \ f_y}{2})}) \end{bmatrix} \tag{3}$$

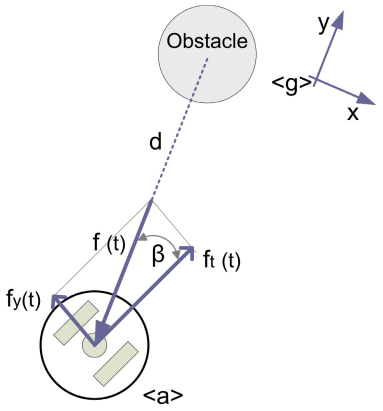


Fig. 3. Fictitious force vector

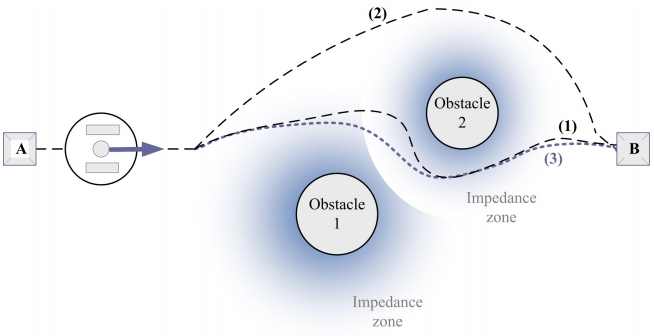


Fig. 4. Conceptual trajectories for different values of K

The gain $K = [k_u \ ; \ k_w]$ is adjusted by the Metric signal computed by the *User's Performance before risk* block. The figure 4 exposes possible paths for three different values of the constant K. Path numbered 1 agree with a non-impedance law ($K = 0$), therefore, the user drives the robot in direct mode. Path number 2 represents the path followed by the robot when the impedance is stuck, with a high K value. Here, the operator commands are highly modified in the nearest of any obstacle detected by the laser. Finally, path number 3 exposes the trajectory of the robot driven by the user helped by a *variable* impedance controller ($K = 0; ..; Max$).

For each of the three possible values of K, the difference between good and bad user's commands can be quantitatively analyzed as follows: For *option* 1, when the command is quite good respect *collisions probability, shortness in path followed and time to complete the task*, the path number 1 is the shorter and the collisions probability will be low. The problem of this controller mode is when the driver generates poor commands in front of risk situations. In one hand, grate *transparency* felling will be present [16]; while on the other hand, these kind of bad commands will generate high levels of collision probabilities with possible crash.

In *option* 2, low risk is insurance despite human's commands performance due to the impedance signal is always activated. The robot travels a long distance and takes a high time to complete the task, despite good commands execution in all senses. Furthermore, the transparency felt by the operator is poor.

In *option* 3, K varies along time. This allows to strongly exploit the stable characteristic of the high impedance signal and the quick and short path followed by the robot driven by a human executing excellent commands. In periods where there is a quite bad operator's command execution before risk, the K variable will take maximum values and will deliver the robot near to the path for case 1 (highly conservative, long and slow trail). Intervals where the operator executes excellent commands ahead of hazard situations, will cause $K \to 0$, which means that the robot will be driven in *direct mode*. Path will be close to number 1 and P_i signals will remain low. When operator executes regular commands in the last sense mentioned, K asumes half values. Therefore, the impedance zone is kept in a medium size taking a trade off between *stability* and *transparency*. The better command's execution over danger, the less impedance and therefore, the system gives a high transparency to the user. If the command's execution became unsafe, then the impedance is high and therefore, the system decreases the transparency. The common characteristic is the secure operation of the robot whether the good or bad human's commands execution.

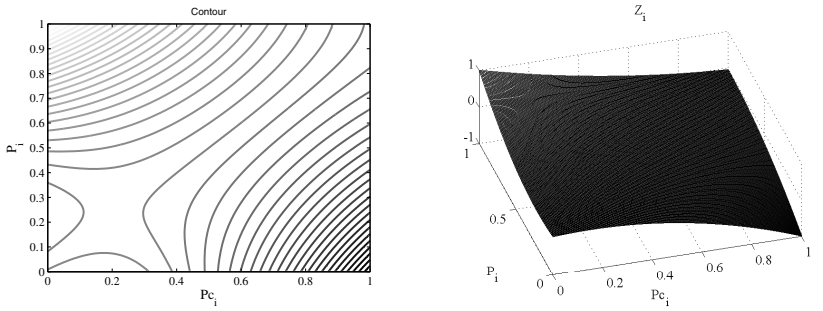
Variation of K is based on the measurement proposal embossed above for the *User's Performance before risk* subsystem. In this work, $K = [k_u \; k_w] = [Me \; 0]$.

3 Definition Proposed for the Metric

In this section, we propose a performance index Me to quantify the *Operator command execution in front of risk*. In order to define Me, the following terms are establish:

- Real risk:
  Level of risk produced by each agent i near to the robot, in the current time $(P_i(t); \quad [low \;\; 0 \ldots 1 \;\; high])$.

– Command risk:
  Level of risk that would appear if the robot controller directly applies the user commands, in the current time ($Pc_i(t)$; $[low \;\; 0 \ldots 1 \;\; high]$)
– Graduation of command's performance:
  Level with respect to the improvement or worsening about *Real risk* as an effect of the *Command risk*, taking into account the level of dexterity that is required by the situation ($Z_i(t)$; $[low \;\; -1 \ldots 1 \;\; high]$).
  For each obstacle i, the proposal is based in the *Real risk (P_i)* and the *Command risk (Pc_i)*. We differentiate when the effect of the operator's command (Pc_i, command risk) potentially *improves*, *preservers* or *worse* the P_i situation of real risk. For example, if the current P_i is valuated in 0.9 and $Pc_i = 0.5$, it means that the command applied by the user, would decrease the real risk.
  Then, in order to valuate the user skill required for the current situation, the more near the crash (high risk level), the more strongly manner the metric should scale the improvement/worsening. For example, the same absolute improvement $P_i = 0.9$ and $Pc_i = 0.5$ or $P_i = 0.6$ and $Pc_i = 0.2$ at a time instant t_n, Z_i should throw a higher value for the first case due to that the improvement demands wiser and quicker commands.



(a) Contour of the surface (b) Surface of the P, Pc and Z.

Fig. 5. Surface for computing $Z_i(t)$

Therefore, in order to quantify $Z_i(t)$ as a function of Pc_i and P_i, a surface is proposed and showed in figure 5. $Z_i(t)$ is bounded between -1 and 1, and includes 3 ranges: *preservation improvements* and *worsening*. Figure 5(b) shows the nonlinear quantification of the $Z_i(t)$ signal. Figure 5(a) pictures the contour of the surface. Green line correspond to $Z_i = 0$. An example is shown in section 3.1.

Now, we define the Me as *"The degree of goodness of choice and execution that a human operator (driver) before risk situations takes"*. Me is computed as the *Distance* between the *Ideal Action* and the *Real Action*, where the Ideal Action (Me_{id}) is the *best possible user's command* and the Real Action (Me_{re}) involved the *Choice and execution* that the operator *effectively* applies through the interface (4). This is,

$$Me = \|Me_{id} - Me_{re}\| \tag{4}$$

where

$$Me_{id} = (Z_{1_id},\ Z_{2_id},\ \ldots\ ,\ Z_{q_id}) \tag{5}$$

with Z_{i_id} calculated like Z_i but considering $P_c = 0$ and

$$Me_{re} = (Z_{1_re},\ Z_{2_re},\ \ldots\ ,\ Z_{q_re}) \tag{6}$$

with Z_{i_re} calculated as Z_i but taking into account the calculated value of P_c. Besides, i corresponds to the identification subindex for each obstacle and the variable q is the amount of obstacles in the current time t_n.

Figure 6 shows the vectorial model proposed in order to compute the metric definition (Me), considering only two obstacles for simplicity. A value of $Me = 0$ implies a perfect Operator Decision Making in Command Execution, and $Me = \sqrt{q\ (1 + z_{bp})}$ confine the worst Decision for the q present obstacles. z_{bp} is the worst value of the range *Preservation*. The upper limit of the Me is non linear and it is altered by the increment of obstacles in the environment.

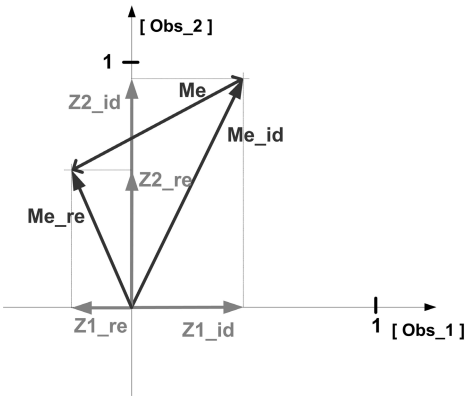


Fig. 6. Quantitative components of the metric proposed

3.1 Implementation of the Metric Block

A function calculates the crash probability P_i based on the information of the laser, while Pc_i signal is computed from laser information, the robot's kinematic model and the direct operator's commands.

Once the calculation of $P(t)$ and $Pc(t)$ is made, $Z(t)$ is set through a neural network. In order to create many points $Z = (P \quad Pc)$ to train the net, a four plane surface is defined as it is shown in Figure 7. The net consist in one hidden layer with 10 neurons, and a linear function output layer. Figure 8 portrait the approximation reached with the neural network used. After computing Z for each obstacle, Me is calculated as in equation 4.

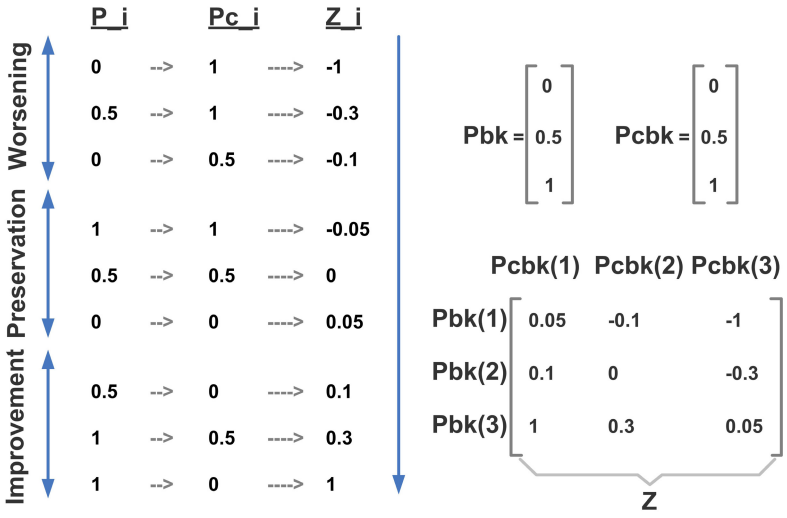


Fig. 7. Example of calculation of Z_i

4 Experiments

In the application proposed, the human drives a mobile robot as quick as possible, from an initial point to a final one, in an environment with two fixed obstacles which must be avoided by the robot. Operator drives the robot by using a joystick while receives the image captured by the camera on board of the robot, with no presence of time delay. Over the robot, it is also considered a 180 degree scanner laser. Experiment are performed using the MobileSim platform, which simulates a Mobile robot Pioneer 3AT.

In order to analyze the control scheme proposed, different k_u are used: constant high value (maximum), null, and k_u variable between 0 and a maximum value, based on the $Me(t)$ signal. For all the cases, k_w is equal to 0 for simplicity.

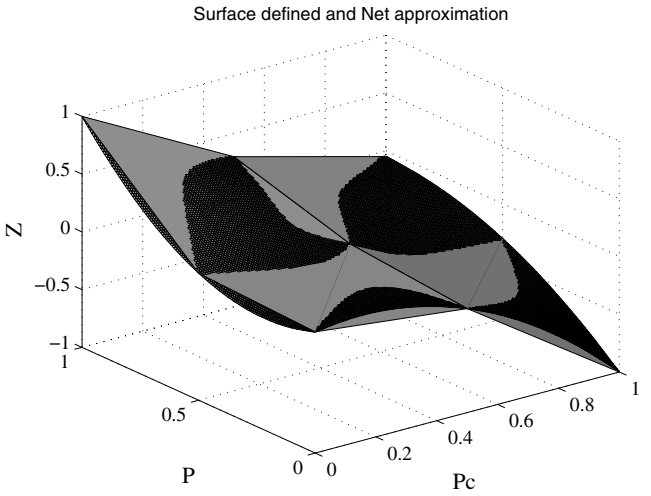


Fig. 8. Test of the neural network

As human users of the system, expert and novice drivers were taken to perform the required task. Drivers who know how the control system works, were classified as expert users. On the contrary, novice users are the operators that do not know anything about the control system.

The next list describe the different cases tested.

Description of the cases	Case
$k_u = Max$	A
$k_u = 0$, Experts and novices	B
$k_u = Me$, Experts and novices	C

Figure 9 shows three robot trajectory, numbered (1, 2 and 3) in concordance with the figure 4, taking one relevant experiment of each case where $k_u = 0$, $k_u = Max$ and $k_u = Me$ respectively.

In order to show results from experiments of trajectory 3, graphics of the collision probability P_1 and P_2 (P for each obstacle), command collision probability Pc_1 and Pc_2 (P_c), metric (Me) and command compensation (u_h against u_r) are shown in figure 10. After time 16.65, the robot overcomes both obstacles which locates out of the laser range, therefore, collision probabilities goes to 0. Experiments correspond to the controller using the linear speed impedance compensation according to the metric proposed. When the driver executes good commands, it can be noted the small differences between *sent* and *command* references. Adversely, when bad commands are performed, mentioned differences are greater, not only in amplitud, even in lapse of time. Computation of P and Pc for both obstacles present in the environment are also shown, as well as the metric signal (Me) along the experiment.

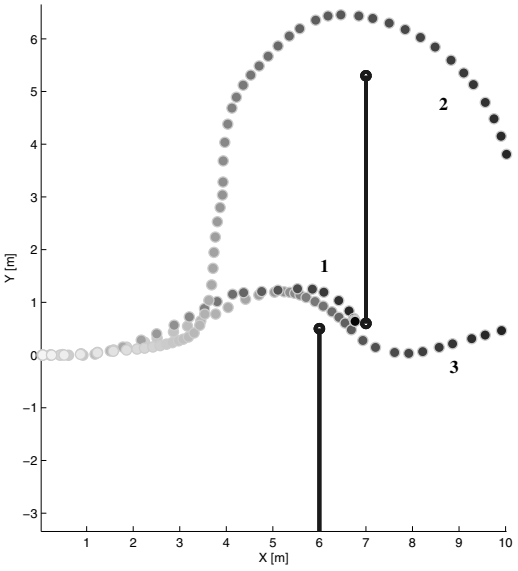


Fig. 9. Trajectories of the robot, case **A**

Table 1. Table of variables measured from experiment

Case		N_{clsn}	T_{task}	$Max(P_1, P_2)$
A ($k_u = Max$)	Average	0	43,241	0,022
	Standard deviation		6,201	0,015
B ($k_u = 0$)	Average	20%	23,50	0,889
	Standard deviation		1,46	0,054
C ($k_u = Me$)	Average	0	26,77	0,778
	Standard deviation		2,28	0,023

Table 1 resumes the average values of the experiments made (10 for each case, where it was considered %50 for novice users and %50 for expert users) in order to test the control system proposed and compare between different cases listed above. Standard deviation is also compute for each case. N_{clsn} represents the percentage of experiments in which the robot crashes against any obstacle. Next, T_{task} measures the time to complete the task. Finally, $Max(P_1, P_2)$ is the maximum collision probability measured along the T_{task}.

Analysis for $k_u = Max$. When a strong impedance control is permanently activated, although the good or wrong operator commands, the robot does not go through the two present obstacle due to high and fix repulsion zone. The

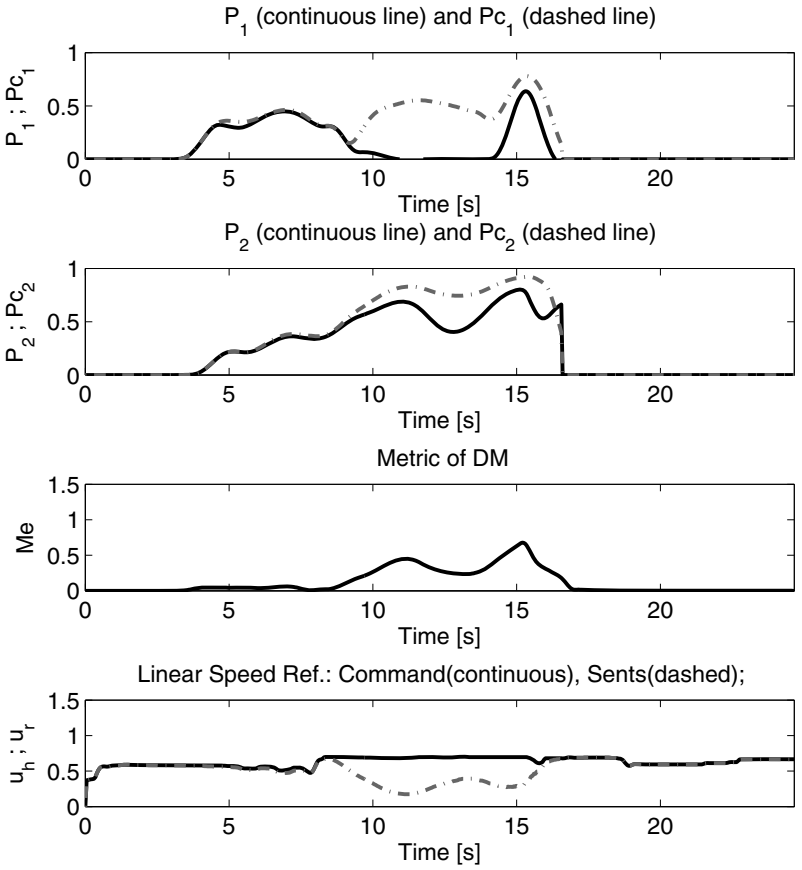


Fig. 10. Curves for one of the cases type **C** (Metric applied and Good commands)

navigation of the robot is always safe for any kind of commands, but it offers bad performance in path and time to complete task for all occasions (about 40% more than the others cases; see T_{task} of table 1). In this case, the *Transparency* is unsatisfactory because the user's command is highly modified by the controller, and the operator perceives that the robot does not respond to his orders. This impedance law do not consider human driver performance.

Analysis for $k_u = 0$. When $k_u = 0$, conditions of *safety* driving are null and depends only on the operator performance. When bad commands occur, high risk situation appear, and collisions could happen. This situation can be seen in the percentage of experiments that ended in collisions as a result of the bad commands. Also, $Max(P_1, P_2)$ is the highest, indicating risk situations were present. All participant express that felt a good sensation when commanding

the robot which exactly follows the joystick (nice *Transparency*). Figure (9, trajectory 1) shows the robot crashing against an obstacle, as one of te examples of this control mode.

Some good drivers achieve good executions, but this condition is difficult to maintain during a long task, or when there appears secondary demanding tasks which overload the attention or cognition of the driver, and therefore, poor decision and execution are taken.

Analysis for $k_u = Me$. When $k_u = Me$, conditions of *safety* driving do not depend on the current operator performance. If bad commands occur, k_u takes high values and the risk situation is bounded. Thus, the collisions are avoided at cost of losing transparency.

Trajectories are similar whether the use or not of the speed impedance (like trajectory in figure 9, trajectory 3). The control system interferes over the commands only when appears danger due to the user executions. Due to this assistance made by the control system, $Max(P_1, P_2)$ is lower (cases C) than the cases without it (B), showing a safer navigation. Thus, no collision cases are found. Furthermore, standard deviation of these signal is approximately the half of the case (B), indicating a more homogeneous trajectory despite the performance of the commands executed.

Time to complete the task depends somewhat on the commands performance, but, for both cases, is much smaller than the case A ($k_u = Max$) and closer to the best one reached.

5 Conclusion

The paper offers a development of teleoperation control scheme taking into account the performance in front of risk of the user's command execution. The metric proposed reaches an on-line value that considers the risk involved, the skill required for the commands and the future effects of the user commands. Multiple obstacles are allowed in the robot operation environment, as well as variant quantity of them. The suggested method for the metric can be applied to other control schemes where the user commands a main task (in this work: driving the robot) avoiding restrictions or threats (in this work: avoiding obstacles). Results shows that, for the teleoperation of mobile robots, the scheme applied present a better trade-off between safety, transparency, path and time to completion.

Acknowledgements. This work was supported by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), the National University of San Juan and the Automatic Institute of the National University of San Juan.

References

1. Fong, T., Thorpe, C.: Vehicle teleoperation interfaces. Autonomous Robots 11, 9–18 (2001)
2. Sheridan, T.B.: Telerobotics, Automation and Human Supervisory Control. MIT Press (1992)

3. Sayers, C.: Remote control robotics. Springer (1999)
4. Slutski, L.: Remote manipulation systems: quality evaluation and improvement. Kluwer Academic Publishers (1998)
5. Fiorini, P., Oboe, R.: Internet-based telerobotics: Problems and approaches (1997)
6. Elhajj, I., Xi, N., Fung, W.K., Liu, Y.H., Hasegawa, Y., Fukuda, T.: Supermedia enhanced internet based telerobotics. Proceedings of the IEEE 91, 396–421 (2003)
7. Lam, T.M., Boschloo, H.W., Mulder, M., van Paassen, M.M.: Artificial force field for haptic feedback in uav teleoperation. IEEE Transactions on Systems, Man, and Cyberneteics-Part A: Systems ans Humans 39, 1316–1330 (2009)
8. Sanders, D.: Comparing ability to complete simple tele-operated rescue or maintenance mobile-robot tasks with and without a sensor system. Sensor Review 30, 40–50 (2010)
9. Dul, J., Bruder, R., Buckle, P., Carayon, P., Falzon, P., Marras, W.S., Wilson, J.R., van der Doelen, B.: A strategy for human factors/ergonomics: developing the discipline and profession. Ergonomics 55(4), 377–395 (2012)
10. Bolton, M.L., Bass, E.J., Siminiceanu, R.I.: Generating phenotypical erroneous human behavior to evaluate humanautomation interaction using model checking. International Journal of Human-Computer Studies (2012)
11. Wickens, C.D., Hollands, J.G.: Engineering Psychology and Human Performance, 3rd edn., vol. 1. Prentence Hall, New Jersey 07458 (2000)
12. Endsley, M.R.: Measurement of situation awareness in dynamic systems. Human Factors 37, 65–84 (1995)
13. Hogan, N.: Impedance control: An aproach to manipulation, part 1: Theory. Journal of Dynamic Systems, Measurement and Control 107, 1–7 (1985)
14. Slawiñski, E., Mut, V., Salinas, L., Garca, S.: Teleoperation of a mobile robot with time-varying delay and force feedback. In: Available on CJO 2011, vol. 201 (2011)
15. Slawiski, E., Mut, V., Postigo, J.: Teleoperation of mobile robots with time-varying delay. IEEE Transaction on Robotics 23 (2007)
16. Slawiñski, E., Mut, V.A., Fiorini, P., Salinas, L.R.: Quantitative absolute transparency for bilateral teleoperation of mobile robots. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans PP(99), 1–13 (2011)

Safe Teleoperation of a Dual Hand-Arm Robotic System

Jan Rosell[1,⋆], Raúl Suárez[1], and Alexander Pérez[2]

[1] Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain
{jan.rosell,raul.suarez}@upc.edu
[2] Escuela Colombiana de Ingeniería "Julio Garavito", Bogotá D.C., Colombia
alexander.perez@escuelaing.edu.co

Abstract. This paper introduces a supervised teleoperation system to cope with some of the main problems that arise in the teleoperation of hand-arm robotic systems. The set-up consists of two magnetic trackers and two sensorized gloves that command, respectively, two industrial robots and the mechanical hands with which they are equipped. The basic mapping aspects both at hand and arm level are discussed, as well as the communication issues related with the implementation done based on ROS (Robot Operating System). The risk of collisions and of reaching singular configurations, either internal or due to the workspace limits, is controlled by monitoring the state of the hand-arm system and stoping it when necessary. An automatic re-synchronization procedure allows to resume the teleoperation as soon as the risk has disappeared, and also permits the user to change the mapping when his/her posture becomes uncomfortable. The resulting system allows an intuitive, simple and safe teleoperation of a dual hand-arm robotic system.

1 Introduction

Despite the advances in the development of autonomous robots, with the robots taking decisions by themselves, the teleoperation of robots has a significant role in applications in which the decisions are still taken by an operator and the robots perform the actions following the movements of the operator. Advances in teleoperation of robots were based on the advances in the robot control systems, the improvements in the velocity and quality of the communications, and the development of devices to capture in a reasonable time and with a reasonable precision the gestures (or intentions) of the operator. A detailed discussion of the problems related with teleoperation as well as a description of typical applications was presented in [1].

⋆ Work partially supported by the Spanish Government through the projects DPI2010-15446 and DPI2011-22471. The authors want to thank Riccardo Rocchio, Armando Palumbo and Dario Davide for the implementation of the ROS nodes of the proposed system.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_44,

One of the problems related with the teleoperation of anthropomorphic devices is the complexity generated by the large number of involved degrees of freedom (*dof*), e.g. up to 27 for the case of a 6 *dof* arm plus 21 of a hand with 5 fingers with 4 independent dof each one plus one additional *dof* in the palm. Therefore, the most intuitive way to command these devices is to capture the movements of a human operator and map them to the robotic system. But even in this case, the kinematic structure of the device is, in general, different from that of the operator. This means that the mapping of the sensorial information obtained from the operator movements onto the movement space of the devices is not an easy issue. This has lead to different mapping approaches, usually depending on the available hardware, but, in any case, quite frequently the operator has to arrive to non comfortable positions in order to command a robotic hand-arm system towards the proper configuration according to a visual feedback. One immediate consequence of this is that the operator has difficulties to properly execute a task with the teleoperated robot, even when the task looks like being trivial, and this easily ends with the teleoperated robot producing undesired collisions. One solution to this problem is the implementation of virtual constraints that do not allow the robot to move beyond some acceptable limits.

Three main approaches have been presented related with the mapping issues in teleoperation of anthropomorphic devices, particularly when anthropomorphic hands are involved [2][3][4]: *Joint-to-joint mapping*, in which each sensor of the operator movements is directly associated with a joint of the robot (e.g. [5]), *Pose mapping*, in which each pose of the operator is associated with a predefined pose of the robot (e.g. [6]), and *Point-to-point mapping*, in which the positions of particular points of the operators arms or hands are replicated by predefined points on the robot (e.g. [7]).

The identification of the pose of the operator to do teleoperation of anthropomorphic devices has been addressed using different types of sensorial information. The two most frequent approaches are based on vision systems, which is useful when information about the gesture of the operator is enough without needing precise positioning and the chance of occlusions is minimal (e.g. [8][9][10]), and based on the used of trackers and sensorized gloves, which quickly provides information about the joints of the operator and can be used for anthropomorphic and non anthropomorphic devices (e.g. [7][11][5]).

In this context, this paper deals with the problem of teleoperating a dual hand-arm robotic system and presents the implementation of a teleoperation approach, including the teleoperation of the arms (with a Point-to-point mapping) and the hands (with a Joint-to-joint mapping). The approach tries to be intuitive and robust at the same time and also cares about security aspects, limiting the movements of the robotic system to avoid collisions or undesired configurations.

After this introduction, the paper is structured as follows. Section 2 briefly presents a system overview and then Section 3 describes the hardware set-up. Sections 4, 5 and 6 deal, respectively, with the mapping, the communication and the supervision issues of the implementation. Finally, Section 7 summarizes the work.

2 System Overview

The teleoperation system proposed in this work consists of two industrial robots equipped with mechanical hands that are to be teleoperated by an operator equipped with wrist trackers and sensorized gloves. The main goal of the proposal is that the system:

- Allows an intuitive teleoperation, both at the hand level (with respect to the mapping between the joints of the glove and those of the mechanical hand), and also at the robot level (assuming the possibility to use different cameras to feed back the image from the remote site).
- Allows a safe teleoperation stopping the robot when necessary to avoid collisions and singular configurations, and automatically and smoothly resuming the teleoperation to recover from these situations.

The logic schema of the system is shown in Fig. 1. The basic modules are:

- *Synchronization*: Required to synchronize the data provided independently by the wrist trackers and the sensorized gloves.
- *Mapping*: Required to transform the data provided by the trackers and gloves into joint values for the robots and the mechanical hands. This module has to verify wether the obtained configuration is kinematically correct, i.e. not singular and within limits.
- *Collision Validation*: Required to verify that the commanded configuration is collision-free.

To control all the devices and manage the communications between them and all the teleoperation modules, the Robot Operation System (ROS, www.ros.org) is used.

3 Hardware Set-Up

3.1 Magnetic Trackers

Two different trackers of different brands were used in this work to capture the position and orientation of the operator wrist with respect to a global reference frame, allowing a mapping of the displacements of the user arm to the robot arm. One tracker is a *flock of birds* manufactured by Ascension Technology and the other a *fastrack* manufactured by Polhemus, both are shown in Figure 1 above the boxes with the corresponding names. Both trackers are prepared to be used together with sensorized gloves (described below). These devices are of magnetic type, measuring the position and orientation of a small sensor with respect to a base fixed in the workspace. The *flock of birds* has a static precision of 1.8 mm for the position and 0.5° for the orientation and allows 144 samples per second, and the *fastrack* has a static precision of 0.8 mm for the position and 0.15° for the orientation and allows 120 samples per second.

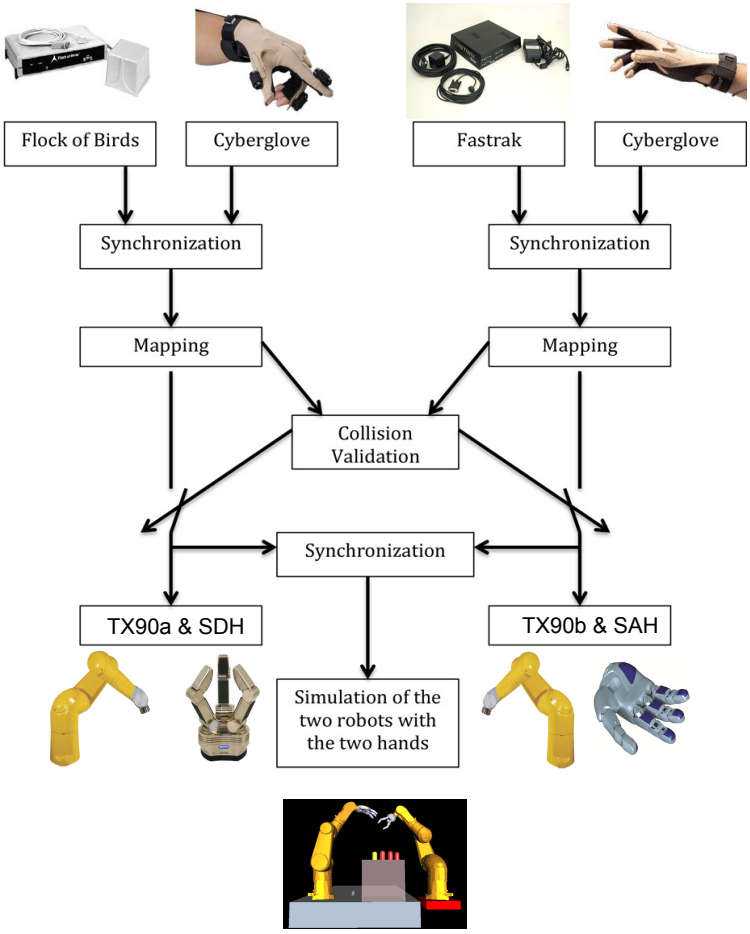


Fig. 1. System overview

3.2 Industrial Robots

The industrial robots are two Stäubli TX90, called TX90a and TX90b, are shown in Figure 1 below the boxes with the corresponding names. These are general purpose 6 *dof* serial robot arms with revolute joints. Each robot is equipped with a CS8C controller, that runs a real time operative system VxWorks, and over it the command interpreter VAL3 language facilitates the robot-level programming. The nominal capacity of this robotic arm is 7 kg and the workspace has a radius of approximately 1 m.

3.3 Sensorized Gloves

Two sensorized gloves Cyberglove from Immersion Corporation were used to capture the hand configurations, i.e. the positions of the operator fingers. Two

pictures of the glove are shown in Figure 1 above the boxes with the corresponding name. Cyberglove is a fully instrumented glove that provides 22 joint-angle measurements using resistive bend-sensing technology, including three flexion sensors per finger, four abduction sensors between the fingers, a palm-arc sensor, and two sensors to measure the flexion and the abduction of the wrist.

3.4 Mechanical Hands SAH and SDH

Two different mechanical hands were used in this work, the Schunk Anthropomorphic Hand (SAH) and the Schunk Dexterous Hand (SDH), which are shown in Figure 1 below the boxes with the corresponding names. Both hands can be attached to the industrial robots through an industrial standard interface EN ISO 9404-1-50.

The SAH has four identical fingers with four joints each one (abduction, proximal flexion, medium flexion and distal flexion), being one of them prepared to work as the opposing thumb, which is equipped with an additional joint (thumb-base joint) that moves the whole thumb with respect to the palm. In all the fingers the distal and medium flexion joints are mechanically coupled, thus there is a total of 17 joints with only 13 independent *dof*. The weigh of the hand is 2.2 kg and the size is around 1.4 times the average human hand. The maximum regular force at the fingertips is approximately 5 N. The hand requires a power supply of 24 V with a regular power of 1 A when it is unload and a maximum power of 2.7 A when it is applying grasping forces.

The SDH has identical fingers with two joints each one (proximal flexion and distal flexion) and two of them have an additional *dof* that makes the fingers rotate around their bases to put them exactly one in front of the other, thus, the whole hand has 7 independent *dof*. Although it is not anthropomorphic it is suitable for many different grasping actions. The weigh of the hand is 1.95 kg and the size is also around 1.4 times the average human hand. The hand requires a power supply of 24 V with a maximum power of 5 A when it is applying grasping forces. It is also equipped with two arrays of tactile sensors per finger, but these are not used in this work.

4 Mapping Issues

4.1 Tracker-Robot

Let H be the reference frame of the workspace of the operator, $\mathcal{W}_H$, and W that of the robot workspace, $\mathcal{W}_W$. Let also HIP be the reference frame of the tracker located at the wrist of the operator, and TCP be the reference frame at the end-effector of the robot. Then, the mapping between the HIP and the TCP is based on the procedure proposed in [12] that maps $\mathcal{W}_H$ into $\mathcal{W}_W$ to obtain the virtual operator workspace ($\mathcal{W}_{\mathsf{vH}}$, with reference frame vH), following two guidelines:

– The orientation of vH must be the same as that of the camera used to feed back the video (Fig. 2 left).

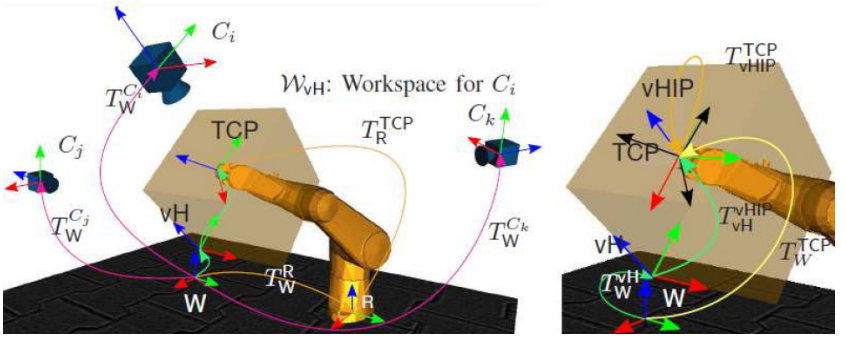


Fig. 2. (left) Model of the remote site showing the virtual operator workspace aligned with camera i; (right) Virtual coupling attained by making the origins of frames vHIP and TCP coincident. Figures reproduced from [12].

– The origin of the HIP within vH, called vHIP, must coincide with that of the TCP (Fig. 2 right).

While teleoperating, the location of the HIP with respect to H, i.e. the transform $T_{\mathsf{H}}^{\mathsf{HIP}}$, is continually read from the magnetic tracker, and the transform $T_{\mathsf{vH}}^{\mathsf{vHIP}}$ is made coincident with it (or modified by a simple scale factor) and used to obtain the location of the TCP with respect to W (Fig. 2 right):

$$T_W^{\mathsf{TCP}} = T_{\mathsf{W}}^{\mathsf{vH}} \cdot T_{\mathsf{vH}}^{\mathsf{vHIP}} \cdot T_{\mathsf{vHIP}}^{\mathsf{TCP}}, \tag{1}$$

where the transforms $T_{\mathsf{W}}^{\mathsf{vH}}$ and $T_{\mathsf{vHIP}}^{\mathsf{TCP}}$ are fixed and set when teleoperation starts:

$$\begin{aligned} T_{\mathsf{vHIP}}^{\mathsf{TCP}} &= \left(\begin{array}{c|c} \mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} & \mathbf{0} \\ \hline 0 & 1 \end{array}\right) \\ \mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} &= (\mathcal{R}_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \cdot (\mathcal{R}_{\mathsf{W}}^{\mathsf{Ci}})^{-1} \cdot \mathcal{R}_{\mathsf{W}}^{\mathsf{TCP}} \\ T_W^{\mathsf{vH}} &= T_W^{\mathsf{TCP}} \cdot (T_{\mathsf{vHIP}}^{\mathsf{TCP}})^{-1} \cdot (T_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \end{aligned} \tag{2}$$

being Ci the reference system of the camera that feeds back the image from the remote site.

Besides the initialization at the start time, the establishment of the virtual connection between the tracker and the robot, as stated by Eq. (2), is also done in the following situations encountered while teleoperation is executed:

– The operator changes the reference camera to have a better view of the robot workspace, or any other configuration parameter like the scaling.
– The operator stops the teleoperation and resumes it in another (more comfortable) position (this is done by pressing the clutch button on the tracker).
– The inverse kinematics cannot be calculated because the robot is placed at a singular configuration.
– The teleoperation was stopped because an imminent collision or a singular configuration was detected, and it is resumed from a different safe commanding position.

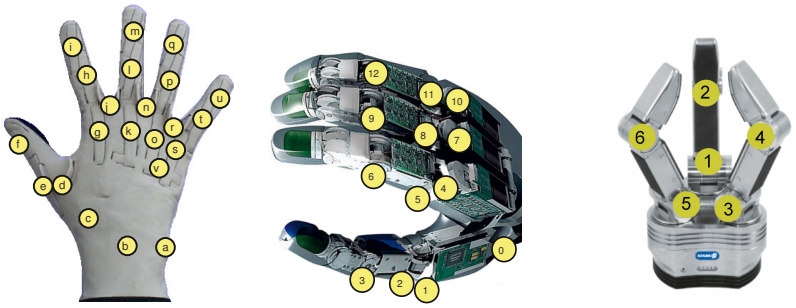


Fig. 3. Cyberglove with the sensor labels and SAH (middle) and SDH (right) with the joint labels

4.2 Glove-Mechanical Hands

A particular mapping was implemented for each of the mechanical hands. The mapping from the glove to the SAH is more direct since it is anthropomorphic, while the mapping to the SDH was thought to allow a simple way of commanding of the hand.

Mapping Glove-SAH. The mapping from the glove sensors to the SAH movements is based on a previous work [13], where different types of mapping are presented for this glove and this hand. Basically, the following issues are considered (the sensors of the glove and the joints of the hand are labeled as shown in Fig. 3):

- The palm of the mechanical hand is rigid and therefore the palm arc sensor v and the wrist flexion and abduction sensors a and b are ignored.
- The mechanical hand lacks the little finger, thus the sensors u, t, s and r are ignored.
- The mechanical hand has a coupling between the medium and distal joints of each finger, therefore the distal sensors i, m and q are ignored.
- The use of sensor c to command the hand joint 1 produces, in practice, a more natural motion of the SAH than using sensor d, therefore sensor c is used for both joints 0 and 1.
- The abduction movements are measured as relative movements in the glove, with sensors j and n providing, respectively, the angle between the index and the middle fingers and the angle between the middle and the ring fingers, while the abduction is measured as the absolute movement of each finger in the mechanical hand. Then, the mapping is done using the middle finger of SAH as a fixed reference (joint 7 is fixed to 0) and then sensors j and n are directly associated to joints 4 and 10 respectively.

Then, with this mapping only 11 out of the 22 available sensor in the glove are actually used to command the hand, even when the hand has 13 *dof*. The complete mapping is shown in Table 1.

Table 1. Mapping from the Cyberglove to the SAH

Cyberglove sensors		SAH joints	
label	name	label	name
c	thumb base	0	thumb base
c	thumb base	1	thumb abduction
e	thumb medium-flexion	2	thumb proximal-flexion
f	thumb distal-flexion	3	thumb medium-flexion
j	index-middle abduction	4	index abduction
g	index proximal-flexion	5	index proximal-flexion
h	index medium-flexion	6	index medium-flexion
-	middle abduction	7	middle abduction
k	middle proximal-flexion	8	middle proximal-flexion
l	middle medium-flexion	9	middle medium-flexion
n	middle-ring abduction	10	ring abduction
o	ring proximal-flexion	11	ring proximal-flexion
p	ring medium-flexion	12	ring medium-flexion

Mapping Glove-SDH. The mapping from the glove sensors to the SDH movements is also a joint-to-joint mapping, but it is not as evident as the previous one because the hand SDH is not anthropomorphic and both the number and the structure of the *dof* are different. The direct association of one finger of the glove with one finger of the mechanical hand was ruled out after some initial experiments, since it is not easy for the human operator to intuitively move the hand to make the mechanical hand doing the desired actions. Instead, for grasping applications, a simple mapping associating the three fingers of the SDH to only one single finger of the glove produced acceptable results. This mapping ignore some possible manipulation configurations of the SDH, but makes it really easy for the operation to exploit some others, the mechanical hand is opened or closed in a predefined grasp configuration when the operator makes an extension or flexion of the index finger. The complete mapping is shown in the Table 2 (see Fig. 3 for the labels of the sensors of the glove and the joints of the hand). The joint 0 of the SDH is not associated with any joint of the glove, instead it is fixed a priori depending on the desired grasp configuration of the hand.

Table 2. Mapping from the Cyberglove to the SDH

Cyberglove sensors		SDH joints	
label	name	label	name
g	index proximal-flexion	1	finger 1 - proximal
h	index medium-flexion	2	finger 1 - distal
g	index proximal-flexion	3	finger 2 - proximal
h	index medium-flexion	4	finger 2 - distal
g	index proximal-flexion	5	finger 3 - proximal
h	index medium-flexion	6	finger 3 - distal

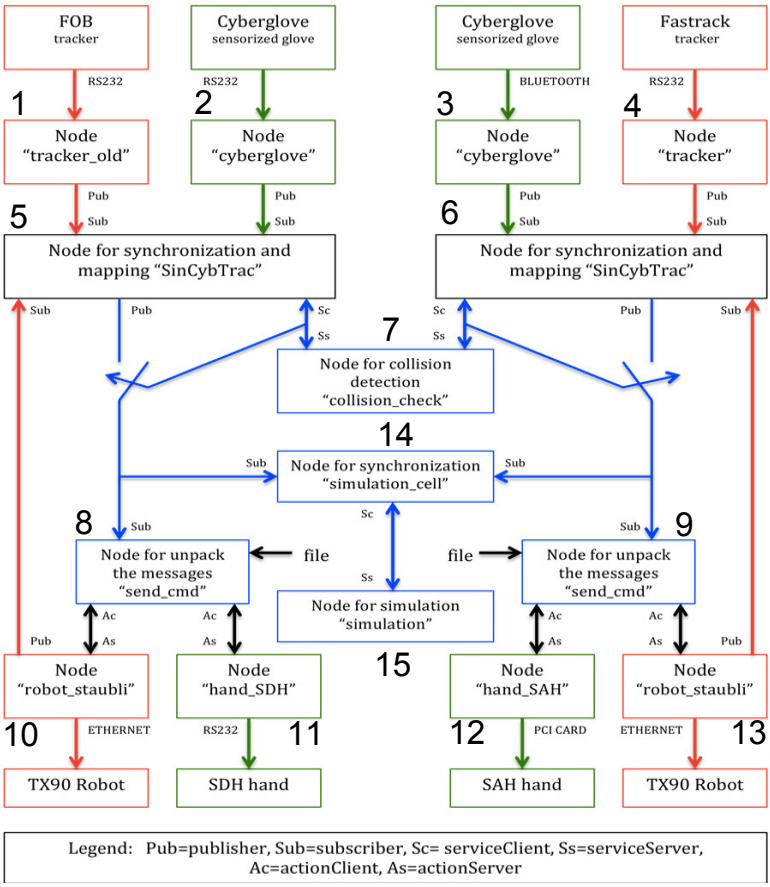


Fig. 4. ROS nodes

5 Communication Issues

5.1 The Robot Operating System (ROS)

The Robot Operating System (ROS) is a meta-operating system for robotic applications that includes hardware abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between processes, package management, and provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

Robotic applications are composed of several processes loosely coupled, connected as nodes of a peer-to-peer network (the ROS runtime graph) using different types of communications:

– **Topic:** Communication based on the Publisher/Subscriber mechanism. It is an asynchronous communication of strongly typed data (the *messages*)

Table 3. ROS nodes, purpose and type of communication used

#	Name	Purpose	Communication
1	tracker_old	broadcasts the data of the FOB tracker	Publish Topic
2	cyberglove	broadcasts the data of the Cyberglove (left hand)	Publish Topic
3	cyberglove	broadcasts data of the Cyberglove (right hand)	Publish Topic
4	tracker	broadcasts data of the Fastrack tracker	Publish Topic
5	SinCybTrac	synchronizes (left) glove and FOB data maps to SDH and TX90a configuration queries collision-check broadcasts a correct configuration	Subscribe Topic – – – – Service Client Publish Topic
6	SinCybTrac	synchronizes (right) glove and Fastrack data maps to SAH and TX90b configuration queries collision-check broadcasts a correct configuration	Subscribe Topic – – – – Service Client Publish Topic
7	collision_check	performs a collision-check test	Service server
8	send_cmd	reads valid hand-arm data commands the SDH hand and the TX90a robot	Subscribe Topic Action clients
9	send_cmd	reads valid hand-arm data commands the SAH hand and the TX90b robot	Subscribe Topic Action clients
10	robot_staubli	moves the TX90a robot broadcasts the TX90a robot position	Action server Publish Topic
11	hand_SDH	moves the SDH hand	Action server
12	hand_SAH	moves the SAH hand	Action server
13	robot_staubli	moves the TX90b robot broadcasts the TX90b robot position	Action server Publish Topic
14	simulation_cell	gets the two sets of hand-arm data sends a move command to the simulator	Subscribe Topics Service Client
15	simulation	moves the simulated system	Service Server

between a node that produces the information and publishes it under a *topic*, and the node(s) that subscribe to that topic to consume this information.

- **Service:** Communication based on the Client/Server mechanism. It is a synchronous communication between a node (the client) that sends a request message to another node (the server) and awaits the reply, i.e. allows to perform a remote procedure call.
- **Action:** Communication based on the Client/Server mechanism with feedback. In some cases the service takes a long time to execute and therefore, in this case, the client needs to get periodic feedback about how the request is progressing and, may be, cancel the request.

The conceptual schema shown in Fig. 1 is redrawn in Fig. 4 showing those ROS nodes that encapsulate each of the hardware devices and those that implement synchronization and supervision functions. The type of communication used by these nodes varies with the communication needs as explained below and summarized in Table 3:

- The glove and tracker nodes (#1 to #4) simply broadcast their data, i.e. the communication between these nodes and the synchronization and mapping nodes (#5 and #6) is done using Topics.
- The synchronization and mapping nodes (#5 and #6) send queries to the collision-check node (#7) using Services, once they have computed the configuration of the robotic system, in order to receive information of whether they are collision-free or not.
- The nodes (#8 and #9) send the motion commands to the nodes that control the mechanical hands and the robots (#10 to #13) using Actions in order to get feedback of their execution to guarantee the correct performance.
- The data computed by the synchronization and mapping nodes (#5 and #6) is broadcasted using Topics; the nodes responsible for sending the commands to the hands and robots (#8 and #9) subscribe to them and orderly manage the queues that receive the information.

5.2 ROS Nodes for Trackers and Gloves

The ROS node of each tracker performs the following steps:

1. Opens a RS232 connection and configures the tracker (the Flock of Birds tracker use the *FOBLIB* library and the Fastrack tracker the *Polhemus* library).
2. Reads data from the tracker (position and orientation).
3. Publishes the data on a Topic, named /tracker_old/tracker_states for the FOB and /tracker/tracker_states for the Fastrack, formatted using the standard ROS messages of type geometry_msgs/TransformStamped, that basically include a time stamp and the information of position and orientation (as a quaternion).

Besides broadcasting the position and orientation data, this node provides two services, one to manage the start and another one to control the publication frequency.

The ROS nodes for the sensorized gloves follow a similar procedure. In this case the data, read through a RS232 connection (for the left-hand glove) or a bluetooth connection (for the right-hand glove), is the set of hand joint values, either raw or calibrated. The data are broadcasted through two topics, named /cyberglove/raw/joint_states and /cyberglove/calibrated/joint_states, formatted using the standard ROS messages of type geometry_msgs/JointState.msg, that basically include a time stamp and a vector of floats to store the joint positions. Besides broadcasting the joint data, these nodes provide three services, one to manage the start, another one to control the publication frequency of the glove, and the last one to change the calibration file, in order to correct the data according to the particularities of the hand of each operator.

5.3 ROS Nodes for Robots and Hands

The ROS node of each robot provides an Action service to control de robot. The connection to the robot controller is done through Ethernet using the CS8Soap

library that allows to set the power on and off, to login and logout, to get the positions of the joints, to move the robot to a desired position, using either joint variables or the robot Cartesian position in the workspace.

Action clients and Action servers communicate following the ROS Action Protocol that define three messages, the goal, the result and the feedback. The Action service is started at the server side and each time the Action client sends a request (a goal), a function named execute is called. This function sends the desired position to the robot controller using the CS8Soap library. When the motion terminates the server responds with a boolean variable (the result) informing wether the motion has been executed successfully or not. The Action client may send a feedback request and the Action server responds with a message indicating the state of the joints (the feedback).

Besides the Action service, these nodes periodically publish their configurations through topics, named /robot_pose, formatted using the standard ROS messages of type sensor_msgs/JointState.msg, that basically include a time stamp and a vector of floats to store the joint positions.

The ROS node of each hand works in a similar way with respect to the Action services although now no topic communication is issued by this node. The communication with both the SDH hand and the SAH hand is done through RS232 using in each case the API of the corresponding drivers. Since the SDH hand has different operation modes, a service has been included in order to change between them.

6 Supervision Issues

6.1 ROS Nodes for Synchronization and Supervision

As shown in Table 3, each of the ROS nodes for the synchronization and supervision, named SinCybTrac (nodes #5 and #6), is responsible for:

- Getting the data from the glove and the tracker by subscribing to the corresponding topics with the information of the values of the joints of the operator hand and of the position and orientation of his/her wrist (since the data from each topic include a time stamp, this is used to synchronize the information of the glove with that of the tracker).
- Mapping the joint values of the operators hand to those of the mechanical hand, as detailed in Section 4.2.
- Mapping the position and orientation of the tracker to the position and orientation of the robot wrist, as detailed in Section 4.1.
- Performing the inverse kinematics, as detailed in Section 6.2, to obtain the joint values of the robot or to detect that the teleoperated goal is out of bounds or at a singular configuration.
- Verifying, if the inverse kinematics ended successfully, whether the mapped hand-arm configuration is in collision or not by sending a query to the collision-check node (#7), that tests for collisions as detailed in Section 6.3.

– Broadcasting a correct configuration using a topic, i.e. the last encountered configuration that was within bounds and collision-free.

In summary, nodes #5 and #6 are the key nodes in the ROS graph, i.e. they are at the core of the teleopeation process and are responsible for having an intuitive, simple and safe teleoperation of the dual hand-arm robotic system.

6.2 Management of Singularities

Inverse kinematics is needed to map the motions commanded in Cartesian space to the motions executed in the robot joint space. We implemented the Jacobian-based method available with the Kinematics and Dynamics Library (KDL). Problems arise at the singularities, those configuration of the manipulator that make the Jacobian matrix of the manipulator to loose rank. Some approaches minimize this problem by taking advantage of redundancy to assure large convex workspaces for singularity-free operation [14]; others try to find approximate solutions to the inverse Jacobian by means of complex robust control schemes based on damped least squares and dynamic weighting [15].

In this work we opted to limit the values of the joint velocities that become very large (which happens when the robot is in a pose very close to a singular configuration and when the desired direction requires the robot to approach the singular configuration). The pose obtained by these joint values differ from the commanded pose. This error is captured by the algorithm that blocks the communication (i.e. keeps publishing the same configuration). When the operator does the appropriate movement(one that does not cause large values of the inverse Jacobian), a re-synchronization is performed and the robot smoothly follows the operator, moving away from the singular configuration.

As an example, Fig 5(top) shows some snapshots of the execution of this supervision. When the commanded configuration makes the robot to reach the limits of the workspace, this is detected and the last configuration within limits is being kept published, thus the robot is stopped there (snapshot b). While the operator moves away from that configuration trying to move the robot further out of limits, the robot remains stopped (snapshot c). When the operator returns to a configuration within the workspace limits then the robot follows him/her again (snapshot d) after an automatic re-synchronization is performed.

6.3 Collision Detection

The remote cell has been modelled using The Kautham Project, the simulation and planning toolkit developed at our lab (sir.upc.edu/projects/kautham/ [16]). This software uses the PQP library for the collision detection. The modules for the modelling of the cell and for collision detection have been encapsulated as a ROS node that, as a service server, offers responses to collision-check queries. Collisions between a robot and the environment, or between the two robots, or possible autocollisions are detected in a very fast manner due to the simplicity and efficiency of the collision-check library used.

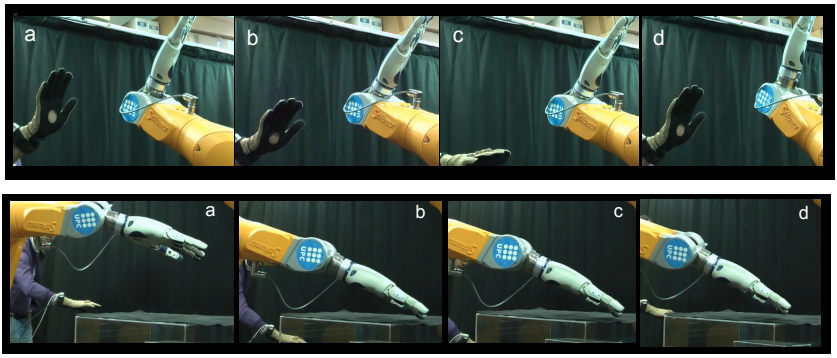


Fig. 5. Example of the supervision of the workspace limits (top) and of the collision detection supervision (bottom). The last commanded valid configuration (b) is being continually published while the operator tries unfruitfully to command an out-of-bounds or a collision configuration (c). When the commanded configuration is valid again (d) a re-synchronization is executed and the teleoperation resumes.

When the commanded configuration makes the robot to collide with the environment, it is detected and the last free configuration is being kept published, thus stoping the robot in a safe place. When the operator detects that the robot is not following him/her, he/she should return approximately to the position where the robot stopped. Once the commanded configuration is again in the free space, a re-synchronization is performed. This is done to assure smooth motions because this new free configuration may not exactly coincide with the last free configuration where the robot stopped.

As an example, Fig 5(bottom) shows some snapshots of an example of this supervision. When the commanded configuration makes the robot to collide with the table, this is detected and the last free configuration is being kept published, thus the robot is stopped there (snapshot b). While the operator keeps moving toward forbidden configurations, the robot remains stopped (snapshot c). When the operator returns to a configuration within the free space the robot follows him/her again (snapshot d) after an automatic re-synchronization is performed.

6.4 Simulation

As an additional support to the operator the visualization of the cell is optionally shown to the user (Fig. 1, bottom). The viewer of the The Kautham Project simulation and planning toolkit has been encapsulated as a ROS node that provides a service called move_robot that moves the robots and the hands to the configurations requested by the client, the Simulation_cell node that synchronizes the data from the two hand-arm sets, i.e. it builds a single message as the concatenation of the joint data of each of the hand-arm systems.

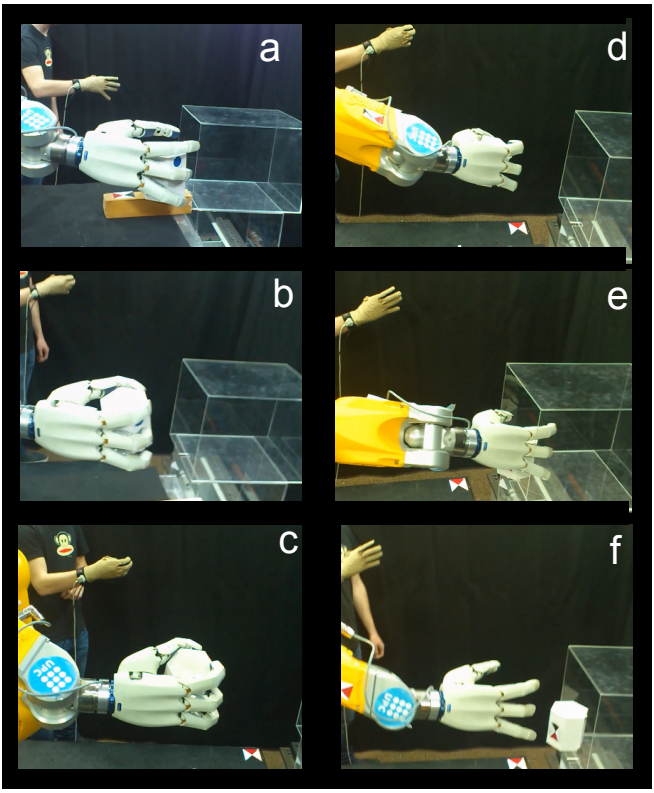


Fig. 6. Pick-and-place real task. A manual re-synchronization is performed at step c in order to command the robot in a more comfortable way.

6.5 Manual Re-synchronization

Fig 6 shows several snapshots of a pick-and-place teleoperated task. The operator must take a prism and place it inside a box. The execution includes a manual re-synchronization at step **c**, done to command the robot in a more comfortable way. This can be seen observing that the transformation between the human arm and the robot changes from the snapshots of the left column and those of the right one.

In order to do a manual re-synchronization the operator stops the teleoperation by pressing the clutch button on the tracker and then, using the same button, resumes the teleoperation with a more comfortable posture. The manual re-synchronization, as the automatic one, is done as described in Subsection 4.1.

7 Summary

The paper has presented the implementation of a teleoperation system for a dual hand-arm robotic system that includes two industrial robots and two mechanical

hands, one of them with anthropomorphic features, commanded with two sensorized gloves and two trackers attached to the wrists of the operator. The paper includes the description of: the mappings used to transform the information captured from the operator movements to the movements of the robotic system, the functional schema of the whole teleoperation system, and the particular modules developed for the communication and supervision issues. Future work includes the application of the approach to an anthropomorphic robotic system.

References

1. Basañez, L., Suárez, R.: Teleoperation. In: Nof, S. (ed.) Springer Handbook of Automation, pp. 449–468. Springer (2009)
2. Speeter, T.: Transforming human hand motion for telemanipulation. Presence 1(1), 63–78 (1992)
3. Rohling, R., Hollerbach, J., JAcobsen, S.: Optimized fingertip mapping: a general algorithm for robotic hand teleoperation. Presence 2(3), 203–220 (1993)
4. Peer, A., Einenkel, S., Buss, M.: Multi-fingered telemanipulation - mapping of a human hand to a three finger gripper. In: Proc. 17th IEEE Int. Symp. on Robot and Human Interactive Communication, pp. 465–470 (2008)
5. Rosell, J., Suárez, R., Rosales, C., Pérez, A.: Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures. Autonomous Robots 31(1), 87–102 (2011)
6. Pao, L., Speeter, T.H.: Transformation of human hand positions for robotic hand control. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1758–1763 (1989)
7. Hong, J., Tan, X.: Calibrating a vpl dataglove for teleoperating the utah/mit hand. In: Proceedings of IEEE Int. Conf. on Robotics and Automation, pp. 1752–1757 (1989)
8. Abe, K., Saito, H., Ozawa, S.: Virtual 3-D interface system via hand motion recognition from two cameras. IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans 32(4), 536–540 (2002)
9. Wachs, J.P., Stern, H., Edan, Y.: Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system. IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans 35(6), 932–944 (2005)
10. Infantino, I., Chella, A., Dindo, H., Macaluso, I.: Cognitive architecture for robotic hand posture learning. IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews 35(1), 42–52 (2005)
11. Tao Geng, M.L., Hülse, M.: Transferring human grasping synergies to a robot. Mechatronics 21(1), 272–284 (2011)
12. Pérez, A., Rosell, J.: An assisted re-synchronization method for robotic teleoperated tasks. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 886–891 (2011)
13. Colasanto, L., Suárez, R., Rosell, J.: Hybrid mapping for the assistance of teleoperated grasping tasks. IEEE Transactions on Systems, Man, and Cybernetics: Systems 43(2), 651–660 (2013)
14. Peer, A., Stanczyk, B., Buss, M.: Haptic Telemanipulation with Dissimilar Kinematics. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3493–3498 (2005)
15. Schinstock, D.: Approximate solutions to unreachable commands in teleoperation of a robot. Robotics and CIM 14(3), 219–227 (1998)
16. Pérez, A., Rosell, J.: A roadmap to robot motion planning software development. Computer Applications in Engineering Education 18(4), 651–660 (2010)

External Force Estimation for Telerobotics without Force Sensor

Enrique del Sol[1,2], Prithvi Pagala[1], Ryan King[2], and Manuel Ferre[1]

[1] Center of Automation and Robotics, Madrid, Spain
{ps.pagala,m.ferre}@upm.es
[2] Oxford Technologies Ltd, Abingdon, United Kingdom
enrique.delsol@oxfordtechnologies.co.uk

Abstract. This paper establishes an approach to external force estimation through the use of a mathematical model and current sensing, without employing a force/torque sensor. The advantages and need for force feedback have been well established in the field of telerobotics. This paper presents the requirement for sensorless force estimation and comparative results between a force sensor and the presented approach using an industrial robot. The approach presents not only a cost effective solution but also a solution for force sensing in hazardous environments, especially ionizing radiation prone environments where the dose rates limit the use of sensing equipment. The paper also discusses the applications and advantages presented by this work in various fields.

Keywords: Industrial robot, sensorless, remote handling, force feedback, master-slave system.

1 Introduction

In teleoperation a human operator manipulates master device, and a slave device follows the motion while manipulating in a remote environment. Providing the operator with various information regarding the remote environment like position, orientation, contact, load, forces and others; improves the task performance and the operator understanding of the environment. This information can be viewed on display screens [1] but, it is more intuitive when provided directly, by reflecting the measured parameters like positions and torques to the master haptic device. When the operator is interacting with the slave with a haptic master then the operator is said to be kinesthetically coupled to the environment. The task being performed is said to be bilateral controlled teleoperation [2], [3]. The continued advances in these various fields of control, communications, haptic systems and others have made possible to have an integrated robotic master slave system that it is able to aid the human operator in effective task execution.

The teleoperation slaves are generally controlled using bilateral control algorithms. Bilateral control algorithms main goals are stability and transparency. Stability assures expected system response for the teleoperation task. It ensures stability and prevents hazards on the master and slave sides of the teleoperation. Transparency is said to be

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_45,

achieved when the human operator interacting with the master device feels as if present in the remote environment. Which means that, the human operator movements are mimicked by the slave in the remote environment and the reaction force from the remote environment is applied to the operator [4].

Haptic devices used in telerobotics are force exerting mechatronic designs to ensure the human operator experiences an immersive interaction with the remote environment. Generally, kinesthetic haptic interfaces not only exert forces to the operator, but behave as bidirectional channel to exchange forces and interactions. In recent years, haptic interfaces have advanced in various interfaces and towards cost effectiveness, like the creation of commercialized equipment [5]. They have been used for several applications in different fields such as telerobotics [6-7], medical surgery [8-10] and others. Dissimilar master-slave systems have been coupled with the use of scaling and other adaptation [11] methods.

This paper will be focussed on hazardous environments with ionising radiation as they present critical need for robot deployments to reduce human intervention. These environments are primarily nuclear facilities and large scientific facilities focused on nuclear research.

Nuclear facilities [12] have continuously used robot deployments since the first developments carried out by Ray Goertz for the U.S. Atomic Energy Commission [13]. Large scientific experiments like CERN (European Organization for Nuclear Research) [14] and JET (Joint European Torus) [15],[16] are also deploying telerobotic solutions in sections where the ionising radiation and hazardous conditions make it difficult or impede human intervention. Ionising radiation hazard depends on the location, dose rates and time elapsed. Some of the robot deployments at CERN for maintenance are, the autonomous source storage robots in the ISOLDE facility [17], the teleoperated TIM robot [18] and Mantis mobile platform [19]. While at CERN and JET complete remote handling solutions for all the tasks were not considered during the design phase but, at ITER (International Thermonuclear Experimental Reactor) [20][21] it is crucial to have complete remote handling solutions due to the hazardous environment.

Traditionally, most of the slaves used in remote handling on radioactive or hazardous facilities have been specifically designed for teleoperation tasks where dexterity is essential, i.e. Mascot manipulators used at JET and CERN [12], the Bilateral Servo Manipulator (BSM) deployed in Japan for the Tokai Vitrification Facility (TVF) and the Recycle Equipment Test Facility (RETF) [22]. These teleoperation slaves are characterized for being easily backdrivables to follow the operator movements smoothly. On the other hand, the robots used for autonomous tasks are industrial robots designed for repetitive tasks where the adaptability for changing from one kind of operation to another could result in lengthy preparation. Industrial robots are usually heavy manipulators with high reduction rate and friction in the gears that make them non-backdrivable or simply the mechanism is non-backdrivable itself. With the large number of industrial robots existing in the world in comparison with dexterous manipulators and their relative lower price, a straight forward adaptation to acquire some characteristics of the custom designed manipulators is desirable. Nevertheless this approach has not been common and an insignificant number of industrial robots have been adapted for teleoperation tasks. This has been done previously by attaching high

proficiency force and torque sensors in the robotic end-effectors [22] for teleoperation tasks and during a shared control study [23].

Two requirements to further increase the use of teleoperated robots during maintenance of hazardous facilities are presented in section 2. Section 3 explains the hypothesis and the preliminary results of the approach. Section 4 details the setup used when applying this research on an industrial robot. In section 5 the mathematical model and the approach is explained. Results of the approach from the industrial robot are compared with the results from force torque sensor are presented in section 6 and in section 7 conclusions and future developments are presented.

2 Requirements

Towards reducing human intervention and as low as reasonable achievable (ALARA) safety measures, more remote handling solutions are explored. A working problem and need has been seen in large scientific facilities where remote handling under ionizing radiation is necessary.

2.1 Force Estimation for Teleoperation of Industrial Robots

As discussed above, the remote handling interventions in hazardous facilities have been typically carried out with low reduction gears and a low weight-to-payload ratio slave manipulator. This is done to achieve force reflection capability. This force reflection is conveyed to the operator using the bilateral control system between master and slave. The positional feedback and the backdrivable design of the manipulator [24] made it possible for the first bilateral control architectures. The robot reflected the position of the operator and the environment making it more secure system for a robot sharing the environment with humans or interact with remote objects. Newer control techniques as force-position control schemes transmit the environmental forces to the operator along with position and other parameters. The force is acquired by specially designed force and torque sensors [25-26] or sensorless force feedback approach determining them from the actuator model like it is showed in [27] where the nominal parameters of a 1 degree of freedom system are used to control a slave in position and acceleration. Both [28] and [27] propose a sensorless approach called disturbance observer to control each robotic joint independently in acceleration and position modes. They differ from the approach presented here since the external forces and parameters variation like inertia are estimated together for control purposes. In these researches there is also no external force determination. Independently of the method used to estimate the forces, the simplest algorithm to teleoperate a non-backdrivable slave is the force-position algorithm described in Fig. 1.

2.2 Force Sensors in Radioactive Environments

Another need for such a force sensorless system is, to be able to obtain force feedback information without the need of designing new force sensors due requirement of size

and application in the remote environment. The findings summarized by Keith E. Holbert et al. in [29] during their performance study of commercial off-the-shelf microelectromechanical (MEMS) systems sensors in a radioactive hazardous environment, shows the limitations imposed by radioactivity over pressure transducers based on MEMS technology. Hence, depending on the different hazardous environments and robot tasks there is a need to redesign sensors due to size and environment requirements. An example of which is the development of hard-rad ATI force sensor used in the AREVA recycling plant [25] robot deployment.

The ability to have force information from the remote environment without a force/torque sensor is beneficial in terms of understanding the remote environments force interactions. Along with no additional large cost, size and development time to adapt the robot with a force senor. Therefore, these requirements are the motivation behind the presented approach of force estimation without the use of force sensor for application in telerobotics.

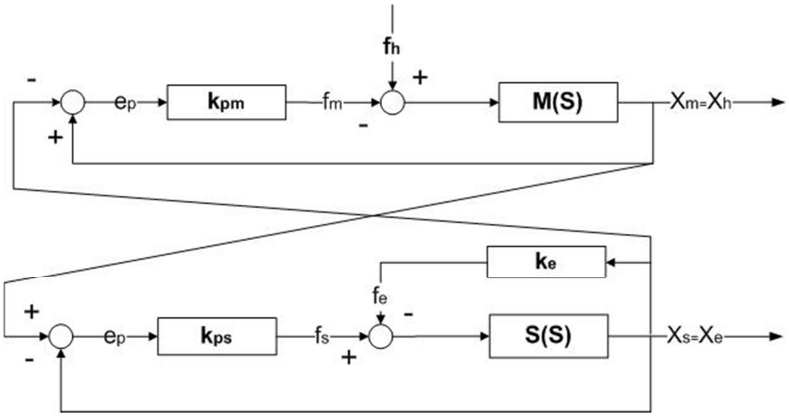


Fig. 1. Force-Position control algorithm

3 Hypothesis

As the forces and torques applied on the master are proportional to those applied to the slave in a bilateral control using force channel, the estimation of the robot end effector torques and forces can be accomplished by means of modelling the robot dynamics. Equation (1) describes the relation between forces and torques on the end effector and joint torques on the robot and equation (2) is the well-known equation of robot dynamics.

$$\tau_m = J^T \cdot T \tag{1}$$

Where,
τ_m: vector of motor torques exerted in each joint.
J: is the robot jacobian.

T: is the vector of forces and torques ejected in the robot end effector and expressed in the base coordinates system.
D: is the robot inertia matrix.
H: is the Coriolis forces vector.
C: is the gravity forces vector.
τ_f: is the friction torques vector.
τ_{ext}: is the external torques on each joint produced by external forces on the end effector.

$$\tau_m = D(q)\cdot\ddot{q} + H(q,\dot{q})\cdot\dot{q} + C(q) + \tau_f(\dot{q}) + \tau_{ext} \tag{2}$$

By combining (1) and (2) the external forces at the tip result on (3).

$$T_{ext} = (J^T)^{-1}\cdot\left(\tau_m - D(q)\cdot\ddot{q} - H(q,\dot{q})\cdot\dot{q} - C(q) - \tau_f(\dot{q})\right) \tag{3}$$

We hypothesise that in electric motors torque can be correctly estimated with equation (4). Based on the current amplitude of each actuator and the mathematical model it is possible to determine the external forces excreted on a robot with multiple degrees of freedom.

$$\tau_m = K_e \cdot I_a \cdot G \tag{4}$$

Where,
K_e : is the motor torque constant in [N/A].
I_a: is the current amplitude in [A].
G: is the gear ratio for each joint.
If the motor currents are sinusoidal the instantaneous amplitude of the waveform can be calculated with the following expression (5) detailed in [30] where *ia* and *ib* are the phase *a* and *b* instantaneous currents respectively.

$$I_a = \left| \frac{ia}{\sqrt{2}\cdot\sin\left(\operatorname{atan}\left(\frac{-\sqrt{3}\,ia}{2ib+ia}\right)\right)} \right| \tag{5}$$

3.1 Preliminary Setup

Fig. 2 shows the preliminary setup used to verify the first hypothesis. It was composed by:

- 1x Aerotech® S-50-86 AC motor.
- 2x TH3A Hall effect current sensors
- 1x RWT410 series manufactured by Torquesense®
- 1x National Instruments NI-USB 6212, 16-Bit resolution and 400 kS/s data acquisition.
- 1x pulley setup with different weights available.

The current sensors were calibrated and placed in shielding to avoid electromagnetic interference and noise. The test rig used in this setup is able to support the motor in a free axis movement as well as allow the weight lifting of several weights with a pulley of 32 mm in radius.

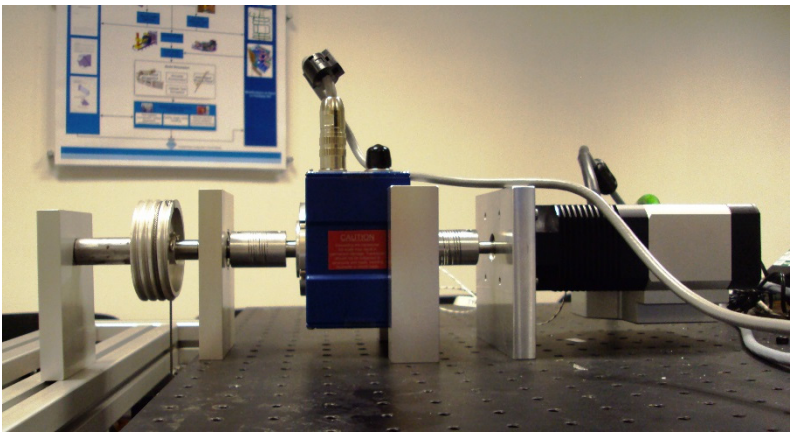


Fig. 2. AC Motor test bench with torque transducer

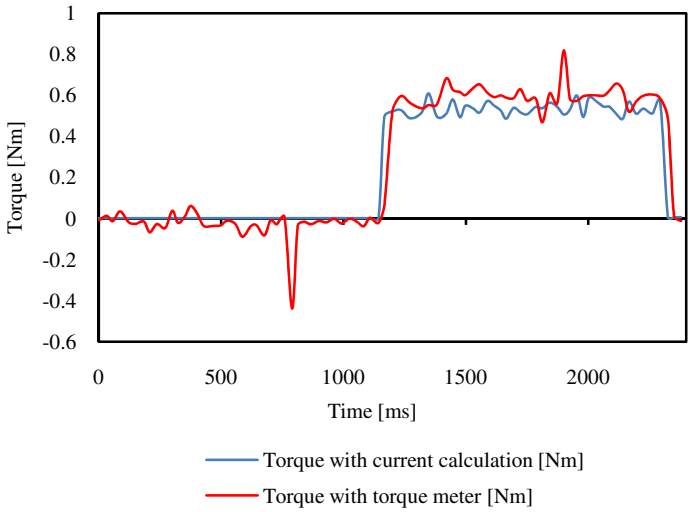


Fig. 3. Single DOF torque experiment

3.2 Preliminary Results

Tests were carried out in the mentioned test rig, lifting varying weights by means of coupling the pulley directly to the motor axis. Fig. 3 corresponding to the external

torque calculated via measuring the two phase currents of the motor and applying the expression in (5) and the measurement with the torque meter inserted between load and motor axis. The calculated torque is proportional to its correspondent torque meter curve and in this case has less error than the torque meter. It indicates the precision acquired with this method as the torque calculation is performed using the instantaneous current data acquisition. The small delay between the two curves is caused by the time that the cabling weight takes in acquiring the necessary tension to lift the weight.

4 Setup

The experimental setup is composed by the following elements:

- 1 x ABB IRB 2400-16 industrial manipulator, Fig 4.
- 1 x ABB SC4+ robot controller able to interface with RS422.
- 1 x NI-USB 6212, 16-Bit resolution, 400 kS/s.
- 1 x EMI shielded measurement box specifically designed for this purpose shown in Fig. 4.
- 1 x Force/Torque sensor, ATI, Gamma SI-130-10.

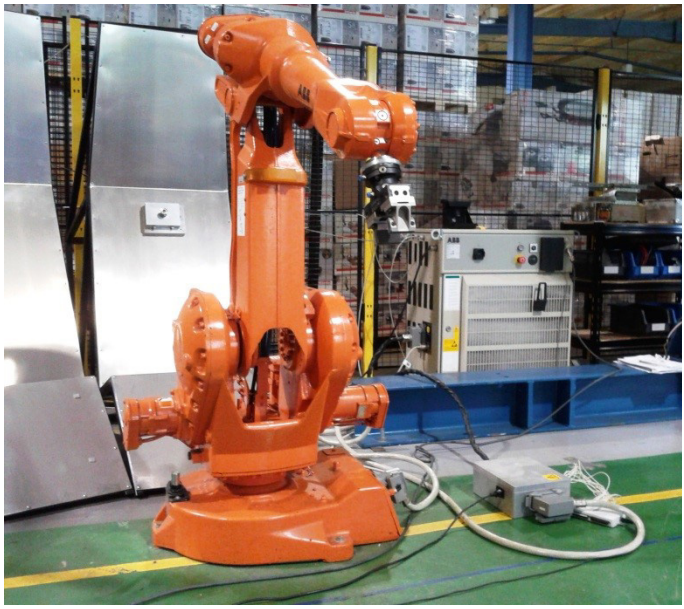


Fig. 4. ABB IRB 2400-16 and data acquisition setup

The ABB IRB 2400-16 is an industrial robot with a payload of 20 kg and a reach of 1.55 m. It is equipped with 6 axis driven by PMSM actuators.

In order to measure the instantaneous current consumption of each actuator an EMI shielded measurement system was designed to capture the outgoing current from the

robot controller to the manipulator. This measurement board is composed by 12 Hall Effect sensors, two for each motor. The current sensors used on this experiment are the inexpensive TH3A for joints 1, 3, 4, 5 and 6 and TH5A for joint 2 with nominal input currents of 3 and 5 A respectively. The output signal of each sensor was sampled at 1 KHz by means of the NI-USB 6212.

5 Experiment Design

5.1 Dynamic Modelling

The Newton-Euler or Lagrangian method is typically used to derive the dynamic equations of kinematic chains of rigid bodies [31]. Both approaches yield the equation described in. (6), which is basically (2) without the friction and external forces terms. τ is an *Nx1* vector expressing the joint torque vector necessary to move a robot with *N* dof with the dynamics described by the right part of the equation.

$$\tau = D(q)\cdot\ddot{q} + H(q, \dot{q})\cdot \dot{q} + C(q) \tag{6}$$

The barycentric parameters [32] or the modified Newton-Euler method [31] can be used to yield a model of the form:

$$\tau = f(q, \dot{q}, \ddot{q}) \tag{7}$$

It is linear in the inertial parameters. Particularizing for a unique point in a certain trajectory that forms results into the following equation:

$$\tau_i = A_i \cdot \phi_i \tag{8}$$

Where τ_i is an *n x 1* vector, A_i is an *n x 10n* matrix and ϕ_i is an *n x 1* vector. For linear models it is possible to apply a parameter estimation by least mean squares by stacking the equation (8) with *P* data points [33] .and obtaining the vector form (9):

$$\tau = A \cdot \phi \tag{9}$$

With:

$$\tau = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_p \end{pmatrix} \text{ and } A = \begin{pmatrix} A_1 \\ \vdots \\ A_p \end{pmatrix} \tag{10}$$

Where τ is now an *nP x 1* vector and *A* is now *nP x 10n* matrix. Determining the joint torques with the method mentioned previously allows an estimation of the unknown parameters with the general solution for least mean squares:

$$\phi = \left(A^T A\right)^{-1} A^T \tau \tag{11}$$

Unfortunately it is not possible to always apply simple least squares estimation since A^TA is not invertible due the loss of rank from restricted degrees of freedom at the proximal links and the inability to measure the forces and torques in every direction [33]. Two main methods are used to cope with the loss of rank, parameters elimination and damped least squares. The parameters elimination technique discards the non-identifiable parameters indicated by zero or small singular values of the regressor matrix. This procedure is not necessary when having an initial parameter estimation ϕ_0 obtained from the CAD model or by different means. The damped least squares leads to a solution by:

$$\phi=(A^TA+\lambda^2 I)^{-1}A^T\tilde{\tau} \tag{12}$$

Where λ is the damping factor and $\tilde{\tau}=\tau-A\phi_0$. The damping factor modifies the singular values and cancels out the effect of very small values. The initial solution will be perturbed over the normal least-squares solution.

5.2 Slave Characterization and Parameters Determination

A dynamic robot model which relates robot motion to joint torques, describes the rigid-body motion of the robot. This model includes the joint friction composed by Coulomb and viscous friction in the joints and inertial parameters for each link. The inertial parameters are the link mass, centre of mass coordinates and inertias with respect to each link frame: *m, cx, cy, cz, Ixx, Iyy, Izz, Ixy, Ixz, Iyz.* As this a priori information was not provided by the manufacturer, the parameters were calculated by using a CAD 3D model representing the robot. A uniform density solid virtual robot was created in order to estimate the mass of each link, the centre of gravity coordinates with respect each joint origin and the inertia matrix. These results are shown in Table 1. The DH parameters were established based on the diagram of Fig. 5.

Tests were performed in order to estimate the torque constant for each joint as they are not usually provided by the manufacturer. The procedure defined in order to estimate these constants was general for every joint but with some differences due the various configurations presented and the non-equal gravity effect on each one. Determining the torque constant allowed calculating the motor torque in real time.

Experimental tests were carried out to determine the friction curves for each actuator which provided a friction torque that was subtracted from the initial motor torque to perform the robot parameters identification. Several elastic impact tests were performed at four different poses and three different speeds with the objective of comparing the external force estimated by the proposed method and the measured force by the ATI force sensor. An elastic interface attached to the robot end effector, based on a spring with elastic constant of 2.52 N/mm and a free length of 125 mm was used to provide an incremental external force.

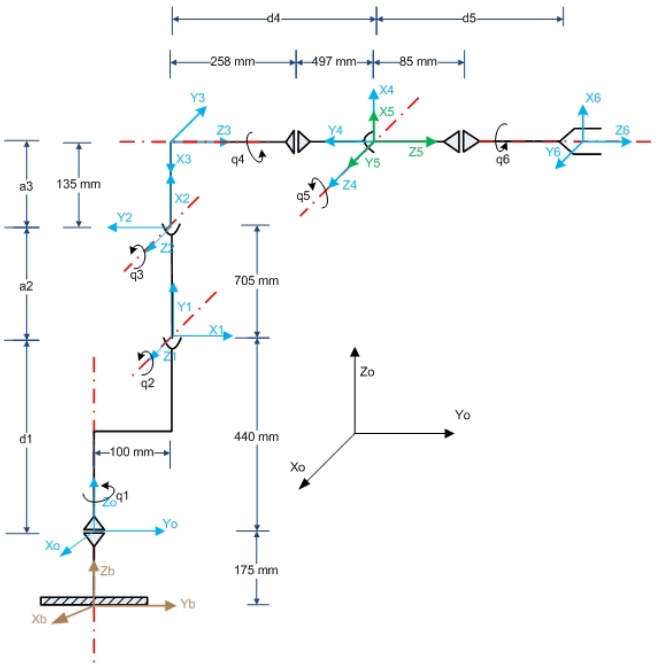


Fig. 5. ABB IRB 2400 links diagram

Table 1. Links parameters for links 1 to 6

	Link 1	Link 2	Link 3	Link 4	Link 5	Link 6
M [kg]	146.1	44.52	46.02	17.89	0.978	0.012
Cx [m]	-69.52	-383.4	95.14	0.301	0.956	0.015
Cy[m]	-169.6	-50.11	-14.9	198.4	0.039	-0.01
Cz[m]	-20.7	-0.493	-14.3	-1.01	-0.24	0.032
Ixx[kg·m^2]	11.84	0.242	1.043	0.519	-0.06	-0.002
Iyy[kg·m^2]	6.571	1.784	3.408	0.043	-0.95	-0.06
Izz[kg·m^2]	9.875	1.758	2.726	0.510	-0.89	-0.03
Ixy[kg·m^2]	-2.746	-0.058	0.119	3E-4	0.036	0.01
Ixz[kg·m^2]	-0.51	-0.005	0.329	4E-5	-0.23	-0.1
Iyz[kg·m^2]	-0.58	0.001	-0.03	-3E-3	-0.01	-3E-4

Table 2. Poses used for experiment on fig. 6

Pose	Q1 [deg]	Q2 [deg]	Q3 [deg]	Q4 [deg]	Q5 [deg]	Q6 [deg]
A	-94.2	10.6	22.7	19.9	12.4	81.4
B	-93.5	25.5	26.8	25.6	8.4	75.9

6 Results

The results presented in Fig. 6 evaluate the model accuracy when calculating the external perpendicular force by using (3) for the trajectory defined between pose A and B and returning to A, shortened for sake of clarity. These points are presented in Table 2. The force prediction error is small except when the velocity changes its sense, as it was expected due the uncertainty of the friction curve at low speeds and the low frequency available for the positional feedback due the controller limitations in comparison with the rate of the actuators current readings.

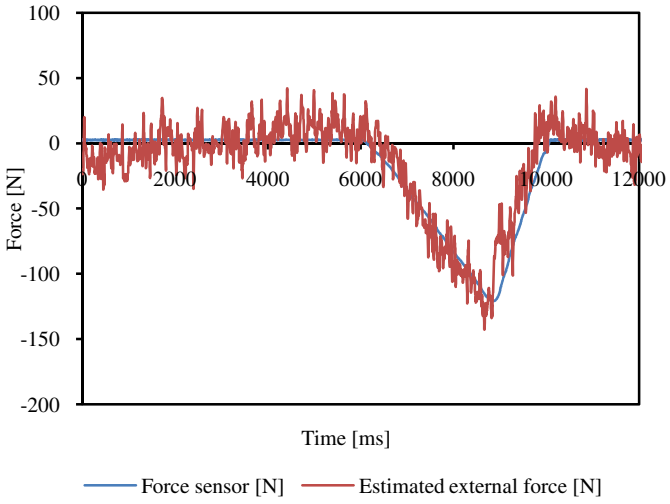


Fig. 6. Model validation based on external force estimation by experiment

An increase in the error of the torque calculations when the motor changes its sense has also been detected which is produced by the phase order change of the currents which creates a non-sinusoidal transitional wave prejudicial for the amplitude calculation described by equation (5).

It has been found than perpendicular forces can be calculated with an average error of 2.47 kg which means 12.35% of the payload. Hence, validating the proposed approach and differentiating it from the state of art presented.

7 Conclusion

The facilities where maintainability and measuring tasks have to be executed in a radioactive or hazardous environment will likely increase the application of remote handling solutions due to the safety measures like ALARA, reducing the human interaction with hazardous environments. Force feedback would be essential to perform remote handling and maintenance. Since backdrivable slaves and torque sensors are not cost-effective an alternative approach has been proposed which estimates the external forces and torques with an acceptable level of accuracy by using the robot model and current information. This method does not require either any modification of the robot or additional wiring but only current sensing at the controller output. It can be employed not only as a substitution of the conventional sensors but also as a redundant solution when other methods are preferred. The applications of the proposed method are not limited to hazardous environments but can also be applied to robot solutions where it is difficult to add a sensor at the end-effectors, like medical telesurgery, and where there is a need to design environment and size specific force sensors. This approach can be extended to other robot manipulators and future work would focus on implementing it on light weight and hydraulic manipulators.

Acknowledgements. This research project has been supported by a Marie Curie Early Stage Initial Training Network Fellowship of the European Community's Seventh Framework Program under contract number PITN-GA-2010-264336-PURESAFE.

References

1. Kotoku, T.: A predictive display with force feedback and its application to remote manipulation system with transmission time delay. In: lEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 239–246. IEEE Press, New York (1992)
2. Hecht, D., Reiner, M.: Sensory dominance in combinations of audio, visual and haptic stimuli. Experimental Brain Research 193(2), 307–314 (2009)
3. Ernst, M.O., Banks, M.S.: Humans integrate visual and haptic information in a statistically optimal fashion. Nature 415(6870), 429–433 (2002)
4. Yokokohji, Y., Yoshikawa, T.: Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment. IEEE Transactions on Robotics and Automation 10(5), 605–620 (1994)
5. Salisbury, J.K., Srinivasan, M.A.: Phantom-based haptic interaction with virtual objects. IEEE Computer Graphics and Applications 17(5), 6–10 (1997)
6. Škorc, G., Zapušek, S., Čas, J., Šafarič, R.: Virtual user interface for the remote control of a nano-robotic cell using a haptic-device. Strojniški vestnik Journal of Mechanical Engineering 56(7-8), 423–435 (2010)
7. Peer, A., Buss, M.: A new admittance-type haptic interface for bimanual manipulations. IEEE/ASME Transactions on Mechatronics 13(4), 416–428 (2008)
8. Waldron, K.J., Tollon, K.: Mechanical characterization of the immersion corp. haptic, bimanual, surgical simulator interface. In: Siciliano, B., Dario, P. (eds.) Experimental Robotics VIII. STAR, vol. 5, pp. 106–112. Springer, Heidelberg (2003)

9. Okamura, A.M.: Methods for haptic feedback in teleoperated robot-assisted surgery. Industrial Robot: An International Journal 31(6), 499–508 (2004)
10. McMahan, W., Gewirtz, J.: Tool contact acceleration feedback for telerobotic surgery. IEEE Haptics 4(3), 210–220 (2011)
11. Colgate, J.E.: Power and impedance scaling in bilateral manipulation. In: Colgate, J.E. (ed.) IEEE International Conference on. Robotics and Automation, pp. 2292–2297. IEEE Press, New York (1991)
12. Nof, S.Y.: Handbook of industrial robotics, vol. 1. Wiley (1999)
13. Hokayem, P.F., Spong, M.W.: Bilateral teleoperation: An historical survey. Automatica 42(12), 2035–2057 (2006)
14. Horne, R.A., Coin, A.Y., Pusenius, M., Hendseth, S., Kallevik, V.: Extended tele-robotic activities at CERN (1991)
15. Rolfe, A.C., Brown, P., Carter, P., Cusack, R., Gaberscik, A., Galbiati, L., Haist, B., Horn, R., Irving, M., Locke, D.: A report on the first remote handling operations at JET. Fusion Engineering and Design 46(2), 299–306 (1999)
16. David, O., Loving, A.B., Palmer, J.D., Ciattaglia, S., Friconneau, J.P.: Operational experience feedback in JET Remote Handling. Fusion Engineering and Design 75, 519–523 (2005)
17. Kugler, E.: The ISOLDE facility. Hyperfine Interactions 129(1-4), 23–42 (2000)
18. Kershaw, K.: Remote Inspection. Measurement and Handling Applied to Maintenance and Operation at CERN Introduction (October 2012)
19. Horne, R.A., Lohmann, K.D., Coull, L., Coin, A.Y., Therville, A., Lips, R., Desrozier, M.: MANTIS: a compact mobile remote handling system for accelerator halls and tunnels. ANS Meeting on Remote Systems and Robotics in Hostile Environments 30 (1978)
20. Honda, T., Hattori, Y., Holloway, C., Martin, E., Matsumoto, Y., Matsunobu, T., Suzuki, T., Tesini, A., Baulo, V., Haange, R.: Remote handling systems for ITER. Fusion Engineering and Design 63, 507–518 (2002)
21. Tesini, A., Palmer, J.: The ITER remote maintenance system. Fusion Engineering and Design 83(7), 810–816 (2008)
22. Lee, J.K., Kim, K., Park, B.S., Yoon, J.S.: Force-reflecting servo-manipulators for remote handling task in a radioactive environment, in Control. In: International Conference on Automation and Systems, ICCAS 2007, pp. 1025–1028 (2007)
23. Dumora, J., Geffard, F., Bidard, C., Brouillet, T., Fraisse, P.: Experimental study on haptic communication of a human in a shared human-robot collaborative task. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5137–5144. IEEE Press, New York (2012)
24. Fischer, P., Daniel, R., Siva, K.V.: Specification and design of input devices for teleoperation. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, pp. 540–545 (1990)
25. Piolain, G., Geffard, F., Coudray, A., Garrec, P., Thro, J.F., Perrot, Y.: Dedicated and standard industrial robots used as force-feedback telemaintenance remote devices at the areva recycling plant. In: 1st International Conference on Applied Robotics for the Power Industry (CARPI), pp. 1–6 (2010)
26. Ferre, M., Buss, M., Aracil, R., Melchiorri, C., Balaguer, C.: Advances in Telerobotics. Springer, Heidelberg (2007)
27. Katsura, S., Matsumoto, Y., Ohnishi, K.: Realization of 'Law of action and reaction' by multilateral control. IEEE Transactions on Industrial Electronics 52(5), 1196–1205 (2005)
28. Nakao, M., Kouhei, O., Miyachi, K.: A robust decentralized joint control. In: IEEE Proceedings on Robotics and Automation, pp. 326–331. IEEE Press, New York (1987)

29. Holbert, K.E., Heger, A.S., McCready, S.S.: Performance of Commercial Off-the-Shelf Microelectromechanical Systems Sensors in a Pulsed Reactor Environment. In: IEEE Data Workshop in Radiation Effects (REDW), vol. 8. IEEE Press, New York (2010)
30. Del Sol, E., Scott, R., King, R.: A sensorless virtual slave control scheme for Kinematically dissimilar master-slave teleoperation. In: HOTLAB (2012)
31. Atkeson, C.G., An, C.H., Hollerbach, J.M.: Estimation of inertial parameters of manipulator loads and links. The International Journal of Robotics Research 5(3), 101–119 (1986)
32. Fisette, P., Raucent, B., Samin, J.C.: Minimal dynamic characterization of tree-like multi-body systems. Nonlinear Dynamics 9(1-2), 165–184 (1996)
33. Siciliano, B., Khatib, O.: Springer handbook of robotics. Springer (2008)

Part XVII

Grasping and Dexterous Manipulation

Contact Detection and Location from Robot and Object Tracking on RGB-D Images

José A. Bernabé, Javier Felip, and Antonio Morales

Department of Computer Science and Engineering, Robotic Intelligence Laboratory,
Universitat Jaume I, 12006 Castellón, Spain
{jbernabe,jfelip,morales}@uji.es

Abstract. Detecting contacts between a robot and the objects on the environment during manipulation activities is challenging problem even when tactile or other contact sensors are available. There are situations in which due to the location or sensitivity of the sensors contacts can be missed. In those cases a complementary sensor that could infer and locate the existence of contact from visual input in the absence of direct touch information would be of great interest.

In this paper, we implemented a method which uses the visual tracking of objects using ICP (Iterative Closest Point) to detect small movements of the object's RGB-D point-cloud which has been previously segmented. Once a movement has been detected, a combination of a 3D occupancy grid of the object and an sphere-based model of the robot is used to probabilistically estimate the locations of contacts. The proposed approach is implemented and experimentally validated for several relevant cases.

1 Introduction

The robot manipulation and grasping of objects on service scenarios is a complex task as a consequence of the uncertainty which appears in such conditions. The limited or complete absence of knowledge about the objects to be manipulated, the limited accuracy of the sensors used to perceive the scene, and the difficult calibration of the kinematics of complex manipulators are the most relevant difficulties. Manipulation planners and controllers must take into account for these uncertainties in their design and implementation.

In such grasping systems, the ability to perceive and detect the contact between the robot and the objects becomes a critical feature to ensure the security and robustness of the execution. The detection of contacts or, in a more broad definition, the sense of touch have been achieved by integrating in the robot hands and arms a variety of contact and proximity sensors. It is not unusual that the most advanced robotic hands include some type of contact sensors. This is the case of the *Shadow Dexterous Hand* [1], the *DLR-HIT-Hand* [2] and many others.

Several technological approaches have been developed to provide the sense of touch. Three main categories can be distinguished depending on their configuration and characteristics. The first one is tactile sensing which aims to imitate the sensitivity of the human skin [3,4]. It usually consists of a single or an array of cells placed on the surface

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_46,

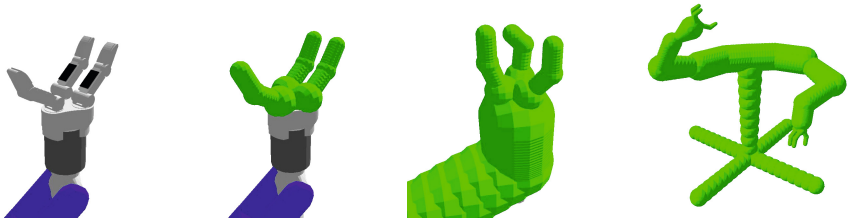


Fig. 1. From left to right: model of a Barrett Hand; spherical model of the fingers; spherical model of the whole hand; and spherical model of the whole manipulator

of a body, which measure the existence of contact and the pressure or related magnitudes in such locations. The main advantage of these sensors is that they are able to determine the location of the contact accurately [5].

A second category is composed by sensors that measure strain or force/torque. They are devices which interface two different bodies and measure the forces and torques transmitted between them. When a contact occurs on one of the bodies, it produces a force or torque that is transmitted to the rest of the connected bodies and thus can be detected by the sensor[6,7]. These sensors can be considered global contact sensors since they are potentially able of detecting any contact on a body, but can hardly identify accurately the location and magnitude of the contact forces involved. Finally, the third category is composed by proximity sensors. Although they are not properly contact sensors, they have been used to identify imminent contacts and thus used in a similar fashion as touch sensors [8]. Their main advantage is that they can be used without perturbing the state of the objects.

Although these approaches have extensively been used, they are not free of limitations. Tactile sensors are unable to detect contacts in surfaces which are not covered by them. Force-torque sensors cannot detect locations of contacts and their sensitivity is low when the contact forces are small. In addition, all these sensors require a wiring and a system integration which causes their implementation difficult. As a consequence, even in heavily sensorized hand-arm systems, it is common that a contact goes unnoticed which compromises the stability of the grasping and manipulation actions.

This paper describes a novel approach which uses vision to detect and analyse contacts. It is based on the basic assumption that if an object moves is because it has been touched. We present a system that simultaneously tracks the robot actuator and the object on the scene. When a movement on the object is detected, it analyses the occupancy information of the object and the robot to infer the probable location of the contact in the surface of the robot.

2 Description of the System and Underlying Assumptions

Our approach assumes an scenario in which a robot system is composed of at least a robot arm and a hand. Its purpose is to approach the hand and manipulate a single rigid object lying on a planar surface. No previous model of the object is available and no assumptions about its shape or aspect are made.

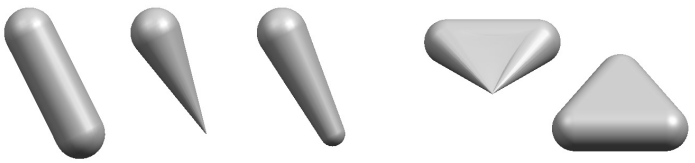
The only sensor modality that our approach is going to use is a *Kinect* sensor which delivers RGB-D images and point clouds describing the scene. Other sensor modalities like tactile sensors are used exclusively for validation purposes in the experimental section (see Sec. 7). Assumptions are made that the kinematics of the arm and hand are known and the correspondence between coordinates of the scene point-cloud and the frame of the robot arm are calibrated accurately.

3 Geometrical Modelling of Robot System

In our approach, we use a geometric model of the robot system to reason about the space occupied by the robot and estimate contacts with objects, especially when the robot is occluded in the RGB-D image.

We have chosen a model based in bounding volume primitives to describe our robot system, in particular we choose the spherically extended polytopes, *s-topes*, as bounding volumes. This representation has been widely used [9,10,11] because of their efficiency in distance computation, specifically in collision detection and path planning. An *s-tope* [12] is the convex hull of a finite set of spheres $s \equiv (c,r)$, where c is the center and r is its radius. Given the set of n spheres $S = \{s_0, s_1, ..., s_n\}$, the convex hull of such a set, S_s, contains an infinite set of swept spheres expressed by Eq. 1 .

$$S_s = \left\{ s : s = s_0 + \sum_{i=0}^{n} \lambda_i (s_i - s_0), s_i \in S, \lambda_i \geq 0, \sum_{i=0}^{n} \lambda_i \leq 1 \right\} \tag{1}$$



(a) s-tope with two spheres, bi-sphere (b) s-tope with tree spheres, tri-sphere

Fig. 2. Examples of simple s-topes: bi-spheres and tri-spheres

Where λ_i is the parameter that determines a specific sphere, radius and center, of the whole set of spheres. To illustrate the previous equation, Figure 2 depicts several examples of s-topes defined by two (*bi-spheres*) and three spheres (*tri-spheres*).

We have modeled our robot as a combination of s-topes. Each link is represented as a bi-sphere and some static parts as singles spheres. In addition, each defining sphere has been attached to the corresponding frame of the kinematic chain. Figure 1,right depicts the complete model of our robot manipulator system and a detail with the model of our three-fingered robot gripper.

4 Object Segmentation

This section describes the identification and the segmentation of the target object in the scene. As it has been described previously, we assume that the scene contains a single object, lying on a planar surface. The scene is obtained by a RGB-D camera that provides a 3D point cloud. This point cloud contains points which belongs to the object, the supporting table and the robot manipulator. It is necessary a procedure to segment the object and isolate the 3D points belonging to it.

The fist step is to transform all points in the point cloud from the camera frame to the robot base frame. Then, several filters are applied to remove points from the point cloud that do not belong to the object. The first one removes the points belonging to the supporting plane. It consists of a predefined frame box which discards all the points outside the box. The dimensions of this box has been calibrated to fit exactly the dimensions of the table in our scenario. This filter keeps all the points in the scene that are over the table, containing the object and the robot manipulator.

The second filter uses the spherical model of the robot to determine which points in the point cloud belong to the robot. A point is considered to belong to the robot if the distance between it and the model is less than zero. This distance is the minimum from the point to all the s-topes which compose the robot model. Since our geometric model is composed only of spheres and bi-spheres, we need to apply only two rules to compute each distance. In the cases of a single sphere, the distance between a point p_i and the sphere $s_i \equiv (c_i, r_i)$ is computed using Eq.2, where c_i is the centre and r_i the radius of the sphere:

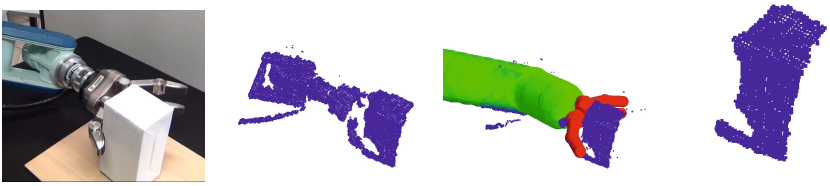
$$distance = \|\overrightarrow{p_i} - \overrightarrow{c_{min}}\| - r_{min} \tag{2}$$

In the case of the distance between a point p_i and a bi-sphere, we first need to determine the closest sphere to the point among the infinite number which define the bi-sphere. Given a bi-shpere defined by the spheres $s_1 \equiv (c_1, r_1)$ and $s_2 \equiv (c_2, r_2)$, Eq. 3 defines the rule to find the closest sphere $s_{min} \equiv (c_{min}, r_{min})$ to p_i. Then, Eq. 2 can be used to compute the distance.

$$\begin{aligned}
&\lambda_{min} = -\frac{(\overrightarrow{c_1} - \overrightarrow{p_i}) \cdot (\overrightarrow{c_2} - \overrightarrow{c_1})}{\|\overrightarrow{c_2} - \overrightarrow{c_1}\|^2};\ \lambda_{min} \in [0,1] \\
&\overrightarrow{c_{min}} = \overrightarrow{p_i} - \overrightarrow{c_1} + \lambda_{min}(\overrightarrow{c_2} - \overrightarrow{c_1}) \\
&\overrightarrow{r_{min}} = \overrightarrow{p_i} - \overrightarrow{r_1} + \lambda_{min}(\overrightarrow{r_2} - \overrightarrow{r_1})
\end{aligned} \tag{3}$$

All the remaining points in the 3D point cloud are evaluated and those which distance in zero or negative are labelled as belonging to the robot and then removed from the point cloud. After this filter, the remaining points are considered to be part of the object.

To illustrate the process of object segmentation, Figure 3 shows results after each step of the process. Picture 3(a) shows the initial image of the scene. Figure 3(c)) shows the points remaining after the box filtering. Finally, the third image on the right (Fig. 3(d)) shows the object segmented from the whole scene. This process is repeated each time step in order to calculate the position of the object.



(a) Scene Example (b) Table:points removed, arm and object segmented (c) arm (green) and hand (red) points labelled (d) object point cloud

Fig. 3. Object segmentation process

5 Contact Detection

Once an object has been segmented from the scene, the next step is to determined when the object moves. As our approach is constrained to rigid solids and non articulated objects, we assume that when an object moves is because a contact has occurred. In order to detect the movement, it is necessary to track the point cloud of the object using consecutive images.

Literature offers algorithms for object tracking based on image descriptors such as SIFT [13], SURF [14], or Harris corners [15]. These methods make the assumption that the objects have a texture, a regular shape or that the descriptors are visible all the time. As we do not have any assumption of shape or texture and the hand of the robot may occlude parts of the object, we have chosen the Iterative Closest Point (ICP) algorithm [16] as our approach to object tracking. In particular, we use the standard ICP algorithm implemented in PCL library [17]. The ICP does not need any characteristic of the object, only two points clouds are needed. The ICP refines iteratively the transform between two consecutive point clouds by repeatedly generating pairs of corresponding points on the meshes and minimizing an error metric. Once the object is segmented as was described in Section 4, the ICP is applied to the object point cloud at instant t^k and t^{k-1}. The algorithm returns a homogeneous transformation matrix that is decomposed in a translation vector $t(x,y,z)$ and a quaternion $q(x,y,z,w)$ from which an angular rotation ω is obtained . When the module of the translation vector $\|\overrightarrow{t}\|$ or the angular rotation ω are grater than a threshold ($\|\overrightarrow{t}\| \geq t_{max}$ or $\omega \geq \omega_{max}$), we conclude that the object has moved and therefore has been contacted by the hand.

6 Estimation of Contact Location

This section describes how the location of the detected contact is estimated. The output is a point cloud containing all the points with high likelihood of being in contact with the object. Contact location estimation from vision is going to find the problem of occlusion, either because the object is occluded by the object or because the robot itself occludes the object. In this sections we describe a procedure that deals with point contact occlusions and gives a guess about where the contact point is.

With the purpose of dealing with the uncertainty of the occluded area we use an Occupancy Grid Map (OGM) that is iteratively updated to estimate which space areas have the greatest probability to be in contact. Initially an OGM of the object is built assuming that the object is not occluded by the hand (Fig. 4). While the hand is moving towards the object to perform any task, the OGM is updated exploiting the robot movement. If a contact is detected (see Sec.5) the intersection between the hand model surface and the OGM is built. For each point of the intersected region its probability of being a contact point is calculated. Section 6.2 tells the details for the generation and updating of the OGM and section 6.2 describes how the contact point likelihood is calculated.

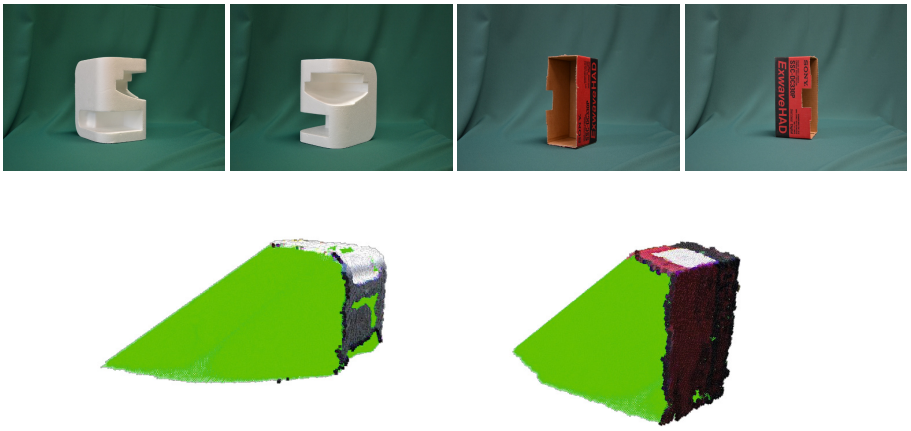


Fig. 4. Real objects and its initial Occupancy Grid Map (OGM)

6.1 Occupancy Grid Map Initialization

The OGM is bulit projecting each point of the initial object point cloud along the direction of the camera until its intersection with the table plane. To project the points, a perfect pin-hole model of the camera is used. We also assume the table plane position to be known in advance.

Then, the occluded area is discretized in cells of $1mm^3$ in each direction (x, y, z). The cells that appear already in the point cloud (i.e. are being seen by the camera) have a probability of being occupied $P(c_i = occ) = 1$. Meanwhile there is no information about the cells that are not being seen its starting likelihood of being occupied is $P(c_i = occ) = 0.5$.

6.2 Occupancy Grid Map Update

In the traditional occupancy map the cell values are updated with a distribution function of the measurement sensor. However no direct sensor to measure contact is considered in this approach. Instead, we propose to use the hand model as a sensor.

Virtual Contact Sensor. In order to delimit the area where the contact is, we have taken into account the direction of movement of the hand $\overrightarrow{u}$. Firstly, the cells of the OGM that belong to the surface of the hand are obtained. Secondly, for each cell, the normal vector to the hand model surface $\overrightarrow{n}$ is computed and the angle α between the normal vector $\overrightarrow{n}$ and the movement vector $\overrightarrow{u}$ is obtained. Finally, if this angle is less than an empirically defined treshold α_{max} the point is considered part of the sensible area and a candidate to be in contact. Fig.5 shows the result of a simulated contact with different hand movement directions and α_{max}, where red points are contact candidates (i.e. have an $\alpha \leq \alpha_{max}$) and blue points are not contact candidates (i.e. points with an $\alpha \geq \alpha_{max}$).

In order to put the above mentioned in a mathematical way we have formulated it in eq. 4. This equation shows the probability function, $P(z(k)|c_i)$, of a given cell to have a contact. This function depends on the distance between the cell and the robot spherical model. If the cell is inside the model $P(z(k)|c_i) = 0$, we have the guarantee that the cell is free of contact. In the case that the cell is outside of the hand model there is no way to have any new information about the occupancy, thus $P(z(k)|c_i) = 0.5$.

Finally if the cell is on the surface of the hand model and $\alpha \leq \alpha_{max}$ a probability function is defined, where $d \in [0,5]mm$, is the variable that models the error in the direction of hand movement. s is the standard deviation of the Gaussian used to take into account the error in the geometrical and kinematic model of the hand. In other words, we consider that the contact surface estimation has a Gaussian error in the direction of the movement.

$$P(z(k)|c_i) = \begin{cases} 0 & p \in \text{inside model} \\ 0.15 \cdot exp(\frac{d^2}{2 \cdot s^2}) + 0.5 & p \in \text{surface}; \alpha \leq \alpha_{min} \\ 0.5 & otherwise \end{cases} \tag{4}$$

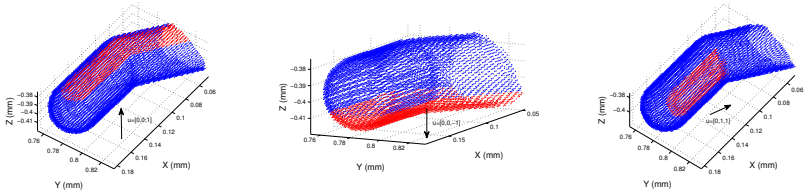
Update Algorithm. The OGM is updated each time step, depending on the contact detection (Sec. 5). If no contact is detected, the cells that are inside or in the surface of the model are updated. The new likelihood value this cells is 0, as can be obtained from eq. 4 and eq. 5. Cells that are outside the model and do not belong to the surface, keep the same value. Finally, if contact has been detected then the surface cells are updated with the eq 5:

$$P(c_i = occ|z(k))^k = \frac{P(c_i = occ|z(k))^{k-1} \cdot P(z(k)|c_i = occ)}{\sum_{c_i} (P(c_i|z(k))^{k-1} \cdot P(z(k)|c_i))} \tag{5}$$

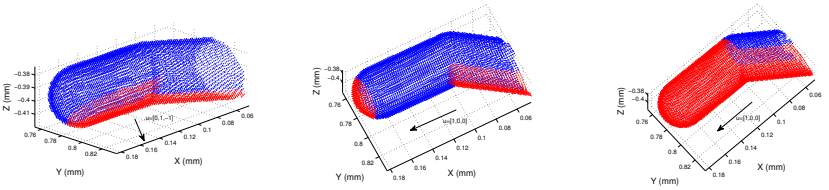
where $P(z(k)|c_i = occ)$ is the probability of a measure given a occupied cell ($c_i = occ$), in other words, the sensor model. $P(c_i = occ|z(k))^{k-1}$ is a priori probability to be occupied given a measure $z(k)$ and $P(c_i = occ|z(k))^k$ is the a posteriori probability to be occupied given a measure $z(k)$, for more information refer to book [18].

When the contact is detected, our algorithm returns a point cloud with the cell that have a high probability level (e.g more than 0.8) and are part of the sensible surface as has been described in section 6.2. The fig. 6 shows all the surface points, the candidate points that belong to the sensible surface are colored from blue to red depending on its contact probability.

The inputs of the alg. 1 are the object point cloud, the joint value of arm and hand, the initialized grid map, and the direction of movement of the hand. The function in line 7 is the detection of contact that was described in section 5.



(a) $\alpha_{max} = 0.6rad$, $u = 0,0,1$ (b) $\alpha_{max} = 1rad$, $u = 0,0,-1$ (c) $\alpha_{max} = 0.6rad$, $u = 0,1,1$



(d) $\alpha_{max} = 1rad$, $u = 0,1,-1$ (e) $\alpha_{max} = 1rad$, $u = -1,0,0$ (f) $\alpha_{max} = 1.6rad$, $u = -1,0,0$

Fig. 5. Sensor surface with different values of α_{max} and hand direction u

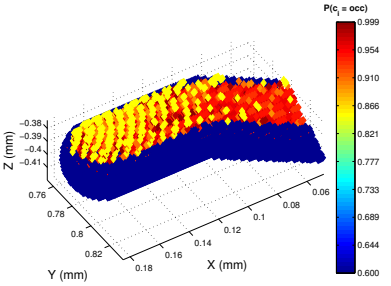


Fig. 6. contact probability distribution

7 Experimental Validation

In order to validate our approach, we have implemented the algorithms for our robot platform and tested them on several cases. Our robot platform is an upper body humanoid composed of two 7 DOF robot arms, a Schunk Dextrous Hand with 7 DOF, equipped with a JR3 Force-Torque sensor on the wrist. The hand has three fingers, each with 2 DOF and two Weiss tactile sensors: one on the fingertip and one on the inner phalanx (see Fig. 1). The vision system is composed of a 2 DOF pan-tilt head with a

Algorithm 1. Grid Map Update

$Object pointCloud \leftarrow INPUT$
$q_{arm} \leftarrow INPUT$ {Arm Joints, Hand Joints}
$Grip\_Map \leftarrow INPUT$
$\overrightarrow{u}_{hand} \leftarrow INPUT$ {Hand movement direction}
$d \in [0,5]$
$distance \in$ {inside, surface, outside}
$contact \leftarrow Contact\_Detection$
for all $cell_i$ **do**
  $[distance, \overrightarrow{n}_{surf}] \leftarrow DistanceToModel(cell_i, q_{arm}, q_{hand})$
  $\alpha = \frac{acos(\overrightarrow{n}_{surf} \cdot \overrightarrow{u}_{hand})}{\|\overrightarrow{n}_{surf}\| \cdot \|\overrightarrow{u}_{hand}\|}$
  if $cell_i \in surface$ **and** contact **then**
    for all d **do**
      $UpdateCellSurface(cell_i, d, \overrightarrow{u}_{hand}, Grid\_Map)$
    end for
    $ContactSurface \leftarrow cell_i$
  else
    if $cell_i \in inside$ **then**
      $cell_i = 0$
    end if
  end if
end for
if contact **then**

  return ContactSurface
end if

Algorithm 2. $UpdateCellSurface(cell_i, d, u_{hand}, Grid\_Map)$

$s = 0.01$
$\overrightarrow{cell}_i = GetCoodinateFromCell(Grid\_Map, cell_i)$
$\overrightarrow{cell}_d = \overrightarrow{cell}_i + d \cdot \overrightarrow{u}_{hand}$
$P(c_i = occ|z(k))^{k-1} = GetCellValue(Grid\_Map, \overrightarrow{cell}_d)$
$P(z(k)|c_i = occ) = 0.15 \cdot exp(\frac{d^2}{2 \cdot s^2}) + 0.5$
$Pc_i = occ|z(k))^k =$
$= \frac{P(c_i=occ|z(k))^{k-1} \cdot P(z(k)|c_i=occ)}{(P(c_i|z(k))^{k-1} \cdot P(z(k)|c_i)) + (1-P(c_i|z(k))^{k-1}) \cdot (1-P(z(k)|c_i))}$

return $Pc_i = occ|z(k))^k$

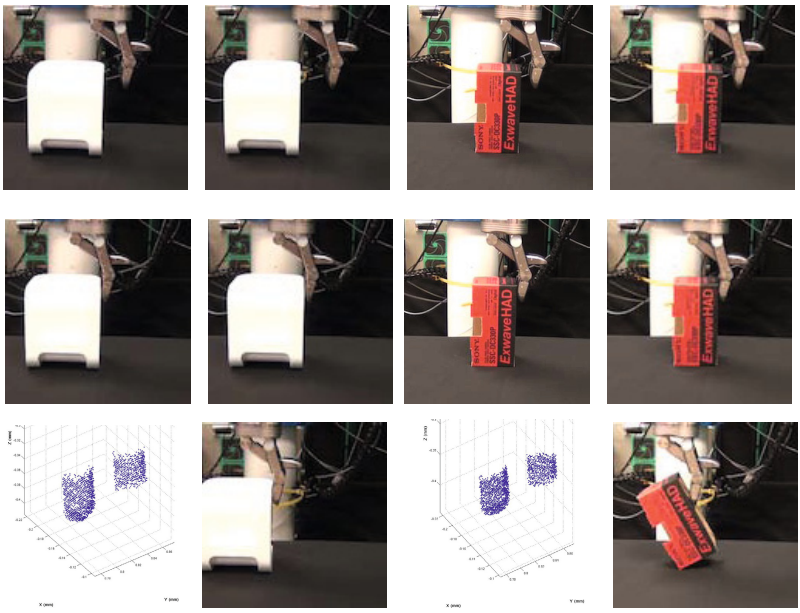
Kinect sensor. For our experiments, the force-torque sensor has not been used and the tactile sensors have been only used to provide ground truth validation data.

Two validation experiments have been implemented. The first one seeks to demonstrate that our approach is more sensitive than the real tactile sensors. The second one evaluates our approach for estimating contacts with occluded parts of non convex objects. The objects used for validation are shown in Fig. 4. The object on the left is a white very light object, with a non-trivial shape. The one on the right is an empty light cardboard box.

7.1 Contact Detection Sensitivity

The first experiments aims to demonstrate that our approach is able to detect contacts that our real tactile sensors can not detect. One single object is in the scene and the hand configuration and approaching movement are set in such a way that the first contact with the object occurs on the fingertip of the middle finger, where there is a tactile sensor. The controller is programmed so that as soon as a tactile contact is detected, the hand stops. The difficulty here is that the object is too light and, in most of the cases, the contacts go unnoticed due to the low sensitivity of the real tactile sensors.

The initial configuration is shown in the upper rows of figures 7(a), 7(b), 7(c) and 7(d). The robot moves in a predefined direction towards the object. When the contact is detected, the robot stops and the contact information is stored. The results of the experiments using the proposed approach are shown in middle rows of figures 7(a) and



(a) Contact estimation object 1 (b) Real tactile sensor object 1 (c) Contact estamination object 2 (d) Real tactile sensor object 2

Fig. 7. sensitivity comparative between contact estimation vs real

7(c) where as soon as a small movement has been observed on the object the hand stopped. Lower rows of figures 7(a) and 7(c)) show a detail of the estimated contact locations on the fingertips.

Middle and lower rows of figures 7(b), 7(d) show the case where the visual contact detection was disabled so the robot relies exclusively on real tactile sensor information. In both cases the contact was not detected and the hand kept pushing the object.

7.2 Validation of the Estimation of Contact Locations

In this set of experiments, we seek to test the validity of our approach to estimate the locations of contacts. In order to validate this, we compared the output of our estimator against the readings given by the real tactile sensors. As the objects are too light to be robustly detected by the tactile sensors, we have loaded the objects with additional weights.

The experiments consist on a single object placed on the table. A set of different hand configurations and directions are predefined to approach the object. The robot moves until a visual contact is detected. Then, the visually estimated contacts locations and the real tactile sensor readings are recorded.

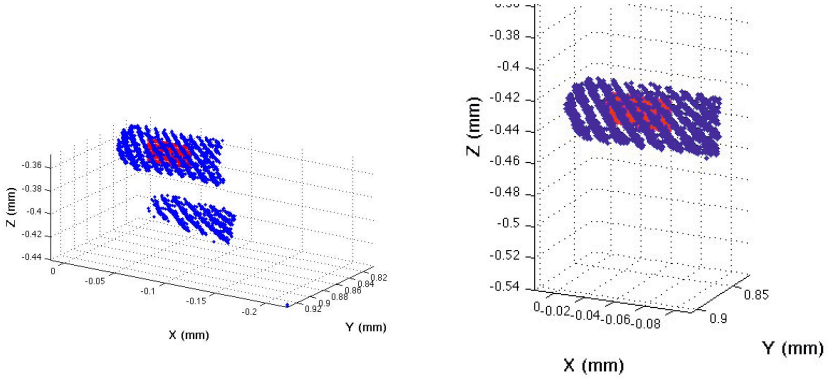
Figure 9 shows the results of these experiments. The graphs in the right of each figure show the estimated contact locations (in blue) and real contact tactile readings (in red). The 3D positions of the tactile readings are reconstructed using the known kinematics and configuration of the robot. As it can be seen, the proposed estimator is able to detect accurately the contact locations. A special case is shown in the fifth row of fig. 9

(a) step 1

(b) step 2

(c) step 3

(d) step 4



(e) Movement to avoid false contacts

Fig. 8. Improvement of the estimation

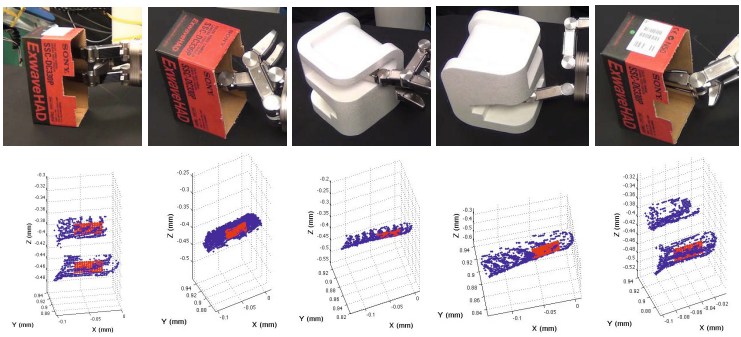


Fig. 9. Contact Estimation Validation. Virtual sensor (blue) vs real sensor (red)

in which not all the occluded robot fingers contact the object, as the readings of the real tactile sensor indicate. However, the visual contact estimator shows that the contact is equally probable on both fingertips. The reason is that when the contact is detected the cells of the occupancy grid map around the finger are unknown, so their probability of being occupied is 0.5.

These type of ambiguities can be solved by further exploration movements of the fingers. An example of this is shown in Figure 8, in which a second movement is performed to get a second contact detection. Figure 8(e) shows the initial estimation (left) and the estimation after the second movement (right) which match with the real sensor readings.

8 Conclusion and Further Work

This paper has described a contact sensor which uses exclusively RGB-D images, without the help of any touch sensors. The whole process is divided in several steps. On the first step the object point cloud is segmented from the whole scene using a spherical geometric modelling of the robot. On the second step, the object point cloud is tracked using an ICP approach in order to detect any relevant movement. As soon as a movement is detected a probability estimator is used to update the occupancy grid of the object and to detect the most likely locations where the contact has happened. This approach is validated through several experiments on a real scenario where the sensitivity of the visual contact sensor to detect contacts that would be unnoticed by real sensors is demonstrated. In addition, the ability of the sensor to detect contacts when the hand is visually occluded is also demonstrated.

The paper presents a first implementation of a contact sensor based exclusively on visual information. This has never been demonstrated to the knowledge of the authors. Regarding the utility of such a sensor, it is not minded to be used alone, but in combination with other touch modalities, in order to obtain a more robust contact detections and, as consequence, a more reliable and robust manipulation of objects. The experiments have shown that the visual sensors can provide information that touch sensors are unable, i.e.: when the robot contacts light objects, or when the contact happens on not-sensorized surfaces.

Fianlly, more technically, the methodology can still be improved. First, methods to speed-up the computations are required. ICP is specially time-consuming and faster alternatives would be necessary to make the approach work at an acceptable frame ratio. Another important improvement is to design schemes to integrate visual and touch modalities of contact detections, in order to design more robust manipulation controllers.

Finally, some secondary parts of the algorithm could also be improved in order to make their use more general. In particular the segmentation phase in the case that the scene contains several objects, stacked objects or articulated bodies.

Acknowledgement. This research was partly supported by Ministerio de Ciencia e Innovación (DPI2011-27846), by Generalitat Valenciana (PROMETEO/2009/052) and by Fundació Caixa Castelló-Bancaixa (P1-1B2011-54).

References

1. Shadow Robot Company: Shadow Dexterous Hand C5 Technical Specification. Technical report (May 2008)
2. Liu, H., Meusel, P., Seitz, N., Willberg, B., Hirzinger, G., Jin, M., Liu, Y., Wei, R., Xie, Z.: The modular multisensory dlr-hit-hand. Mechanism and Machine Theory 42(5), 612–625 (2007)
3. Tegin, J., Wikander, J.: Tactile sensing in intelligent robotic manipulation a review. Industrial Robot: An International Journal 32(1), 64–70 (2005)
4. Dahiya, R., Metta, G., Valle, M., Sandini, G.: Tactile sensing - from humans to humanoids. IEEE Transactions on Robotics 26(1), 1–20 (2010)
5. Felip, J., Morales, A.: Robust sensor-based grasp primitive for a three-finger robot hand. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1811–1816 (October 2009)
6. Platt, R., Fagg, A., Grupen, R.: Null-space grasp control: Theory and experiments. IEEE Transactions on Robotics 26(2), 282–295 (2010)
7. Prats, M., Sanz, P.J., Pobil, A.P.: A framework for compliant physical interaction. Autonomous Robots 28(1), 89–111 (2009)
8. Hsiao, K., Nangeroni, P., Huber, M., Saxena, A., Ng, A.Y.: Reactive grasping using optical proximity sensors. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2098–2105 (2009)
9. Tornero, J., Hamlin, J., Kelley, R.: Spherical-object representation and fast distance computation for robotic applications. In: International Conference on Robotics and Automation, vol. 2, pp. 1602–1608 (April 1991)
10. del Pobil, A., Serna, M., Llovet, J.: A new representation for collision avoidance and detection. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 246–251 (May 1992)
11. Gilbert, E., Johnson, D., Keerthi, S.: A fast procedure for computing the distance between complex objects in three-dimensional space. IEEE Journal of Robotics and Automation 4(2), 193–203 (1988)
12. Hamlin, G., Kelley, R., Tornero, J.: Efficient distance calculation using the spherically-extended polytope (s-tope) model. In: IEEE International Conference on Robotics and Automation, vol. 3, pp. 2502–2507 (May 1992)
13. Lowe, D.: Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)

14. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Computer Vision Image Understanding 110, 346–359 (2008)
15. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. Fourth Alvey Vision Conference, pp. 147–151 (1988)
16. Zhang, Z.: Iterative point matching for registration of free-form curves and surface. International Journal of Computer Vision 13(2), 119–152 (1994)
17. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 9-13 (2011)
18. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents) (2001)

A New Extended SDLS to Deal with the JLA in the Inverse Kinematics of an Anthropomorphic Robotic Hand

Choukri Bensalah[1], Javier González-Quijano[1], Jianwei Zhang[2], and Mohamed Abderrahim[1,⋆]

[1] Roboticslab, Systems Engineering and Automation Department, University Carlos III of Madrid, Spain
[2] Faculty of Mathematics, Informatics and Natural Science, Department Informatics, Group TAMS, University of Hamburg, Germany
{cbensala,jgonza1,mohamed}@ing.uc3m.es, zhang@informatik.uni-hamburg.de

Abstract. This paper proposes Extending the Selectively Damped Least Squares method to cater for joint limits avoidance (JLA) when solving the inverse kinematics of anthropomorphic robotic hands. Here, the SDLS is considered as the main task allowing to avoid singularity behaviors by computing for each singular value an appropriate damping factor of a Jacobian matrix concatenating the proper kinematics of the robotic hand with an additional task (JLA) projected into the null-space of the main task. A comparative study of different orders-norm of the objective function defining the JLA task has been done with the aim to select the suitable order which is able to overcome the mentioned constraints with less instability and errors. After an initial tuning to choose the right parameters, the method was applied to calculate the kinematics of all fingers' joints of a robotic hand to fulfill a grasping task. The method achieves the solution of all fingers together as one complex multibody system.

Keywords: Hand Inverse kinematics, singularity and joint limits avoidance, redundancy, SVD .

1 Introduction

In order for robotics systems to be able to perform complex manipulation tasks like humans do, they should be endowed with similar mechanical dexterity capabilities and the accompanying sensory and control requirements. The development of anthropomorphic robotic hands and achieving higher autonomy in the manipulation of objects and tools are considered among the important challenges

⋆ This work was partially supported by the European project HANDLE, FP7-231640. Choukri Bensalah is supported by the Spanish ministry MICINN through FPI scholarship.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_47,

to overcome in nowadays robotics [1]-[3]. The correct grasping and in-hand manipulation including rolling and sliding motions are specific features which are necessary to achieve dexterous manipulation tasks. An important research effort has been dedicated to implement these types of motions [4]-[6]. Even in the more recent manipulation solutions based on the use of the synergies and set of given hand postures the mentioned motions are required to move the hand fingers between successive synergies. These methods can be classified in three types. Those based on the stochastic laws as learning [6], or the heuristic methods [7] or the numerical approaches using the Jacobian matrix [8]-[21]. It is the inverse kinematics problem. The inverse kinematics is a method which consists in computing the posture of the manipulator via the estimation of each individual degree of freedom for which the end-effector reach a desired position. During the last three decades, the calculation of the inverse kinematics numerically using the Jacobian matrix of the manipulator has been widely studied. Several mathematical concepts have been introduced to describe the manipulator kinematic configuration. One of these concepts is the ill-conditioned Jacobian matrix. In this case, the manipulator is situated at the singularity configuration or in the neighborhood of singularity. In this configuration, a small deviation of the end-effector position provides a high deviation of the angular joints. A robust inverse kinematics approach is extremely important to ensure a good behavior in such situation. Many researchers have proposed methods to overcome this critical situation. The most popular and successful method is to damp the norm of the joint velocity [9],[10] in order to prevent the infinity tendency. This was implemented in the Damped Least Squares (DLS) method. The Jacobian matrix should be regularized by adding to it a corrective component, called the damping factor. This factor must be carefully chosen to guarantee good behavior without disturbing the tracking task of the end-effector. Using the Singular Value Decomposition (SVD) of the Jacobian matrix, the choice of this factor is easy to understand. However, if the manipulator is far from the singularity configuration, this factor should be zero to allow a behavior similar to the pseudoinverse method. In such critical configuration, the damping factor is computed using some comparative criteria [11],[12],[14],[15],[16] and [17]. Some authors have calculated a damping factor as a function of the smallest singular value [10], while others have chosen its value according to the condition number [13]. There are also several ways to compute this factor according to the conditions imposed on the maximum allowable values of the joint velocity, end-effector velocity or by minimizing the residual function with respect to this factor between the exact and the damped solution [12] [18]. There are other proposed solutions to overcome the singularity problem based on the measure of the smallest singular value. The solution consists in regularizing this component by truncating it [19] or by applying a numerical filter [18]. The latter proposed solution means that the smallest singular value has been damped more than the others. However, due to the necessity to the SVD process, the main drawback of these methods is the heavy computation cost. In order to avoid this problem, Maciejewski and Klein [18] proposed a recursive numerical algorithm allowing the estimation of solely

the smallest singular value and its associated singular vectors. This algorithm uses the symmetry property of the damped Jacobian matrix and the Cholesky decomposition. An extension of this algorithm has been proposed in [14], where the objective consists in estimating the first and the second smallest singular values. The failing of DLS method, including in the case of the smallest singular value estimation, results from applying the obtained damping factor uniformly to all singular values, and therefore affecting considerably those components with good behavior. In order to avoid this undesirable effect, Buss and Kim [17] have devised a new algorithm that applies selectively different damping factors to the different singular values. This method designated as Selectively Damped Least Square SDLS is the base of our proposed work. Although SDLS can be considered as an extension of [18], it uses suitable values of damping allowing a minimum tracking error of the end-effector position. The method analyzes each joint angle separately and computes the required value of this joint to allow the end-effector to reach its desired position. In the case of unreachable target position, the SDLS method evolves better than DLS where the end-effector points toward this position without oscillating. The contribution of this paper consists in using this technique to calculate the inverse kinematics of an anthropomorphic robotic hand, and extending it to account for more features. Although robotic hands possess a high number of DOF, most methods consider each finger as a single manipulator and calculate the inverse kinematics of the fingers separately, while our proposed method calculates simultaneously the solution for all the DOF of the hand together as one multi-body system. Taking advantage of the hand redundancy, in addition to the singularity avoidance of the original SDLS, our extended version is able to deal with constrained motions. Here, the motions of all joints are constrained by their mechanical limits. This new additional task is projected into the null-space of the main task. Various ways of defining the null-space and the projection methods are discussed in [25] and [26].
This document is organized as follows. Section II gives a description of the original SDLS method and introduces the modification which defines the optimal incremental step. The Extended Selectively Damped Least Squares method (ESDLS) that deals with joint limits avoidance task is discussed in section III. Section IV presents the simulated results of ESDLS applied to the redundant index finger for simplicity and to the overall hand as a single multi-body system. The model of the hand used is the *Shadow hand* since it is part of our experimental platform in the HANDLE project [28].

2 Selectively Damped Least Squares with an Optimal Increment Steps for Robotic Hand

The SDLS method can be considered as the most robust inverse kinematics scheme in terms of singularity configurations, and particulary for unreachable targets. For the human-hand, this configuration is quite frequent for example using the index finger to point towards an unreachable objet (see Fig.1). In this critical situation, both methods DLS and pseudoinverse exhibit oscillations

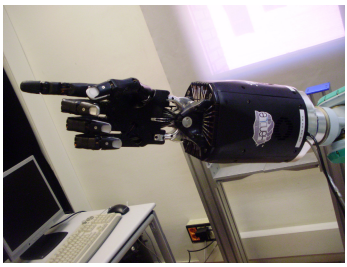


Fig. 1. Positioning posture of Shadow hand

brought by the instability of the system with less effect for first one. As for SVD, this instability is interpreted by the existence of at least one zero singular value. Let J be the jacobian matrix, then, in discontinuous time, the inverse kinematics solution given by the pseudoinverse method is defined as: [23][24].

$$\delta q = J^{\dagger}\delta x \tag{1}$$

Where $\delta q \in \mathbb{R}^n$ are the joint variables, $\delta x \in \mathbb{R}^m$ is cartesian position of the fingertips and $J^{\dagger}$ is the pseudoinverse of the matrix $J(m \times n)$. The associated SVD of Jacobian matrix allows to rewrite the system (1) as :

$$\delta q = \sum_{i=1}^{r} \frac{1}{\sigma_i} v_i u_i^T \delta x \tag{2}$$

with r represents the rank of the Jacobian matrix, u_i and v_i are the output and the input singular vectors, respectively. The singular values of the Jacobian matrix σ_i are generally ordered so that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$.
Here, the target positions, denoted by δx_d, will be defined the same way as in [17].

$$\delta x_d = \begin{cases} x - x_{goal} & \text{if } \|x - x_{goal}\| \leq \delta x_{max} \\ \frac{x - x_{goal}}{\|x - x_{goal}\|} \delta x_{max} & \text{Otherwise} \end{cases} \tag{3}$$

However, unlike the fixed choice of the allowable maximum value of δx_{max} in [17], this value will be selected optimally using the definition of condition number.

$$\frac{\|\delta q\|}{\|q\|} \leq \kappa(J) \frac{\|\delta x\|}{\|x\|} \tag{4}$$

where $\kappa = \frac{\sigma_1}{\sigma_r}$. Then, in the extreme case corresponding to the maximum allowable joint angle $\delta q = \delta q_{max}$, it is evident to verify that:

$$\delta x_{max} = \frac{\sigma_1}{\sigma_r} \frac{\|\delta q_{max}\|}{\|q\|} \|x\| \tag{5}$$

Suppose here that all joint angles have the same maximum incremental value and the Jacobian matrix should be initially well-conditioned. Usually, the contact

points of the fingertips of robotic hands are defined only by the position without the orientation. This limitation is due to the planar kinematic structure of each finger. Whereas, the contact points' orientation can be achieved by means of the arm to which the hand is mounted. Let $x_i \in \mathbb{R}^3$ be the position of each fingertip. To ensure a simultaneous movements of all fingertips, the hand is considered as one multi-body mechanical system. Then, $x = (x_1, \cdots, x_k)^T$ $(x \in \mathbb{R}^{3k})$ where k is the finger number and $3k = m$. The SDLS method treats each singular value individually by computing for each one its own damping factor. In the algorithm 1, the SDLS with an optimal incremental step for five fingered robotic hand is presented $(k = 5)$.

3 Extended Selectively Damped Least Squares for Joint Limits Avoidance (JLA)

In this section, the Jacobian matrix of the main task has been extended to deal with an additional task in order to prevent all joint angles from reaching their mechanical joint limits. This process makes sense since the robotic hand is assumed redundant, where the Jacobian matrix exhibits more columns than rows, allowing an infinity of possible solutions. In order to obtain a unique solution, the Jacobian matrix has been extended to obtain a full rank matrix. Let $\psi(q)$ be the objective function defining the additional task. Let extended Jacobian matrix

$$J_{ext} = \begin{pmatrix} J \\ \frac{\partial}{\partial q}\left(N_e^T \frac{\partial \psi(q)}{\partial q}\right) \end{pmatrix} \tag{6}$$

with $J_{ext} \in \mathbb{R}^{n \times n}$ and $N_e \in \mathbb{R}^{n \times r}$ is a full rank matrix whose columns span the r-dimensional null-space of J. The null-space matrix can alternatively be obtained taking advantage of the SVD result, which represents the last r input singular vectors. To avoid the mechanical joint limits, the objective function can be suggested as:

$$\psi(q) = \sum_{i=1}^{n} \left(\frac{q_i - \overline{q}_i}{q_{iM} - q_{im}}\right)^2 \tag{7}$$

where q_{iM} and q_{im} denotes the upper and lower bounds of each joint q_i, respectively, and $\overline{q}_i$ is the middle value of the joint range. Thus, by maximizing this distance, all joint angles try to reach the center of their ranges. Instead of computing the 2-norm cost function, another alternative could be introduced to this formulation using high order norm. To make the objective function meaningful by activating the joints which are close to limits more than those which are far from the limits, a weighting factors has been introduced. Taking into account the p-norm cost function, the new objective function has the following form:

$$\psi(q) = \sum_{i=1}^{n} K_i \left\| \frac{q_i - \overline{q}_i}{q_{iM} - q_{im}} \right\|^p \tag{8}$$

Algorithm 1. SDLS with Optimal Increment Steps for Robotic Hand

Input: $\delta x_{allowable}$, δq_{max},σ_i,u_i,v_i,q,x,x_{goal},ε,J
Output: δq

$\kappa \leftarrow \frac{\sigma_1}{\sigma_r}$
$\delta x_{max} \leftarrow \kappa \frac{\|\delta q_{max}\|}{\|q\|} \|x\|$
if $\delta x_{max} > \delta x_{allowable}$ **then**
  $\delta x_{max} \leftarrow \delta x_{allowable}$
end if
$\delta x \leftarrow x_{goal} - x_{current}$
while $\|\delta x\| \geq \varepsilon$ **do**
  if $\|\delta x\| \geq \delta x_{max}$ **then**
    $\delta x \leftarrow \delta x_{max} \frac{\delta x}{\|\delta x\|}$
  end if
  for $i = 1$ to r **do**
    if $\sigma_i = 0$ **then**
      $\varphi_i \leftarrow 0$
    else
      $\alpha_i \leftarrow u_i^T x$
      $N_i \leftarrow \alpha_i \sum_{j=1}^{5} \|u_{j,i}\|$, where $u_{j,i} \in \mathbb{R}^3$
      $\rho_{k,i} \leftarrow \|\frac{\partial x_k}{\partial \theta_i}\|$
      $M_{i,k} \leftarrow \alpha_i \sigma_i^{-1} \sum_{k=1}^{5} \sum_{j=1}^{n} |v_{j,i}| \cdot \rho_{k,j}$
      $D_i \leftarrow \frac{N_i}{M_i}$
      if $D_i \leq 1$ **then**
        $\gamma_i \leftarrow D_i \cdot \Delta q_{max}$
      else
        $\gamma_i \leftarrow \Delta q_{max}$
      end if
      $\varphi_i \leftarrow \alpha_i \sigma_i^{-1} v_i$
      $\eta_i \leftarrow \|\varphi_i\|_\infty$
      if $\eta_i \geq \gamma_i$ **then**
        $\varphi_i \leftarrow \gamma_i \frac{\varphi_i}{\eta_i}$
      end if
    end if
  end for
  $\delta q \leftarrow \sum_{j=1}^{r} \varphi_j$
  if $\|\delta q\| \geq \delta q_{max}$ **then**
    $\delta q \leftarrow \delta q_{max} \cdot \frac{\delta q}{\|\delta q\|}$
  end if
  $x \leftarrow x + J \cdot \delta q$
end while
return δq

where K_i are the weighting factors. In order to introduce a smooth incorporation of the JLA task into the additional task, the weighting factors K_i are chosen as function depending on q_i, q_{iM}, q_{im} and τ_i which is a margin region defined between both bounds and the center. When the joint variable is evolving outside of this region, the weight is zero, and on entering the region, the weight increases gradually towards its maximum value. Basing on [27], and bringing some modifications, it is possible to formulate the weighting factor selection procedure by:

$$K_i = \begin{cases} 0 & q_i \leq q_{iM} - \tau_i \text{and} q_i \geq q_{im} + \tau_i \\ \frac{K_{max}}{2}(1 + \cos \pi \frac{q_{iM} - q_i}{\tau_i}) & q_{iM} - \tau_i \leq q_i \leq q_{iM} \\ \frac{K_{max}}{2}(1 + \cos \pi \frac{q_i - q_{im}}{\tau_i}) & q_{im} \geq q_i \geq q_{im} + \tau_i \\ K_{max} & q_i > q_{iM} \text{ or } q_i < q_{im} \end{cases} \tag{9}$$

Where K_{max} is a user-defined constant representing the maximum weight. Under the new definition of the extended Jacobian matrix, the global algorithm dealing with the joint limits avoidance will be the same as the Algorithm.1, with some modifications beginning from the line 11, whose the SVD of the new extended Jacobian matrix must be recomputed to be applied after by the SDLS method. In the following part, the experimental equipment will be described. The choice of the different parameters of the SDLS with JLA algorithm will be also justified in order to fulfill the target positions in less computation time with a minimum of tracking errors.

4 Simulation Results

The Shadow hand is one of the most relevant anthropomorphic robotic hand available in the market with 24 joints driven by tendons transmission system, allowing it to perform a large scale of human-hand like tasks. Each joint is connected to a DC motor by a couple of tendons through a mechanism that makes it behave as an antagonistic system. The wrist two joints, and the 22 joints distributed on the five fingers are equipped by Hall-effect sensor to measure their angles. The Shadow robotic hand and its kinematic model are shown in Fig.2. The Shadow hand exhibits a redundancy in DOF where ($m = 15$ and $n = 24$). Here, all 20 joints of fingers evolve simultaneously including both wrist joints. The forward kinematics of the overall hand is obtained using Denavit-Hartenberg (D-H) parametrization of each finger from the wrist to fingertip frame. The Jacobian hand matrix will be computed concatenating the Jacobian matrices of each finger according to the determined order. In order to simplify the choice of the different parameters of the SDLS with JLA, the index finger colored in red line given in Fig.2 has been used as a tuning simulation platform. Later, the obtained results are used for the whole hand. The corresponding D-H parameters of the index finger are regrouped in the Tab.I. According to this configuration, where $q \in \mathbb{R}^6$ and $x \in \mathbb{R}^3$, the SVD of the associated Jacobian matrix ($J \in \mathbb{R}^{3\times6}$) provides three columns forming the null-space matrix $N_e \in \mathbb{R}^{6\times3}$. On the basis of SVD of J given by eq.2, the null-space matrix is formed by $N_e = (v_4, v_5, v_6)$.

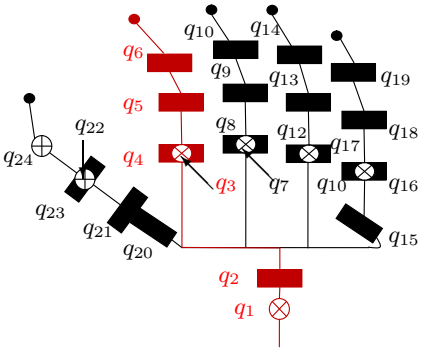


Fig. 2. Kinematic model of Shadow hand and distributed joints naming

Clearly, the objective function depends essentially on the norm order p and the weighting constant K_i. To illustrate the effectiveness of the proposed algorithm, the index finger performs a tracking task for a reachable point in which one of the joint angle evolve close to its bound. In the Fig.3, the good performance of the JLA process has been illustrated in which a virtual upper bound of the joint q_5 has been imposed ($0 < q < 35°$), whereas, mechanically, the range of this joint is defined by ($0 < q < 90°$). Taking into account margin τ, the critical zone starts from 29°. The right picture shows that this joint is well-kept approaching the limit $q_5 = q_{max} + \tau$ without violating it. When the joint comes close to this bound, the rest of the joint variables change their curvatures trying to compensate the generated lack by q_5. These results are obtained performing 2-norm objective function with the fixed values of $\tau = 6°$ and $K_{max} = 10$.
Using this kind of norm, the joint variables are not necessarily closer to the center range of each joint angle. In order to improve the JLA performance and make all joint variables evolve near to their centers, a high order norm has been applied to the objective function. Patel and Shadpey in [18] propose a comparative study between $p = 2$ and $p = 4$. Next, the JLA will be computed and compared using $p = 2$, $p = 4$ and $p = 6$ to achieve a reachable point. As

Table 1. D-H parameters for red colored chain (see Fig.2)

i	a_i	d_i	α_i	q_i	q_{iM}	q_{im}
1	20	0	$-\pi/2$	q_1	10	-30
2	80	33	$\pi/2$	q_2	35	-45
3	0	0	$-\pi/2$	q_3	25	-25
4	45	0	0	q_4	90	-10
5	25	0	0	q_5	90	0
6	24	0	0	q_6	90	0

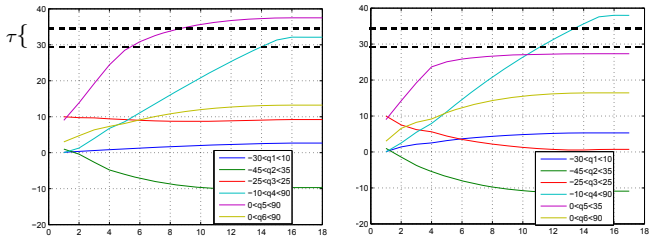


Fig. 3. Joint Limits Avoidance Response for Index Finger

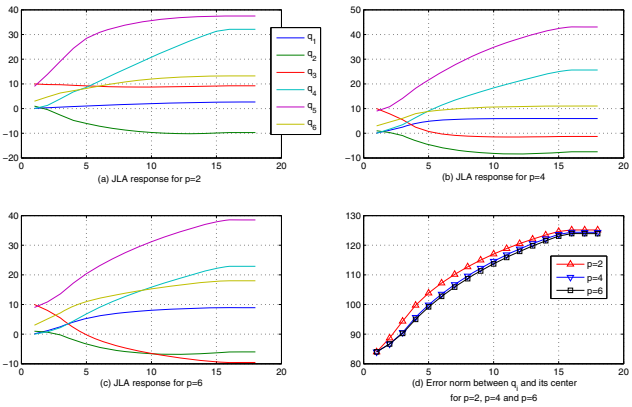


Fig. 4. JLA Response for $p = 2$, $p = 4$ and $p = 6$

shown in Fig.4, for $p = 2$ (Fig.4-a), the joints motion evolve farther from their centers than $p = 4$ (Fig.4-b) and $p = 6$ (Fig.4-c), whose the last order norm lead to less errors, which can be considered as a suitable experimental order. This can be illustrated in Fig.4-d which shows the errors evolution between each joint angle and its corresponding center for all orders norm through the tracking. Here, the index finger does not pass by any singularity. However, our principal aim is not only to avoid the joint bounds but also to ensure a good behavior near or at the singularity configuration. Using the same simulation platform of the index finger, it is necessary to find a suitable configuration from which both joint limits and singularity phenomena could arise. One of critical existing postures allowing both phenomena is pointing toward an unreachable point. In this configuration, in addition to obvious singularity, at least the two last joints finger q_5 and q_6 are at their limits. The associated weighting factor to these two joints takes high value which affects considerably the main task quality. This described case simulations is reproduced to track an unreachable point as shown in Fig.5. The joint motions for $p = 2$ exhibit less oscillations than $p = 4$ and $p = 6$ in stable phase. However, during the tracking, for $p = 6$ the joint motions

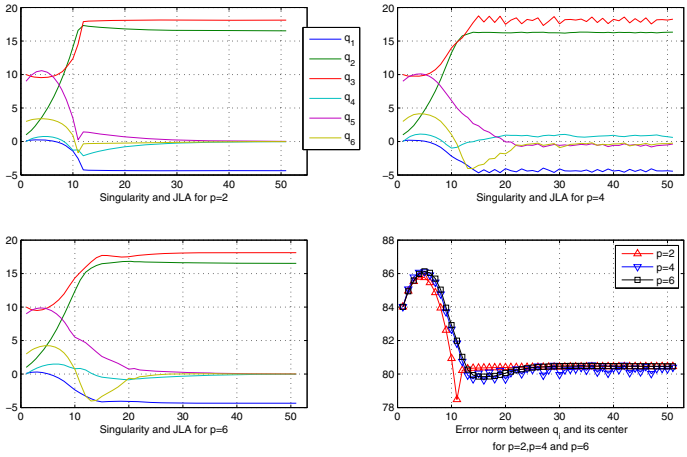


Fig. 5. Extended SDLS response for $p = 2$, $p = 4$ and $p = 6$

are smooth while for $p = 2$ there are some undesired behaviors. Regarding the error norm (Fig.5-d), the 2-norm gives less error norm and it stabilizes faster. Finally, according to these obtained results, $p = 2$ can be considered as a suitable order-norm bringing a good trade-off between the stability, tracking error and time response while avoiding the singularity configurations and joint limits.

5 Experiment Results

The obtained results using the index finger can be successfully extended to deal with all fingers' joints, including both wrist joints. The used robotic hand exhibits a degree of redundancy $r = 9$, which represents also the number of columns of the null-space matrix N_e. The Jacobian matrix of the whole hand is $J \in \mathbb{R}^{15\times24}$, the extended matrix is $J_{ext} \in \mathbb{R}^{24\times24}$ and $N_e \in \mathbb{R}^{15\times9}$. If another constraint need to be added in addition to the JLA, such as the collision avoidance between fingers themselves for example, then, this matrix should be divided, $N_e = [N_{e1}^T, N_{e2}^T]$ where each block matrix is dedicated to project each constraint into null-space. In order to test experimentally the success of the our proposed Extended SDLS algorithm, we used the real Shadow robotic hand to carry out the necessary motion needed to turn a knob. All algorithm steps and generated trajectories are designed in Matlab environment. The hand and the turned cylindrical knob are also plotted in 3D allowing to show different postures of the hand while performing the turning motion (see Fig.6). The generated joint angle trajectories are transmitted to the real Shadow hand by means of ROS (Robot Operation System) framework. In the Fig.7, all sequences of the expected turning motion have been reported which are split in two steps. The first row of Fig.7 shows the grasping task while in the second one, the index and the thumb perform

a circular motion. The knob object is placed far from the palm to make both fingers in singularity configuration and in which some joint angles evolve near to their bounds. As can be shown, the three last joints of the index finger are approximatively aligned.

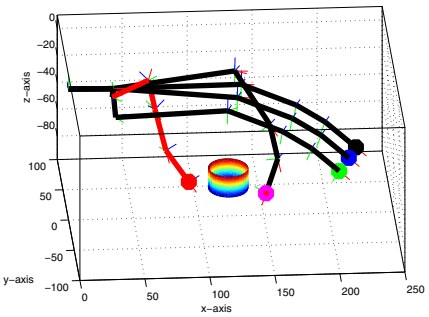


Fig. 6. Simulated kinematic model of the Shadow robotic hand

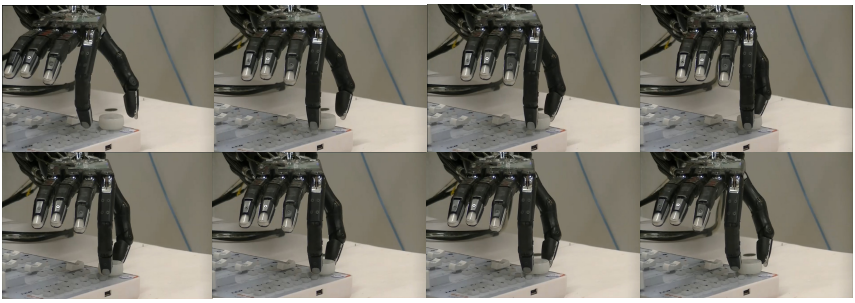


Fig. 7. Experiment application of the Extended SDLS approach performing circular motion for cylindrical knob

6 Conclusion

In this work, the inverse kinematics for an anthropomorphic robotic hand under constraints was investigated. The adopted Selectively Damped Least Squares approach has been improved by applying an optimal incremental step and has been extended to avoid the mechanical bounds of each joint. The proposed algorithm has been tested first on one finger to perform critical postures. The encouraging results were extended to cover all hand fingers including the two wrist joints as one multi-body system. The reported experimental tests demonstrate the efficiency of the proposed algorithm to perform a complex task even in critical configuration such as the singularity and near to joint bounds.

It is well known that the main drawback of the SDLS approach is the expensive computation cost due to the full singular values computation and their corresponding singular vectors. As a future work, instead of computing all singular values, we will adopt an estimation procedure of those singular values which generate singularity. This is expected to reduce substantially the execution time for the whole process.

References

1. Kichan, A.: Shadow Delivers First Hand. Industrial Robot 32, 15–16 (2005)
2. Gaiser, I., Schulz, S., Kargov, A., Klosek, H., Bierbaum, A., Pylatiuk, C., Oberle, R., Werner, T., Asfour, T., Bretthauer, G., Dillmann, R.: A New Anthropomorphic Robotic Hand. In: 8th IEEE-RAS Int. Conf. on Humanoid Robots, Humanoids 2008, pp. 418–422 (2008)
3. Butterfass, J., Grebenstein, M., Liu, H., Hirzinger, G.: DLR-Hand II: Next Generation of a Dextrous Robot Hand. In: IEEE Int. Conf. on Robotic and Automation Robotics and Automation, Proceedings 2001 ICRA, vol. 1, pp. 109–114 (2001)
4. Kent, B., Engeberg, E.D.: Biologically Inspired Posture Control for a Dexterous Robotic Hand. In: Int. Conf. on Adv. Inte. Mechatronics (AIM 2011), Budapest, Hungary, July 3-7 (2011)
5. Xu, J., Li, Z.: A Kinematic Model of Finger Gaits by Multifingred hand as Hybrid Automation. IEEE Tra. on Auto. Science and Engeneering 5(3) (July 2008)
6. Pelossof, R., Miller, A., Allen, P., Jebara, T.: An SVM Learning Approach to Robotic Grasping. In: Proceding of the 2004 IEEE. Int. Conf. on Rob. and Aut., New Orleans, LA (April 2004)
7. Wong, L.C.T., Chen, C.C.: A combined Optimization method for solving the inverse kinematics problem of mechanical manipulators. IEEE Transactions on Robotics and Automation 7, 489–499 (1991)
8. Choukri, B., Mohamed, A., Juan, G.G.: A New Finger Inverse Kinematics Method for an Anthropomorphic Hand. In: IEEE International Conference on Robotic and Biomimetics (ROBIO), Thailand, pp. 1314–1319 (2011)
9. Phuoc, L.M., Martinet, P., Lee, S., Kim, H.: Damped least square based genetic algorithm with Gaussian distribution of damping factor for singularity-robust inverse kinematics. Journal of Mechanical Science and Technology 22, 1330–1338 (2008)
10. Egeland, O., Sagli, J.R., Spangelo, I., Chiaverini, S.: A Damped Least Squares Solution to Redundancy Resolution. In: Proceeding of 1991 IEEE Conference of robotics and Automation, Sacramento, CA, pp. 945–950 (April 1991)
11. Maciejewski, A.A., Klein, C.A.: The Singular Value Decomposition: Computation ans Applications to Robotics. The International Journal of Robotics Research 8(6) (Decembre 1989)
12. Deo, A.S., Walker, I.D.: Robust Subtask performance with Singularity robustness using Optimal Damped Least-Squares. In: Proceeding of the 1992 IEEE International Conference of Robotics and Automation, Nice, France (May 1992)
13. Wampler, C.W.: Manipulation Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. IEEE Trans. of Systems, Man, and Cybernetics SMC-16(1) (January/Februry 1986)
14. Chiaverini, S., Siciliano, B., Egeland, O.: Review of Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator. Trans on Control System Technology 2(2) (June 1994)

15. Phuoc, L.M., Martinet, P., Lee, S., Kim, H.: Damped Least-Square based genetic algorithm with Gaussian distribution of damping factor for singularity robust inverse kinematics. Journal of Mechanical Science and Technology 22, 1330–1338 (2008)
16. Buss, S.R.: Introduction to Inverse Kinematics with Jacobian transpose, Pseudoinverse and Damped Least Squares Method. Typeset manuscript (April 2004), http://math.ucsd.edu/~sbuss/ResearchWeb
17. S.R.: buss and J.S. Kim, Selectively Damped Least Squares for Inverse Kinematics. Journal of Graphics Tools, 37–49 (2005)
18. Patel, R.V., Shadpey, F.: Control of Redundant Robot Manipulators, Theory and Experiments. Springer, Heidelberg (2005)
19. Hansen, P.C.: The truncated SVD as a method for regularization. Numerical Analysis Project Manuscript. NA-86-36 (October 1986)
20. Wolvich, W.A., Elliott, H.: A computation technique for inverse kinematics. In: Proceeding of 23rd Conference on Decision and Control, Las Vegas, NV (December 1984)
21. Sciavicco, L., Siciliano, B.: A Solution Algorithm to the Inverse Kinematics Problem For Redundant Manipulators. Journal of Robotics and Automation 4(4) (August 1988)
22. Guo, G., Gruver, W.A.: Dynamic Grasp Planning of Multifingered Robot Hands Based on Asymptotic Stability. IEEE Trans. on Sys. Man and Cybernetics 26(5) (October 1996)
23. Sun, L.J.D., Liu, H.: An Inverse Kinematics Table-Based Solution of Humanoid Robot Finger With Nonlinearly Coupled Joints. IEEE/ASME Transactions on Mechantronics 14(3) (June 2009)
24. Montana, D.J.: The Kinematics of Multifingered Manipulation. IEEE Transactions On Robotics and Automation 11(4) (August 1995)
25. Seraji, H., Colbaugh, R.: Singularity-Robustness and Task-Prioritization in Configuration Control of Redundant Robots. In: Proceeding of the 29th Conference on Decision and Control, Honolulu-Hawail, pp. 3089–3095 (December 1990)
26. Baillieul, J.: Kinematic programming Alternatives for Redundant Manipulators. In: IEEE Conf. on Robotics and Automation, pp. 722–728 (1985)
27. Shah, M., Patel, R.V.: Inverse Jacobian Based Hybrid Impedance Control of Redundant Manipulators. In: Proceeding of the IEEE, Inter. Conf. on Mecha. and Auto., Niagara Falls, Canada (July 2005)
28. http://www.handle-project.eu
29. http://www.shadowrobot.com

Analysis and Experimental Evaluation of an Object-Level In-Hand Manipulation Controller Based on the Virtual Linkage Model

Javier González-Quijano[1], Thomas Wimböck[2], Choukri Bensalah[1], and Mohamed Abderrahim[1]

[1] RoboticsLab, University Carlos III of Madrid, Spain
[2] Institute of Robotics and Mechatronics, DLR, Germany
javier.gonzalez-quijano@uc3m.es,thomas.wimboeck@dlr.de,
{cbensala,mohamed}@ing.uc3m.es

Abstract. Robot grasping and in-hand manipulation are nowadays very active research fields. Their study has been carried out since many years ago. A very important topic is object-level impedance controllers. While many different articles have been published during the last two decades, it seems not clear that most of the newer methods have taken the advantage over some older ones. Main reasons are related to the problem of validating such approaches due both to the limited number of existing robotic hands and to the high limitations of the simulation technology. It is the aim of this work to review one of this powerful approaches, with some extensions, and demonstrate that they still represent a strong basis for developing robust and efficient state-of-the-art object-level grasp and in-hand manipulation controllers. A review on the concept of the Virtual Linkage null-space parametrization, and its application to the implementation of object-level manipulation controllers will be given. Furthermore, a validation using the Openrave simulator configured with the ODE physical dynamics engine has been performed. The DLR-Hand II has been used in these experiments.

Keywords: in-hand dexterous manipulation grasp control.

1 Introduction

Controlling internal forces in multi-finger and multi-arm robotic systems has been a topic of interest since many years. In multiple-arm manipulation, the presence of internal forces allows to create robust manipulation controllers capable of compensating for modelling errors and, indirectly, to balance different types of object loads, either having an internal origin (i.e. gravity and object inertia) or an external one (i.e. disturbance forces). Nevertheless, during grasping and in-hand manipulation, internal forces are also necessary to fulfil friction constraints, thus preventing slippage.

In the past, research concerning internal forces and moments was focused basically on two issues: on one hand, minimizing internal forces during manipulation [9]; on the other hand, calculating the grasp forces that satisfy frictional and stability constraints for point contact without applying excessive force to

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_48,

the manipulated object [1,3]. Generally, such approaches determine the relationship between the forces applied to an object and the resultant wrench at some reference frame, which is finally gathered in what is called the grasp map. The inverse relationship is then solved using the Moore-Penrose generalized inverse of such grasp map. Here, the null-space basis of this generalized inverse is used to describe the internal forces. This type of methods allows the minimization of internal forces, despite the fact that they are not very suitable for maintaining specific non-zero internal forces. The special case in which the contact model takes into account only contact forces, and not moments, is called "point contact". The Moore-Penrose generalized inverse solution for such model is quite interesting. It yields a solution vector such that the difference between the forces at any two contact points projected along the line joining the two points vanishes, thus with no component in this internal force subspace [6]. Nevertheless, the work done in [11] proposed a method for characterizing the internal forces based on a physical model, called the Virtual Linkage model. Such model may be added to the grasp map to allow to obtain a solution for not only a desired resultant wrench, but also a desired set of internal forces.

It is the aim of this work to describe the implementation and test the performance of an object-level in-hand manipulation controller based on the already mentioned Virtual Linkage model. We will demonstrate the usefulness of such approach, even under real conditions, by analysing the the results of different control experiments with the DLR-Hand II. Such real conditions, concerning namely errors in the contacts model (i.e. contact position, contact normals, contact surface instead of contact points, soft contact conditions, existence of slippage, etc) have been simulated using the Open Dynamic Engine integrated in the OpenRAVE simulator.

1.1 Definition of Grasp Constraints

Defining the dynamics of a precision grasp controller implies, in first place, defining a set of physical constraints describing the relation among all the kinematic variables of the system. Generally, it is enough to represent them as function that relates the differential evolution of the object pose variables x_r with respect to the fingers joints position q, that is to say $\dot{x}_r = f(\dot{q})$. In second place, it is also necessary to find a function that relates the torques applied on the fingers joints τ to the equivalent resultant generalized force at some specific reference frame on the object F_r, in other words $F_r = f(\tau)$.

The analysis depends on the model assumptions. Complex contact models (i.e. taking into account soft-contact [4], rolling effects [5,8] or sliding effects [10]), are very difficult to analyse. Furthermore, their complexity makes them inefficient, from the computational point of view, in real-time control applications. For that reason, the grasp constraints equations will be derived considering a simplified model under the following assumptions [12]:

- The object is manipulated using precision grasps, that is to say using just the fingertips. It is assumed that the contact points in the fingertips do not vary (no rolling or sliding effects).

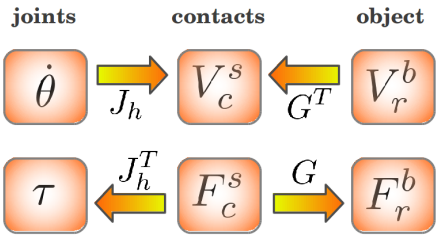


Fig. 1. Relation between applied forces and resultant object wrench

- The contact model type is point contact with friction (PCWF). Under the assumption of such model, the contacts take place only in one point and only forces, not torques, are transmitted from the finger to the object.
- The internal forces calculated by the controller will ensure at any time that the friction constraints are fulfilled for every contact point (no sliding effects).
- The fingers configurations are never singular, having at least three degrees of freedom.

The whole analysis is divided into two subsequent analysis: one studies the relation between the variables at the joint level and the variables at the contact points; the other, the relation between the variables at the contact points and the variables at the the specific object reference frame (Figure 1). This study involves the analysis not only of the kinematic variables but also the force domain variables. The force domain relations are given by:

- The relation between the generalized forces exerted by the fingertip at the contact point $F_{f,i}$ and the torques exerted on the finger joints τ, which is given by the hand Jacobian Matrix J_h^T.
- The relation between the forces exerted by the fingertip on the object at the contact point $F_{c,i}$ and the resultant generalized force (wrench) on the object F_r, which is given by the grasp description matrix G_r.

It is possible to apply the Principle of Virtual Work to obtain the relations among the kinematic variables. Such relations are defined in reverse order to the force domain relations by using the transpose matrices that define such force domain relations:

- The relation between the joint angular velocities $\dot{\theta}$ and generalized velocities of the fingertip contacts reference frames $V_{f,i}$ is given by the hand Jacobian matrix J_h.
- The relation between the generalized velocities of the object reference frame V_r and the generalized velocities of the object contacts reference frames $V_{c,i}$ is given by the grasp description matrix G_r^T.

1.2 The Grasp Description Matrix

The already mentioned grasp description matrix G_r relates the forces exerted by the fingertip on the object at the contact point f_c and the resultant generalized force on the object F_r (Figure 2).

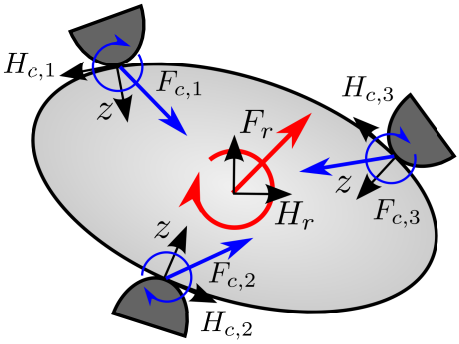


Fig. 2. Description of applied forces $f_{c,i}$ and resultant wrench F_r when manipulating an object

This matrix may be calculated by applying the Screw theory. This theory makes use of generalized force and generalized velocity vectors [1]. It is important to notice the notation that will be used in the following formulation, where the superscript b in a vector denotes that such vector is defined in the body frame (H_r), unlike the superscript s that denotes that it is defined in spatial coordinates (in this case, we our spatial reference is the palm reference frame H_p). According to this, it is possible to define the resultant generalized force as:

$$F_r^b = \sum_{i=1}^{N} G_{r,i} f_{c,i}^s = G_r f_c^s$$

, where N is the number of contact points, $G_r = \begin{bmatrix} G_{r,1} \, G_{r,2} \, ... \, G_{r,N} \end{bmatrix}$ ($\mathbb{R}^{6x3N}$) and $f_c = \begin{bmatrix} f_{c,1} \, f_{c,2} \, ... \, f_{c,N} \end{bmatrix}$. In addition, each of the components $G_{r,i}$ in G_r are calculated using the already mentioned Screw theory:

$$G_{r,i} = \mathrm{Ad}_{H_{r,c_i}^{-1}}^T B R_{c,i}^T$$

, where $\mathrm{Ad}_{H_{X,Y}}$ denotes an adjoint transformations of a coordinate frame X w.r.t. a coordinate frame Y and B is the wrench basis that represents the contact model. Three main contact models exist. In the case of grasping and in-hand manipulation with most majority of robotic hands and gripers, it is possible to consider the PCWF (point contact with friction) model. In such cases, B belongs to $\mathbb{R}^{6x3}$) and is defined as:

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

[1] A generalized force consists of a linear force f and a moment τ ($F = \begin{bmatrix} f \, \tau \end{bmatrix}^T$). A generalized velocity consists of a linear velocity v and an angular velocity ω ($V = \begin{bmatrix} v \, \omega \end{bmatrix}^T$)

When this version of the B matrix premultiplies a generalized force, the force component of the new generalized force is the force component of the old generalized force, unlike the torque component which becomes the zero vector. This matches the PCWF model, where only forces are transmitted.

1.3 The Hand Jacobian

The hand Jacobian $J_h \in \mathbb{R}^{m \times n}$ is responsible for relating the fingers joints angular velocities $\dot{\theta}$ to the linear velocities of the fingertips v_f. As mentioned before, and according to the Principle of Virtual Work, its transpose is used to relate the transmitted forces on the the contact points f_c to the joint moments τ:

$$v_c = J_h \dot{\theta} \qquad \tau = J_h^T f_c$$

, The hand Jacobian J_h is defined as follows:

$$J_h = \begin{bmatrix} B^T J^s_{pf_1}(\theta_{f_1}) & & 0 \\ & \ddots & \\ 0 & & B^T J^s_{pf_k}(\theta_{f_k}) \end{bmatrix}$$

, where $J^s_{pf_i}$ is the Jacobian for the i^{th} fingertip in spatial coordinates (in this formulation, the palm reference is used).

1.4 The Fundamental Grasp Constraint

The fundamental grasp constraint equation gathers all the already described concepts, relating the finger joints angular velocities to the object generalized velocity:

$$J_h(\theta)\dot{\theta} = G^T(x_r)\dot{x}_r,$$

, where the vector $\dot{\theta} \in \mathbb{R}^n$ denotes the finger joint angular velocities and $x_r := H_r$ is the configuration of the object frame w.r.t. to the palm frame[2].

2 Object-Level Impedance Controller Using the Virtual Linkage Parametrization of the Null-Space

Even in the case of non-redundant fingers, the contact forces applied over the object conforms a redundant system. This implies that the null-space of the G matrix is not the null vector, which in other words means that there exist infinite combinations of contact forces that can satisfy a desired object wrench. Most grasp controllers presented in the literature solve this redundancy by means of the inversion of the grip map [7]. The grip map is a matrix Q formed by

[2] The implementation presented in this work uses the palm reference frame as the global reference frame. It is possible to choose any other reference frame

the grasp description matrix G_r stacked together with the pseudo-inverse of a parametrization E of the null-space of the grasp map:

$$Q = \begin{bmatrix} G \\ E^{\dagger} \end{bmatrix}$$

2.1 Virtual Linkage Null-Space Parametrization

One method of parametrizing the null-space of the grasp map is the virtual linkage [11]. Such approach is based on construction of a closed chain mechanism formed by virtual springs that links all contact points (Figure 3). These virtual springs produce forces on the object that are translated into a null net wrench in the object frame, which makes the object be at static equilibrium (if no other forces are being applied). This way, the controller can control, in addition to the desired resultant wrench on the object, the tensions of the links in the Virtual Linkage model. Such tensions contribute directly to the application of internal forces without affecting the resultant wrench on the object.

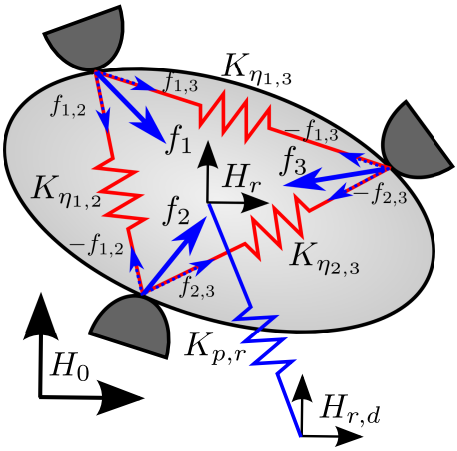


Fig. 3. Detail of the virtual springs considered in the controller. $K_{\eta,ij}$ corresponds to the springs of the Virtual Linkage model, while K_r corresponds to a spatial spring that pulls the object reference frame towards the global reference frame.

2.2 Derivation of the Grip Map Using the Virtual Linkage Null-Space Parametrization

Forces[3] acting on the contact points f_c may be decomposed into internal forces f_{ci} which cause only tensions and forces f_{ce} which do not produce any internal forces and, therefore, produce a total net wrench on the object.

[3] In this section, a contact model based on contact point with friction has been assumed. Nevertheless, a similar derivation can be obtained for other contact models

$$f_c = f_{ci} + f_{ce}$$

We denote E as the matrix that relates the tensions t in the Virtual Linkage model with the produced internal forces f_{ci}. In the case of having four contact forces, E is:

$$E = \begin{bmatrix} e_{12} & e_{13} & e_{14} & 0 & 0 & 0 \\ -e_{12} & 0 & 0 & e_{23} & e_{24} & 0 \\ 0 & -e_{13} & 0 & -e_{23} & 0 & e_{34} \\ 0 & 0 & -e_{14} & 0 & -e_{24} & -e_{34} \end{bmatrix}$$

, where $e_{j,k} = (p_{c,j} - p_{c,k})/\|p_{c,j} - p_{c,k}\|$ represent all the unit vectors that point from each contact point to all the others.

Taking into account that $f_{ci} = Et$ and $f_{ce} = G^{\dagger}F_r$ [4], thus:

$$f_c = Et + G^{\dagger}F_r$$

It is possible to compact this expression in the following way:

$$f_c = \begin{bmatrix} G^{\dagger} & E \end{bmatrix} \begin{bmatrix} F_r \\ t \end{bmatrix} = Q^{-1} \begin{bmatrix} F_r \\ t \end{bmatrix}$$

It also possible to formulate this deduction the other way round, what would end up with the following expression:

$$\begin{bmatrix} F_r \\ t \end{bmatrix} = \begin{bmatrix} G \\ E^{\dagger} \end{bmatrix} f_c = Q f_c$$

The matrix Q is called the grip map. In the general case, it relates the generalized forces at the contact points f_c with the resultant wrench at some reference point on the object plus some additional variable. In the case of the Virtual Linkage parametrization model, this additional variable refers to the tensions on the links that conform such model. This matrix is square and full-rank, thus invertible, under some conditions [13]:

- If having two contact forces, the contact points must not be coincident.
- If having three contact points, the contact points must not be co-linear.
- If having four contact points, the contact points must not be co-planar.

2.3 Object-Level Impedance Controller Implementation

The implementation of the object-level impedance controller was performed in a very similar way to the one described in [12]. The general control scheme is presented in Figure 4.

[4] Taking the pseudo-inverse ensures that $G^{\dagger}F_r$ is a solution with minimum norm. It should be obvious to deduce that a minimum norm solution does not produce any internal force

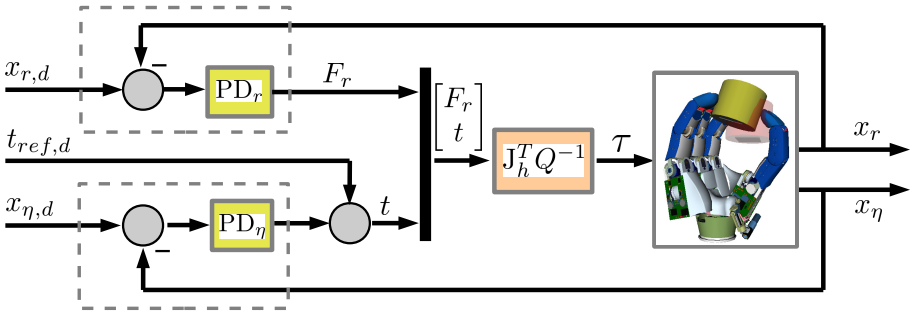


Fig. 4. General control scheme of the implemented object-level impedance controller

The task states x is formed by the coordinates of the object x_r and the coordinates of the anchors of the virtual linkage (contact points coordinates) $x_\eta \in \mathbb{R}^{(N1)N/2}$, that is $x = \begin{bmatrix} x_r & x_\eta \end{bmatrix}^T$. The vector $x_d = \begin{bmatrix} x_{r,d} & x_{\eta,d} \end{bmatrix}^T$ contains the corresponding desired values. The object-level control scheme uses two proportional-derivative controllers to reduce the task space error $\Delta x = x - x_d$. The objective of the first PD controller is to calculate the necessary joint torques to move the object to the desired pose $x_{r,d}$. In addition, it is necessary to add some joint torques to fulfil the specified tensions requirements t_{ref}. Nevertheless, a second PD controller modifies the final applied tensions to ensure that $x_{\eta,d}$, that is the positions of the contact points, is also achieved. Generally, the desired positions of the contact points is fixed. This is important as prevents sliding due to inappropriate specified tensions. Notice that is the rest lengths of the corresponding springs. The control law calculates the necessary joints torques τ for reducing the task space error x and, at the same time, fulfilling the tensions constraints t. Such control law is derived using the hand Jacobian J_h and the grip map Q:

$$\tau = J_h^T F_c \tag{1}$$

$$\tau = J_h^T Q^{-1} \begin{bmatrix} F_r \\ t \end{bmatrix} \tag{2}$$

$$\tau = J_h^T Q^{-1} \begin{bmatrix} F_r \\ t_{ref} + t_{x_\eta} \end{bmatrix} \tag{3}$$

$$\tau = J_h^T Q^{-1} \begin{bmatrix} K_{p,r} \Delta x_r + K_{d,r} \frac{\partial \Delta x_r}{\partial t} \\ t_{ref} + K_{p,\eta} \Delta x_\eta + K_{d,\eta} \frac{\partial \Delta x_\eta}{\partial t} \end{bmatrix} \tag{4}$$

3 Experimental Results

For the purpose of validating the implemented object-level controller based on the Virtual Linkage parametrization, a simulation set-up using the DLR-Hand

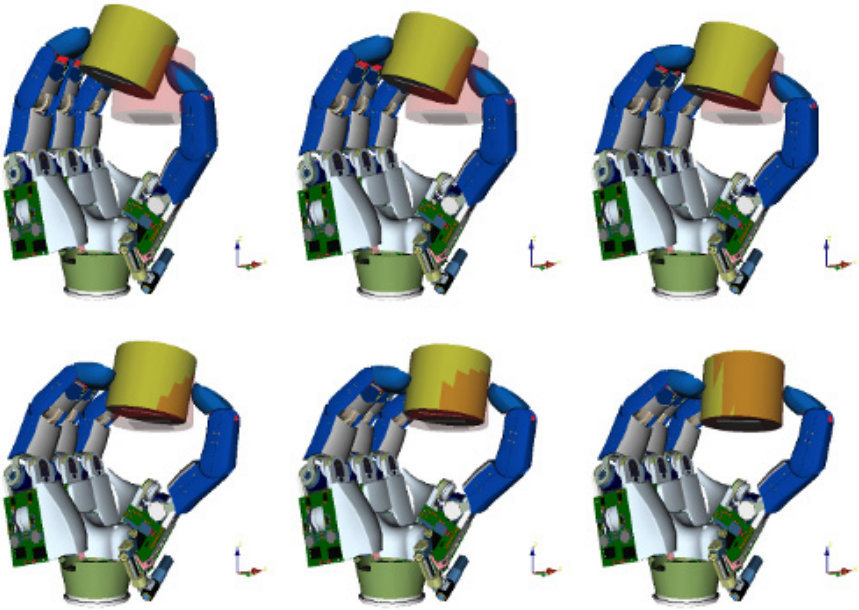


Fig. 5. Simulation sequence of the in-hand manipulation task using the DLR-Hand II. Changes both in position and orientation were part of the task.

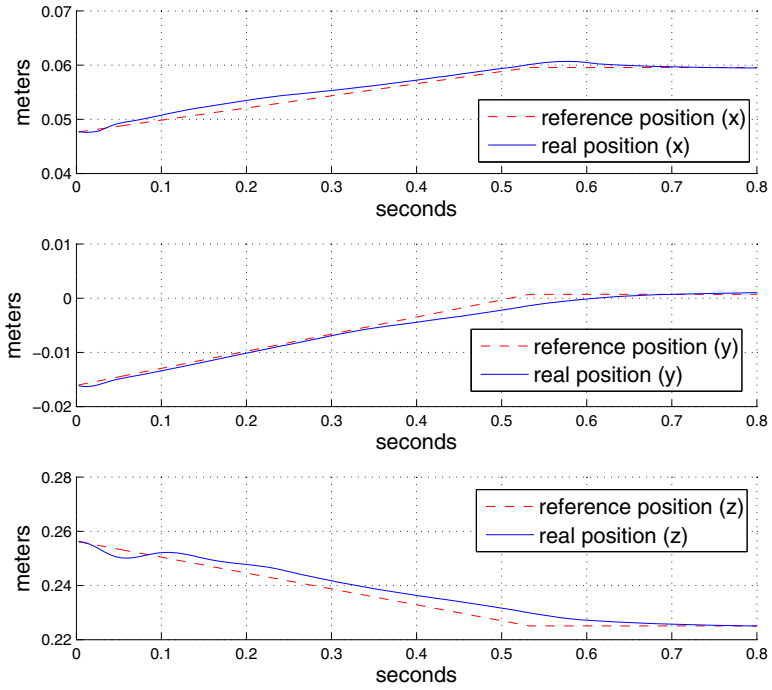


Fig. 6. Response of the tracking controller concerning position values of the reference trajectory

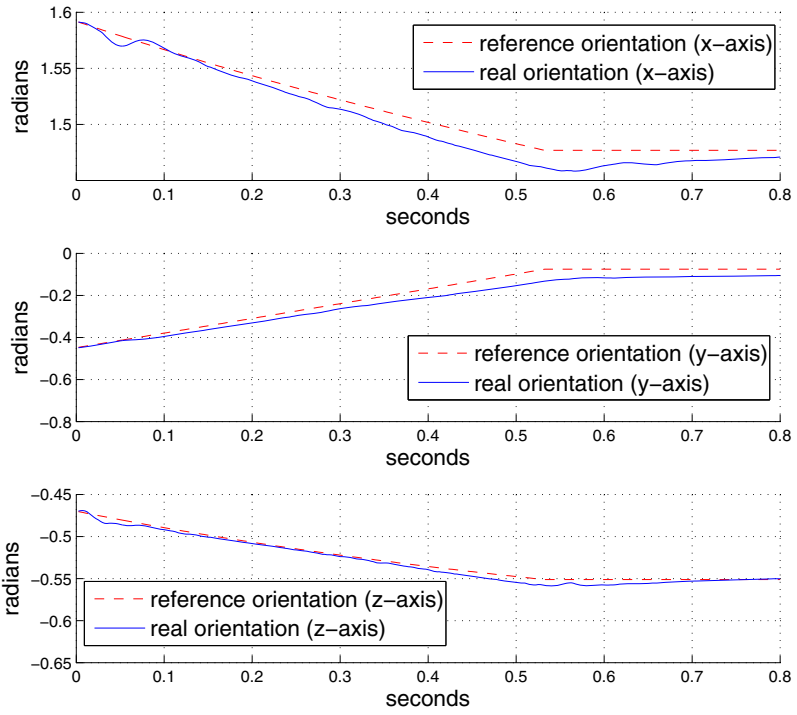


Fig. 7. Response of the tracking controller concerning position values of the reference trajectory

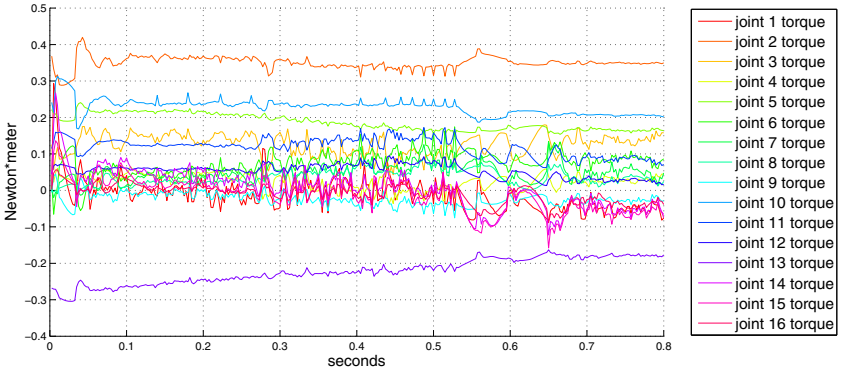


Fig. 8. Evolution of the joint torques of the finger along the control experiment

II[5] was employed. The Openrave simulator [2] configured with the ODE physics dynamics engine was used. An initial grasp configuration, which was known to be stable, was used to define the initial state of the system (Figure 5). The goal of the experiment presented here was to change both the position and orientation of the object within the hand, while maintaining some specific internal forces (parametrized by the tensions of the Virtual Linkage model). In this specific experiment, the desired tensions were fixed to some appropriate values calculated by a grasp force optimization procedure. As it is possible to see in Figure 7, the controller is capable of tracking the desired reference trajectory (red dashed lines). Obviously, the performance of the controller depends on the tuning of all the parameters of the controller, mainly t_{ref}, $K_{p,r}$, $K_{d,r}$, $K_{p,\eta}$ and $K_{d,\eta}$. Nevertheless, there exist some small errors in the response. These errors are caused mainly, but not only, by the assumption that the contact points coincide with the fingertip positions. However, the geometry of the fingertips prevents this assumption to be completely true, thus introducing some errors in the contact points positions that are used in the calculation. Furthermore, sliding and rolling is also produced, what makes this error to vary along the manipulation.

The evolution of the torques applied to each joint are shown in Figure 8.

4 Conclusions and Future Work

The implementation of object-level impedance controllers, based on a Virtual Linkage model, seems to be, in the opinion of the authors, the most reasonable solution to achieve dexterous in-hand manipulation with robotic hands. For the purpose of supporting this affirmation, an experimental validation of a specific controller implementation based on the Virtual Linkage model has been performed. The experiments were carried out in simulation with the DLR-Hand II. The Openrave simulator and the ODE physics dynamics engine were employed in the experiments. The results show that it is possible to achieve a reasonable response in position and orientation tracking tasks. Furthermore, it is also feasible to maintain some desired internal forces and contact points. Nevertheless, the tracking error may still be improved by adding integral control capabilities to the PD controllers that were used in the implementation.

Acknowledgments. The research leading to these results has been partially supported by the HANDLE project, which has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013)

References

1. Chen, C., Walker, I., Cheatham, J.: A new approach to force distribution and planning for multifingered grasps of solid objects. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, vol. 1, pp. 890–896 (1991)

[5] It is necessary to remark that, unlike the real DLR-Hand II, the employed model did not have any coupled joints. The original coupled joints were substituted by fully actuated joints

2. Diankov, R.: Automated Construction of Robotic Manipulation Programs. Ph.D. thesis, Carnegie Mellon University, Robotics Institute (August 2010), http://www.programmingvision.com/rosen_diankov_thesis.pdf
3. Guo, G., Gruver, W.: Fingertip force planning for multifingered robot hands. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, vol. 1, pp. 665–672 (1991)
4. Kao, I., Yang, F.: Stiffness and contact mechanics for soft fingers in grasping and manipulation. IEEE Transactions on Robotics and Automatio 20(1), 132–135 (2004)
5. Kerr, J., Roth, B.: Analysis of multifingered hands. The International Journal of Robotics Research 4(4), 3–17 (1986), http://ijr.sagepub.com/content/4/4/3.abstract
6. Kumar, V., Waldron, K.: Force distribution in closed kinematic chains. IEEE Journal of Robotics and Automation 4(6), 657–664 (1988)
7. Mason, M.T., Salisbury, J.K.J.: Robot hands and the mechanics of manipulation (1985)
8. Montana, D.: The kinematics of multi-fingered manipulation. IEEE Transactions on Robotics and Automatio 11(4), 491–503 (1995)
9. Nakamura, Y.: Minimizing object strain energy for coordination of multiple robotic mechanisms. In: American Control Conference, pp. 499–509 (1988)
10. Peshkin, M., Sanderson, A.: Manipulation of a sliding object. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 233–239 (1986)
11. Williams, D., Khatib, O.: The virtual linkage: a model for internal forces in multi-grasp manipulation. In: Proceedings of the 1993 IEEE International Conference on Robotics and Automation, vol. 1, pp. 1025–1030 (1993)
12. Wimböck, T., Ott, C., Albu-Schäffer, A., Hirzinger, G.: Comparison of object-level grasp controllers for dynamic dexterous manipulation. Int. J. Rob. Res. 31(1), 3–23 (2012), http://dx.doi.org/10.1177/0278364911416526
13. Yoshikawa, T.: Virtual truss model for characterization of internal forces for multiple finger grasps. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2389–2395 (1998)

Affordance-Based Grasp Planning for Anthropomorphic Hands from Human Demonstration

Norman Hendrich[1] and Alexandre Bernardino[2]

[1] Computer Science Dept., University of Hamburg, Germany
hendrich@informatik.uni-hamburg.de
[2] Institute of Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal
alex@isr.ist.utl.pt

Abstract. Robust grasping and manipulation using multi-finger hands remains one of the key challenges of service robotics. So far, most theoretical approaches and simulators have concentrated on the search for static stable grasps, but without consideration of the task context or the different grasp types to be used during object manipulation.

In this paper we describe a simple approach to derive human-like grasps for anthropomorphic hands from human teleoperation on a set of test objects. Experimental data is presented for the Shadow Dexterous Hand and eight different precision grasps that explore a large part of the reachable hand workspace. Analysis of the data shows a surprisingly simple correlation of the hand-object poses with the object size. Given a target object of known size, our algorithm provides a suitable hand-object pose and approach direction, and both the finger pre-shapes and approximate grasp-shapes required for grasp execution.

Keywords: robot grasping, grasp planning, dexterous manipulation, grasp synergies, robot workspace analysis.

1 Introduction and Related Work

The capacity of the human hand to grasp and manipulate objects, known or unknown, and of widely different sizes, shapes and materials is unmatched. Despite significant progress in the design and control of multi-finger robot hands, their application in service-robotics is still limited by the complexity of grasp-planning for a given object and the task context.

In analytical approaches to grasp-planning, a *grasp* is formally defined as a set of contact points on the surface of the target object together with friction cone conditions [1]. The traditional solution to this problem is divided into two stages: first, suitable grasping points on the object are determined, and in the second step a robot hand posture is computed via inverse kinematics to reach those points with the fingertips. See [2] and [3] for extensive reviews. Several contact models and grasp quality measures have been proposed to identify and evaluate different candidate grasps, and an elaborate theory of the kinematics

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_49,

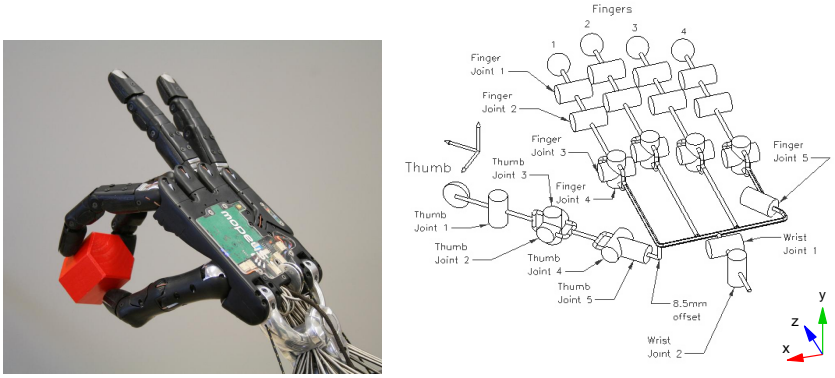


Fig. 1. The 24-DOF Shadow Hand. The fingers have three flexure joints and one abduction joint each, with the distal joint underactuated from the medial joint. The thumb has five independently controllable joints.

and dynamics of finger and objects movements is available. Given the kinematics structure of a robot hand, any finger posture is fully specified by the joint angles, and can be thought of as a point in a high-dimensional joint space. For example, the Shadow Dexterous Hand [4] is designed to match a typical human hand in size and all its movements. Successful grasping of everyday objects with the Shadow Hand has been demonstrated [5], but in-hand re-grasping or the use of tools is far beyond the state of the art. With 24 degrees of freedom for the hand itself and another 6-DOF to specify the hand-object pose, grasp-planning for the Shadow hand corresponds to searching a 30-dimensional space.

The second approach to grasp planning is motivated by the way that humans grasp, and relies on empirical studies and classification of human manipulation tasks Typically, the manipulation task is divided into different phases, e.g. pre-shape, grasp, and stabilization of an object [6]. Analyzed human strategies can then be mapped to a robot hand, and complex behaviour is created by sequencing and combining basic motion primitives [7]. In their classical study, Santello et. al. [8] asked several test subjects to shape their hands as if to manipulate imaginary everyday objects, while the hand poses were recorded with a data-glove. The study clearly demonstrated that the fingers were shaped using certain patterns, or *postural grasp synergies*. A Principal Component Analysis of the recorded data showed that the two first principal components accounted for more than 80% of the variance, strongly suggesting that the grasp postures used by the humans could be approximated by a 2-dimensional basis. This fact is also refleced in the classical grasp taxonomies [9], where only a handful of different poses are sufficient to explain the hand motions used by humans for grasping. Based on this idea, the concept of *eigengrasps* for robot hands for grasp-planning was introduced in [10]. The approach was refined in [11] and can dramatically reduce the effective dimension of the parameter space for grasp

generation. Running the GraspIt! simulator [12] with this search technique, grasps could be generated for thousands of 3D models, using a model of the human hand as well as different robotic hands. This collection of pre-calculated form closure grasps has been published as the Columbia Grasp Database [13]. The authors also suggest to use an object's 3D geometry as an index into the database, so that finding suitable grasps for a new object turns into a database lookup.

Regarding the task context, another important area of current research is the modelling of object affordances, and the integration of the affordances and relevant object properties into the grasp generation and grasp quality evaluation [14]. For example, object shape is used in [15]. A flexible grasp quality estimation based on a weighted sum of different criteria is described in [16] and has been implemented in our in-house grasp simulator [17]. Recently, [18] introduced a framework for compliant physical interaction, connecting manipulator motion with task context, objects attributes, and constraints. A multi-layered approach to grasp planning is described in [19], which also includes an extensive manipulator workspace analysis of two-arm manipulation.

In this paper we concentrate on the problem of predicting suitable hand-object poses for the Shadow hand as a function of object diameter and the selected grasp-type. Unlike simple parallel grippers, where the grasp point is clearly defined, very different parts of the hand workspace may be used on multi-fingered hands, depending on grasp type and object size. In our approach, reference grasps are generated using a teleoperation-based methodology, where the human experimenter controls and adjusts the robot hand using a calibrated data-glove, while data is recorded directly from the robot joint angles. The key idea is that the human experimenters learn to compensate errors in the glove-to-robot map, thus bypassing the correspondence problem from the unknown hand kinematics of the experimenter to the kinematics of the robot. We recorded grasp data for eight different dexterous grasp classes, including pinch, tripod and lateral grasps. The final grasp datasets only include *human-like* grasps on the target objects, and the grasps correspond to those parts of the overall hand workspace that the humans considered best for in-hand manipulation of the objects. As shown below, we can directly calculate an appropriate grasp position (x, y, z) for objects of given size, reducing this part of the grasp-planning problem from 6-DOF to 3-DOF. Additionally, symmetry and task constraints can be used to generate hand-object orientation, and a suitable grasp center point can be calculated immediately. As reported in a companion paper [20], we also extracted postural synergies for all studied grasp types from the demonstrated grasps, with 2..6 synergies required to re-create the original finger poses.

The rest of this paper is organized as follows. First, section 2 describes the experiment setup and the approach to derive human-like grasps from teleoperation. Section 3 provides the workspace analysis of the hand and shows that linear-regression is sufficient to recover the hand-object pose for given grasp-type and the size of the target object.

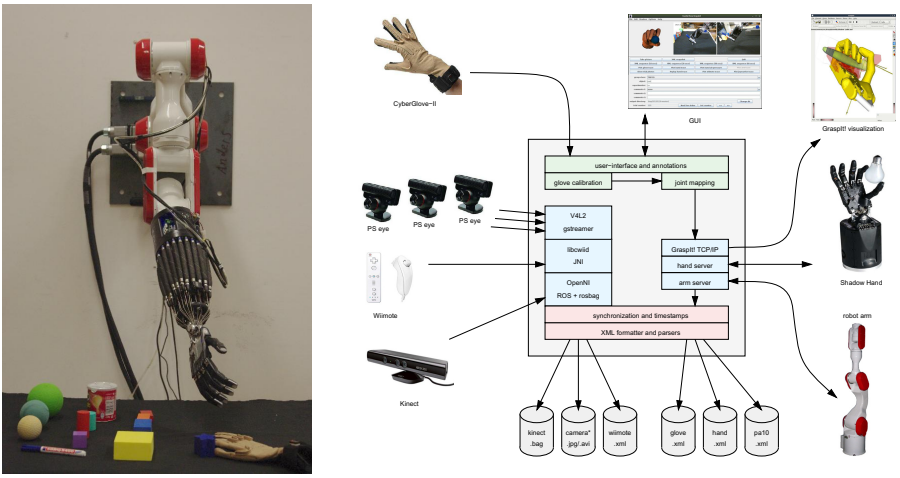


Fig. 2. (a) The experiment setup with the Shadow C5 air-muscle hand mounted on the PA10-6C robot arm and teleoperated via the CyberGlove-II. The multi-camera system for pose-tracking is not shown. (b) Overview of the main components. The recording software synchronizes the sensor data, controls the robot, and generates the grasp datasets.

2 Experiment Setup and Approach

This section describes the experiment setup used to record the human demonstrations of the grasps required for the subsequent robot workspace analysis and for extraction of postural synergies. We list the hardware components and describe the recording software, including the data-glove calibration approach.

2.1 Hardware

See figure 2 for a photo of our experiment setup. The Immersion *CyberGlove-II* data-glove [21] is used to teleoperate the hand. The vision system used for the experiments consists of three *Sony PSeye cameras* in a co-linear setup which are used as two stereo-camera pairs [22]. The cameras provide RGB images of 640x480 pixels at frame-rates of up to 60 fps. Additionally, a Kinect camera is used to record the experiments.

The *Shadow C5 air-muscle hand* closely matches the size and shape of a human hand and provides a total of 24 degrees of freedom, with 4-DOF per finger and 5-DOF for the thumb, as well as 2-DOF for the wrist and 1-DOF for palm flexure. See figure 1 for a photo of the hand and the kinematics diagram. The distal phalanges of the fingers are underactuated from the medial joints, resulting in a total of 20-DOF controllable. All joints are tendon-driven and the tendons on the C5-type hand are actuated by a pair of McKibben-style air-muscles for each controllable joint. The muscles are elastic and provide passive compliance, resulting in good grasp stability for a large variety of static grasp poses.

Fig. 3. The seven hand poses used for the calibration sequence, as visualized by the Shadow hand model in GraspIt!. The set includes stretched and fully flexed fingers, finger abduction, and selected thumb poses. The last pose is used to calibrate the palm-arch sensor and the extreme range of the thumb.

The newer C6-type hand has the same mechanical structure, but the tendons are driven by electric motors, resulting in faster actuation. The Shadow hand is mounted onto a *Mitsubishi PA10-6C* robot arm. The arm has six rotational joints, has a total reach of approximately one meter and a payload of 10 kg.

2.2 Software

To simplify the recording of the teleoperation experiments, a dedicated software tool was developed. The software provides a graphical user-interface that allows the user to set up, control, and analyse the experiments. The software architecture of the tool is shown in figure 2, where the basic data-flow is from the sensors on the left to the actuators on the right. The joint-angle data from the data-glove is first calibrated and mapped to the kinematics of the Shadow hand. This data is then sent via TCP/IP to the Shadow hand server, and optionally also to the GraspIt! simulator for visualization. All hand and arm data is timestamped and recorded together with the raw data from the glove and the vision sensors. We decided to store all experiment data as XML-files in the format specified in [23]. The data corresponding to an experiment is stored as individual files, one per sensor or device, with a header including device information and calibration information, followed by the data section with the time-stamped raw data.

2.3 Dataglove Calibration

In principle, any means to control the robot hand can be used to perform the grasps required for the analysis, and we have also experimented with marker-based visual tracking of the hand and with direct kinesthetic teaching of grasps on the Shadow C6 hand. However, direct teleoperation of the robot from a calibrated data-glove has proven to be the most efficient way to perform the experiments, because this approach exploits the experience of the humans (and their own grasp synergies).

The *CyberGlove-II* used for the experiments provides a total of 22 sensors, with three flexure sensors per finger, three abduction sensors placed between the fingers, and one palm-flexure sensor. Four sensors measure the thumb position, and two sensors are provided for the wrist. Due to the different sensor layout and sensor resolution, a mapping is required to translate from the CyberGlove

object name and grasp pose	width	height	length	material
big green ball	**86**	**86**	**86**	sponge
medium green ball	**64**	**64**	**64**	rubber
small white ball	**54**	**54**	**54**	sponge
big red cylinder, top	**64**	76	76	metal
big red cylinder, side	64	**76**	**76**	metal
medium green cylinder, top	**38**	38	38	sponge
medium green cylinder, side	38	**38**	**38**	sponge
small red cylinder, top	**59**	27	27	wood
small red cylinder, side	59	**27**	**27**	wood
pen, side	150	**12**	**12**	metal
small purple cube	**30**	**30**	**30**	wood
large blue box, long side	**77**	39	39	sponge
large blue box, short side	77	**39**	**39**	sponge
medium orange box, long side	**60**	30	30	wood
medium orange box, short side	60	**30**	**30**	wood
small red box, long side	**60**	14	29	wood
small red box, short side	60	**14**	29	wood
small red box, medium side	60	14	**29**	wood
large yellow box, short side	80	80	**38**	sponge
large yellow box, long side	**80**	**80**	38	sponge

Fig. 4. Attributes of the prototype objects from the IST object set used for the recordings. The objects include three basic shapes (sphere, cylinder, box), different object diameters matched to typical human grasping tasks, and also different materials (wood, metal, rubber, sponge). Values in bold indicate the object dimensions along which opposition forces are applied. Dimensions are in millimeters.

sensor values to the corresponding Shadow hand joint angles. Even for a single test-person, a slightly different fit of the glove will result in differing measured joint angles between different experiment runs.

The glove calibration used for this work is based on [24]. As a perfect match between human demonstration and the Shadow hand proved impossible, the goal of the calibration is to provide a mapping that allows the user to adjust the requested grasp poses quickly, and with acceptable precision. For the finger flexure sensors, a 1:1 mapping was found to be sufficient to translate recorded CyberGlove angles to robot joint angles. For other sensors, in particular the finger abduction, hand-flexure, and thumb positions, a 2:1 mapping from CyberGlove values to Shadow hand joints is used. During the *interactive calibration sequence* the software displays images of a set of carefully selected hand poses, see figure 3, and optionally demonstrates the hand poses on the Shadow hand itself. For each hand pose, the user is asked to shape her hand accordingly, and the joint-values from the glove are sampled and stored. Once all hand poses from the calibration set have been recorded, the software uses regression to calculate the mapping from the glove to the robot hand.

2.4 Benchmark Object Set

For the recordings targeting the postural synergies, we used a set of twelve prototype objects. See figure 4 (b) for a photo of all objects and refer to table 4 for the relevant object attributes and dimensions. As we are interested in relations between object affordances and the grasps that humans use, the set includes three different shapes (sphere, cylinder, box) and also typical materials.

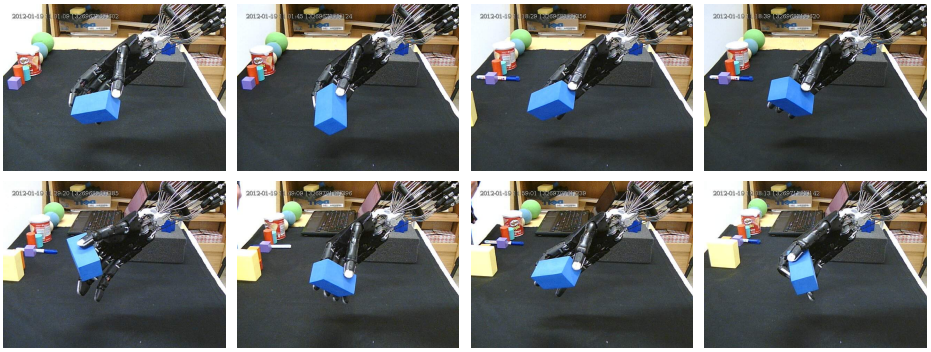


Fig. 5. Example grasps performed on the *large blue box* object: (a) tripod, long side, (b) tripod, short side, (c) palmar-pinch, long side, (d) palmar-pinch, short side, (e) lateral, short side, (f) parallel extension, short side, (g) tip-pinch, long side, (h) lateral-tripod

3 Workspace Analysis of the Shadow Dexterous Hand

The teleoperation approach described above was used to record several complete sets of grasps on the benchmark objects. The experiments concentrated on eight different dexterous fingertip grasp-classes (palmar pinch, tip pinch, tripod, writing tripod, lateral, lateral tripod, adduction grip, parallel extension).

For most of the grasp classes and test objects, the control of the fingers and thumb was straightforward, and the experimenters quickly found stable and robust grasps with the Shadow hand. However, the fine control of the thumb left room for improvement for some grasps (e.g. tripod) and the experimenters learned to tweak their grasp in order to reach suitable poses for the Shadow hand. Figure 5 shows several example grasps performed on a simple box object that were accepted as suitably human-like.

The final dataset for analysis of hand workspace consists of the recorded joint-angles for 20 different object configurations and 8 different grasp-types, controlled by 4 experimenters. A total of 438 samples were acquired, as not all grasp types could be applied to all objects and some subjects missed or repeated some grasps. These samples were split into data matrices A_i each corresponding to a different grasp type g_i, and containing in each row the vector of N joint angles acquired in each trial.

Running a Principal Component Analysis on the recorded grasps then provided us with the synergy matrices, as well as correlations of grasp types with object size and shape, and therefore a first correlation to extract object affordances. Please refer to the technical report [25] for details and the full analysis. As expected, the results show that most of the variance of the grasp data is concentrated in the first few synergies. To explain 90% of the variance, and thus achieve mean square approximation errors below 10%, 6 principal components are sufficient for all studied grasps [20]. For the adduction grip type, due to its simplicity, we can achieve the same low approximation errors with only the

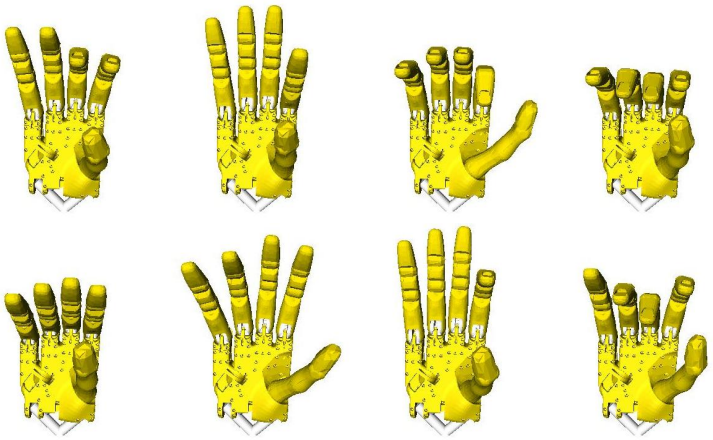


Fig. 6. Grasp poses corresponding to the origin of eigenspace for the eight reconstructed precision grasps: *tripod*, *palmar pinch*, *lateral*, *writing tripod*, *parallel extension*, *adduction grip*, *tip pinch*, and *lateral tripod*.

first two principal components. Correspondingly, the complexity of searching a suitable finger configuration is reduced from 24-DOF in joint-space to about 2..6-DOF in synergy space. It should be noted that standard PCA was chosen both for its simplicity and the ease of converting from joint-space to synergy-space and back. Results on more complex techniques for dimensionality reduction and synergy extraction will be reported in an upcoming paper.

The grasp poses corresponding to the origin of the eigenspace for the eight grasp classes are shown in figure 6, as visualized in the GraspIt! simulator. These grasp poses can be understood as the mean hand shapes averaged over all recordings for the given grasp type, and provide the reference hand-shape for interpolation.

3.1 Reconstruction of Grasp Center Point

To execute reach-to-grasp motion, the relative hand-object pose must be calculated, which as usual can be split into a translation (x, y, z) and an orientation $(\theta_x, \theta_y, \theta_z)$ between the center of the object and the palm-coordinate frame of the hand. Once this information is available, traditional inverse kinematics calculations can be performed to find solutions for the arm and wrist configuration to reach a given object at position (x, y, z) in the world coordinate system. Additionally, we also have to find a suitable approach direction of the hand towards the object. The reconstruction of the hand-object pose and estimation of approach vectors is described now.

No absolute object tracking system (e.g. Polhemus) was available for the human demonstration recordings, and therefore the hand-object pose could not be measured directly. Instead, we have to reconstruct the hand-object pose from the existing datasets. All objects used during the training sessions are simple

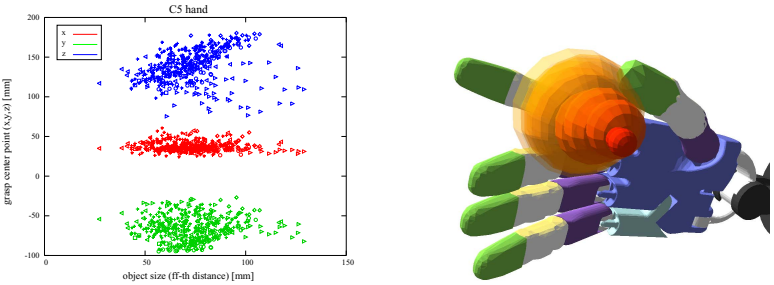


Fig. 7. Reconstructed hand-object pose as a function of the object size. The cluster plots show the (x, y, z) coordinates of the object center relative to the palm coordinate system for each human-demonstrated grasp. Left: all grasps from the demonstration session. Right: Visualization of hand-object pose for the lateral tripod grasp, where the diameter of the semi-transparent spheres corresponds to object size.

geometric shapes (sphere, cylinder, box), and the size of the objects is known. We now assume that all experimenters adjusted the grasps to be approximately symmetrical according to the object shape, before accepting the grasps as human-like. Also, we assume that all fingers taking part in a grasp according to the given grasp class did actually contact the object.

Under these two assumptions, we can run forward-kinematics on the recorded grasp poses to find the cartesian coordinates of the phalanges and the fingertips of the hand, and then find the geometric center between the fingertips involved in the grasp. For example, the pinch-grasps calculate the center between the tips of the first finger and the thumb, while the variations of the tripod grasps use the thumb, first finger, and middle finger. Of course, for the lateral and lateral-tripod grasps, the medial phalange of the first-finger is used as the relevant coordinate frame in the calculations. The FK solver is available as part of the ROS software stack for the Shadow hand. Based on the kinematics library of the Orocos project [26], the solver provides the forward kinematics directly and also an iterative inverse kinematics solver useful for grasp adaption.

The result of this analysis is shown in figure 7, where the reconstructed hand-object pose is plotted for all grasps of the recorded dataset, and markers indicate the eight different grasp types. The (x, y, z) coordinates are indicated by colors (red, green, blue) and are plotted against the object-size estimated from the distance between the fingertips of the thumb and first finger. For the adduction grip, the distance between the fingertips of the first and middle fingers is used. A right-handed coordinate system aligned with the fingers is assumed, compare Fig. 1. When viewing from the back of the palm,

- the *origin* $(0, 0, 0)$ is at the WRJ1 joint of the Shadow hand,
- x runs to the left (along the stretched thumb),
- y is up (from the palm), and therefore actual y values are negative,
- z runs along the palm from the wrist to the tip of the middle finger.

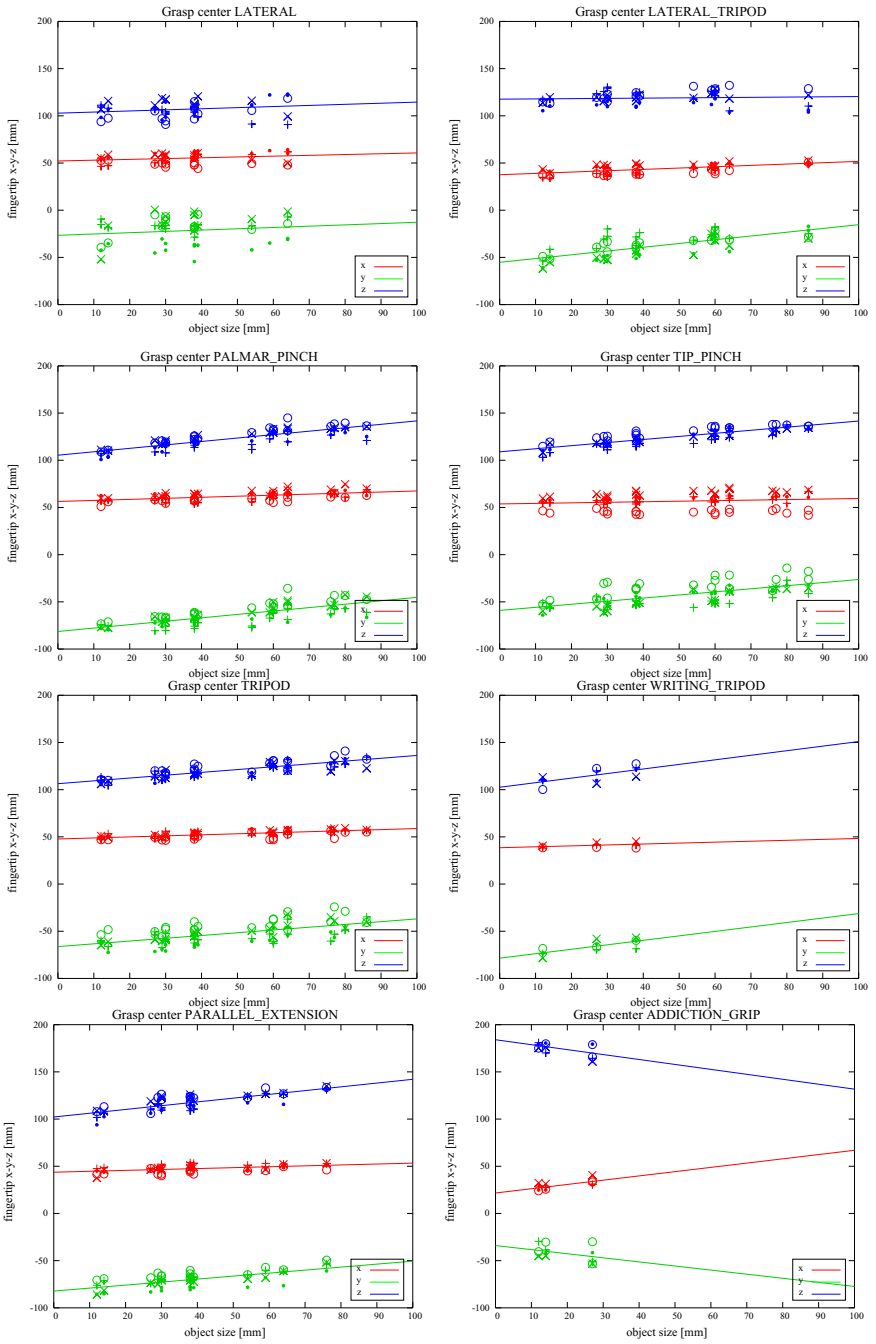


Fig. 8. Grasp center point vs object size for the eight grasp types. The plots show the reconstructed object centers for the human demonstrations and the linear least-squares regression for the data. The (x, y, z) coordinates are plotted in (red, green, blue), while the different markers identify the test persons performing the grasps.

Table 1. Estimation of the grasp center point (x, y, z) as a function of object size and grasp type for the Shadow C5 hand. The parameters give the linear least-squares approximation to the grasp-center point from the teleoperation experiments.

ID	Grasp class	x_0	Δx	y_0	Δy	z_0	Δz
08	PALMAR_PINCH	56.32	0.11	-81.42	0.36	105.38	0.36
13	TRIPOD	47.72	0.11	-66.28	0.29	106.32	0.30
15	LATERAL	52.14	0.09	-26.60	0.14	102.82	0.12
19	WRITING_TRIPOD	38.49	0.10	-78.52	0.47	102.54	0.48
21	PARALLEL_EXT.	43.71	0.10	-82.29	0.32	102.17	0.40
22	ADDICTION_GRIP	21.64	0.45	-33.99	-0.43	184.01	-0.52
23	TIP_PINCH	53.70	0.06	-59.12	0.33	108.95	0.33
24	LATERAL_TRIPOD	37.54	0.14	-55.24	0.40	117.61	0.03

Table 2. Estimated hand approach vectors for the studied eight grasp-classes

ID	Grasp class	(a_x, a_y, a_z)
08	PALMAR_PINCH	$(0, -1, 0)$
13	TRIPOD	$(0, -1, 0)$
15	LATERAL	$(0.71, 0, 0.71)$
19	WRITING_TRIPOD	$(0, -1, 0)$
21	PARALLEL_EXT.	$(0, -0.71, 0.71)$
22	ADDICTION_GRIP	$(0, -0.71, 0.71)$
23	TIP_PINCH	$(0, -1, 0)$
24	LATERAL_TRIPOD	$(0, -1, 0)$

As expected, the human demonstrated grasps cover a large part of the overall hand workspace, with about 100 mm range in z and 50 mm range in y. However, the rather small variance for the x coordinates is surprising, given that the thumb of the Shadow hand has a total working range of almost 200 mm.

Figure 8 shows a separate analysis for all eight recorded grasp-classes. Again, the (x, y, z) components are plotted in (red, green, blue) color, while the object size is now taken from the known dimensions of the test objects, and markers correspond to the experimenters. The figure also plots the linear regression through the dataset, averaged over all demonstrations for each grasp type. The inter-subject differences are small, but some interesting differences can be seen. For example, there are two different choices for the x coordinates in the tip-pinch grasp, taken by two experimenters each. The large variance in y for the lateral grasp is easily explained, as the exact value of y doesn't matter much for this grasp type. There are only very few data points for adduction grip, but this grasp uses the first and middle fingers basically as a standard gripper and is the least interesting for this study.

Note the clear correlation between grasp type, object size, and grasp point, which is the main result of this paper. Despite the large reachable workspace of the individual fingers and thumb, only a small and well-defined area was selected by the human experimenters, indicating the actual useable workspace

for the different grasp types. The numerical values for the regression parameters are summarized in table 1. This provides us with the (x, y, z) components of the relative hand-object pose as a function of object diameter for all of the recorded grasp types. Using the approximation, the relative hand-object pose does not need to be hardcoded in the object description, but can be calculated on-the-fly for any given object with known or approximately known size.

3.2 Approach Vectors

Additionally, we can also estimate approach vectors for reach-to-grasp motions from our dataset. The proposed solution is to use fixed approach vectors which are orthogonal to the main close-grasp direction for the eight grasp-types. For example, a tripod grasp touches the target object with the thumb, index finger, and middle finger, and the object orientation will be almost parallel to the x-z plane. Therefore, $-y$ is the most suitable approach direction. For the lateral grasps, the human demonstrations show the largest variance, as object of middle size can be grasped in many different orientations. Here, we select an approach vector slightly along the x-z plane, with the hand mostly moving forward but also a bit to the left. Table 2 gives the approach vectors selected for the eight grasp classes studied. Here, the approach vector is considered a property of the grasp type, and as such only needs to be included in the grasp database for objects that impose extra geometric constraints on the reach motion.

4 Integrated Grasp Planning

When combining the postural synergies extracted from the human demonstrations [20] with the grasp center point analysis described in the last section, we arrive at an efficient integrated grasp planning algorithm. The basic idea of grasp-planning based on synergies is to combine a quick search of the reduced subspace spanned by the relevant eigengrasps with a later adjustment phase. This can also be thought of as a hierarchical approach, where the synergies pre-shape the hand with approximate finger positions around the object. The following refinement phase closes the fingers until contact with the object is established, and then adjusts the contact positions and finger-forces to stabilize the object.

Given a grasp-class, hand pre-shape, and the approach vector, we can plan the approach and close-to-grasp motions for the hand and fingers. While this can be done using open-loop control with pre-planned trajectories, the integration of tactile feedback from the fingertip sensors will provide a much more robust solution. We are currently implementing a two-level scheme where the approach motion is slowed down and adjusted whenever the tactile sensors indicate a hand-object contact during the early approach phase, while the fingers are still in the pre-shape pose.

We have updated the Shadow hand model in GraspIt! to match the current C5/C6-type hands, and our software converts the postural synergies into the required .egr file format. The standard user interface is then used to load a

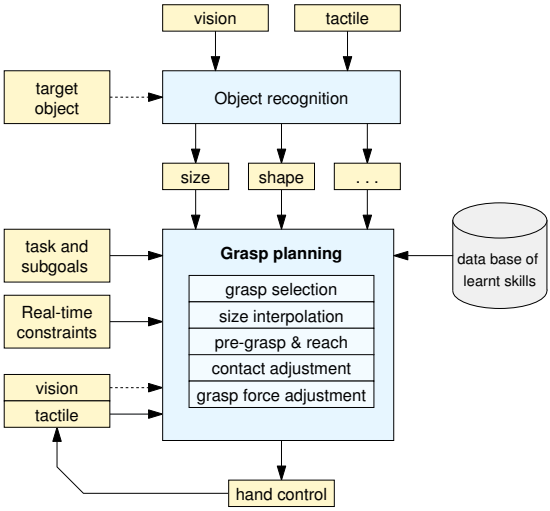


Fig. 9. When facing a new task, the robot looks-up a prototype grasp matching the approximate object-type and -size, as well as any additional task contraints. Hand-object pose and approach vector are derived from the hand workspace analysis, while the initial hand-shape is derived from interpolation of the recorded prototypes. Grasping is performed using tactile feedback and local finger position adaptation using a suitable goal function (e.g. independent contact regions).

3D-model of the target object, to define the desired contact locations, and to specify the parameters (synergies and hand position/orientation) to include in the grasp search.

5 Summary and Future Work

This paper introduces a simple approach for grasp and pre-grasp planning planning for anthropomorphic robot hands, based on the analysis of human demonstrations on a set of test objects of different shapes and sizes. For each object, the human experimenter adjusts the robot fingers using teleoperation until a *human-like* grasp is reached.

Eight different precision grasp types were recorded in our experiments on both the Shadow C5 (air-muscle) and C6 (electric motors) Hands. Our analysis recovers the hand-object pose from symmetry considerations and the forward kinematics of the hand. For all recorded grasps, the reconstructed object position shows a clear correlation to grasp-class and object diameter, and also indicates the useable workspace of the Shadow hand. In fact, a simple linear regression was found sufficient to predict the human-like grasps for all data in our experiments.

The algorithm provides the hand-object grasp point for a given object, as well as the hand approach direction for the selected grasp-type and a finger

pre-shape. We are currently working on integrating postural synergies extracted from the human demonstrations to arrive at low-complexity motion primitives suitable for in-hand manipulation.

For the grasping of complex-shaped objects, we propose to initalize a grasp search with the hand-object pose corresponding to grasp type and object size, followed by local grasp-adjustment. This results in a large speedup when compared to a blind search of the large state-space.

Acknowledgements. This work is partially supported by the European project HANDLE ICT-236410, www.handleproject.eu. The authors acknowledge discussions and feedback from all partners of the HANDLE project.

References

1. Ferrari, C., Canny, J.: Planning Optimal Grasps. In: Proceedings of the IEEE Int. Conference on Robotics and Automation, Nice, France, pp. 2290–2295 (1992)
2. Shimoga, K.B.: Robot grasp synthesis algorithms: a survey. International Journal of Robotics Research 15, 230–266 (1996)
3. Bicchi, A., Kumar, V.: Robotic grasping and contact: A review. In: IEEE International Conference of Robotics and Automation, pp. 348–353 (2000)
4. Shadow Robot Dexterous Hand, `http://www.shadowrobot.com`
5. Röthling, F.: Real Robot Hand Grasping using Simulation-Based Optimisation of Portable Strategies. Ph.D Thesis, Technische Fakultät, Universität Bielefeld (2007)
6. Iberall, T.: Human prehension and dexterous robot hands. International Journal of Robotics Research 16, 285–299 (1997)
7. Kondo, M., Ueda, J., Ogasawara, T.: Recognition of in-hand manipulation using contact state transition for multifingered robot hand control. Robotics and Autonomous Systems 56, 66–81 (2008)
8. Santello, M., Flanders, M., Soechting, J.F.: Postural hand synergies for tool use. Journal of Neuroscience 18(23), 10105–10115 (1998)
9. Cutkosky, M.R.: On grasp choice, grasp models, and the design of hands for manufacturing tasks. IEEE Transactions on Robotics and Automation 5, 269–279 (1989)
10. Ciocarlie, M., Goldfeder, C., Allen, P.K.: Dimensionality reduction for hand-independent dexterous robotic grasping. In: Proc. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3270–3275 (2007)
11. Ciocarlie, M., Allen, P.K.: Hand Posture Subspaces for Dexterous Robotic Grasping. International Journal of Robotics Research 28, 851–866 (2009)
12. Miller, A.T.: GraspIt!: A Versatile Simulator for Robotic Grasping. Ph.D. Thesis, Department of Computer Science, Columbia University (June 2001)
13. Goldfeder, C., Ciocarlie, M., Dang, H., Allen, P.K.: The Columbia Grasp Databse. In: IEEE Internatioonal Conference on Robotics and Automation, pp. 1710–1716 (2009)
14. Haschke, R., Steil, J., Steuwer, I., Ritte, H.: Task-Oriented Quality Measures for Dexterous Grasping. In: Proc. IEEE Conference on Computational Intelligence in Robotics and Automation (2005)
15. Miller, A.T., Knoop, S., Christensen, H.I., Allen, P.K.: Automatic grasp planning using shape primitives. In: IEEE International Conference on Robotics and Automation, pp. 1824–1829 (2003)

16. Baier, T., Zhang, J.: Resuability-based Semantics for Grasp Evaluation in Context of Service Robotics. In: IEEE International Conference on Robotics and Biomimetics (2006)
17. Baier, T., Zhang, J.: Learning to Grasp Everyday Objects using Reinforcement-Learning with Automatic Value Cut-Off. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2007)
18. Prats, M., Sanz, P.J., del Pobil, A.P.: A framework for compliant physical interaction. Auton. Robot. 28, 89–111 (2010)
19. Siciliano, B. (ed.): Advanced Bimanual Manipulation. STAR, vol. 80. Springer, Heidelberg (2012)
20. Hendrich, N., Henriques, M., Bernardino, A.: Dexterous Postural Synergies from Teleoperation of the Shadow Robot Hand. In: Workshop on Hand Synergies, ICRA 2013 (2013)
21. Cyberglove systems, `http://www.cyberglovesystems.com`
22. Richter, E.: Hand pose reconstruction using a three-camera stereo vision system. diploma-thesis, University of Hamburg (2011)
23. HANDLE project, D4 — Protocol for the corpus of sensed grasp and handling data (2009), `http://www.handleproject.eu`
24. Sciuto, L.: Robotic Hand and Sensorized Glove: A Calibration for Managing Robotic Grasp in Teleoperation. MSc. thesis, University of Siena (2011)
25. HANDLE project, deliverable D24, Parameterizing and creating new actions (2012), `http://www.handleproject.eu`
26. Smits, R.: KDL: Kinematics and Dynamics Library, `http://www.orocos.org/kdl`

Reachability and Capability Analysis for Manipulation Tasks

Oliver Porges, Theodoros Stouraitis, Christoph Borst, and Maximo A. Roa

Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
82234 Wessling, Germany
{firstname.lastname}@dlr.de
http://www.robotic.dlr.de

Abstract. An offline analysis of the reachability of a robotic arm saves time for online queries like grasp selection or path planning. Reachability data is complemented with indices that quantify the goodness of one region in space to create a capability map, which can be computed based either on forward or inverse kinematics. This paper discusses the advantages and limitations of those methods, and proposes a hybrid method to improve the generation time while guaranteeing complete exploration of the space. The correctness of the results is studied with a prediction accuracy test. To illustrate the utility of a capability map, real-time visual information is incorporated to the map to help in the selection of grasp poses, or in path planning from an initial to a final pose.

Keywords: arm dexterity, reachability, manipulability, capability map, path planning.

1 Introduction

The ability of a mobile manipulator to grasp and manipulate objects depends on the location of the arm in the physical space. Arms with different kinematical structures provide different capabilities, and knowing the limits and capabilities of the arm allows a proper location of the robot to successfully fulfill a desired manipulation task. Either if the mobile robot is located in a suitable position or if the robot arm has a fixed base, an offline analysis of the robot's workspace is helpful to speed up the online solution of planning tasks. The representation of the regions where the robot tool frame (TCP) can be moved to is known as a reachability map [3]. It is in general computed as a spatial grid in the 6D space (position and orientation), where each cell has a binary value that indicates if it is reachable or not. The cells can also have an associated quality index that measures the dexterity of the robot when located in this position, thus creating a capability map (Fig. 1) [15].

An analytic representation of the reachable volumes for an arm can be obtained [7]. However, the information of the reachability map requires not only the bounding volumes, but a more complete description of the workspace, usually described with a voxelized structure. Two methods can be used to generate

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_50,

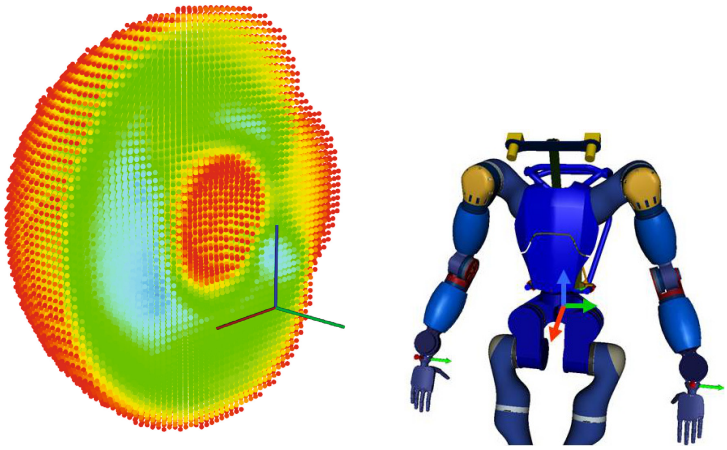


Fig. 1. Capability map for the right arm of the DLR humanoid Toro [8]

reachability maps, using either forward or inverse kinematics. With forward kinematics (FK), the arm configurations are randomly sampled in joint space, the TCP is computed and then assigned to one bin in the map, which changes value from 0 to 1. This method, however, has problems with zones close to a singularity, as too many samples tend to accumulate in a small region of the space, and it does not guarantee the completeness of the map [14]. With inverse kinematics (IK), the 6D space is uniformly sampled, and for each sample an IK solver runs to try to find a solution that leads to that configuration. This method guarantees a complete exploration of the workspace, but the use of an IK solver makes the generation of a new sample much slower [15].

The reachability or capability maps can be used for online computation and selection of reachable grasps for performing manipulation tasks [3,16]. Given a grasp pose, the position of the TCP is checked against the capability map to verify if it is reachable. To speed up query times, the reachable directions in each voxel can be represented with a simplified geometrical shape (cylinder, cone) that approximately describe the reachability properties, although at the cost of losing some precision in the analysis [16]. The map can also be applied to locate the base of a mobile manipulator for execution of 3D trajectories, like for instance opening a drawer or a door [3,5,12,14]. The associated indices that indicate the quality of a voxel help to define an optimal solution for the location of the mobile robot.

Even though the generation of maps is performed offline due to the large amount of data that must be processed, an efficient computation of the data is desirable to reduce the computational times to a minimum, while providing a data structure that facilitates online queries on the map. In cases when an FK approach is used, no criteria for stopping the sampling have been suggested, nor the validity of the map thus generated has been evaluated. This paper compares the performance of forward and inverse kinematic methods, and proposes

a hybrid method to generate the map; the formalization of the computation is presented in Section 2. Section 3 presents different application scenarios for the capability map. The efficiency of the map and its data structure is shown with an online analysis of reachability for an arbitrary scene captured with a Kinect-like sensor, which can be processed at the same rate of data acquisition (30 Hz). An A* path planner that takes advantage of the voxelized structure is implemented to efficiently compute paths in the workspace. The integration of the planner with a grasp database allows the quick generation of a feasible TCP path for grasping a target object. Finally, Section 4 discusses the findings and solutions obtained with the presented methods.

2 Efficient Computation of the Capability Map

For $\mathcal{Q}$ the configuration space of a manipulator, and a forward kinematics map $g_{fk} : \mathcal{Q} \to SE(3)$, the manipulator workspace W is defined as the set of all the end-effector configurations that can be reached with some choice of joint angles:

$$W = \{g_{fk}(\theta) : \theta \in \mathcal{Q}\} \subset SE(3) \tag{1}$$

where an element of $SE(3) = \mathbb{R}^3 \times SO(3)$ is described by (t, R), with t the translational and R the rotational part. The set of positions that can be reached with some choice of joint angles is called the reachable workspace W_R ($W_R \subset W$), defined as

$$W_R = \{t(\theta) : \theta \in \mathcal{Q}\} \subset \mathbb{R}^3 \tag{2}$$

An efficient data structure for representing $SE(3)$ is required to get a convenient memory allocation, so that online requests on the map can be processed very fast. The reachability map is then represented using a voxelized structure in $SE(3)$, which can be interpreted as the voxelized reachable workspace in $\mathbb{R}^3$ where each cartesian voxel has an associated rotational grid that discretizes $SO(3)$. The 3-dimensional Euclidean space $\mathbb{R}^3$ is divided into cubes $C_{x,y,z}$ of a user-defined side length. This allows us to map the TCP's translational component to a cubical volume, $t_{TCP} \to C_{x,y,z}$, with $x, y, z \in \mathbb{N}$.

Each cartesian voxel carries a rotational occupancy grid to map the orientation R_{TCP}. To obtain this grid, a virtual sphere is associated to each voxel; the spherical surface represents the discretization of the approach direction (pitch, yaw of the TCP), i.e. it is a division of $SO(2)$. It is important to stress that the spherical bins need to be of equal areas (Fig. 2) to avoid imposing an artificial weighting function that affects the reachability index. Such organized division allows creating a precomputed table of positions, that is fast to determine the bin that corresponds to a particular R_{TCP}. Each surface bin of the virtual sphere has an attached discretization of the missing parameter (roll) that completes the $SO(3)$ division. Note that if the last joint of the robot can freely move from 0 to 2π, only the approach directions in $SO(2)$ need to be discretized, as the roll angle is irrelevant for the map. With this structure, each R_{TCP} is uniquely mapped to one of the bins.

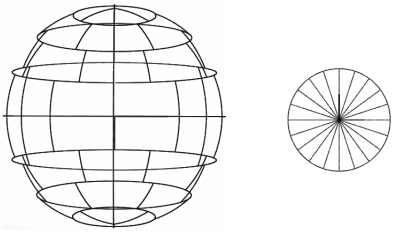


Fig. 2. Example of the virtual sphere for discretizing $SO(2)$, in this case with 50 equal-area bins organized in 8 layers. Each bin in the surface has an attached discretization of the roll angle, as shown in the right, in this case with 20 divisions.

The coordinates of the TCP can then be efficiently mapped to a set of 6 indices $M = \{i_1, ..., i_6 \in \mathbb{N}^6\}$ that describe the $SO(3)$ voxel where it is contained. Also, the voxels are finally stored in a one-dimensional, connected field called a linear voxel space [1], that provides a quick access to any element in the map while the map is kept in RAM. Therefore, memory size is the main limitation to the maximum resolution that can be achieved with the map. Table 1 summarizes the memory requirements for different map resolutions. For instance, a map with a linear resolution of 5cm, with 200 approach directions and 30 rolls per approach direction occupies 30 Mb, and a map with a linear resolution of 1cm, with 100 approach directions and 20 roll directions occupies 1100 Mb.

Table 1. Memory requirements for different map resolutions (in MB)

	50-10	100-20	200-30
7.5cm	0.69	2.9	8.6
5.0cm	2.3	9.7	30
2.5cm	18	73	219
1.0cm	258	1100	3301

2.1 Generation of the Capability Map

Two main approaches have been proposed for the computation of the reachability map, either based on forward or inverse kinematics. For the forward kinematics approach a large set of random configurations $q \in \mathcal{Q}$ is generated, for each one of them the TCP is computed using the map g_{fk}, and the bin that corresponds to the TCP is marked as visited [13]. This approach computes very quickly a large number of samples, but it does not guarantee a complete exploration of the robot workspace. Besides, a uniform sampling in the joint space will be reflected in a concentration of samples in some regions of the space, in particular

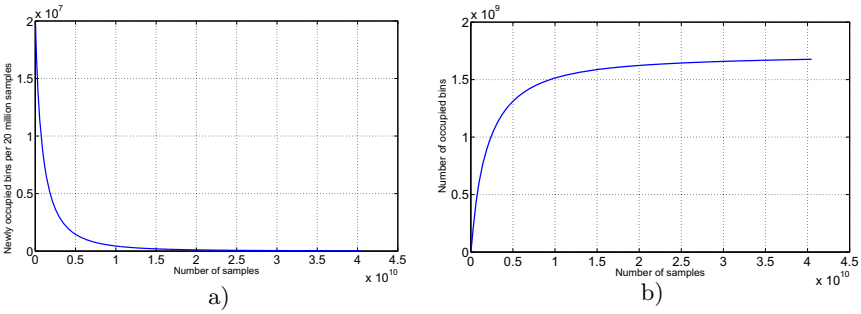


Fig. 3. Behavior of samples added for the FK generated map (convergence): a) Number of newly occupied bins per 20 million samples explored; b) Evolution of the total number of occupied bins with the number of explored samples

close to singular configurations where large link motions result in small cartesian displacements [14]. Therefore, a biasing of the sampling distribution could be helpful to improve the coverage of the space. Also, the pseudo random number generator must have a large period, as a large number of samples are required to get a reasonable prediction accuracy.

The second method takes a representative point per bin (its center) and computes the inverse kinematics solution for that TCP pose [3,15]. If the solution exists, the bin is marked as visited. This method guarantees that the whole workspace is explored, although the cost per configuration is higher for the IK versus the FK computation. The difference in computational times can become really important for kinematic chains with a high number of DoFs.

The method based on forward kinematics is cheap. Fig. 3 illustrates the evolution of the coverage of the capability map with the number of samples for the 7 DoF Kuka LBR arm, using a voxel size of 1cm, with 100 approach directions and 10 rolls per approach direction (from now on, the resolution data will be summarized as 1x100x10). The figure also shows the number of newly visited bins against the number of samples; the time required to generate the data was 26.7 hours, and it stopped when a predefined threshold of 1400 newly visited bins per million samples (0.14%) was reached (from now on, the threshold will be simply summarized as δ=0.14% newly visited bins). The method is very efficient in the first part (before 5×10^9 samples, generated in 3.3 hours), when most of the samples visit a new bin. However, as the number of samples grows, the number of newly visited bins decreases. At the end of the represented data, the procedure is still visiting about 1400 new bins per million of samples. In practice, it would take too long time to complete the map. The evolution of the IK-based method, on the other hand, is more predictable, as every new valid configuration belongs to a new bin.

Fig. 4a shows the capability map for the LBR arm, and Fig. 4b and 4c illustrate the differences between two maps with different resolution (5x200x30 and 1x100x10) generated with IK and FK approaches. A more intense red color in each 3D voxel represents a larger difference in the number of visited bins per

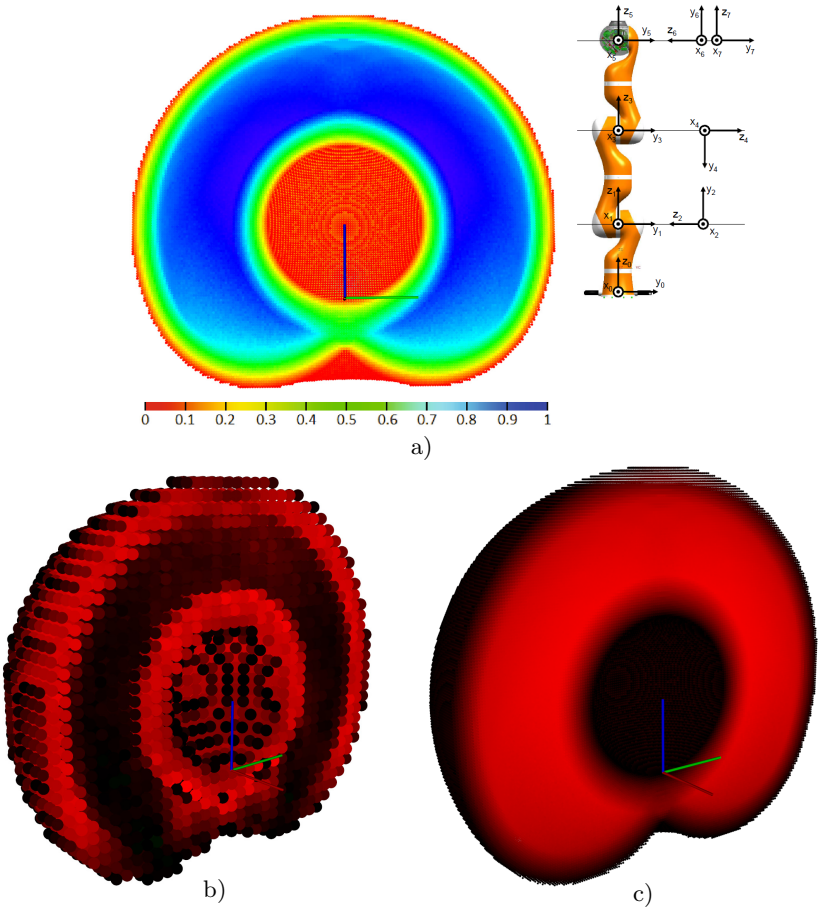


Fig. 4. Capability map for the Kuka LBR arm: a) Cross-section of the complete capability map, colored according to the reachability index $\boldsymbol{R}$ (1 means a perfect reachability). Difference between maps generated with IK and FK for two different resolutions: b) 5x200x30; c) 1x100x10.

voxel between the FK and IK maps, while a black color represents no difference between the number of visited bins. In both cases, the FK map stopped at 0.6% newly visited bins (6000 new bins per million samples). Note that the differences for the coarser resolution map tend to accumulate in the inner and outer area of the map. However, for a finer resolution the differences tend to accumulate in the most dexterous area. In both cases, the FK map contained more visited bins than the IK map, although differences could happen in the other direction as well. Effectively, for the IK approach only one TCP pose (the central one) is assumed to be representative of the complete bin; if no IK solution is found, then the bin is considered not visited. Of course the FK approach can by chance obtain a TCP pose inside that bin that has valid IK solution, so it marks the bin

as visited. On the other hand, even having a large number of samples, nothing guarantees that the FK approach visits at least once each one of the truly valid bins (there is no uniform distribution of the samples in the space).

For any robot, the cost of computing one IK solution can be easily estimated, and a simple switching mechanism can be implemented such that the FK based method is stopped when the performance (measured as number of newly visited bins per second) decreases to the level of performance of the IK method. At this point, the IK method is used to verify if the bins that have not been visited so far are reachable or not. To provide an idea of the performance in our case, with the current C++ implementation and using a commodity Linux PC the FK method can process 450 samples per ms, while the IK method processes 10 samples per ms. The hybrid approach profits of a good performance to cover a large portion of the space with FK, while still guaranteeing a complete exploration of the space by using IK. Some simple heuristics can be added to, for instance, avoid IK computations on 3D voxels that have never been visited (which avoids numerous IK computations in empty areas, like internal holes of the workspace).

Given that both FK and IK methods do not provide the ground truth for the reachability map, the question missing is how close is the map description to reality. To test this point, a number of random TCP directions inside the voxelized space were generated, an IK solution was computed to verify if the TCP is reachable or not, and then the reachability of the samples was estimated using the FK, IK and HK (hybrid) capability maps. For querying the map, a given TCP frame is mapped to its corresponding bin in the 6D space. If the bin is marked as visited, then the TCP is considered to be reachable.

The accuracy of the prediction depends on the criteria used for stopping the FK method, or for switching methods (from FK to IK) in the hybrid approach. To illustrate this relationship, Fig. 5 shows the results of the accuracy test using the capability map for the LBR with a resolution of 5x100x20. The stopping or switching criteria δ is specified by the user, and indicates the lower threshold of newly occupied bins when verifying a certain number of samples. This number of samples is adjusted automatically by a linear function. The interval increases with the map's resolution in order to prevent frequent quality checks which would slow down the generation process. Once the number of newly occupied bins drops below the percentage specified, the FK generation is stopped, or the hybrid method switches from FK to IK to verify the remaining unexplored portions of the space. Note that smaller values of δ increase the accuracy for the FK method, but decrease it for HK. The accuracy of the IK method does not depend on any additional parameter, it is shown as a dotted line for reference. To give an idea of the computational times, for $\delta = 5\%$, FK required 4.85 min, HK requires 18.4 min, and IK requires 41.6 min.

The errors in accuracy are caused by false positives (FP), when a TCP is reported as reachable but it is not, and true negatives (TN), when a TCP is reported as not reachable but it actually is. Fig. 6 presents the behavior of FP and TN for the three methods; once again, the values for IK are plotted as a reference, although they do not actually depend on the value of δ. To understand

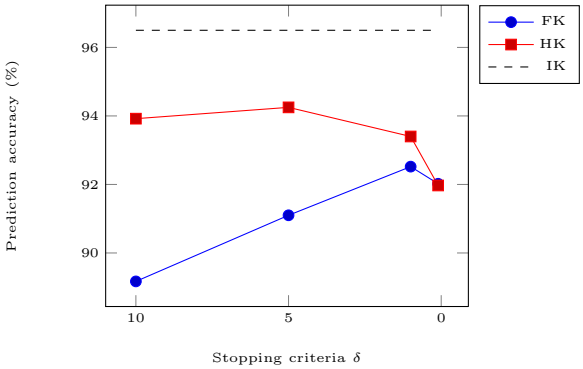
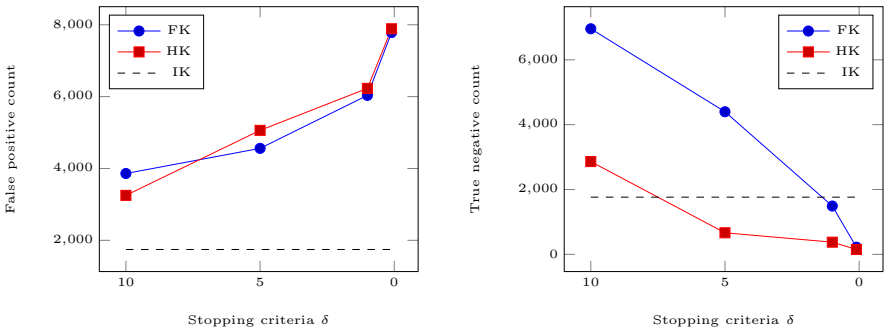


Fig. 5. Prediction accuracy of FK and HK methods in a LBR capability map of resolution 5x100x20, with varying δ as a stop criterion. The accuracy obtained for the IK method is plotted as a reference.

the origin of the errors, it is important to remember that the map is a binary representation of the discretized $SE(3)$ space, where each bin represents a range of directions by only one binary value. As a result of the discretization, the map contains bins that are in an ambiguity region, i.e. the bin covers both reachable and unreachable TCP poses, so the binary state cannot represent the bin state accurately. The portion of reachable and unreachable poses within one bin's range varies. With ongoing FK pose generation, the probability of visiting more and more bins which are mostly unreachable and yet contain a small region of reachability is increased; by setting them to 1, they are considered to be fully reachable. This explains the increased number of false positives when the value of δ is decreased. However, also with a longer running of FK generation (lower δ), the number of true negatives decreases. In the case of the IK generation, the number of false positives and true negatives is roughly the same.

Naturally, the resolution of the map also influences the computational times and prediction accuracy of the map. Tables 2 to 4 summarize the results of computations for different combinations of linear and angular resolutions; the results for the prediction accuracy test are obtained after querying 300000 poses. Based on the insights gained form the above analysis of the methods, we use $\delta = 1\%$ for the FK method and $\delta = 10\%$ for the HK method. One first result is that even for very coarse resolutions, like 7.5x50x10, the map still has an accuracy of about 90%. Obviously for higher resolutions the accuracy increases, getting to around 94% for the best analyzed cases using the FK or HK methods. However, the gain in accuracy comes to the cost of an increase in the memory required to store the map; Table 1 shows the memory requirements for the same resolutions). Note also that the IK based method has in general a slightly better accuracy compared to the other methods for the same resolution, although the time to generate it is roughly two times the time required for the other methods.



(a) (b)

Fig. 6. Errors in the prediction test, according to different values of δ: a) False positives, and b) True negatives, for the LBR capability map with a resolution 5x100x20

Table 2. Results for maps generated with inverse kinematics (IK)

(a) Generation time (min)

	50-10	100-20	200-30
7.5cm	3.1	12.49	37.9
5.0cm	12.66	41.62	125.43
2.5cm	75.52	318.33	969.38
1.0cm	1237.0	5179.7	

(b) Prediction accuracy (%)

	50-10	100-20	200-30
7.5cm	94.85	95.47	95.65
5.0cm	95.83	96.49	97.54
2.5cm	96.18	97.07	97.48
1.0cm	96.2	97.34	

Table 3. Results for maps generated with forward kinematics (FK)

(a) Generation time (min)

	50-10	100-20	200-30
7.5cm	0.83	3.33	9.21
5.0cm	2.33	10.26	28.91
2.5cm	21.08	78.82	217.6
1.0cm	260.6	7093.6	

(b) Prediction accuracy (%)

	50-10	100-20	200-30
7.5cm	89.55	90.48	90.88
5.0cm	91.54	92.52	92.94
2.5cm	91.51	93.65	94.22
1.0cm	93.11	94.33	

Table 4. Results for maps generated with hybrid method (HK)

(a) Generation time (min)

	50-10	100-20	200-30
7.5cm	0.25	8.65	23.6
5.0cm	4.78	20.71	64.21
2.5cm	20.66	153.82	579.6
1.0cm	678.0	2267.0	

(b) Prediction accuracy (%)

	50-10	100-20	200-30
7.5cm	92.26	93.09	93.5
5.0cm	93.77	94.74	95.2
2.5cm	92.92	93.92	94.28
1.0cm	94.22	95.17	

2.2 Evaluation of Voxel Quality

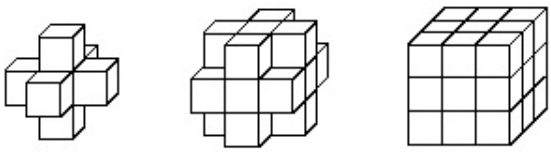
Applications of the map include finding the best region inside the workspace for executing a predefined path, or the evaluation of the final pose for the end effector, which appears for instance when looking for the best TCP pose for grasping an object. A quality value associated to the voxel can be helpful in this evaluation process, by providing information on the local dexterity of the robot around a TCP location. To quantify the reachability of a region in $\mathbb{R}^3$, a reachability index $\boldsymbol{R}$ is proposed, based on the discretization of the associated rotational voxel that approximates $SO(3)$. Let n_b be the number of bins considered in $SO(3)$, and v_i be the binary value of the occupancy of the i-th bin. $\boldsymbol{R}$ is defined as

$$\boldsymbol{R} = \frac{\sum_{i=1}^{n_b} v_i}{n_b} \tag{3}$$

Therefore, $\boldsymbol{R}$ represents the fraction of occupied bins in the voxel. The higher $\boldsymbol{R}$, the more dexterous is the robot in that region of the workspace. The capability map of Fig. 4a is colored according to this index.

In additional applications like planning a path from an initial to a final TCP pose, it is useful to know also the dexterity of the neighboring voxels, so that the search can be guided to avoid regions with poor dexterity. Different voxel adjacencies can be considered for creating this regional dexterity index $\boldsymbol{D}$, according to the number k of predefined neighbors (Fig. 7). Let $co(i, j)$ be a function that evaluates the number of common orientations between two voxels i and j. The regional dexterity index is defined for the central voxel (0) as

$$\boldsymbol{D}_0 = \frac{\sum_{i=1}^{k} co(0, i)}{k \cdot co(0, 0)} \tag{4}$$



a) b) c)

Fig. 7. Different adjacencies of a voxel in the 3D discrete space: a) 6-adjacent voxels; b) 18-adjacent voxels; c) 26-adjacent voxels

Additional indices that provide extra information on the manipulator can be included. For instance, a classic manipulability measure like the inverse of the condition number of the Jacobian J for a given configuration indicates how close the robot is to a singularity [11]. The measure is defined as

$$\boldsymbol{S} = \frac{\sigma_{\min}(J)}{\sigma_{\max}(J)} \tag{5}$$

with $\sigma_{\max}$ and $\sigma_{\min}$ being the maximum and minimum singular values of J.

A measurement of the distance to joint limits is also helpful to keep the robot with enough movement capacity in situations like an online path change to avoid collisions with a dynamic object. The measure is defined as [10]

$$\boldsymbol{L} = 1 - \frac{4}{l} \sum_{i=1}^{l} \left(\frac{\theta_i - \theta_{0i}}{\theta_{\max_i} - \theta_{\min_i}} \right)^2 \tag{6}$$

where l is the total number of joints in the arm, and θ_i, θ_{0i}, $\theta_{\max_i}$, and $\theta_{\min_i}$ are the actual, middle-range, maximum and minimum position of the i-th joint, respectively (the index is simplified when $\theta_{0i} = 0$).

Note that the last two measures depend on the arm configuration, and in the case of methods that use FK for the map generation several configurations can lead to the same bin, so one should keep track of the maximum and minimum value of the measure for any given bin. All the measures are normalized to obtain 0 in the worst case and 1 in the best case. Of course, for a given application a combination of the measures can be considered to properly weigh the importance of each aspect in the voxel evaluation.

3 Applications

3.1 Integration with Visual Information

An initial test of the efficiency of the map is the online evaluation of the reachability of points in a current scene. For the implementation, the scene is perceived with an Asus Xtion Pro Live, which provides depth data at 30 Hz with VGA resolution. The current scene is projected onto the voxelization of the capability map, which allows a quick labeling of free and occupied voxels in the 3D space. The voxels that contain points coming from the point cloud are marked as occupied, which serves later, for instance, to discard them from further processing in path planning applications. There still remain voxels where there is no information, as they are not in the line of sight of the camera. For common objects, for instance, only a portion of the object is represented on the pointcloud. For typical environments like tabletop scenarios, a clustering process is applied to isolate objects, and then simple heuristics can be employed to generate a better description of the occupied voxels in the current scene; for instance, all the voxels lying inside the bounding box of the clusterized object are considered as occupied.

As an example of the integration of visual data with the capability map, Fig. 8 shows an online evaluation of reachability for all the points in the scene. If the point and its associated normal direction are reachable according to the capability map (meaning that the robot could apply forces pushing that specific point), then they are represented in green. Identifying the occupied voxels for a given pointcloud takes about 0.48 ms in our current implementation, and the evaluation process can answer up to 2000 queries per ms.

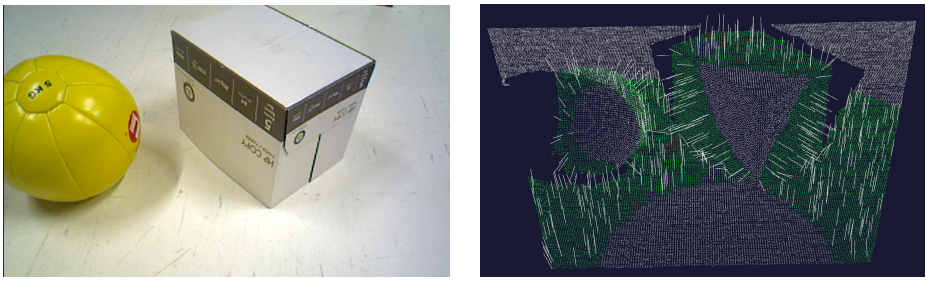


Fig. 8. Reachable points in a scene. The scene is shown in the RGB image, and the point cloud shows green points when the robot can apply forces that push that point along its normal direction.

3.2 Grasp Evaluation

For grasp and manipulation tasks with known objects, an offline database of grasps can be generated with different feasible poses for the end effector. This database is in general created for a free floating object, in our current implementation using a modified version of the grasp planner in Openrave [3]. After the object is recognized in the current scene, usually several tens or hundreds of feasible TCP locations can be loaded from the database and used as potential grasp locations. A first filtering step is performed once the supporting surface of the object is recognized from the current scene; the surface is reproduced in the virtual scene, and all the grasps in the database are verified so that grasps where the hand collides with the surface are pruned out (Fig. 9). The implementation of this step uses a collision check based on the Voxmap-Pointshell algorithm (VPS) [9]; this allows a filtering of about 50 grasps in 0.2s. This filtering is only performed once, at the beginning of the grasp evaluation process, right after the pose estimation for the object (assuming that the stable pose of the object on the supporting surface does not change during the task).

The potential TCP locations for grasps that survive the filtering step can be further processed. There is information on the grasp quality coming from the grasp planner, so the grasps are prearranged from higher to lower quality. Using the capability map, the reachability index $\boldsymbol{R}$ can be used to generate a new ranking of grasps that takes into account the arm dexterity for the grasping locations. The combined position from the two rankings can be used to generate a list of good grasps that take into account both criteria.

3.3 Path Planning

An interesting application for the capability map is the generation of a path between an initial and a final pose, for instance, to generate a reach motion to grasp and pick up an object. Offline computations of reachable poses has been traditionally used in probabilistic roadmaps (PRM) for path planning [6]. The integration of the capability map containing reachability data can speed up the

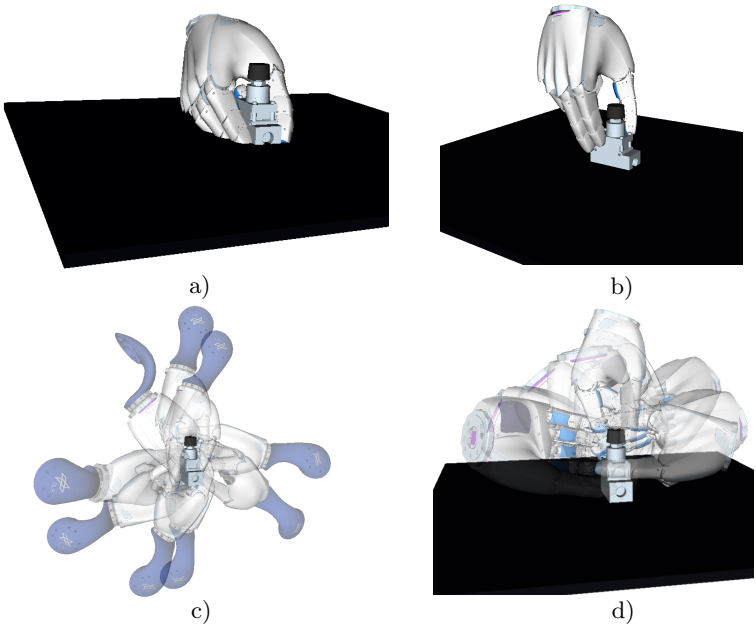
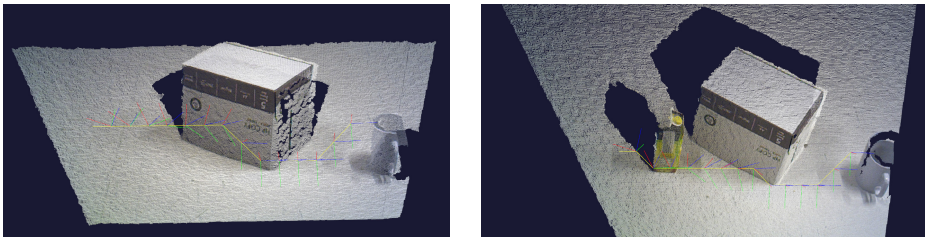


Fig. 9. Filtering of grasps according to the supporting surface of the object: a) Unfeasible grasp; b) Feasible grasp; c) Initial database; d) Filtered database

path planning generation process, and also allows a quick generation of a path for the robot TCP in the 6D workspace. This initial guess for the TCP trajectory can be used in a complete path planner that checks for collision between the arm and possible obstacles, and that deals with redundancies of the manipulator, like for instance [2,4].

The TCP path search can be solved efficiently in the 6D space using the information of the map integrated with the visual information coming from the scene. The identification of occupied voxels discards them automatically from the path planning process. For solving the TCP planning task, an A* approach was chosen [6], as the path on the capability map is obtained by moving from voxel to voxel, using the discretized structure that provides additional information (indices) on the quality of each point in the trajectory.

To solve the path, it is decoupled in the space of translations and orientations. The translation from the beginning of the path is measured using euclidean distance, and the estimation of the distance to the goal uses Manhattan distance. The path progresses through the workspace using 18-adjacent voxels. The orientation is coupled to the translation, and the planner tries to get to the final orientation when the path is half-way between the start and goal positions. Fig. 10 shows one example of the path planning process for an initial query (Fig. 10a), and a replan after a new obstacle is added to the scene (Fig. 10b). Each plan generation takes about 40ms when using the 5x200x30 map, and about 8s using the 1x100x10 map.



a) b)

Fig. 10. Example of path planning using capability map: a) Initial plan; b) Path recomputed after adding an obstacle

4 Results and Discussion

This paper deals with the offline generation of capability maps, that synthesize information on reachable positions and orientations for a given robotic arm. Classical methods for the map generation use either forward or inverse kinematics. Methods based on forward kinematics are fast to approximate a large portion of the reachable workspace, but they do not guarantee a uniform exploration. Methods based on inverse kinematics are slower, but provide a uniform exploration of the space. A hybrid method that uses both approaches is also employed here, using as a control parameter δ the percentage of newly discovered bins to decide the moment to change from forward to inverse kinematics. Evaluation of the performance of the methods for real queries is provided, and even with very coarse resolutions the capability maps obtain accuracies of over 90%. All the methods lead to false positives and true negatives, and the influence of δ on the aaccuracy was analized.

Traditionally used resolutions of the map in $\mathbb{R}^3$ are in the order of 4-5 cm [12,15], but no previous consideration had been given to the required resolution of the map. The present study showed how the prediction accuracy changes with different resolutions, leading in some cases to accuracies of about 95%. Marginal difference was perceived between the accuracy obtained with the methods, due to the imprecision that each one of them entails. However, the FK and HK maps are generated much faster while leading to a performance comparable to the IK method. Another interesting issue to study is the relation between the resolution of sampling in $\mathbb{R}^3$ and $SO(3)$. From the memory management point of view, smaller cartesian resolutions go hand in hand with a relaxation of the directional resolution. However, the influence of these parameters on real applications like path planning should be further studied.

One practical note is how to choose the map's resolution for a particular task. In applications that are restricted by hardware resources, one might desire to achieve the best resource to performance ratio. For high DoF systems it might be more useful to have better reachability information, as an IK solution is computationally more expensive. Memory requirements were summarized in Table 1 to help in choosing this resolution. In cases where the map is combined

with real time visual data, as presented in this work, it is important to be able to dynamically allocate reachable and unreachable voxels in the map. The voxel size is then chosen based on the sensor refresh rate and maximum expected/allowed velocity for dynamic obstacles in the scene.

The computation of the capability map for a complex robot, like the humanoid in Fig. 1, presents additional challenges. Usually, planning for mobile platforms is conceived in a two-stage approach: first position the mobile base in a convenient location, and then move the arm/upper torso for executing the manipulation task. This classical approach is compatible with computing the capability map only for the robotic arm, and when the base is displaced the capability map can be moved around the physical space to obtain a good position for executing the task. Although this approach has proven effective, perhaps a more natural (human-like) motion of the robot can be obtained when the capability map considers additional DoFs in the offline computation. In these applications of high-DoF kinematic chains it is possible that a difference in performance between the presented methods is more evident. Additional applications in bimanual systems include the definition of the dexterous space for both arms, and the proper task planning exploiting such information [14].

The structure of the capability map allows a seamless integration with the scene information coming from a depth sensor. This allows applications such as a fast evaluation of possible grasp poses coming form a grasp database. The grasp poses usually are ranked according to a suitable grasp quality measure, and this ranking can be complemented with the additional dexterity information stored in the capability map. For path planning, traditional methods based on RRTs that work directly in the configuration space usually generate rough paths, which require a post processing step to smooth out the obtained path. Using the voxelized structure of the capability map, it is in theory possible to obtain softer paths in the joint space. For this work, an A* planner to obtain TCP trajectories was implemented, to make an initial proof of the validity of the approach. The complete integration of the capability map in a path planner and its evaluation against a standard bi-RRT planner, and the test of the planner in a real robotic platform, are next steps that the authors will soon tackle.

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287787, project SMERobotics.

References

1. Bodenmueller, T.: Streaming Surface Reconstruction from Real Time 3D Measurements. Ph.D. thesis, Technical University of Munich - TUM, Lehrstuhl fuer Reakzeit - Computersysteme (2009)
2. Cohen, B., Sucan, I., Chitta, S.: A generic infrastructure for benchmarking motion planners. In: Proc. IEEE Int. Conf. Robotics and Automation - ICRA, pp. 589–595 (2012)

3. Diankov, R.: Automated Construction of Robotic Manipulation Programs. Ph.D. thesis, Carnegie Mellon University, Robotics Institute (2010)
4. Dragan, A., Ratliff, N., Srinivasa, S.: Manipulation planning with goal sets using constrained optimization trajectory (2011)
5. Guilamo, L., Kuffner, J., Nishikawi, K., Kagami, S.: Efficient prioritized inverse kinematic solutions for redundant manipulators. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS, pp. 3921–3926 (2005)
6. Kallmann, M., Mataric, M.: Motion planning using dynamic roadmaps. In: Proc. IEEE Int. Conf. Robotics and Automation - ICRA, pp. 4399–4404 (2004)
7. Kee, D., Karwowski, W.: Analytically derived three-dimensional reach volumes based on multijoint movements. Human Factors: The Journal of the Human Factors and Ergonomics Society 44(4), 530–544 (2002)
8. Ott, C., Eiberger, O., Roa, M.A., Albu-Schaeffer, A.: Hardware and control concept for an experimental bipedal robot with joint torque sensors. Journal of the Robotics Society of Japan 30(4), 378–382 (2012)
9. Sagardia, M., Hulin, T., Preusche, C., Hirzinger, G.: Improvements of the voxmap-pointshell algorithm - fast generation of haptic data structures. In: Proc. 53rd Int. Wissenschaftliches Kolloquium (2008)
10. Suárez, R., Roa, M.A., Cornellà, J.: Grasp quality measures. Technical University of Catalunya. Technical Report IOC-DT-P-2006-10 (2006)
11. Togai, M.: An application of the singular value decomposition to manipulablity and sensitivity of industrial robots. SIAM Journal of Algebraic and Discrete Methods 7(2), 315–320 (1986)
12. Vahrenkamp, N., Asfour, T., Dillmann, R.: Robot placement based on reachability inversion. In: Proc. IEEE Int. Conf. Robotics and Automation - ICRA, pp. 1962–1967 (2013)
13. Vahrenkamp, N., Asfour, T., Metta, G., Sandini, G., Dillmann, R.: Manipulability analysis. In: Proc. IEEE-RAS Int. Conf. on Humanoid Robots, pp. 568–573 (2012)
14. Zacharias, F.: Knowledge Representations for Planning Manipulation Tasks. Springer, Berlin (2012)
15. Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure: Representing robot capabilities. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS, pp. 3229–3236 (2007)
16. Zacharias, F., Borst, C., Hirzinger, G.: Online generation of reachable grasps for dexterous manipulation using a representation of the reachable workspace. In: Proc. IEEE Int. Conf. Advanced Robotics, pp. 3229–3236 (2009)

Dexterity Optimization of a Three Degrees of Freedom DELTA Parallel Manipulator

Vitor Gaspar Silva, Mahmoud Tavakoli, and Lino Marques

Department of Electrical and Computer Engineering,
Institute of Systems and Robotics, University of Coimbra, Portugal
v.gaspar.silva@gmail.com, {mahmoud,lino}@isr.uc.pt

Abstract. This paper demonstrates dexterity optimization of a Delta-like three degrees of freedom (3 DOF) spatial parallel manipulator. The parallel manipulator consists of three identical chains and is able to move on all three translational axes. In order to optimize the manipulator in term of dexterity, a Genetic Algorithm (GA) global search method was applied. This algorithm aims to propose the best design parameters such as the length of the links which results in a better dexterity. Results of the optimization are presented.

1 Introduction

Parallel Manipulators are closed-loop mechanical systems, used on the industry and many other activities of high technical demand like aerospace, for their good performances in terms of accuracy, rigidity, manipulation speed and ability to manipulate large loads [1,2,3]. One of the first parallel manipulator was developed by Stewart [4]. The Stewart platform is a 6 DOF. manipulator with 6 arms. Despite the many advantages of parallel mechanisms, they also has several drawbacks [2,5], demonstrate some disadvantages that are shared by many parallel manipulators: complex direct kinematics; position and orientation of the moving platform are coupled; small workspace; expensive joints, namely the spherical. Other platforms were proposed to tackle some of these disadvantages. The DELTA platform designed by Clavel [6] solves the first two items. Nonetheless, it is a 3-4 DOF manipulator. Delta manipulators are popular manipulators that are mainly being used for pick and place applications. Their main advantage is their moving speed within their workspace. Figure 1 shows a 3 DOF version of the Delta robot, which was constructed in our lab, and the schematic of its kinematic chain. An advantage of delta robots is that the links a and b can be easily extended or shortened, while keeping the main platform and its actuators unchanged. In this way one can change the workspace of the robot for a desired task. However changing these parameters has also an effect on the dexterity of the robot. In contrast with serial articulated arms, where singularities are located on the border of the workspace, parallel manipulators suffer from singularities inside their workspace, and thus some areas in the workspace of the manipulator should be avoided [7]. Dexterity of a manipulator may be viewed

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_51,

as a degree of farness or distance from a singularity. On the other hand the location of such singularities in the workspace changes with changing length of the links of the manipulator. It is desired to choose the best design parameters in order to increase the dexterity of the manipulator. Considering the complex nature of the parallel manipulators kinematics, global search methods are usually applied for optimization of the dexterity of the parallel manipulators [8]. Stamper et al. [9], proposed a method for optimization of a three degree of freedom translational parallel manipulator both total workspace and global conditioning index. The total workspace of the manipulator is determined by a Monte Carlo technique. The optimization of the manipulator design for the global condition index results in a manipulator where the lower leg comprises 44% of the total leg length and the upper arm comprises 56% of the total leg length, while legs are installed on a circle with an angular separation of 120′. They showed that optimization of the total workspace volume results in significantly different design parameters compared to the optimization performed for a well conditioned workspace. Moreover, these results showed that a manipulator of this type that is designed to maximize total workspace volume will result in an ill conditioned workspace. Tavakoli et. al. considered optimization of the design parameter for a multi objective object function which comprises of both dexterity and workspace volume [10] using a genetic algorithm method. This work addresses the problem of optimizing the dexterity of this spatial three degrees of freedom parallel manipulator with a Genetic Algorithm method.

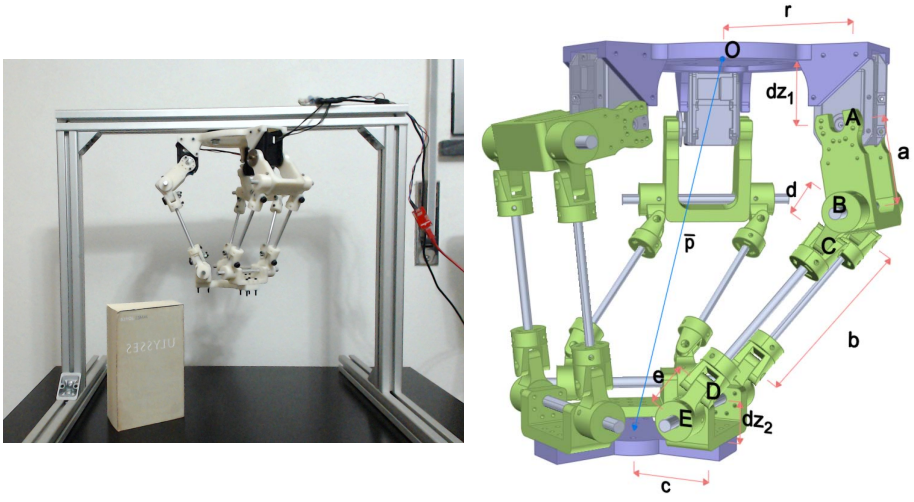


Fig. 1. The spatial three degrees of freedom parallel manipulator and schematics of its kinematics chain

2 Optimization

Dexterity of a manipulator may be viewed as a degree of farness or distance from a singularity. To have a dexterity measurement of a parallel manipulator the Global Conditioning Index (GCI) represented by η in equation 1 is used. This was first proposed in [11] and expands the condition number [12] of the manipulator's to the entire workspace W.

$$\eta = \int_W \frac{1}{\gamma} dW \tag{1}$$

Where γ is the *condition number*, which defines the amplification factor that the error has on the end-effector velocity, using the inverse kinematics Jacobian (J). This number is dependent on the norm, which in our case is the Euclidean norm in the 3D space. The smallest possible value of the condition number is 1, meaning that the error is not amplified. The inverse of this value is used, $s = 1/\gamma$, to transform the problem in a maximization one in the domain of [0 1].

$$\gamma = \|J^{-1}\|\|J\| \tag{2}$$

Some limitations on this condition number approach are pointed in [1]. The most significant is that if the manipulator has rotational DOF on the Cartesian space the number does not translate in a clear physical meaning. They are instead transformed on "equivalent" translations. As the analyzed manipulator has only translational DOF no additional methods are to be taken into account. Now that the condition number is defined for one point, there is the need to expand this to the whole workspace in order to evaluate the manipulator. The integration of a manipulators workspace is complex, hence a numeric method was used. The Monte Carlo method was selected, since it has been successfully applied in similar tasks [9,13]. The Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. In this work, a known workspace is defined and points are randomly generated according to a uniform distribution. Those points are tested on the kinematic function of the manipulator to check if they rely on the manipulator's workspace. Finally the relation between the number of points tested and the number of points that are valid on the manipulator workspace, provides the relation between the known workspace and the manipulator workspace. Using the Monte Carlo method, a number of points is created inside a volume that is confined by half of a sphere on the positive Z axis. This sphere has a maximum theoretical radius limited by the sum of the manipulator parameters $[a+b+d+e]$ (Figure 1). The created points are then checked if they are on the workspace, and, if so, their condition number is calculated. The volume of a sphere is given by $V = \frac{4\pi r^3}{3}$. Considering only half of the sphere and the proportion of the points on the Monte Carlo method where the n_{valid} are the points inside the workspace, where $r_s = a + b + d + e$ is the sphere radius limited by the manipulator links lengths. The sum of all valid points will have a relation to the workspace volume.

Thus the global conditioning index calculated with the Monte Carlo method for the parallel manipulator in study is defined by:

$$GCI = \frac{2\pi {r_s}^3 \sum_i \frac{1}{\gamma}}{3n_{total}} \quad \text{for } i \text{ valid points.} \tag{3}$$

In this formula the term $\sum_i \frac{1}{\gamma}$ sums the condition number of the "i" valid points. When it is divided by "n" total points, it results in the ratio of valid points to total points. If this ratio is multiplied by the initial considered volume of the half sphere, it results in an estimation of the actual workspace volume of the manipulator.

It is worth mentioning that the GCI, relates to the volume of the workspace, presenting a normalised value for all lengths of the arms and workspaces. That is, if for instance, the sum of the lengths of the links are considered as 1, the actual volume of the workspace can be avoided in the formulation. Dexterity depends on the length of a parallel manipulator's links. Since these manipulators contain a large number of links and their configuration imposes dependencies between the links, the study of the optimal configuration to achieve a better dexterity in the workspace is a complex problem, which is usually studied by global search algorithms such as Genetic Algorithm.

2.1 Genetic Algorithm

In order to optimize the dexterity of the manipulator within its workspace Genetic Algorithms (GAs) based search was applied. GAs based on natural evolution are a search heuristic commonly used to find a solution for complex problems, and have been used with success across several problem domains [14]. In our work it was built a floating point GA to optimize the GCI, where the length of the links compose the genome of the individuals presented in the population. In floating point GA method, the traditional binary representation of the solutions is substituted by real numbers. Thus mutation and cross over functions are different with the binary and most common GA optimization methods.

To guarantee convergence and avoid local maxima, the operators that guide the search have to be tailored to the encoding of the solution and the origin of the problem. In the binary GA, the mutator function is defined based on swapping one or more binary digits of each chromosome. In floating point GA methodology, the mutation function is performed by a gaussian mutator. On the gaussian mutator a gaussian function with a specified standard deviation is applied to the allele, that way is possible to produce a new value within a certain range of the old value. This operator has an important role in the search on the nearest neighbourhood. Due to the nature of the problem and the fact that the mutation is so tailored to the problem the crossover operator will promote the population diversity, so it may be used without regarding the structure of the genome, making two new individuals from the recombination of two parents' genome.

It was ran some initial simulations in order to tune the parameters of the GA in order to guarantee convergence of the solutions. As a result it was initiated a population of 150 individuals through 250 generations. In each generation, individuals are selected in a tournament of 5 to generate the next generation, these individuals have 0.9 probability to be crossbreed and 0.2 probability to be mutated. The next generation is composed by the crossbreed and mutated individuals and 2% of the best individuals of the previous generation (Elitism operator in Figure 2). It was also introduced a novel operator, called armageddon. When the best solution does not suffer alteration on 10 consecutive generations the operator armageddon replaces 92% of the population with new individuals, creating new diversity on the search space. Figure 2 shows the structure of the GA built for the optimization of the GCI of the manipulator.

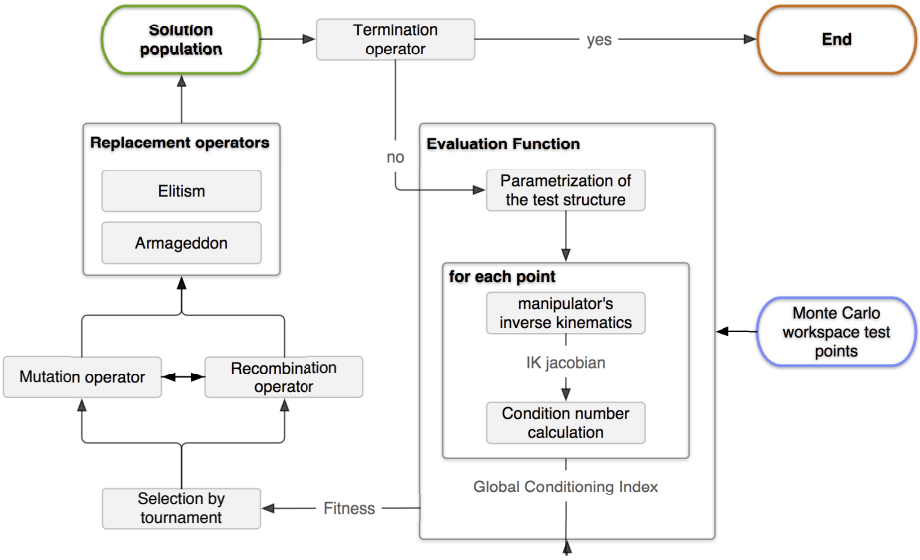


Fig. 2. Genetic Algorithm Implementation

3 Results

After the tuning the operators that compose the GA and guaranteeing its convergence, 30 runs of the algorithm each with 250 generations were made in order to have a statistic validation of the results, allowing to verify the validity of the algorithm solution to the optimization. In the Monte Carlo method 100000 test points were considered.

Table 1 shows the fitness of the GA to the problem. in our approach this means the GCI of the possible structures created by the GA. Table 1 shows the

Table 1. Test 1 solutions summary of the 30 runs

	Mean	Median	Max	Min	Std
a	0.31682	0.31620	0.39609	0.23653	0.05354
b	0.66159	0.65047	0.74893	0.58653	0.05527
d	0.00533	0.00233	0.03226	0.00000	0.00994
e	0.00748	0.00326	0.02066	0.00000	0.00854
c	0.39494	0.43838	0.74988	0.00000	0.26823
r	0.43295	0.51821	0.76442	0.00000	0.27619
fitness	0.09407	0.09482	0.09966	0.08400	0.00457

structure parameters of the last generation in each run representing the statistic parameters of the best solutions. Table 2 shows the best solution obtained for the Test 1. Based on the fitness low standard deviation (0.00457), in table 1, and the high standard deviation (0.26823) on c and (0.27619) r, on table 1, it can conclude that parameters c and r do not have a direct relation with the manipulator's GCI. The values for the parameters d and e are close to 0, with a low standard deviation.

Table 2. Tests' best solutions

	a	b	d	e	c	r	c-r
Test 1	0.35440	0.62724	0.01023	0.00813	0.74988	0.76442	
Test 2	0.34689	0.65170	0.00065	0.00075			0.00178
Test 3	0.34506	0.65494					

Therefore as a first result it was considered $e = d = 0$. Also as it can be seen that c and r values are not directly connected to the improvement of the manipulator's GCI, and thus in the next test it was considered "the difference between c and r" as a solution parameter. Based on the results (Table 2), it can be seen that for a better GCI the $c - r$ should be 0. The focus will then be on the parameters a and b, considering $c = r = 1$, and $d = e = 0$.

Therefore it was ran the optimization for the third time with the results of the previous two steps. The results obtained confirm the ones obtained in the previous tests with $a = 0.34506$ and $b = 0.65494$. With the lowest standard deviation (0.00748) on these parameters on the tests conducted as expected. Meaning that for our symmetric manipulator the link length $a = 35\%$ of the total length of the each arm, and $b = 65\%$ is the optimal.

As shown on previous tests, in order to improve the GCI of the manipulator the structure parameters d, e, difference $c-r$ must be 0. Allowing us to set those values, to better explore the impact the structure parameters a and b have on the manipulator's GCI, as done in the third test. But as this is done the complexity of the problem decreases. So it is now possible to apply a "brute force" method

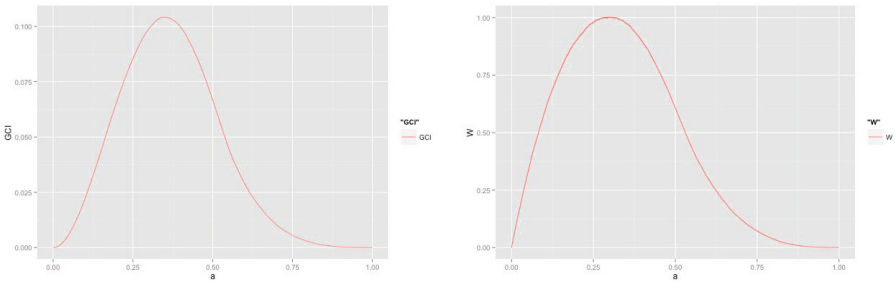


Fig. 3. Variation of the GCI (left) and the workspace volume (right) against the length of the link "a"

exploring all the combinations of a and b. As imposed before, the length of the arm composed by $a + b + d + e$ is 1. This means that for this test $b = 1 - a$.

Figure 3, shows the evolution of the GCI by its value S as a function of a. The GCI maximum value is reached when the manipulator configuration as the structure parameters: $a = 0.35102$; $b = 0.64898$. The evolution of the workspace volume is shown in Figure 3, validating our previous results with the GA.

4 Conclusions

The evolutionary algorithm was developed and validated, and with it the optimization of the dexterity of the manipulator's workspace was performed. This optimization showed a structure configuration of the manipulator where: the length of the link $a = 35\%$ of the total arm's length; $b = 65\%$ of the total length; the lengths d and e equal to 0; and the difference between the length of the base and manipulator $r - c = 0$. It could also derive the effect of length of each parameter on the workspace volume and dexterity of the manipulator (Figure 3).

References

1. Merlet, J.P.: Parallel Robots. Solid Mechanics and Its Applications, vol. 128. Springer, Heidelberg (2006)
2. Pierrot, F., Fournier, A., Dauchex, P.: Towards a fully-parallel 6 dof robot for high-speed applications. Robotics and Automation (April 1991)
3. Gosselin, C.: A new architecture of planar three-degree-of-freedom parallel manipulator. Robotics and Automation, 3738–3743 (April 1996)
4. Stewart, D.: A platform with six degrees of freedom. Proceedings of the Institution of Mechanical Engineers 180(1), 371–386 (1965)
5. Hunt, K.H.: Structural Kinematics of In-Parallel-Actuated Robot-Arms. Journal of Mechanisms Transmissions and Automation in Design 105 (1983)
6. Clavel, R.: DELTA, a fast robot with parallel geometry. In: Burckhardt, C.W. (ed.) Proc of the 18th International Symposium on Industrial Robots, pp. 91–100. Springer, New York (1988)

7. Agrawal, S.: Workspace boundaries of in-parallel manipulator systems. In: Fifth International Conference on Advanced Robotics, Robots in Unstructured Environments, ICAR 1991, vol. 2, pp. 1147–1152 (1991)
8. Gallant, M., Boudreau, R.: The synthesis of planar parallel manipulators with prismatic joints for an optimal, singularity-free workspace. Journal of Robotic Systems 19(1), 13–24 (2002)
9. Stamper, R., Tsai, L.W., Walsh, G.: Optimization of a three dof translational platform for well-conditioned workspace. In: Proceedings of the1997 IEEE International Conference on Robotics and Automation, vol. 4, pp. 3250–3255 (April 1997)
10. Tavakoli, M., Zakerzadeh, M., Vossoughi, G., Bagheri, S., Salarieh, H.: A novel serial/parallel pole climbing/manipulating robot: Design, kinematic analysis and workspace optimization with genetic algorithm. In: 21st International Symposium on Automation and Robotics in Construction, Korea (2004)
11. Gosselin, C., Angeles, J.: A global performance index for the kinematic optimization of robotic manipulators. Journal of Mechanical Design 113, 220 (1991)
12. Salisbury, J.K., Craig, J.J.: Articulated hands force control and kinematic issues. The International Journal of Robotics Research 1(1), 4–17 (1982)
13. Alciatore, D., Ng, C.: Determining manipulator workspace boundaries using the monte carlo method and least squares segmentation. In: 23rd ASME Mechanisms Conference, pp. 141–146. American Society of Mechanical Engineers, Minneapolis (1994)
14. Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G.: Genetic programming IV: Routine human-competitive machine intelligence. Springer (2005)

High Speed Fragile Object Manipulation

Javier Sarria[1], Hector Montes[1,2], Manuel Prieto[1], and Manuel Armada[1]

[1] Centre for Automation and Robotics, CSIC-UPM
[2] Faculty of Electrical Engineering, Technological University of Panama
j.sarria@csic.es

Abstract. This paper describes the optimization of picking up soft goods moving on a belt conveyor using a high speed manipulator driven by linear motors and equipped with a vacuum gripper structure and a computer vision system. A special gripper with several grasping devices and with a rotational degree of freedom has been designed to cope with object orientation. The vision system determines the location and orientation of the objects in the task space of the robot and the control system is in charge of the correct manipulation of the products. Also, this work presents the development and results of a pastry industrial project, which has a particular emphasis, given its complexity, the Cartesian robot manipulator, which has been designed to optimize the process of handling and palletising in this industrial area.

Keywords: Handling robot, palletising robot, gripper, linear motors, artificial vision, control system, vacuum system.

1 Introduction

The industry at present, stand facing a number of challenges when it comes to handling and packaging of various kinds of products [1], [2]. This is due to environmental conditions and hygiene that are involved in this process, among other technical features. Additionally, the efficiency and speed in carrying out the processes are other matters that must be improved. This difficult has prompted many companies to use manipulator robots in much of their production lines [1], [3-5]. Since the emergence of industrial robots in 1964, their main task has been to replace the human operator in repetitive actions and often dangerous tasks [6-7]. Noticeably, this is to increase efficiency in the scheduled tasks and the safety for the human operator.

Food industry requires a very high speed manipulation of soft goods at an affordable cost. However, it is well known that industrial processes involved in food product manipulation are very difficult to automate because of many practical constraints [1-5]. The packaging of products in many companies is done by hand, this form of process leads to a low performance and too much production time lost. Additionally, there is a lack of guarantee of hygienic of the product. This increases the costs of company productivity.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_52,

Consequently, for diminish the losses in productivity and enhance efficiency and hygiene in the handling of products, it must adapt and integrate different technologies in the original system, for the purpose of enabling the optimal use of its inherent characteristics, in applications dedicated to the processes implementation of the company.

In this paper, motivated to solve a problem of industrial application, we focused, particularly, in the food industry (bakery industry), and specifically the area at high speed packaging of various products (pastries), with the aim of achieving significant improvements with respect to the traditional system of packaging manual. The technologies integrated to the original system for manipulation of pastries include artificial vision systems, pneumatic system, magnetic linear motors and its control architecture, as well as a several sensors in all system. The current computer vision systems [8-10] and the designs of high speed robot manipulators are making possible such kind of applications. The integration of both technologies to accomplish with real production restrictions will lead to an important technological breakthrough.

The heavy objects manipulation it is well known in the industrial process [6-7], [11-14], and the manipulators robots were fabricated with that goal. In addition to these kinds of industries there are others one that use special manipulator to handle small objects [15-18]. On the other hand, the robots have giving response to the pastry industries [1], [3-5]. In this case the robot must carry out the handling and palletising in soft way and fast way. It is suitable to avoid the damage of pastries.

The Department of Automatic Control of Industrial Automation Institute has developed a new integrated system for high speed manipulation of pastries that incorporates the recent technical advances. This system is composed by one three degrees of freedom Cartesian robot, driven by linear motors [19-20], one special gripper with one extra degree of freedom, one vision system, and an overall control system. The objective of the vision system is to determine the centre of the objects and their orientation, being the acquired data, subsequently used by the control system to enable real time object manipulation. The fundamental characteristics of the developed system are based in the robustness of its construction, its versatility, and the reliability of the overall control system.

2 Overall System Design

The full system is made of a robotic arm, a belt conveyor, a vision system, an industrial computer with the control algorithm and its dedicated hardware, a pneumatic system, and others support devices. The high speed manipulator is designed as a robotic arm of three linear degrees of freedom in the Cartesian space X, Y, Z. The robotic arm configuration is prismatic, but, added to this, the manipulator has an additional rotational degree of freedom in Z axis (Fig. 1(a)).

A study was realized by simulation to determine the best method for the capture of pieces in the minimum possible time. After that the mechanical configuration more acceptable according to these results was proposed.

The samples used in the modelling stage were products of industrial pastry with some different characteristics: variable size, composition, different shape (e.g. presenting

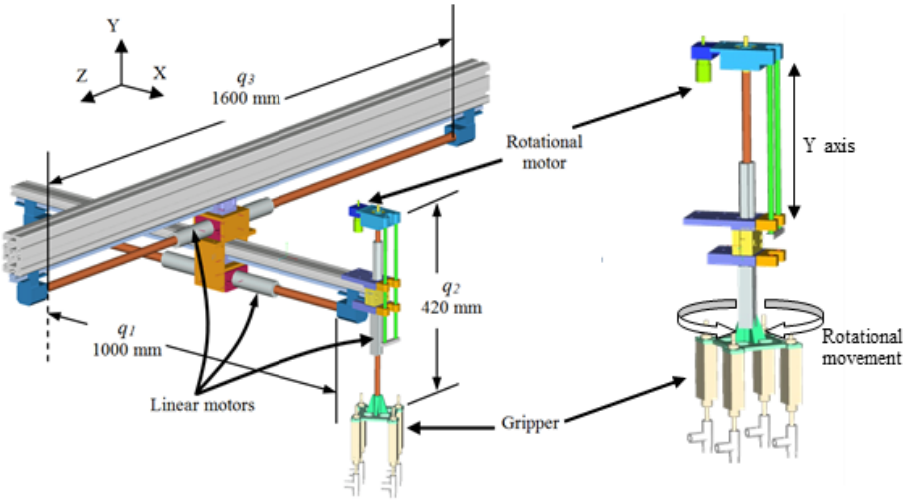


Fig. 1. (a) Robotic arm and kinematics parameters, (b) Robotic arm with four vacuum grippers

holes in the centre), etc. All these data were entered in a simulation environment using MATLAB®, because this presented several advantages from the calculation point of view.

The first commercial systems for product manipulation are using only one gripper per robot [3], however, nowadays there are some manipulators with more than one gripper per robot, but with one alone displacement for all grippers [1], [4-5]. In our approach we proposed to use several grippers, and the problem was to determine how many and their physical location according to the application requirements. The grippers are of two kinds: vacuum cups and vacuum grippers. The simulations were carried out in MATLAB®. It was tested different number of grippers for optimising the efficiency of the system. The first estimations indicate that with three or four grippers the objectives could be achieved.

The Fig. 1(b) shows that the gripper has one prismatic degree of freedom. More of four grippers could be employed, but the resulting mechanical design will more complex and could result in mechanical vibrations due to the high accelerations it will be subjected, what could be hazardous for the robustness of the structure.

2.1 Mechanical System

The robotic Cartesian arm is placed in frame assembly by special aluminium profile, and it is appropriately linked to diminish the vibration in tasks of pastry handling (Fig. 1(a)). The gyratory degree of freedom is placed in the superior part of the

especial gripper. Additionally, there are four pneumatic cylinders placed in the gripper (robot end-effectors). They are considered as additional on-off prismatic axes. The actuators of the manipulator are magnetic linear motors used for prismatic movements [21], and rotational motor used for rotational degree of freedom, both driven by PWM signals. This kind of design carried out a compact and light manipulator robot, and with very good features in speed, mobility, and maintenance.

The move very fast above the pieces to be grasped and to be activated just when the corresponding vacuum cup will be flying over the selected piece. With the first two of the positioning of the special gripper on the working zone is achieved. The rotational DOF is used for its orientation before the grasping of the object.

The system of the robot manipulator, as mentioned above, uses linear motors manufactured by LinMot®. These linear motors have two main parts: a slider and a stator. The stator is made up winding, bearings and sensors (to detect the slider position and temperature), integrated within a metal cylinder. The slider is formed by a set of small permanent magnets whose poles, positive and negative, are adjacent, and covered with a metal casing. Finally each party is protected against injurious elements of the environment.

The linear motors are associated to a servo controller. The communication of the servo controller with the industrial computer is through of RS485 interface. Likewise, the control system is provided of software that allows controlling the robot trajectories and monitoring several characteristics of the system like temperature, stopping effects, errors signals, etc. The communications between the control systems (in the industrial computer) with servo controller, it is carried out using the ASCII communication protocol.

The manipulator robot arrangement is based in the line robot concept; understand it like the realization of works about of objects supplied on transport line. Thereby, it was decided employ dc linear motors for the prismatic displacements. However, the linear motors assembly shown several difficulties. For example, the linkage of the axes X and Y, it should assurance, light weight and robust joint. Thus, several assemblies were designed and simulated in Pro-engineer software of mechanical design. As a result of this work, we concluded in attach the linear motor sliding to the frame and allow the mobility of the linear motor stator.

In this way, the array formed by the X and Z axes are coupled to special sliding rails. This system allows support high torques produced in different rail points. This kind of attaches permit the assembly stability and it avoid the disturbances in the system produced by its dynamic and the mechanical configuration of the robot. Since to the importance of the slider parallelism and the support structure of the robot, it was designed one support with inner damping system Fig 3. The main characteristic is the great capacity to absorb disturbances along the axis X, besides of reduces possible inconveniences of parallelism between the slider and fixed part of Z axis.

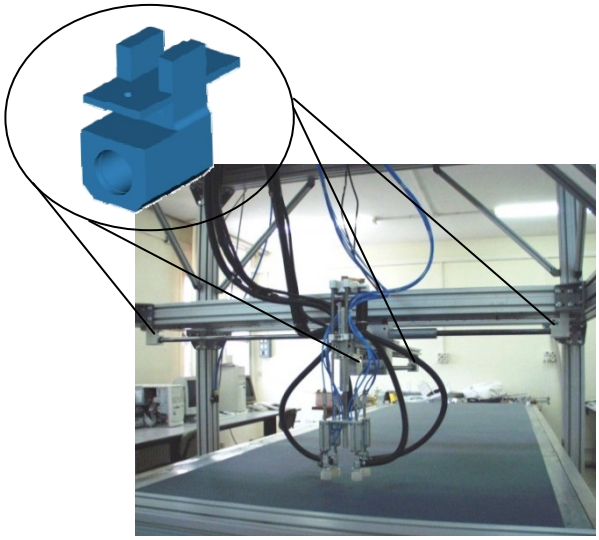


Fig. 2. Support structure with damping device

The inverse kinematics of the handling machine is determined by its Cartesian configuration [6-7], [12], [22]:

$$q_1 = -p_x; \qquad q_2 = -p_y - l_4 + a_1; \qquad q_3 = p_z \tag{1}$$

where, l_4 is the maximum motor stroke of the Y axis; a_1 is the orthogonal distance between the Z and X axes; and q_i are Cartesian DOF of the manipulator (refer to Fig. 1).

2.2 Electronic and Sensors System

The electronic and sensors system of the manipulator Cartesian robot consist in an industrial computer, frame grabber, data acquisition board, control boards, servo controllers, power driver boards, several sensor for linear and rotational displacement, temperature, velocity, etc.

The hardware for the data acquisition and control consists of master/server single board computer Pentium III 700 MHz with 4 PID/slave processors (control boards). Also, there is one data acquisition board with 16, 12.5 KHz effective, A/D channels each of them. The output of control boards are PWM ports to 20 KHz. The same control board is able to drive both motors, the linear and rotational motors.

The sensorial system of handling machine consists in: three linear position sensors located in each linear motor, two incremental optical encoders placed in rotational motor axis and conveyor belt, temperature sensors placed in each motor, velocity sensor placed in the driver of conveyor belt, and one CCD ½" monochrome camera.

In the machine had been used two different kinds of controller boards. One of them is used to control the linear motors through of interface. The other one had been designed in the Institute of Industrial Automation for purpose generals to control of various dc motors. By other hand, the pneumatic system has been controlled through a power electronics switching. This board serves as an electronic interface between the computer industry and valves of the pneumatic system.

2.3 Pneumatic System

The handling machine also consists of a pneumatic system, which is used to drive of pneumatic cylinders that are extreme to the manipulator gripper (end-effector) and the vacuum process which allow the handling of the pastry through the vacuum suction cups. This gripper is installed at the bottom part of the Y axis of the manipulator. The gripper consists of four pneumatics cylinders and its electrovalves of control Fig. 3.

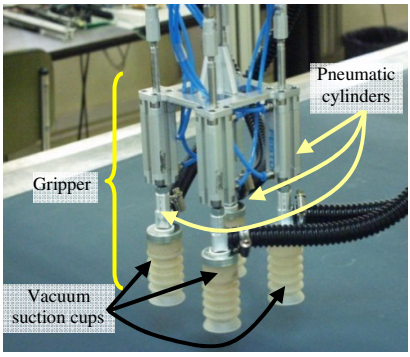


Fig. 3. Gripper of the manipulator

The electrovalves that control the air flow of the pneumatic cylinders and the vacuum system are low pressure (0-2 bar), of fast response time (30 ms) for this process, of low energy consumption (12W). The vacuum suction cups are used for suitable manipulation of breads, cakes, rolls, pies, cookies, and other similar pieces of pastry. These cups are made of silicone used, generally, to handle the pastry with low vacuum to avoid the risk of broken. In addition, these suction cups are able to handle heavy cakes.

2.4 Computer Vision System

The artificial vision system consists of a camera (JAI) monochromatic ½" CCD of high sensitivity for industrial applications, with functions of automatic gain adjustment, output for external video display and external trigger shutter mode. In addition, this system of artificial vision, has a video acquisition card (Frame grabber PXC200), featuring PCI bus to acquire images in real time, in memory or VGA display, capture resolution of 640 x 480 (NTSC), 768 X 576 (PAL and SECAM) and application software developed in Borland.

The computer vision system realises the image capture and its transformation and analysis in order to proportion to the control system of the robot the location and orientation of the objects that it encounters meet in their working area. The implemented algorithms for image processing are summarised by the execution sequence shown in Fig. 5, [9], [23].

It is well known that one of the most important stages, not only of the vision system, is the data acquisition. In this case the image acquisition, for it must take into account several aspects related the hardware and software. In this case we are referring to the camera, the illumination, and subsequent calibration process of the camera in relation to the work space [8].

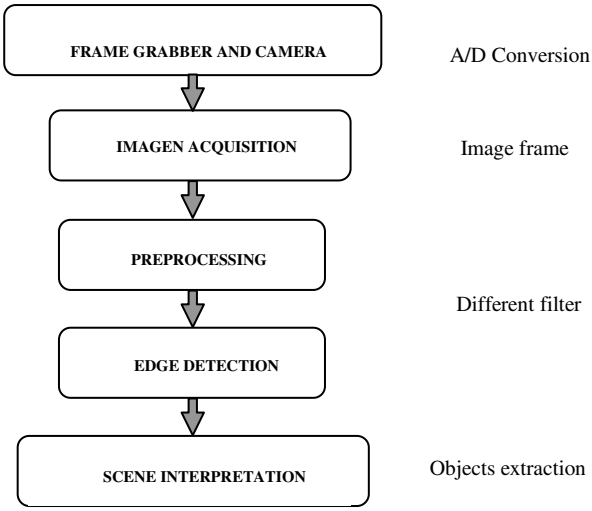


Fig. 4. Steps sequence in artificial vision system

2.5 Edge Detection and Scene Interpretation

In this stage the principal objective is to evaluate if each pixel of the image belongs to or not to the object of interest. Pixels with values of "1" belong to the object, but pixels with values of "0" are below the threshold binarization, then they do not belong to the object. In this step was considered several criteria (e.g. size, holes of the product, types of pastries, etc.) In Fig. 5 is shown a flow diagram illustrating the algorithm to be followed for the search and determination of the contour of the object through the computer vision system [10], [24-26].

The final stage refers to the calculation of the coordinates (x, z) of the objects and their orientation by mean of the vision system.

The strategy to follow in this procedure is to determine the estimate closer to the centre. Therefore, two data arrays in a structure are used for storage them at the moment when the tracking of image is performed, these arrays are formed of the values of x and y coordinates [27]. Subsequently, to obtain the centre of the object, is the term applied (2), where tcont is the size of the data structure around the contour of the object.

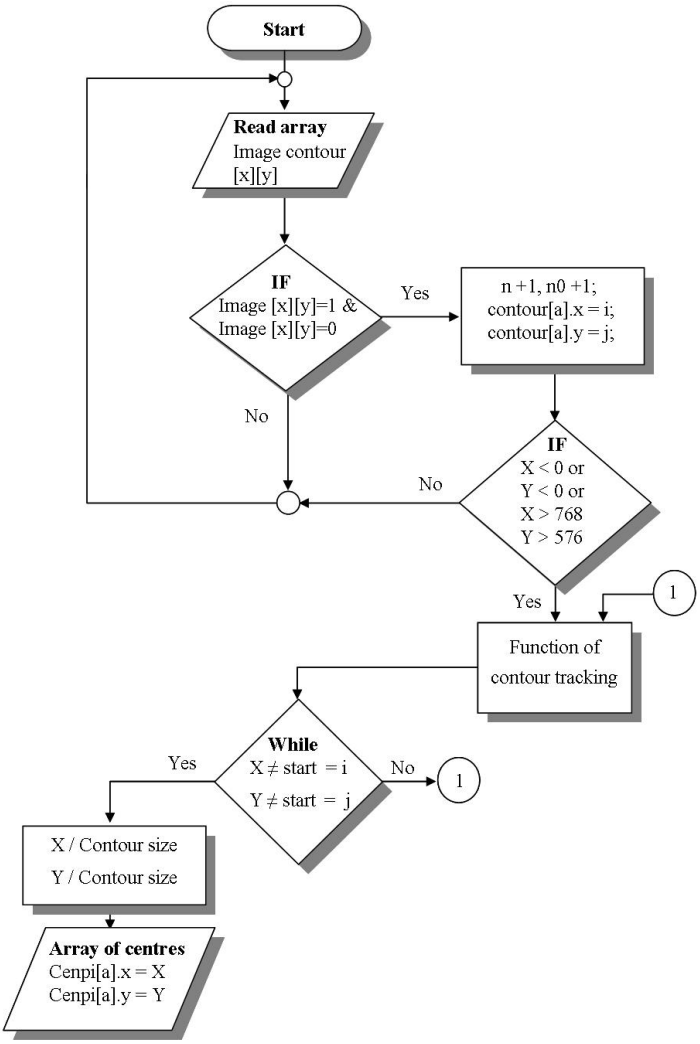


Fig. 5. Flow chart to search of objects

$$\hat{x} = \sum x / tcont; \qquad \hat{z} = \sum z / tcont \tag{2}$$

Afterwards to get the orientation of the object were proceeded to assess the distance between pixels. We have used the expression known as the Euclidean distance [12], applied on the contour points from the centre of the object (3). With these results, the pixels closer and farther of the objects located. In Fig. 7 presents the pixels closer and the centre of each object.

$$D_e\left(p_i, p_j\right) = \left[\left(x_i - x_j\right)^2 + \left(z_i - z_j\right)^2\right]^{1/2} \tag{3}$$

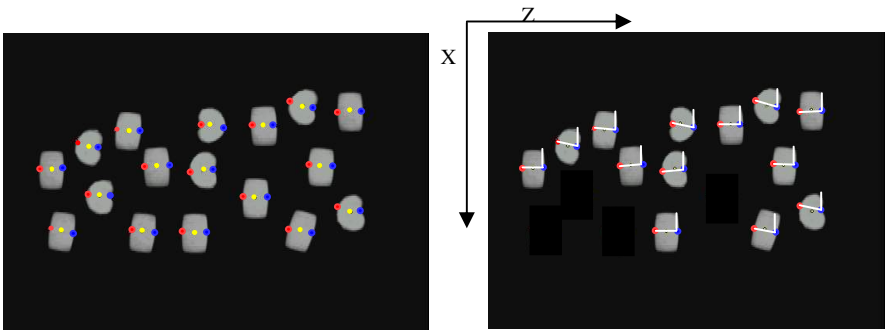


Fig. 6. (a) Representation of the centre and closer points of the objects, (b) Orientation of objects

The representation of the orientation of objects is shown in Fig. 8. It is possible to observe an imaginary line between the pixels closer. Additionally, it is extracted the slope of the imaginary straight line to get the orientation of the object. The determination of coordinates and orientation of objects (pastries) is sent to the robot manipulator control system, to locate the pastries on the conveyor belt.

In Fig. 7 is shown a flow diagram illustrating the algorithm for the determination of the minimums points. This algorithm is implemented in the control system to know the orientation of the objects, too. With basics mathematical routines it is possible to calculate the orientation of the objects with respect to the X axis.

2.6 Control System Architecture

The handling and palletising machine has been designed in the Institute of Industrial Automation to solve the problem of manipulation of pastry of one industrial bakery of the region [20]. The main objectives for the development of this project were: the fast manipulation of the products and the delicate handle of the different types of cakes that are transported.

Fig. 8 shows a block diagram illustrating the control system architecture of the handling and palletising machine. It can be noticed the modularity in the selected approach, which would permit a suitable expansion of the existing capabilities of the machine. The main characteristics of the electronic hardware and the sensor system were described above. The entire control algorithm is running on Neutrino II real time operating system, which assures high levels of performance and reliability in this type of application.

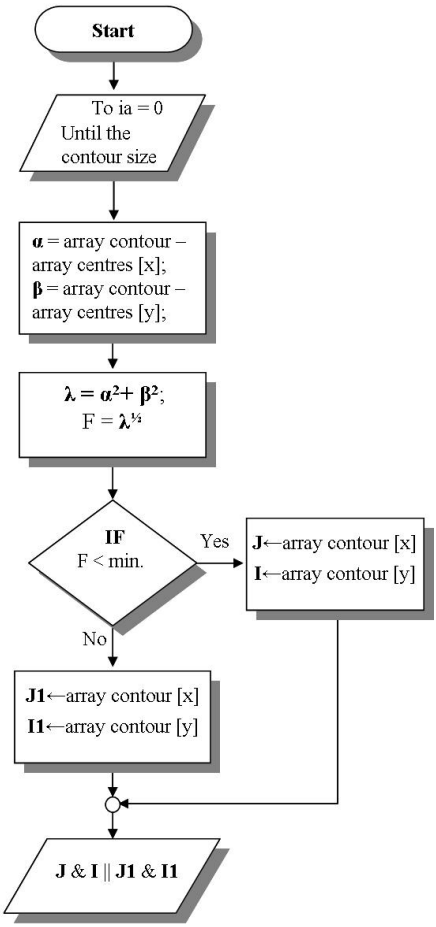


Fig. 7. Flow chart for determination of minimums points

In this machine are implemented some control systems that depend on each actuator. This is dc linear motors, dc rotary motor and pneumatic cylinders. Each of them operated with their own drivers, and with their particular control strategies.

First, the dc linear motors are linked with a servo-controller. The servo-controller is communicated to industrial computer through interface RS485. Similarly, the control system is equipped with software that allows to control and monitor trajectories, temperature, stops, errors, etc. The control system for linear motors uses several feedback loops to control the position, the velocity, and electric current

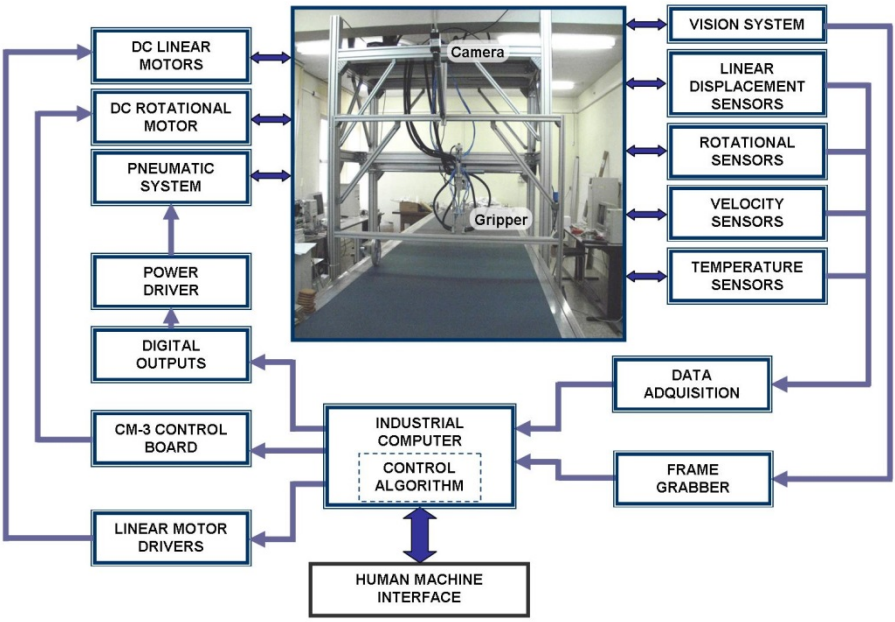


Fig. 8. Control system architecture of the manipulating machine

The dc linear motor operation on the vertical axis (Y axis) may result to asymmetrical behaviour of the system, by the distribution of mechanical devices and the movement continued to go up and go down. In classical literature in robotics such problems are solved applying a control with gravity compensation. For compensate any presence of disturbances, an algorithm of compensation of electric current was implemented. The function is presented in equation (4). This electric current compensates the problem of asymmetry, and together with the tuning of a PID controller, the displacement of the dc motor in the vertical position is compensated. The PID controller was tuning with P = 0.234 A/mm, I = 0.045 A/s/mm, and D = 1.5 A·s/mm. In Fig. 15 is shown the control system diagram for the vertical prismatic articulation.

$$Offset\_Current = \frac{m \cdot G}{c_f} \tag{4}$$

where,

m, load mass [Kg]; G, gravity [m/s^2]; c_f=$22N/A$, force constant of the motor [N/A].

On the other hand, uses the CM-3 control board to drive the dc rotational motor. This control board is designed by Institute of Industrial Automation to control dc motors for implementation of robotic systems and industrial automation systems. This control subsystem has a digital PID controller and its output is PWM. Likewise, this subsystem has a power driver to condition the PWM output to a voltage signal suitable for the dc motor.

3 Planning of the Manipulator Motion

When the vision system ends its calculations, the recognition and extraction of characteristic, the system calculates the viable trajectory for the grasping of the objects, and all data is sent to the controller of the robot, where the system examines different strategies depending on the collected data. These criteria are described as follows:

Central strategy: It is a process when the robot takes the product inside all working zone. This method is possible to use if the total time for the product does not exceed a time threshold, and depends upon the speed that can reach the manipulator and also depends on the total travel distances required to attain the deposit place.

For achieve the coordination of the distances travelled and time of flight, the original control algorithm had to be adjusted with some variables found experimentally. Primarily were implemented variables adjusted to the trajectory of the manipulator on the Z axis (see expressions (5), (6), and (7)). In this way the flight time, with security and reliability, allowed the grasping objects on the coordinates determined by the vision system.

In addition, to achieve a coordinated movement, the algorithm that determines the position for the axis X was modified. The equation (8) determines the correction of position error in X coordinate of the piece, which is linked to the expression (9). This assessment depends on the object position with regard to the work area delimited by the robot.

$$distZ = \left| Coord.z - Coordprev.z + 155 \right| \tag{5}$$

$$t\_flight = \sqrt{distZ \cdot 1.5 / acc} \tag{6}$$

$$distX = 3.833333333 \cdot t\_flight \tag{7}$$

$$distX = distX + 40 \tag{8}$$

$$Pos\_X = 410 - \left| \Delta Position - coord.x \right| - distX \tag{9}$$

Right and left strategy: It allows the product grasping, in any side of the working zone left or right one (along the axis Z), having like centre the axis and of the full working area (Fig. 9). All this can be made online, so that the system does not generate failures unnecessarily.

Thus, it defines a control strategy depending on the position of objects, stating that: "By having an object gripped in the terminal part of the robot manipulator and the next object to capture will come close the position of the last taken on the same area, it must wait until it reaches the working area of the robot manipulator to his capture. The next step, the robot will move to the area of deposit; otherwise it will leave the object in the deposit area and will return to grasp the next object."

The discrimination of objects is realized collating the position of the last object with the first position that found in the structure of coordinates, implemented for the type of coordinates of the axis Y. It also matches the position of the object on the X axis, which has been divided between two parties to delimitate the left side of the right side.

This strategy allows the use of a greater number of manipulators, to which it will determine by what side of the band it will work. Additionally, it is possible to execute this strategy online. The time to interrupt this process is short, avoiding unnecessary stops or avoiding that system reach the collapse.

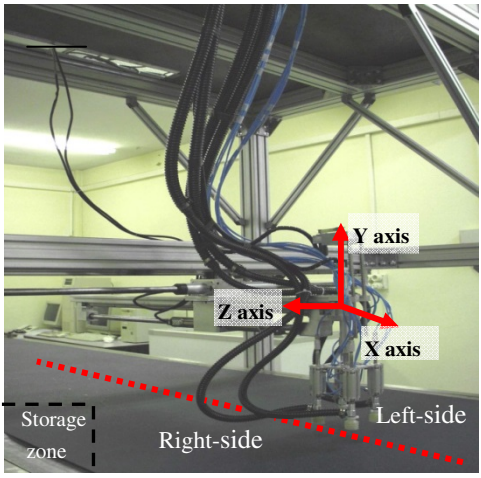


Fig. 9. Areas of work of the manipulator

Deposit strategy: This strategy is based in the way and distribution of the product on every pallet or storage box. The main program has annexed a palletizing algorithm. Its function is to generate coordinates of deposit, which are transferred to a table of positions palletizing.

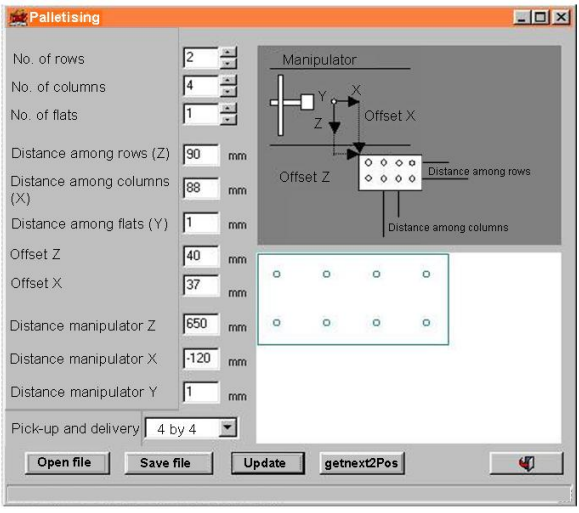


Fig. 10. Definition of palletising coordinates

Thus, the user can manually configure what will be the order and distribution of objects in pallet storage. As show in Fig. 10, at the top of the window, the terminal piece of the robot is the reference point of the coordinates of deposit. In this way, the values to enter each table are in agreement on this point. These parameters can be stored and then loaded by the system for each piece.

3.1 Orientation of the End-effector

The products are transported in a disorderly way through the conveyor belt. These pieces are placed on storage boxes, ordered and directed, in groups. The order is obtained by the prismatic degrees of freedom of the robot. However, the orientation is done by the end-effector of the manipulator. This orientation is obtained by rotating the end-effector. The angles of each piece are obtained by the vision system. Subsequently, it is estimated a new angle, referenced to the X axis of the robot.

Theoretically, the orientation of the end-effector can be considered quite simple. However, if the end-effector has four small grips, to solve the problem is not so simple. Thus, it should be considered the angle of each of the small grips and their subsequent tasks. For carry out this task a new angular coordinate by each small gripper was established.

In Fig. 11 shows an imaginary circle that serves as a reference for determining the angle per each individual gripper. Expression (12) facilitates the calculation of these angles for both axes. The value obtained in the previous expression is added to the coordinate of each piece and assigned to each small gripper according to a stipulated order in the control strategy.

$$X' = hip \cdot sen(\alpha); \qquad Z' = hip \cdot \cos(\alpha); \tag{10}$$

where, '*hip*' is the square root of twice the radius of the circle squared.

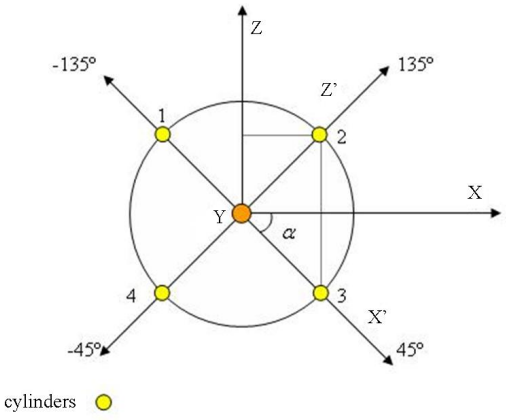


Fig. 11. Gripper orientation

However, in the first experiments, the angle of orientation obtained was inaccurate. The variation was ±5 degrees. This error caused loss of pieces by the robot manipulator. In equation (11) there are two variables, ang1 and ang2, which correspond to the angle at the first image capture and the angle at the third image capture, respectively. As referred in the previous expression, it was necessary to carry out image captures consecutively. The values ang1 and ang2 are compared in the equations (12) and (13), to obtain a new angle and its subsequent comparison between them. The variable ang1a corresponds to the angle of the piece in the second capture of vision.

$$\left|ang1 - ang2\right| > 30 \tag{11}$$

$$angA = ang1 - ang1a \tag{12}$$

$$angB = ang2 - ang1a \tag{13}$$

$$\left|angA\right| < \left|angB\right| \tag{14}$$

Afterwards, if angA is less than angB (14), it is used the angle found in the equation (15) to locate the object that has been evaluated. If the above evaluation is contrary, the expression (16) is applied; in this case ang1 and ang2 are matched. All this is correct if in equation (17) has been true.

If the expression (11) is false, it must determine the difference between ang1 and ang2, checking if the result is less than 30 degrees (17). If so, ang1a will become equal to the value of ang2 (18). This new value is regarded as the correct angle of the object.

$$angC = ang2 = \left(ang1 + ang1a\right)/2 \tag{15}$$

$$ang1 = ang2 = \left(ang2 + ang1a\right)/2 \tag{16}$$

$$angA1 = \left(ang1 - ang2 > 30^{\circ}\right) \tag{17}$$

$$ang1a = ang2 \tag{18}$$

3.2 Approach of Implementation

At the moment when the system starts its work, the vision system captures one image. At the same time, the system reads the encoder position located on the belt conveyor. Subsequently, the robot takes four objects, and then goes to the area of deposit; the objects are stored and run the procedure again. This process is performed every 2.5 seconds. In Fig. 14 shows a block diagram of error correction. These corrections are made before the coordinates are sent to the table of positions.

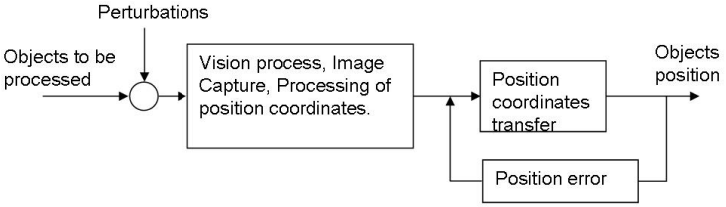


Fig. 12. Block diagram correction of errors of final position of the pastries

It is possible, using the information acquired, to estimate the position of the pieces when they enter the working area of robot manipulator, and thus anticipate the manipulator movements and reduce the time to grasp the objects. This process is carried out comparing the information that is in the table of positions with the data obtained of the encoder. In this way the system anticipates the possible position of the pieces.

Several experiments were carried out to test the performance of the manipulating machine and its control system. In Fig. 15 shows a series of images that demonstrate the operation of the manipulator. The task illustrated in Fig. 15 a tough time of 20 seconds. It is understood that the cookies were placed correctly inside the storage box, and they were not destroyed by the gripper of the manipulator.

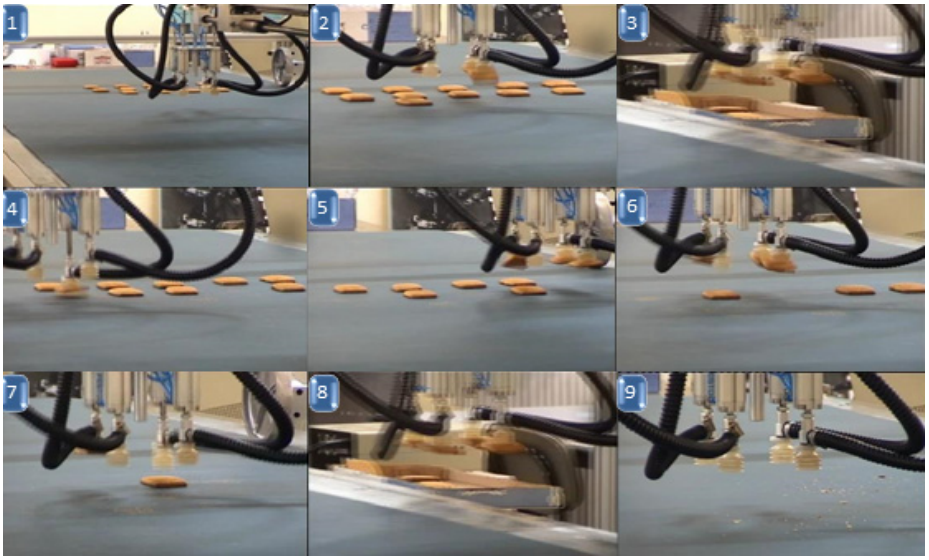


Fig. 13. Process of pick-up and delivery of cookies

4 Conclusions

This paper has focused on the design and implementation of a robotic system whose purpose is to manipulate at high speed a wide range of pastry products that are transported on conveyor belt. The system made has involved the integration of diverse technologies including the linear dc motors. The end result is a robot equipped with computer vision capable of manipulating pieces of different shapes and sizes that are placed in disorder on the conveyor belt and travelling at relative high speed. The system made has been tested in practice with very satisfactory results. Additionally, it opens the possibility of adapting the robot and the technology developed to other areas of production.

Several test were realized with different types of pastry products, where in a 98% it was taken efficiently, and the 2% residual were products that presenting holes in their surfaces. For this kind of products, other method must be implemented, for example, the interchange of a mechanical clamp. However, with 98% of efficiency with the use of four grips implemented in one robotic arm the system is very useful.

With this manipulating machine the high speed grasping of the soft objects (pastries) has been optimised. These objects can be several forms and sizes, and they can move disorderly on the conveyor belt. In addition, the bakery industry imposes rigorous restrictions for the realisation of these processes. However, this robot manipulator can perform the tasks of handling and palletizing of different kinds of cookies, with high efficiency and low cost.

References

1. Ixapack: Packaging Machine, http://www.ixapack.com
2. Komatek Robotics and Conveying Systems, http://www.komatec.de/en
3. Holtem, H.: Packaging a piece of cake, from Packaging Magazine, http://www.abb.com/robotics
4. Sayoa maquinarias a medida, http://www.sayoa.com
5. Walch, H.: Salami under control, from Packaging Magazine, http://www.abb.com/robotics
6. Asada, H., Slotine, J.J.E.: Robot Analysis and Control. Wiley & Sons, Inc., USA (1986)
7. Barrientos, A., Peñín, L.F., Balaguer, C., Aracil, R.: Fundamentos de Robótica. McGraw Hill, Interamericana de España (1997)
8. De la Escalera, A.: Visión por Computador. Prentice Hall, Spain (2001)
9. González, J.: Visión por Computador, Paraninfo. ITP An Internacional Thosom Publishing company, Madrid (1999)
10. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision. Addison-Wesley, U.S.A (1992)
11. Engelberger, J.F.: Robotics in Practice. Kogan Page, London (1980)
12. González de Santos, P., Jiménez, M.A.: Robótica industrial, Monografía de Master TIF. Universidad Politécnica de Madrid, Madrid (2000)
13. Lewis, F.L., Abdallah, C.T., Dawson, D.M.: Control of Robot Manipulators. MacMillan, New York (1993)

14. Luh, J.Y., Lin, S., Lin, C.S.: Approximate joint trajectories for control of industrial robots along Cartesian paths. IEEE Transactions on System, Man, and Cybertics 14, 444–450 (1984)
15. Makita, S., Kadono, Y., Maeda, Y., Miura, S., Kunioka, I., Yoshida, K.: Manipulation of submillimeter-sized electronic parts using force control and vision-based position control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 1834–1839 (2007)
16. Randall, J., Bowman, M., Johansson, R.: Control strategies in object manipulation tasks. Current Opinion in Neurobiology 2006 16, 1–10 (2006)
17. Sato, T., Koyano, K., Nakao, M., Hatamura, Y.: Novel manipulator for micro object handling as interface between micro and human worlds. In: Proc. of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, Yokohama, Japan, pp. 1674–1680 (1993)
18. ScienceDaily: Fastest Industrial Robot on the Market, http://www.sciencedaily.com/releases/2007/05/070503102529.htm
19. Armada, M.: Control de procesos por ordenador. Monografía de Master TIF. Universidad Politécnica de Madrid, Madrid (2000)
20. Sarria, J., Prieto, M., Montes, H., Armada, M.: High Speed Object Manipulation Using Computer Vision. In: Proc. of the 13th International Symposium on Measurement and Control in Robotics, ISMCR, Madrid, Spain, pp. 271–274 (2003)
21. Jacek, F., Gieras, Z.J.: Linear synchronous motors. CRC Press (2000)
22. Fu, K.S., Gonzalez, R.C., Lee, C.S.G.: Robotics: Sensing, Vision and Intelligence. McGraw-Hill, New York (1987)
23. González, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, USA (1992)
24. Efford, N.: Digital Image Processing. Addison Wesley (2000)
25. Parker, J.R.: Practical Computer Vision using C. Wiley Computer Publishing, USA (1993)
26. Parker, J.R.: Algorithms for Image Processing and Computer Vision. Wiley Computer Publishing, USA (1997)
27. Seul, M., O'Gorman, L., Sammon, M.: Practical Algorithms for Image Analysis, University of Cambridge (2000)

Author Index