Advances in Intelligent Systems and Computing 252

Manuel A. Armada
Alberto Sanfeliu
Manuel Ferre *Editors*

ROBOT2013: First Iberian Robotics Conference

Advances in Robotics, Volume 1

Springer

Advances in Intelligent Systems and Computing

Volume 252

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 1

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 978-3-319-03412-6 ISBN 978-3-319-03413-3 (eBook)
DOI 10.1007/978-3-319-03413-3
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013952937

Printed on acid-free paper

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

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 V: Intelligent Systems and Robotics

Part VI: Agricultural Robotics

Part VII: Humanoid Robots

Part VIII: Legal and Ethical Aspects in Robotics

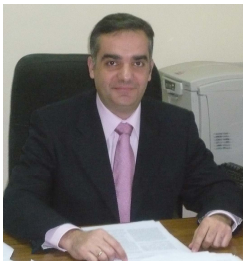
Part IX: Robot Vision

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 I
Marine Robotics

RANSAC Based Data Association for Underwater Visual SLAM*

Antoni Burguera, Yolanda González, and Gabriel Oliver

Dept. de Matemàtiques i Informàtica, Universitat de les Illes Balears
Ctra. Valldemossa, Km. 7'5, 07122 Palma de Mallorca, Spain
{antoni.burguera,y.gonzalez,goliver}@uib.es
http://srv.uib.es

Abstract. This paper presents an approach to perform data association in a monocular visual SLAM context. The proposed approach is designed to avoid the detection of false associations by means of RANSAC, and is well suited to help in localizing a robot in underwater environments. Experimental results embed the data association in a *trajectory-based SLAM* in order to evaluate its benefits when localizing an underwater robot. Qualitative and quantitative results are shown evaluating the effects of dead reckoning noise and the frequency of the SLAM updates.

Keywords: Underwater robotics, Visual SLAM.

1 Introduction

Thanks to recent technological advances, the sub-aquatic world is more accessible for exploration, scientific research and industrial activity. At present, *Remotely Operated Vehicles* (ROVs) are commonly used in a variety of underwater applications like surveying, scientific sampling, rescue operations or infrastructure inspection and maintenance.

Trying to overcome some of the intrinsic limitations of ROVs, such as their limited operative range or the need for a support vessel, *Autonomous Underwater Vehicles* (AUVs) are progressively being introduced, especially in highly repetitive, long or hazardous missions. Because they are untethered and self-powered, AUVs offer a significant independence from surface support ships and weather conditions. This can notably reduce the operational costs when compared to missions conducted with tethered ROVs.

1.1 Underwater Vision

Light attenuation and scattering, non-uniform lighting and shadows, colour filtering or suspended particles are frequent difficulties when using optical sensors

* This work is partially supported by the Spanish Ministry of Research and Innovation DPI2011-27977-C03-03 (TRITON Project), Govern Balear (Ref. 71/211) and FEDER funds.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_1,

underwater [3]. Contrarily, acoustic sensors have interesting properties in these scenarios and that is why sonars are the most common choice for AUVs [15].

However, acoustic sensors have low spatial and temporal resolution. For example, the *Mechanically Scanned Imaging Sonar* (MSIS) used in [15,4] among others, has a linear resolution of 10 cm and spends more than 13 seconds to gather a full view of its surroundings.

Conversely, cameras have high spatial and temporal resolution. Although cameras are not appropriate in open waters where it can be difficult to see the seabed or other reference points, they are convenient for surveying or intervention applications, where the vehicle can navigate close to the bottom or it has to stay near an object of interest [3]. Examples of such applications are mosaicking and station keeping. Also, recent literature shows that underwater cameras are being used more and more to perform visual *Simultaneous Localization and Mapping* (SLAM) [9].

1.2 Underwater SLAM

Nearly all advanced mobile robotic tasks require some knowledge of the robot location in the environment. The most successful approach to estimate the robot pose is the SLAM [16]. A wide variety of SLAM algorithms exists, mostly based on feature maps composed of straight lines and corners in the environment. Although this approach has proved to be robust and accurate in structured environments it has two important problems. On the one hand, this kind of features can be easily found by means of range sensors, but they are error prone when using a single monocular camera. On the other hand, depending on geometric features reduces the scenarios where the robot can be deployed, especially in underwater scenarios where man made structured environments are uncommon or can be hidden by flora and fauna grown on them.

Common approaches to visual SLAM [6] do not rely on the existence of geometric features in the environment but on the presence of visual features in the images. These approaches extract visual features from the images provided by the on-board camera and try to match them against those stored in the SLAM map. If matching is possible, then map and robot pose are improved. Otherwise, non matching features are stored in the SLAM map. Thus, the feature matching method is crucial when adopting this approach.

In order to perform visual SLAM, the feature matcher, which is in charge of performing the data association, has to meet two opposed requirements. First, it has to be fast enough to reach the desired frame rate. Second, it has to be robust enough to work with images gathered from arbitrarily separated places. This last requirement, which makes possible to detect previously visited places, is known as *loop closing*.

Some authors focus on the first of the aforementioned problems and develop very fast feature matchers providing good results as long as matched images are similar [11,12]. Unfortunately, these techniques are not well suited to close loops. Some other authors focus on the second problem using complex features

relying on complex descriptors [13,2] which reliably detect loops at the cost of large execution times.

This paper proposes a data association method which is well suited to perform visual SLAM in underwater environments although it could be also used in terrestrial or aerial scenarios. The presented approach is able to decide whether two images overlap or not and, in case they do, it estimates the roto-translation that better explains the overlap. Thanks to that, the proposed data association method is especially well suited to detect loops. Moreover, a *trajectory based Extended Kalman Filter* SLAM (EKF-SLAM) approach is described and used to evaluate the proposed data association. Experiments show the ability of our approach to detect and measure roto-translation between two images even if the overlapping area is very small.

2 Data Association

2.1 Feature Detection and Tracking

Although detection and tracking of visual features is a growing area in the computer vision community nowadays, there are several widely used and well established methods, each one with its own advantages and drawbacks. Our proposal is based on the *Scale Invariant Feature Transform* (SIFT) which is robust to changes in scale and rotation, performs well with different illuminations and camera viewpoints and has proved to be a good choice to register images in the localization and SLAM contexts [13]. A description of SIFT is out of the scope of this paper. The reader is directed to the literature for a detailed description of the method.

There are some optimized and publicly available SIFT implementations, such as VLFeat [17], which has been used in the experiments shown in this paper. However, we would like to emphasize that our proposal does not rely on any specific feature detector, and it can be applied with minor changes to any existing feature detector and matcher. Consequently, although during the rest of the paper we are going to refer to SIFT features, other feature detectors can be used.

2.2 Underwater Scenarios

A common task in underwater surveying missions is to sweep over a certain area in order to map the terrain or to look for an object, such as archaeological or shipwreck remains. In general, this kind of tasks are performed at an approximately fixed distance to the bottom and, thus, can be considered a 2D problem.

Figure 1-a exemplifies a typical sweep trajectory followed by a robot which is endowed with a bottom-looking camera. A circle is depicted over the trajectory every ten camera frames. The rectangles illustrate the camera field of view,

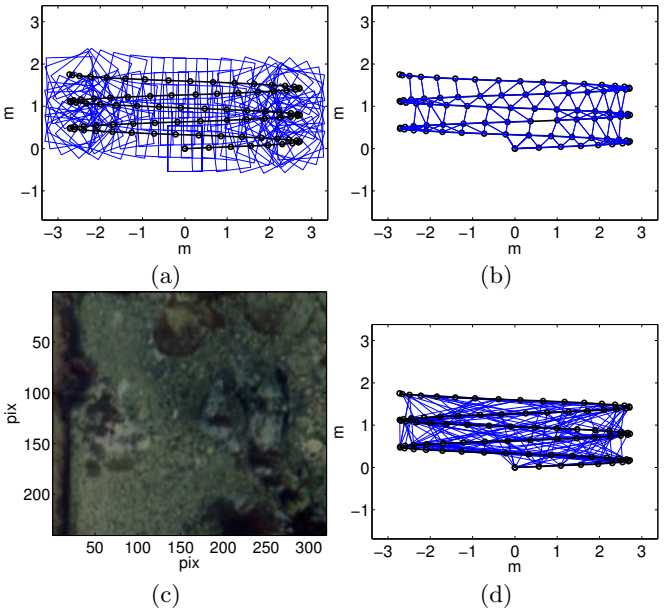


Fig. 1. (a) Sweeping trajectory. The rectangles illustrate the camera's field of view. (b) Existing loop closures. (c) Example of underwater image. (d) Loops detected using plain SIFT.

so that the overlap between frames can be appreciated. Figure 1-b depicts the possible loop closures that can be found in the previous example taking into account the camera's field of view. A line between two circles means that the two corresponding images overlap.

Typical sea floor images pose two important problems to long term matching and, thus, to loop closures. First, it may be difficult to find reliable features because of the low contrast and the lack of distinctive image regions. Second, most of the gathered images look similar, leading to similar sets of features and descriptors even between non overlapping images. Figure 1-c shows an example of the sea floor images used in the different experiments throughout this paper.

Figure 1-d illustrates the effects of these problems. In this example, every image was matched against each of the previously gathered ones assuming a sweep trajectory and emulating a typical SLAM data association. Matching was performed using SIFT. Two images were assumed to overlap if the number of SIFT feature matchings between them exceeds a certain threshold. In this data set the number of SIFT matchings ranged from 3 to 515 with a mean of 67.4. Just as an example, the lines shown in Figure 1-d denote detected overlaps according to this criteria using a threshold of 150. It can be seen a large amount of errors, mainly false positives, especially when comparing to the actual matches shown in Figure 1-b.

2.3 The Algorithm

By comparing Figure 1-b and Figure 1-d it can be observed that the most part of the errors is due to false matchings. Our goal is to detect and remove these errors as much as possible from an initial guess provided by SIFT. By doing so, our algorithm, which is fed with two images, not only decides whether they overlap or not, but also estimates the displacement and rotation between them. In this paper only 2D motions in the image plane are allowed. Although this may seem an important limitation, the 2D assumption is affordable in the aforementioned sweeping tasks. Moreover, our approach is not conceptually limited to two dimensions and, thus, can be generalized to 6 DOF.

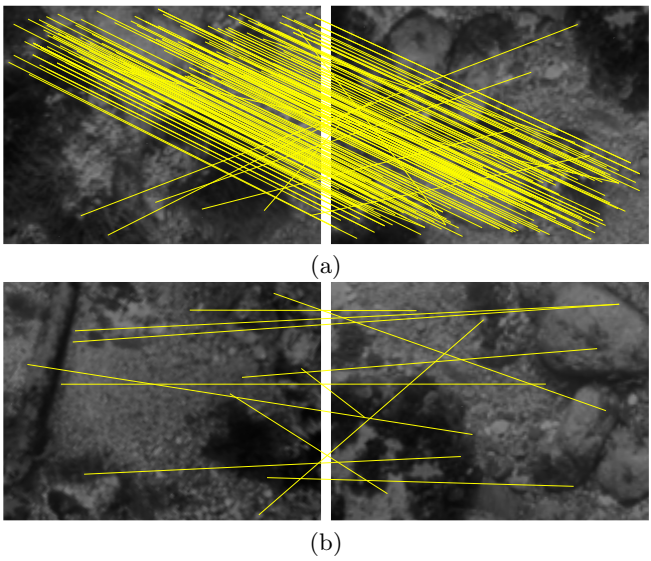


Fig. 2. Candidate matchings. The yellow lines denote SIFT matchings between the left and right grayscale images. (a) Two images that actually overlap (b) Two images with no overlapping.

The first step is to compute SIFT features and matchings for the two images under analysis. Of course, SIFT features previously computed for one of the images are not to be searched again. Figure 2 shows two examples of matchings. Figure 2-a illustrates a situation in which, after a forward motion of the robot, there is still overlap between the two images. It can be observed how the most part of the matchings actually capture the motion. However, some false matchings appear that may corrupt further motion estimates. In the situation depicted in Figure 2-b, although there is no overlap between the two images, several wrong matchings have been detected.

Our proposal is based on the following premise: correct matchings tend to propose a single roto-translation between the two images, whilst incorrect matchings

do not. In other words, correct matchings can be explained by a single model whilst wrong matchings constitute outliers. The goal of *Random Sample Consensus* (RANSAC) [10] is, precisely, to find a model where inliers fit while discarding outliers and that is why RANSAC has been adopted in this study.

Figure 3 shows our proposal to compute the roto-translation between two underwater images using RANSAC. The symbol $\oplus$ denotes the compounding operator, as described in [16]. Roughly speaking, this algorithm randomly selects a subset C of the SIFT matchings M and then computes the roto-translation $X = [x, y, \theta]^T$ that better explains them. Next, each of the non selected matchings is tested to check if it fits X with an acceptable error level. If so, it is selected too. Finally, if the number of selected matchings $|C|$ exceeds a certain threshold, the roto-translation that better explains all the selected matchings is computed. After a fixed number of iterations, the best of the computed roto-translations constitutes the output of the algorithm.

The algorithm relies on the so called $find\_motion$ function, which takes the SIFT feature matchings C and the SIFT feature coordinates in the first (F_{ref}) and second images (F_{cur}) as inputs. This function provides the roto-translation X that better explains the overlap between the images. This is accomplished by searching the roto-translation that minimizes the sum of squared distances between the matchings in C. More specifically, the roto-translation and the associated error are computed as follows:

$$X = \underset{x}{\operatorname{argmin}}\, f(x) \tag{1}$$

$$\varepsilon = f(X) \tag{2}$$

where

$$f(x) = \sum_{(i,j)\in C} ||p_i - x \oplus q_j||^2 \tag{3}$$

As an example, our proposal has been applied to the same data used in Figure 1. Matchings corresponding to images that cannot overlap according to the camera's field of view have been rejected. Figure 4-a shows the result of this. By comparing this image to Figure 1-b it can be observed that all the RANSAC estimates correspond to actual matchings. Also, it is easy to observe that there are very few positions with no matchings. This means that, when our proposal is used in a SLAM framework, most part of the time there will be measurements available.

In order to provide a fair comparison, Figure 4-b shows the raw SIFT matchings depicted in Figure 1-d but now using the abovementioned rejection criteria. As it can be seen, only a very reduced number of matchings remains and the most part of the frames have no matchings. Thus, using this method to perform SLAM will lead to no measurement data during most part of the time.

Figure 5 shows an example of loop detected and closed. The robot first obtained the image shown on the top left and then, when going back because of the sweep trajectory, the second image was obtained. The bottom image shows both

Input :
- F_{ref}: Feature coordinates $\{p_1, p_2, \cdots, p_m\}$ in the first image
- F_{cur}: Feature coordinates $\{q_1, q_2, \cdots, q_n\}$ in the second image
- M: Matchings $M = \{(i,j) | visual\_matching(p_i, q_j)\}$
- $nIter$: Number of iterations to perform
- N: Number of matchings to be randomly selected
- α: Maximum allowable error per matching
- β: Minimum number of selected matchings to consider a model

Output:
- X_{best}: The estimated roto-translation
- ε_{best}: The error of the estimated roto-translation
- $found$: Boolean stating if reliable matching found

```
begin
    k ←0;
    ε_best ← ∞;
    found ← false;
    while k < nIter do
        C ← random selection of N items from M;
        (X, ε) ← find_motion(F_ref, F_cur, C);
        foreach (i, j) ∈ (M − C) do
            if ‖p_i − X ⊕ q_j‖ < α then
                C ← C ∪ {(i, j)};
            end
        end
        if |C| > β then
            (X, ε) ← find_motion(F_ref, F_cur, C);
            if ε < ε_best then
                ε_best ← ε;
                X_best ← X;
                found ← true;
            end
        end
        k ← k + 1;
    end
end
```

Fig. 3. RANSAC algorithm to estimate motion from underwater images.

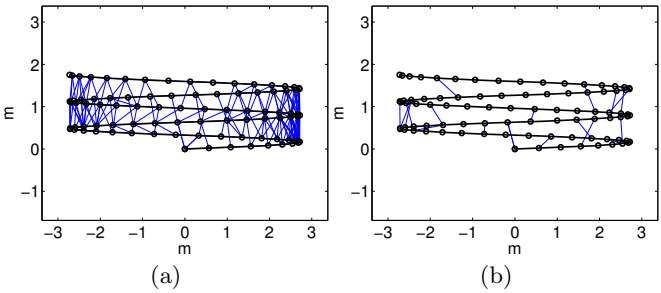


Fig. 4. (a) RANSAC matchings pruned according to a distance criteria. (b) SIFT matchings pruned according to a distance criteria.

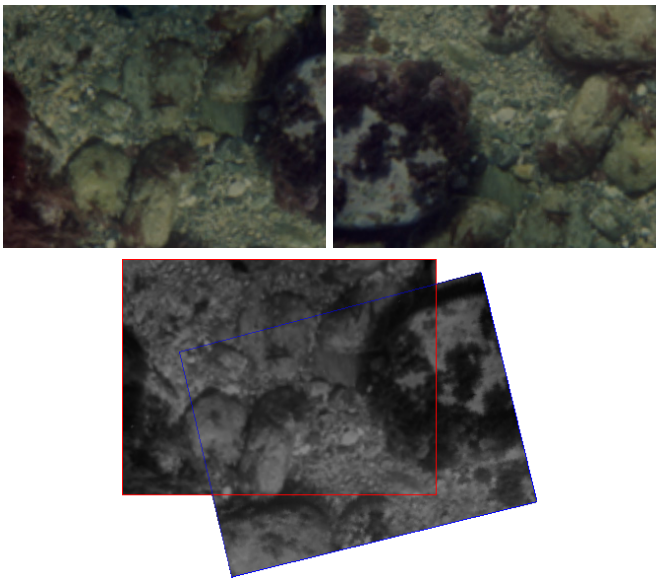


Fig. 5. Example of loop closure. The detected motion between the two images involves a rotation of 166.3^o.

pictures overlayed according to the motion estimate provided by our algorithm. It can be observed how both images match almost perfectly.

3 Visual SLAM

This section describes a SLAM framework where the aforedescribed loop closure approach can be used. The term *keyframe* will refer to an image that is going to be used to perform SLAM. For example, if one image every ten is used, that image is a keyframe.

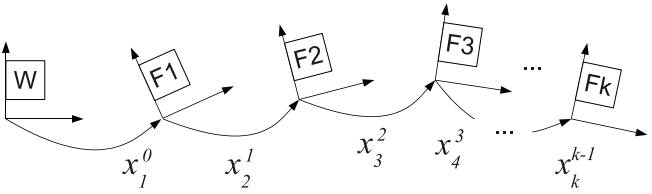


Fig. 6. Relationship between keyframes (F_i) and the state vector components (x_j)

Our proposal is based on the *delayed-state* filtering [9] and the *trajectory-based SLAM* [5]. Accordingly, the EKF state vector contains the relative positions between consecutively gathered keyframes.

$$X_k = [x_1^0, x_2^1, x_3^2, \cdots, x_k^{k-1}]^T \tag{4}$$

where each $x_i^{i-1} (2 \leq i \leq k)$ denotes a roto-translation from keyframe $i-1$ to keyframe i and x_1^0 represents the initial robot pose relative to a world fixed coordinate frame. Let us assume, without loss of generality, that $x_1^0 = [0, 0, 0]^T$. Figure 6 illustrates the relationship between the state vector components and the keyframes, which are labelled as F_i. As the state vector stores relative poses between consecutively gathered keyframes, the pose of the most recent keyframe with respect to the world fixed coordinate frame can be easily computed as $x_k^0 = x_1^0 \oplus x_2^1 \oplus x_3^2 \oplus \cdots \oplus x_k^{k-1}$.

3.1 Prediction and State Augmentation

Under the assumption of static environment, the state vector does not change during the EKF prediction step. Also, as the state vector stores the motion between consecutively gathered keyframes, it has to be augmented every time a new keyframe is available. Taking this into account, the prediction and state augmentation steps are only performed after gathering a new keyframe and are as follows:

$$X_k^- = [X_{k-1}^+, x_k^{k-1}]^T \tag{5}$$

At this point, the motion estimate x_k^{k-1} is provided by dead reckoning. Also, the keyframes have to be stored.

3.2 The Update Step

Every time a new keyframe is gathered, it could be compared with all the previously gathered ones using the data association algorithm proposed in Section 2. However, performing such exhaustive test can be extremely time consuming. Different approaches can be found in the literature regarding this issue [7]. In this regard, our proposal to speed up the process is as follows. First, the position of each keyframe is computed from the state vector. Using this information,

those keyframes that are within a certain radius of the new one are selected. The radius depends on the robot pose uncertainty and on the camera field of view. Finally, only the selected frames are compared with the new one using the proposed data association method. In other words instead of comparing the new keyframe with all the previously gathered, it is only compared with those that are close enough to allow a match.

The data association tells whether the new keyframe matches each of the previously gathered ones and, if so, it provides an estimate of the roto-translation between them. This information constitutes the measurement vector Z_k:

$$Z_k = [z_k^{C1}, z_k^{C2}, \cdots, z_k^{Cn}]^T \tag{6}$$

where $C1, C2, \cdots, Cn$ denote the keyframes thet actually match the current one and z_k^{Ci} represents the motion estimated by our RANSAC based approach.

The observation function h_i is in charge of predicting z_k^{Ci} from the state vector X_k^-. In other words, the observation function estimates the displacement and rotation from F_{Ci} to F_k using the state vector. Because of the state vector format, this can be computed as follows:

$$h_i(X_k^-) = x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_k^{k-1} \tag{7}$$

One of the advantages of the trajectory based approach is that the whole trajectory followed by the robot is explicitly stored and, thus, improved with each observation.

The observation matrix H_i is as follows:

$$H_i = \left.\frac{\partial h_i}{\partial X_k}\right|_{X_k^-} = \left[\left.\frac{\partial h_i}{\partial x_1^0}\right|_{X_k^-} \left.\frac{\partial h_i}{\partial x_2^1}\right|_{X_k^-} ... \left.\frac{\partial h_i}{\partial x_k^{k-1}}\right|_{X_k^-} \right] \tag{8}$$

It is straightforward to see that

$$H_i = \left[\begin{matrix} 000 \\ 000 \\ \underbrace{000}_{\times Ci} \end{matrix} \left.\frac{\partial h_i}{\partial x_{Ci+1}^{Ci}}\right|_{X_k^-} \left.\frac{\partial h_i}{\partial x_{Ci+2}^{Ci+1}}\right|_{X_k^-} ... \left.\frac{\partial h_i}{\partial x_k^{k-1}}\right|_{X_k^-} \right] \tag{9}$$

By applying the chain rule, the non-zero terms of this Equation are as follows:

$$\begin{aligned} \left.\frac{\partial h_i}{\partial x_j^{j-1}}\right|_{X_k^-} = & \left.\frac{\partial h_i}{\partial x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_j^{j-1}}\right|_{X_k^-} \cdot \\ & \cdot \left.\frac{\partial x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_j^{j-1}}{\partial x_j^{j-1}}\right|_{X_k^-} \end{aligned} \tag{10}$$

According to [8] this can be computed as follows:

$$\left.\frac{\partial h_i}{\partial x_j^{j-1}}\right|_{X_k^-} = J_{1\oplus}\{g_j, \ominus g_j \oplus h_i\}|_{X_k^-} \cdot J_{2\oplus}\{g_j \ominus x_j, x_j\}|_{X_k^-} \tag{11}$$

where $J_{1\oplus}$ and $J_{2\oplus}$ are the Jacobians of the composition of transformations [16] and

$$g_j = x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_j^{j-1} \tag{12}$$

At this point, the observation function h and the observation matrix H considering all the matched keyframes are as follows:

$$h(X_k^-) = \begin{bmatrix} h_1 \\ h_2 \\ ... \\ h_n \end{bmatrix} \; H = \begin{bmatrix} H_1 \\ H_2 \\ ... \\ H_n \end{bmatrix} \tag{13}$$

By means of Z_k, h and H, the standard EKF update step can be performed. However, the effects of linearizations may be especially important due to Equations 10 and 11. A possible way to alleviate such problem is the use of an *Iterated EKF* (IEKF). Roughly speaking, the IEKF consists on iterating an EKF and relinearizing the system at each iteration until convergence is achieved. The detailed description of the IEKF is out of the scope of this paper. The reader is directed to [1] for a complete explanation.

4 Experimental Results

4.1 Experimental Setup

The experiments conducted in this section have been performed using the underwater simulator UWSim [14]. This simulator visualizes a virtual underwater scenario where one or more underwater robots can be configured and deployed. Moreover, UWSim easily interfaces with ROS [18], which makes possible to test under simulation exactly the same code that is going to be executed on a real robot. Also, ROS log files gathered from real underwater missions can be played and analyzed using UWSim. The environment where this vehicle was deployed consisted of a photo-realistic mosaic of a real sub-sea environment. Pictures shown in Figures 1-c, 2 and 5 are examples of the imagery used in the experiments.

The simulated mission consisted on a sweeping task, which is very common in underwater robotics. During the mission execution, images coming from a monocular bottom looking camera were gathered as well as the real robot pose. However, only the images have been used to perform SLAM whilst the real robot poses have been gathered with the sole purpose of providing a ground truth to evaluate the SLAM pose estimates.

In order to evaluate the proposed data association, the trajectory-based SLAM described in Section 3 has been executed. Dead reckoning used in the prediction step was computed by means of basic 2D visual odometry: SIFT features were searched and matched between consecutive frames and the roto-translation between them computed using least squares. The resulting visual odometry is used to feed the SLAM prediction step.

Two different configurations have been tested: 5 and 10 frames of separation between keyframes. In our particular test case, a separation of 5 means that, in the straight parts of the trajectory, the overlap between consecutive keyframes is close to 55% of the image. A separation of 10 frames leads to an overlap close to a 10%.

Also, in order to test the robustness of our approach in front of odometric noise, we have added synthetic noise to odometry estimates. Five experiments using different synthetic noise levels have been tested for each configuration. The noise used is additive zero mean Gaussian and the covariance ranges from a $[\Sigma_x, \Sigma_y, \Sigma_\theta] = [0, 0, 0]$ in experiment 1 to $[\Sigma_x, \Sigma_y, \Sigma_\theta] = [4{\cdot}10^{-5}, 4{\cdot}10^{-5}, 5{\cdot}10^{-4}]$ in experiment 5. The random noise was added to each visual odometry estimate. For each configuration (5 or 10 frames of separation between keyframes) and experiment (five noise levels), 100 trials have been performed in order to obtain statistically significative results. The resulting SLAM trajectories, 1500 in total, have been compared to the ground truth in order to quantitatively measure their error. The error of a SLAM trajectory is computed as the mean distance between each of the SLAM estimates and the corresponding ground truth pose.

4.2 Qualitative Results

Figure 7 provides some of the obtained trajectories to illustrate the capabilities of our proposal. The figure shows two trajectories provided by the described SLAM approach which makes extensive use of the proposed data association. The ground truth is also shown, as well as the corrupted visual odometry. The noise level added to visual odometry corresponds to experiment two. Two keyframe separations, 5 and 10, are shown. It can be observed that, in both cases, the SLAM trajectory is very similar to the ground truth one, even in presence of significantly corrupted odometry.

4.3 Quantitative Results

Figure 8 shows the results of the aforedescribed experiments. It can be observed that our SLAM proposal is able to significantly reduce the odometric error. For

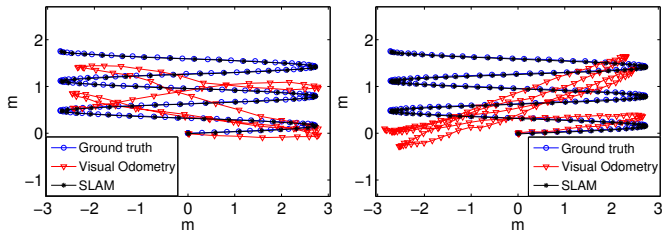


Fig. 7. Results for visual inspection. (a) Experiment 2, Keyframe separation 10. (b) Experiment 2, Keyframe separation 5. Even though the depicted odometric trejctories look different, they have been generated using the same amount of synthetic noise.

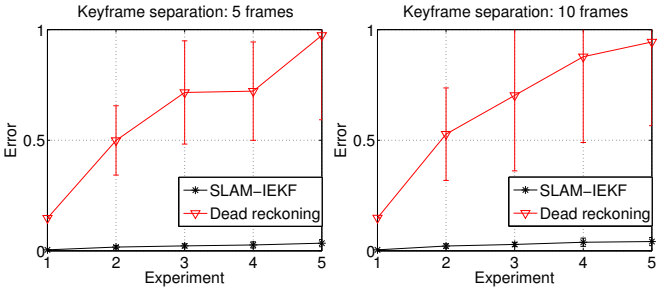


Fig. 8. Errors and 2σ bound

example, in experiment one, where no synthetic noise has been included, the mean odometric error is 0.15m whilst the SLAM error is 0.004m when using one every five frames as keyframe. When using one every ten frames as keyframes, the differences are barely appreciable. This suggests that, for low odometric errors, the quality of the pose estimates is almost independent of the keyframe separation being five or ten.

In experiment 5, where the largest amount of synthetic noise has been used, the mean odometric error raises to 0.99m. When using a keyframe separation of 5 frames, our approach is able to reduce the mean error to 0.035m. If one keyframe is used every 10 frames, then the mean error is reduced to 0.042m. In this case, using one every 5 frames reduces the mean error a 20%.

It can also be observed that, in both cases, the SLAM error and the SLAM error covariance barely increase with the odometric ones. For example, if one every five frames is used the SLAM error increases 0.031m from experiment one to experiment five whilst the odometric error increases 0.837m. If one every ten frames is used, the increase in the SLAM error is 0.038m, which is significantly inferior to the odometric error increase.

5 Conclusion and Future Work

This paper presents an approach to perform data association in a monocular visual SLAM context. The proposed approach has been designed to avoid the detection of false associations by means of RANSAC, and has shown to be well suited to help in localizing a robot in underwater environments. Experimental results embed the data association in a trajectory based SLAM in order to evaluate its benefits when localizing an underwater robot. Qualitative and quantitative results have been shown evaluating the effects of dead reckoning noise and the frequency of the SLAM updates.

Given the good results provided by this method, the next step is to adapt this method to 6DOF SLAM using an stereo pair. By doing so, SLAM will neither be constrained to constant height missions nor to almost flat terrains.

References

1. Bar-Shalom, Y., Rong Li, X., Kirubarajan, T.: Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley and Sons, Inc. (2001)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Computer Vision and Image Understanding 110, 346–359 (2008)
3. Bonin, F., Burguera, A., Oliver, G.: Imaging systems for advanced underwater vehicles. Journal of Maritime Research 8(1), 65–86 (2011)
4. Burguera, A., González, Y., Oliver, G.: The UspIC: Performing scan matching localization using an imaging sonar. Sensors 12, 7855–7885 (2012)
5. Burguera, A., Oliver, G., González, Y.: Scan-based slam with trajectory correction in underwater environment. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010), Taipei, Taiwan (2010)
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6) (June 2007)
7. Elibol, A., Gracias, N., Garcia, R.: Augmented State Extended Kalman Filter combined framework for topology estimation in large-area underwater mapping. Journal of Field Robotics 27, 656–674 (2010)
8. Estrada, C., Neira, J., Tardós, J.D.: Hierarchical SLAM: real-time accurate mapping of large environments. IEEE Transactions on Robotics 21(4), 588–596 (2005)
9. Eustice, R.M., Pizarro, O., Singh, H.: Visually augmented navigation for autonomous underwater vehicles. IEEE Journal of Oceanic Engineering 33(2), 103–122 (2008)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
11. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: Dense 3d reconstruction in real-time. In: IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany (June 2011)
12. Huang, A.S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., Roy, N.: Visual odometry and mapping for autonomous flight using an rgb-d camera. In: Proceedings of the International Symposium on Robotics Research (ISRR), Flagstaff, Arizona, USA (August 2011)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
14. Rauvi, Trident: UWSim: The underwater simulator. Web (accessed April 22, 2013)
15. Ribas, D., Ridao, P., Neira, J.: Underwater SLAM for Structured Environments Using an Imaging Sonar. Springer Tracts in Advanced Robotics, vol. 65. Springer (2010)
16. Smith, R., Cheeseman, P., Self, M.: A stochastic map for uncertain spatial relationships. In: Proceedings of International Symposium on Robotic Research, pp. 467–474. MIT Press (1987)
17. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), `http://www.vlfeat.org`
18. WillowGarage. Ros.org. Web (accessed April 22, 2013)

Increasing the Autonomy Levels for Underwater Intervention Missions by Using Learning and Probabilistic Techniques

Jorge Sales[1], Luís Santos[2], Pedro J. Sanz[1], Jorge Dias[2,3], and J.C. García[1]

[1] Computer Science and Engineering Dept., University of Jaume-I, Castellón, Spain
salesj@uji.es
[2] Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Coimbra, Portugal
[3] Robotics Institute, Khalifa University, Abu Dhabi, UAE

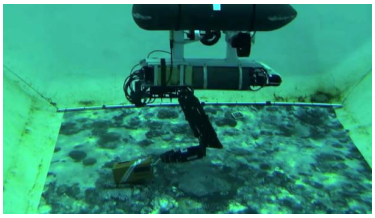
Abstract. This paper represents research in progress in autonomous manipulation for underwater intervention missions within the context of the GRASPER project. This project focuses on developing manipulation skills for an Autonomous Underwater Vehicle (AUV). Current research in underwater robotics intends to increase autonomy for all kinds of robotic intervention operations that require physical interaction. Very few underwater systems have the capacity to carry out intervention without any kind of umbilical cables for tele-operating the actions. This article aims to investigate new approaches to follow with the aforementioned challenges, with the inclusion of learning and probabilistic techniques to increase the autonomy levels of an underwater manipulation system. With this goal, a collaboration research action has been established between the IRS-Lab at UJI (Spain), as experts in the underwater robotic manipulation domain, and the Institute of Systems and Robotics from University of Coimbra (Portugal), experts in learning by interaction within a robotic manipulation context.

Keywords: Underwater Autonomous Intervention, Bayesian Learning, Dynamic Bayesian Network, UWSim underwater realistic simulator.

1 Introduction

This paper discuss the research in progress, under development by UJI-ISR cooperation action, in the context of autonomous underwater intervention missions. Current research in the underwater robotics intends to increase autonomy for all kinds of robotic intervention operations requiring physical interaction. Despite the fact that autonomous robotic intervention on land remains in development and with some valuable achievements, the current state-of-the-art in underwater intervention missions is currently in a very primitive stage where the majority of the systems are tele-operated by an expert user. This paper addresses this challenge through research that stills under development, within the context of a project, funded by the Spanish Ministry, titled GRASPER. GRASPER

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_2,



(a) search and recovery of an object of interest (e.g. a "black-box mockup" from a crashed airplane).

(b) the intervention of an underwater panel in a permanent observatory.

Fig. 1. TRITON Spanish coordinated project proposed scenarios

(under the responsibility of University of Jaume-I, UJI, and addressing the problem of the "Autonomous Manipulation") represents only a sub-project inside a Spanish Coordinated Project, entitled: TRITON[1], "Multisensory Based Underwater Intervention through Cooperative Marine Robots", which includes two other sub-projects: COMAROB ("Cooperative Robotics", under the responsibility of University of Girona, UdG), and VISUAL2 ("Multisensorial Perception", under the responsibility of University of Balearic Islands, UIB). In summary, TRITON is a marine robotics research project focused on the development of intervention technologies really close to the real needs of the final user and, as such, it can facilitate the potential technological transfer of its results. The project proposes two scenarios to test the concept, and to demonstrate the developed capabilities: (1, Figure 1a) the search and recovery of an object of interest (e.g. a "black-box mockup" from a crashed airplane), and (2, Figure 1b) the intervention of an underwater panel in a permanent observatory.

The specific objectives for GRASPER are the following:

(a) To develop the user interface and simulation capabilities needed for TRITON.
(b) To generate all the mechatronics and sensor improvements to succeed in the autonomous manipulation requirements.
(c) To develop new planning and control strategies, making use of range and visual information, finally leading to visual free floating manipulation.

This paper highlights the potential benefits of including a new approach based on the "learning by demonstration" paradigm, in order to increase autonomy in the required grasping and manipulation skills. Because initially the experimental validation will be carried out in virtual reality (i.e. by using the 3D simulator UWSim [1] described below), some contributions are expected in the aforementioned objectives (a) and (c).

[1] Multisensory Based Underwater Intervention through Cooperative Marine Robots (TRITON), available: `http://www.irs.uji.es/triton/`

1.1 Initial Strategy and Roadmap

The activities developed in this research activity follow a methodology where the core techniques can be designed, developed and prototyped with support of a simulator named UWSim [1]. The research results generated by this activity are after, tested on real scenarios with different levels of complexity. The Figure 2 provides a graphical perspective of this strategy.

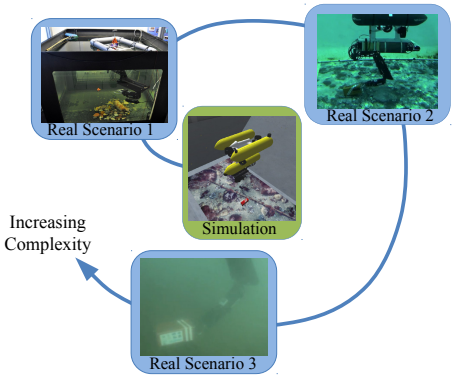


Fig. 2. Development strategy: the core techniques can be designed, developed and prototyped inside the UWSim simulator. Then, the research results generated by this activity are tested on real scenarios with increasing scales of complexity.

This methodology and the modular computational architecture is based on the *Robot Operating System (ROS)* and provides the support for prototyping a solution based on a simulator that can be used to target the real robot, in different real scenarios. The architecture allows us to switch from the simulated environment to a real scenario at any moment and test the prototyped system (manipulation, new algorithms, learning, etc.). The real test scenarios include different physical complexities with increasing degree of realism and hard conditions, when compared with open sea conditions:

- Testbed 1: Water Tank (described below) (UJI, Castellón, Spain)
- Testbed 2: CIRS pool at Girona (UdG, Girona, Spain)
- Testbed 3: Roses Harbour (Roses, Spain)

For each development step or research outcome it is possible to introduce more complex scenarios by simulating them on UWSim system and test the results in different testbeds that convey real hardware in real environments with increasing number of uncontrolled variables (disturbances, visibility, noise, etc.).

1.2 Related Work

In the field of the underwater intervention it is worth mentioning previous projects like SAUVIM [2], intended for deep interventions, which demonstrated

the autonomous recovery of seafloor objects by using a very bulky and expensive system; and TRIDENT[2] [3], that demonstrated the first multipurpose object search and recovery strategy in 2012, able to operate in shallow waters. Nowadays, two ongoing projects are running in the underwater intervention context funded by European Commission: MORPH[3] and PANDORA[4]. It is also noticeable, that the ongoing TRITON project is an extension of the previous Spanish founded project RAUVI[5] [4]. RAUVI was the origin of TRIDENT, demonstrating in 2011 a successful approach for the search and recovery problem but in a more limited manner.

Manipulation Learning: One trend in the current state of the art concerns learning based on geometric properties of manipulation movements to identify the different manipulation stages [5], or continuously learning constraints in a Gaussian Mixture Model approach [6]. Kondo [7], Bernardin [8] and Kruger [9] use symbolic representations encoding hand-object contacts states, temporally represented in Hidden Markov Models (HMM) or Markov Decision Processes. Bekiroglu [10] proposes an approach, using Support Vector Machine (SVM) and HMM to learn and assess robotic grasping stability. Lin [11] applies GMM to learn required fingertip force and pose, to obtain a stable grasp during dexterous manipulation tasks. A different approach is presented in [12], which applies inverse reinforcement learning techniques to infer the underlying task, which is being executed by the demonstrator. Another example comes from Jetchev [13] which adapts inverse optimal control techniques [14] to a single grasping task on a real robotic platform. Beyond the terrestrial applications, there are manipulation platforms working on space to fix satellites [15], where the robot is taught remotely by human operators using an immersive interface with sensorial feedback. Underwater scenarios have only recently been addressed, e.g. in [16] autonomous mobile manipulation in shallow water using a single robotic arm is presented. Recently, Carrera et al. [17], propose a learning solution for autonomous robot valve turning, using Extended Kalman Filtering and Fuzzy Logic to learn manipulation trajectories via kinaesthetic teaching.

1.3 Our Previous Approach

With the aim of increasing the autonomy levels of the underwater manipulation systems, we have recently been working in a multisensory based manipulation

[2] Marine Robots and Dexterous Manipulation for Enabling Autonomous Underwater Multipurpose Intervention Missions (FP7-TRIDENT), available: `http://www.irs.uji.es/trident/`

[3] Marine Robotic System of Self-Organizing, Logically Linked Physical Nodes (FP7-MORPH), available: `http://morph-project.eu/`

[4] Persistent Autonomy through learNing, aDaptation, Observation and Re-plAnning (FP7-PANDORA), available: `http://persistentautonomy.com/`

[5] Reconfigurable Autonomous Underwater Vehicle for Intervention (RAUVI), available: `http://www.irs.uji.es/rauvi/`

approach[6]. This approach allows the grasp of different known-a-priori objects in a water tank, but still requires the user intervention in order to specify the grasp. Some important pieces of this approach are now described:

UWSim: The Underwater Simulator: UWSim is a software tool for visualization and simulation of underwater robotic missions [1]. The software is able to visualize an underwater virtual scenario that can be configured using standard modeling software. Controllable underwater vehicles, surface vessels and robotic manipulators, as well as simulated sensors, can be added to the scene and accessed externally through network interfaces. UWSim do the interface with external control programs through the *Robot Operating System (ROS)* (see additional details in section 5). UWSim has been successfully used for simulating the logics of underwater intervention missions and for reproducing real missions from the captured logs [1]. UWSim is currently used in different ongoing projects funded by European Commission (MORPH and PANDORA) in order to perform HIL (Hardware in the Loop) experiments and to reproduce real missions from the captured logs.

3D Reconstruction of the Scene: The aforementioned approach requires the reconstruction of the geometry of the objects laying on the floor. To achieve this, a scan of the scene is performed using a structured laser beam attached to the forearm of the manipulator. The scan is done by moving the elbow joint of the manipulator at a constant velocity. At the same time, a digital video camera is used to capture the scene with the laser beam projected on the object. A visual processing algorithm runs in parallel: the laser peak detector, which is in charge of segmenting the laser stripe from the rest of the image and computing the 3D points [18]. With these points, a 3D point cloud of the scene is built and represented on the simulator.

2 Problem Statement and Definitions

The challenging problem addressed in this manuscript is within the context of underwater robotics. Assume there is an object O represented by a set of characteristics C_O, located in a 3 Dimensional underwater space $U \in \mathcal{R}^3$. Consider a n DoF manipulator M, operating in U. The challenge is to give M a set of skills S, such that M is able to **reach**, **grab** and **manipulate** O into reaching a user specified goal G. At a first stage, this knowledge S is taught by a human expert. Posteriorly, the manipulator exploits the skill space S, in order to operate autonomously into solving $D(G, O)$. From this scenario (Figure 3), we are able to identify the following main problems:

1. The development of a realistic simulation environment, that a human operator can control and, simultaneously, from which it is able to get realistic feedback while teaching, via "tele-operation", a virtual representation of M.

[6] Underwater semi-autonomous grasping experiments using laser 3D reconstruction can be seen on-line: Experiment 1: `http://youtu.be/VOLNBWfeoLs`, Experiment 2: `http://youtu.be/c62FTTycxsQ`, Experiment 3: `http://youtu.be/42ZklVwNaqc`.

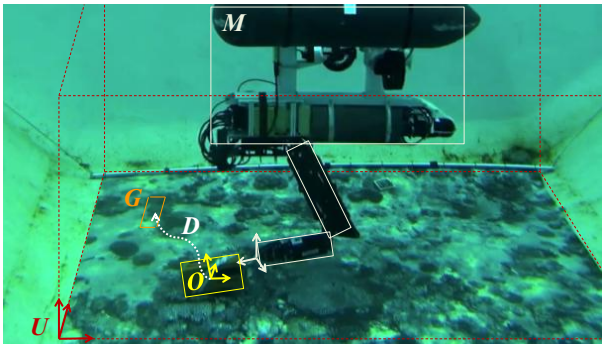


Fig. 3. Underwater intervention scenario. A manipulator M detects and object O in a workspace U. Upon a user specified goal G (or task), it should be capable of, autonomously, estimate a solution D for successfully accomplish its mission.

2. Find a suitable, probabilistic knowledge representation, which accurately models the relation between a set of manipulator sensed information I and a set of skills S, such that M can interpret scene information, identify objects of interest and decide the best course of action D into satisfying G. The solution D, should be updated every time new information is available, so to be able to cope with dynamic and difficult underwater operation conditions.
3. Given a solution for G, project D into a set of motor primitives, allowing the mechanical system M to operate the different steps of its intervention mission.
4. Define a metric to evaluate the success of each intervention, so the system has the capability to decide whether or not the new proposed solution for G should be incrementally added to existent knowledge S.

The proposed solution to this problem can be easily stated: an autonomous manipulator M should be able to decide the best solution D for a given user specified goal G, based on the information I its sensors are able to acquire from the environment U. Such information is, at its most basic forms, identity and pose of objects O, obstructions and its relative End-Effector M pose towards a specified goal G. The solution and integration of these problems are expected to provide an intelligent system, capable of autonomously perform underwater tasks, with minimum human intervention, while being able to constantly adapt to the difficult underwater conditions.

Data Acquisition Via Realistic Underwater Simulator. Implementation of the HRI simulator, ensuring that it will acquire the necessary data for learning the manipulation skills. The data acquired from the user controlling the simulator will be used to develop a filter for assessing what is considered a good trial or not, deciding which trials can be included in the learning.

UWSim do the interface with external control programs through the *Robot Operating System (ROS)*. This architecture provides message-passing and communication between nodes in a transparent manner, thus allowing both local and remote localization of executing nodes (the simulator itself, the learning and database modules, the user interface, etc.). As a consequence of this, we

are able to run the whole system in a single computer but also in a distributed system, allowing thus **remote** learning.

3 Manipulation Skills: Phases, Information and Tasks

In our manipulation scenario we parametrize the execution in 4 different stages:

1. **Initial:** The initial stage is a stage where the robot acquires initial information from the scene, and starts the first iterations to identify scene properties and find initial solution D for the proposed user defined task T.
2. **Approach:** in this stage, the system will refine its assessment of the environment conditions and will gather extra scene information, adjusting its behavior during the reach to grasp trajectory.
3. **Reach/Contact:** once in the neighborhood of the target object, the manipulator needs to decide the best pose and force parameters to enter in contact with the object.
4. **Contact/Manipulation:** at this stage, the manipulator needs to operate the gripper in order to move the object from an initial to a final position, i.e. a second goal G, which is defined by a user specified goal OR automatically assessed from the available sensed information.

We propose a log-spherical intermediate defined key points, at which the manipulation should verify its own attitude towards intermediate and final goals. An example is show in Figure 4. We define **Attitude** as the End-Effector pose, velocity and gripper state, with respect to a specific goal. This attitude should be inferred based on information acquired from the laser scanner and vision system.

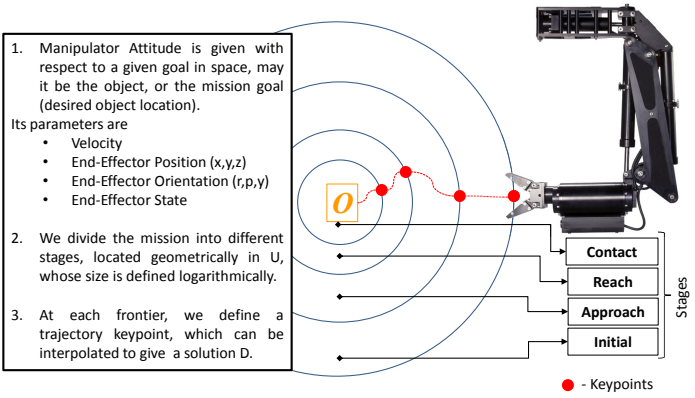


Fig. 4. Proposed log-spherical space for manipulation stage division. The closer M is to its goal, it should be assessing its attitude more frequently.

Solutions are addressed from **different system perspectives.** At each of these stages a supervised learning process is applied, using a tele-operated realistic simulator environment, from which data from the scene, from the manipulator

and from the user controlling the simulator will be recorded in a database for posterior analysis. Our goal is to map a set of sensed information I into a set of skills S observed during tele-operated execution. The data will then be associated by means of probabilistic density functions, into developing an autonomous decision making framework. We propose a system which will make its decisions according to information from different perspectives:

1. Sensing Solution: We start by defining a workspace region, which can be reachable by the manipulator. Sensed information will be projected into an occupancy grid space and processed for developing an interpretation model of the scene, objects and actions. Segmented information is complemented and associated with the manipulator attitude parameters.
2. "Egocentric" Solution: With this approach, we will project all information, as it would be seen by the gripper perspective. We aim at comparing the egocentric approach, which will encompass possibly less and different data, to the proposed sensing solution in terms of manipulation efficiency.

4 Learning Manipulation Skills: Probabilistic Modelling

Learning the adequate end-effector attitude M should exhibit at the different phases, will encompass the definition of approach, contact and manipulation skills based on sensed information and using a learning approach. Unknown sequences will be conducted in order to assess the framework scene interpretation and decision capabilities. We propose a Dynamic Bayesian Model as a support methodology into solving the decision making framework. The development of our model follows the formalism of Bayesian Programming [19], which allows efficient and coherent model development. It defines 4 main stages, which are described in the forthcoming subsections.

Stage 1::Variable Definitions: Bayesian Programming formalism starts by identifying the relevant variables to the problem. Let us recall we are interested in estimating the End-Effector attitude, at the Different phases of the intervention, based on sensed information and considering a user-specified goal. We will address each goal as an independent problem, which can be posteriorly generalizable. This is due to the fact there are two different goals for our manipulator to succeed in a mission: grabbing and object of interest AND move/manipulate it into a specified goal. The mission goals is defined as a tuple $\{Object, Action\}$, where the user manually orders the system to find an object and act on it. From this description we can define the following variables:

- **Attitude Velocity** V**:** this variable defines the velocity which the end effector should exhibit at each given key point.
- **Attitude Position** $X \in \mathcal{R}^3$**:** this random variable vector defines the spatial location (x, y, z), relative to object O coordinate system, in which the end-effector must be positioned, so to perform a correct approach. This location must be unobstructed, otherwise the system should select next candidate location. In the presence of singularities, human tele-operation is required.

- **Attitude Orientation** $\Gamma \in \mathcal{R}^3$**:** this random variable vector defines the orientation of the End-Effector relative to the estimated O pose.
- **Attitude End-Effector** E**:** this random variable has two different states {*open*, *closed*}. While scenario complexity increases, it will become a real valued tupple, representing both opening width and pressure.
- **Object Identity** O**:** this variable space state is $\{Box, \neg Box\}$.
- **Object Location** $O_L \in \mathcal{R}^3$**:** this random variable defines the location of O.
- **Object Pose** $O_P \in \mathcal{R}^3$**:** this random variable vector states the object pose with respect to U.
- **Estimated End-Effector to Object Distance** D**:** random variable with the estimation for the relative distance of the end-effector to the object, and at a subsequent decision stage, from the end-effector to the target Location.
- **Occupancy** V_o**:** this variable represent the state of a given cartesian location in U. For simplicity purposes, we consider this to have two possible states: Occupied or Empty. The information of this state is retrieved from an occupancy map taken from the readings of a Laser Range Finder, but will be held into consideration to be estimated from a stereo vision system.
- **Laser Information** I_L**:** this random variable vector will represent laser range finder information.
- **Image Information** IM**:** this random variable vector contains image features and characteristics of segmented objects of interest.
- **Task** T**:** is a random variable vector defining the end pose of O.
- **Goal** G**:** is a tupple containing information about an object identity O and a specified task T.

We can now define the joint distribution J of our model as:

$$J = P(V, X, \Gamma, E, O_L, O_p, O, V_o, I_L, I_i, T, G) \tag{1}$$

Stage 2::Decomposition: The second stage of the Bayesian Program is to define the decomposition of the joint distribution. This is a simplification process, where the joint distribution is parametrized in a multiplication of simpler conditional distributions. Directed Acyclic Graphs (DAG) can be used to assist this step. The following Figure 5 represents the proposed DAG of our model. We break the DAG into different abstraction levels for easier comprehension. As can be seen the attitude space is where the attitude variables are. As mentioned previously, the attitude of the end-effector should depend on the available information about the object, the environment and also from the user specified goals. The information about the object may be retrieved from the vision information as well as from the laser scanner information. We introduce a simplification step at an initial stage, where the user identifies the object in the occupancy grid, which at that particular stage resolves into a problem of object verification, in a classification process. Assuming all variables are independent and identically distributed, we obtain the following decomposition:

$$\begin{aligned} &P(V, X, \Gamma, E, O, O_L, O_P, G, T, D, V_O, I_L, IM) = \\ &= P(V)P(X)P(\Gamma)P(E)P(O|V)P(T|V)P(G|V)P(O_L|X)P(G|X) \\ &\quad P(T|\Gamma)P(O_P|\Gamma)P(O|E)P(V_O|O)P(IM|O)P(I_L|V_O) \\ &\quad P(V_O|O_P)P(IM|O_P)P(V_O|O_L)P(D|O_L)P(I_L|D)P(IM|D). \end{aligned} \tag{2}$$

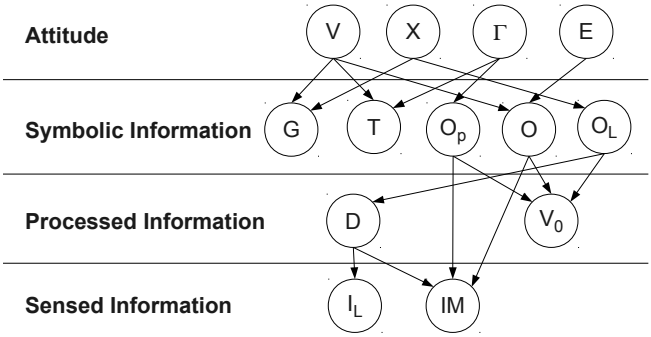


Fig. 5. Directed Acyclic Graph of the proposed Dynamic Bayesian Network. Nodes represent variables and directed arcs represent variable dependencies.

Stage 3::Formulation: This represents the final stage in the specification process. Upon formulating each of the conditional distributions of our decomposition, we have a complete model definition. At this stage, we need to assign parametric or non-parametric probability density functions to each term of the previous step. We will tendentiously assign parametric distribution functions, as these will provide a closed form solution to our problem, and therefore making the inference process solvable analytically, a desirable property. For variables in which the space is $\mathcal{R}^k$, it is a common and efficient solution to assign **Gaussian** distributions. In case a variable lies in a subspace of $\mathcal{R}$ such as $\mathcal{R}^+$, one can decide to assign Poisson distributions. For discrete variables with two states, we propose either **Binomial** or **Bernoulli** distributions, whose parameters might be user specified or learned from sets of experimental data. Other discrete variables whose space state is bigger than 2, may be represented by m **Multivariate Stochastic Matrices**, in which $(m-1)$ dimensions defined parameters and states and the other represents the probability values. Considering these guidelines, we now present the complete specification of our Bayesian Program in Figure 6, where the decomposition in Equation 2 is compacted for simplicity purposes. The functions in the formulation, are likelihood distributions, which can be learnt incrementally or in batch, from experimental data. In our problem, we will consider the *batch* approach. Likelihoods are probability density functions which are defined in terms of a the first random variable, given the knowledge of the outcome. This constitutes a supervised learning process, and requires a user to give the learning process information about the outcome of the variable argument on the right side of the conditional probability.

Stage 4::Bayesian Inference: Once the model is fully specified we can now inquire for information, based on observable evidence. The questions to our model follow the Bayes Rule formalism, i.e., what is the most likely attitude of the end-effector which will allow to fulfil the desired goal, considering the information about the environment. For simplicity purposes, variable A is multivariate and contains $\{V, X, \Gamma, E\}$, O generically defines $\{O, O_P, O_L\}$ and I is composed of

$$
\textit{description}\begin{cases}
\textit{specification}\begin{cases}
\textit{Variables:}\\
V, X, \Gamma, E, O, O_L, O_P, G, T, D, V_O, I_L, IM\\
\\
\textit{Decomposition:}\\
P(V, X, \Gamma, E, O, O_L, O_P, G, T, D, V_O, I_L, IM) =\\
= P(V, X, \Gamma, E)P(O, T, G|V)P(O_L, G|X)\\
P(T, O_P|\Gamma)P(O|E)P(V_0, IM|O, O_P)\\
P(V_O, D|O_L)P(I_L|V_O)P(I_L, IM|D)\\
\\
\textit{Formulation:}\\
P(V)P(X)P(\Gamma)P(G|V)P(O_L|X),\\
P(G|X)P(O_P|\Gamma)P(IM|O)P(IM|O_P),\\
P(D|O_L), P(I_L|D), P(IM|D) = \textit{Gaussian}\\
P(E) = \textit{Bernoulli}\\
P(O|V), P(T|V), P(T|\Gamma), P(O|E), P(V_0|O),\\
P(V_0|O_P), P(V_0|O_L), P(I_L|V_O) = \textit{Stochastic Matrix}
\end{cases}\\
\\
\textit{identification: Bernoulli, Binomial parameters based on } \Theta.
\end{cases}
$$

Fig. 6. Bayesian Program description: it enumerates the relevant variables, the joint distribution decomposition and the formulation of the conditional distributions in parametric forms. The identification stage refers to the parameters of the Bernoulli distribution that will be estimated from the experimental data Θ.

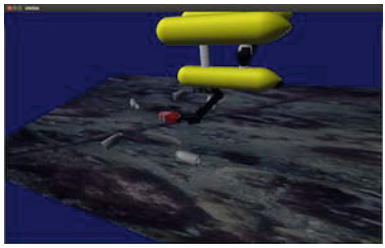
$\{I_L, IM\}$. The term $P(A)$ represents the Prior distribution, which is the estimated attitude of the end-effector before new evidence is taken into account. This distribution is what states the difference between frequentist and subjectivist approaches. It forces regularization for the posterior probability, avoiding overfitting and ensuring that at least one optimal solution exists and it is unique. The terms $P(I|D), P(I|V_O), P(D|O), P(V_O|O), P(O|A), P(G|A)$ and $P(T|A)$ represent a set of likelihood distributions, which reflect how the evidence affects the estimation for A. The normalization term is omitted for mathematical simplification as it does not affect local maximum values for A and its only purpose is ensuring the posterior density integrates to 1. There are various algorithms allowing to perform inference. Perhaps the most popular is the Maximum A Posteriori (MAP). It is a point estimate, which allows finding the maximum value for a given variable based on observable evidence, given by equation 3.

$$
\begin{aligned}
&\hat{A}_{\mathrm{MAP}}(G, I, V_O, T, O, D) =\\
&arg_A max\ P(I|D)P(I|V_O)P(D|O)P(V_O|O)P(T|A)P(O|A)P(G|A)P(A)
\end{aligned}
\tag{3}
$$

5 Experimental Arrangements

In order to validate the proposed learning paradigm, the real scenario 1 shown in Figure 2 (and described below) has been modelled into the UWSim simulator (see Figure 7b). The real scenario includes an underwater vehicle equipped with a robotic arm. In our initial validation arrangement, we will consider that the vehicle is docked to the tank in a fixed position (fixed base manipulation configuration), so the manipulation training will be done by using the degrees

(a) User interacting with the UWSim.



(b) Black-box intervention scenario.

Fig. 7. User interaction with the UWSim simulator using a gamepad controller to train the system. The user gets complete 3D visual information and contact feedback from the observed scene.

of freedom of the underwater arm. The interaction between the human and the virtual robot involves (see Figure 7a): (1) the use of a gamepad, (2) the complete 3D visual information of the scene observed in the computer screen, and also (3) contact feedback, that is, when the robotic arm contacts an object. Right now, this binary contact condition is displayed as a simple indicator on the interface, but later, we will require force feedback implemented on the gamepad.

5.1 Testbed 1: The Water Tank Scenario

The scenario implemented on the simulator corresponds to the real scenario 1 depicted in Figure 2. It consists on a 2m x 2m x 1.5m water tank, whose floor recreates a real seafloor. The underwater vehicle can move with the aid of four thrusters in the horizontal plane, but can also be docked to the water tank to perform fixed base manipulation. Attached to it is a 4 D.O.F. robotic arm (CSIP Light-weight ARM5E [20]) with the possibility to mount different grippers (like the UJIOne, a sensorized gripper containing tactile sensors based on strain gauges in its end-effector[7]). The vehicle is equipped with an underwater camera (Bowtech 550C-AL) that is placed near the base of the arm and is looking downwards. As previously described in section 1.3, the system is able to perform a 3D reconstruction of the scene by using a laser stripe emitter (Tritech SeaStrip) attached on the forearm of the manipulator [18].

5.2 The HRI with the UWSim Simulator

The UWSim simulator is an open project in continuous development. It is divided in different specific branches, where the main one containins the simulator itself. A second branch, named QtUWSim, has the objective of improving the user interaction with the simulator, integrated with the Qt library (windows manager framework). This new environment, allows the user to work with the simulator through buttons, menus and dialogues. Some options will use the the menus to load the scene characteristics, topics, objects and vehicles. The possibility to

[7] See a reactive tactile sensor test on-line: `http://youtu.be/42ZklVwNaqc`.

the user to move the end effector with *interactive markers* [21] also exists. This end effector is defined in a URDF file and loaded into the 3D scene. After the user selects the "grasp specification 3D" option from the menu, the end effector is surrounded by 6 interactive markers (3 translational and 3 rotational). So, the user moves these interactive markers to indicate the end effector position and orientation to reach the target. Another branch integrates the PCL library[8] for 3D reconstruction. A laser is attached to the forearm of the manipulator, reconstructing the target object as a point cloud image. This data may be used also to specify grasping points manually for learning. An algorithm can use this information to find the best grasping points, considering maximum width [18].

5.3 Adapting UWSim for the Proposed Learning Paradigm

As mentioned above, the UWSim simulator do the interface with external control programs through the *Robot Operating System (ROS)*. This means that all the inputs and outputs to and from the simulator are done through ROS topics. A specific ROS node outside the Simulator architecture named `arm_joy_control` allows the user to interact with it using a game controller, by using a specific launch file. Specifically, it allows the user to move the robot arm in joint space ($q1$=Slew, $q2$=Shoulder, $q3$=Elbow, $q4$=JawRotate) and also controlling the Jaw opening ($q5$=JawOpening).

A second node named `arm_joy_cartesian_control`, currently under development, will allow the user to control the robot arm in Cartesian space, using numerical inverse kinematics solvers from the *Orocos KDL library*, to calculate the successive trajectory key-points. By using these values, the algorithm calculates the direct kinematics to verify the user desired point. Due to the few arm degrees of freedom (4 D.O.F.), the algorithm does not check the arm orientation.

Our validation relies on an accurate representation of the real world in our realistic simulator. This property allows us to have a 1 on 1 correspondence between variables on both environments.

5.4 Preliminary Results

The first step in the learning process conveys data acquisition from several trials, upon user demonstration by using the simulator. The acquired variables are the joint values $q_i = \{q_0 \dots q_4\}$; the relative distance to the target object d; binary data indicating when the target object has been picked; and collision information (any collision between the arm and/or the end-effector and the target object). The second step is the automatic execution phase P determination, where $P \in \{approach, reach, manipulation\}$ is a random variable which identifies the current phase of a given action. We divide an action into three different phases: the approach phase in which the manipulator identifies the object of interest and starts moving in its direction; in the reach-to-contact the end-effector is required to take the grasp configuration needed to perform the action; the

[8] Point Cloud Library, available: `http://pointclouds.org/`.

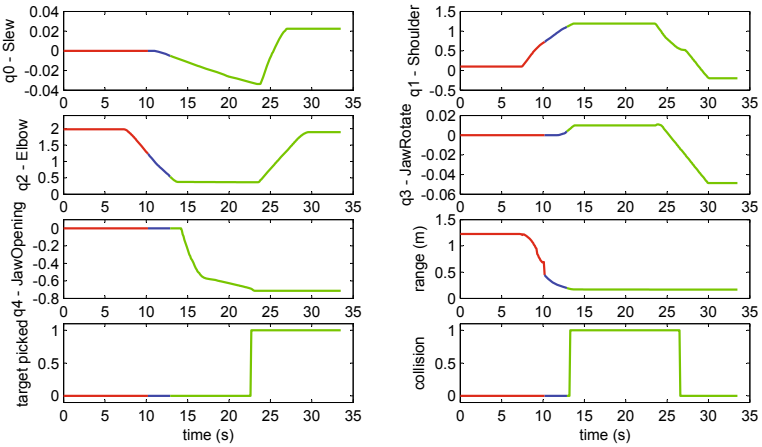


Fig. 8. Preliminary results showing captured data from simulator (q_i, *range*, *target picked*, *collision*) over time and automatic determination of the execution P phase, where $p_1 = \{approach\}$ is represeted in *red*, $p_2 = \{reach\}$ is represeted in *blue* and $p_3 = \{manipulation\}$ is represeted in *green*

manipulation stage happens when the end-effector contacts the object to perform a specific task. The captured data from simulator (q_i, *range*, *target picked*, *collision*) and the distance-based phase P selection can be seen in Figure 8.

6 Conclusions and Future Work

After very successful research achievements through previous projects, like TRIDENT or RAUVI, following a semi-autonomous strategy, we are trying to increase now the autonomy levels, under GRASPER project context, by means of learning. This new approach is supported by the ongoing cooperation between UJI (Spain) and ISR (Portugal). In particular, upon successfully implementing the learning/classification framework, we will achieve two main goals. The first, is to have a knowledge representation, in which probability density functions are used to efficiently map sensed information into manipulator action parameters, dependent on specific underwater mission goals. The second objective is having the manipulator accessing the knowledge information in a decision making/classification framework and, autonomously, decide the correct course of action into solving a given user-defined task. The estimated optimal solution is based on the information it is capable of acquiring from itself and the environment. This process is intended to increase AUV autonomy capabilities, while simultaneously reducing, or even assisting human intervention. The model will include additional information, the state of the AUV in the representation of the model, to maintain the apropriate arm configuration if the vehicle is not in a static position. To maintain the vehicle position on top of the object to manipulate, we will use our already developed visual tracking method [22].

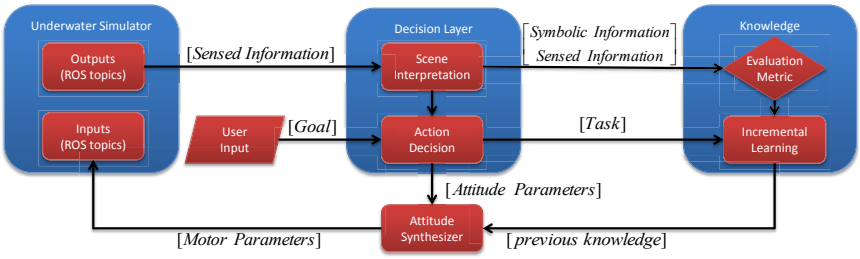


Fig. 9. Execution and Incremental Learning Block Diagram

The final goal of the project is to have an AUV (Autonomous Underwater Vehicle) performing an underwater task autonomously, with minimum human intervention. To that purpose, autonomous execution will be used to incrementally update existing knowledge with new trials (Figure 9). While executing, the acquired information will be interpreted, and forwarded to a module which will decide whether the trial will be added to memory for future interventions, or not. This decision will be based on a comparison between expected and real mission outcomes. This is a continuous process where the generalized action information is synthesized into low-level control primitives acting on the manipulator itself.

Acknowledgment. This research was partly supported by Spanish Ministry of Research and Innovation DPI2011-27977-C03 (TRITON Project), by Foundation Caixa Castelló-Bancaixa PI.1B2011-17, by Universitat Jaume I under grant E-2013-33, by the FCT - Portuguese Foundation for Science and Technology, Grant # 65935/2009, by the Institute of Systems and Robotics from University of Coimbra, Portugal and Khalifa University, Abu Dhabi, UAE.

References

1. Prats, M., Pérez, J., Fernández, J., Sanz, P.: An open source tool for simulation and supervision of underwater intervention missions. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2577–2582 (2012)
2. Marani, G., Choi, S.K., Yuh, J.: Underwater autonomous manipulation for intervention missions AUVs. Ocean Engineering 36, 15–23 (2009)
3. Sanz, P.J., Ridao, P., Oliver, G., Casalino, G., Insaurralde, C., Silvestre, C., Melchiorri, C., Turetta, A.: TRIDENT: Recent improvements about autonomous underwater intervention missions. In: 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2012), Porto, Portugal (April 2012)
4. Sanz, P., Prats, M., Ridao, P., Ribas, D., Oliver, G., Ortiz, A.: Recent progress in the RAUVI project: A Reconfigurable Autonomous Underwater Vehicle for Intervention. In: ELMAR, 2010 Proceedings, pp. 471–474 (2010)
5. Faria, D., Martins, R., Lobo, J., Dias, J.: Extracting data from human manipulation of objects towards improving autonomous robotic grasping. Robotics and Autonomous Syst., Elsevier: Sp. Issue on Autonomous Grasping 60, 396–410 (2012)

6. 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, Special Issue on Robot Learning by Observation, Demonstration and Imitation 37, 286–298 (2007)
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(1), 66–81 (2008)
8. Bernardin, K., Ogawara, K., Ikeuchi, K., Dillmann, R.: A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. Trans. Rob. 21(1), 47–57 (2005)
9. Krger, V., Herzog, D., Baby, S., Ude, A., Kragic, D.: Learning actions from observations. IEEE Robot. Automat. Mag. 17(2), 30–43 (2010)
10. Bekiroglu, Y., Laaksonen, J., Jrgensen, J.A., Kyrki, V., Kragic, D.: Assessing grasp stability based on learning and haptic data. IEEE Transactions on Robotics 27(3), 616–629 (2011)
11. Lin, Y., Ren, S., Clevenger, M., 0004, Y.S.: Learning grasping force from demonstration. In: IEEE Int. Conf. on Robotics and Automation ICRA, pp. 1526–1531 (2012)
12. Lopes, M., Melo, F., Montesano, L.: Active learning for reward estimation in inverse reinforcement learning. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part II. LNCS, vol. 5782, pp. 31–46. Springer, Heidelberg (2009)
13. Jetchev, N., Toussaint, M.: Task space retrieval using inverse feedback control. In: ICML, pp. 449–456 (2011)
14. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Proceedings of the 23rd International Conference on Machine Learning (ICML 2006), pp. 729–736. ACM, New York (2006)
15. Guizzo, E.: Humanoid robot justin learning to fix satellites. IEEE Spectrum, `http://spectrum.ieee.org/automaton/robotics/industrial-robots/humanoid-robot-justin-learning-to-fix-satellites`
16. Prats, M., García, J., Fernández, J., Marín, R., Sanz, P.: Advances in the specification and execution of underwater autonomous manipulation tasks. In: 2011 IEEE OCEANS, Spain, pp. 1–5 (2011)
17. Carrera, A., Ahmadzadeh, S.R., Ajoudani, A., Kormushev, P., Carreras, M., Caldwell, D.G.: Towards Autonomous Robotic Valve Turning. Journal of Cybernetics and Information Technologies (CIT) 12(3), 17–26 (2012)
18. Prats, M., Fernández, J., Sanz, P.: Combining template tracking and laser peak detection for 3D reconstruction and grasping in underwater environments. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 106–112 (2012)
19. Bessiere, P., Ahuactzin, J.M., Mekhnacha, K., Mazer, E.: Bayesian Programming. Chapman and Hall/CRC (2013)
20. Fernández, J.J., Prats, M., García, J.C., Marín, R., Peñalver, A.: Manipulation in the seabed: A new underwater robot arm for shallow water intervention. In: Embedded Systems, Computational Intelligence and Telematics in Control (2012)
21. García, J.C., Peñalver, A., Prats, M., Sanz, P.J.: Recent progress in HRI for underwater robotic intervention. In: 2013 IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany (May 2013)
22. Prats, M., García, J.C., Fernández, J.J., Marín, R., Sanz, P.J.: Towards Specification, Planning and Sensor-Based Control of Autonomous Underwater Intervention. In: Proceedings of the 18th IFAC World Congress, Milano, Italy, pp. 10361–10366 (2011)

Squirtle: An ASV for Inland Water Environmental Monitoring

Jorge Fraga, João Sousa, Gonçalo Cabrita, Paulo Coimbra, and Lino Marques

Institute of Systems and Robotics,
University of Coimbra,
3030-290 Coimbra, Portugal
{goncabrita,acoimbra,lino}@isr.uc.pt

Abstract. This paper presents the Squirtle, an Autonomous Surface Vehicle (ASV) for autonomous navigation and capable of manoeuvring in tight river environments. Its main features include precise localization with RTK GPS and inertial sensors, high manoeuvrability and autonomy, depth sensing and obstacle detection. Its purpose is the exploration and mapping of the bottom of rivers and of their shores. The article includes a brief description of the systems developed to fulfil this task. Software is based on the Robot Operating System (ROS). Results from preliminary field trials are presented and discussed.

1 Introduction

ASVs are an important tool for studying bodies of water such as rivers, oceans or lakes. An ASV is able to undertake long term missions of environmental sampling or bathymetric surveying which can be laborious and tedious tasks for human workers. The absence of the human element means that they can have a reduced size and navigate on shallower and more dangerous waters. As such, they can also be employed as a cost effective coast monitoring tool. Further common uses for ASVs include providing support for an autonomous underwater vehicle by improving their communication and localization capabilities, as well as retrieving data from sensors deployed in buoys without long range communication equipment. To be able to complete these tasks, ASVs must be able to react to dynamic environments and modify their course accordingly. Obstacles such as other vessels, swimmers, buoys, tree branches and other debris are a challenge that must be dealt with in order to achieve full autonomy. In addition, this type of obstacles will often be mobile due to currents or even by their own propulsion.

This work focuses on the Squirtle ASV, developed at Institute of Systems and Robotics (ISR) from the University of Coimbra, and its capabilities to explore and map a river. This task requires that the ASV is capable of bathymetry surveying to map the bottom of the river and of detecting shores and small river islands. These include situations where the shore is defined by vegetation or pontoons that do not reach the bottom of the river, and as such will remain undetected by a sonar. Finally, the ASV should also be able to avoid dynamic obstacles.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_3,

1.1 Related Work

With the rise in popularity of ASVs as a tool for environmental monitoring, there are a few examples of successfully developed platforms. A review of robotics for environmental monitoring can be found in [1] and a more specific review of ASVs developments until 2006 can be found in [2]. In this review, the development challenges of an ASV are divided in three main categories: (i) design and construction; (ii) control and navigation; and (iii) path planning. Some of the most relevant examples detailed are the SCOUT [3], the DELFIM [4] and the Charlie [5]. The literature shows that the focus for civilian ASV design is on producing stable, electric powered vessels, designed for navigating at low forward speeds. This comes in contrast to military vessels, where the focus is on fast vehicles and teleoperation is a higher priority rather than full autonomy. An example on this category is presented in [6].

Some of the most recent works have been developed in Australia, focusing on environmental missions like the detection of greenhouse gas emissions [7]. Other modern applications of ASV technology include monitoring invasive fish species in lakes, as reported in [8].

On the more specific problem of shore and obstacle detection, some approaches can be found in the literature. In [9] overhead imagery is used to detect edges and obstacles on water bodies. Since up to date aerial footage is not usually available, different methods have been developed that make use of cameras and computer vision. A camera and knowledge of the horizon position are used to determine the coast position in [10]. Although this is a powerful approach for vessels operating in the ocean or large estuaries, it is less efficient in settings where the ASV is going up or downstream on smaller rivers where the shore is not an horizontal line. This particular setting is considered in [11], where a State Vector Machine (SVM) is trained to recognize the river with an aerial vehicle. The ideas presented here, due to their applicability to our particular case, were used as a starting point for the development of the Squirtle vision system, further discusses later in this article.

2 The Squirtle ASV

2.1 Hardware

The Squirtle ASV, as shown in figure 1, is a catamaran built from two kayak hulls adjoined by a structure of aluminium profiles. The catamaran design was chosen for its load capacity and stability, the latter of which is important for bathymetric surveying. Two boxes are fastened between the hulls, one for power electronics and batteries and the other for instrumentation and processing material. Figure 2(a) shows the hardware systems of the ASV.

Propulsion is created by two Yamaha M12 brushed DC motors with $13.6\,kgf$ of static thrust each. It is possible to rotate the motors through a linear actuator connected to the shafts of the motors in an Ackermann steering scheme. For control purposes, an Arduino micro-controller is connected to the H-bridges driving

Fig. 1. The Squirtle ASV

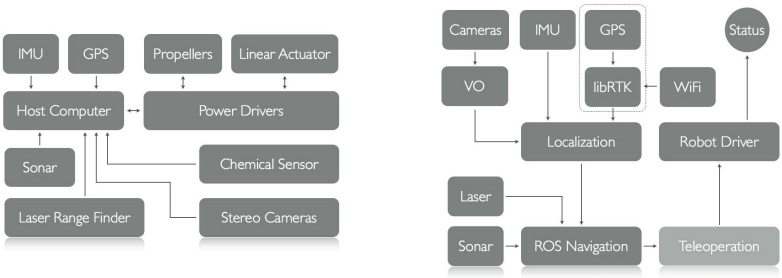
the motors. The Squirtle has two methods of steering. Differential steering allows low speed manoeuvring for docking or dynamic positioning purposes. The second method, rotating the motors, is preferable for higher speed manoeuvres, since differential turning limits the amount of forward thrust available. With these three degrees of freedom the Squirtle is able to perform a larger amount of manoeuvres than most rudder based ASVs, such as rotating in place.

Power is supplied through a pack of $12\,V$ gel electrolyte lead acid batteries. Each battery has $18\,Ah$ of capacity, and up to four batteries can be installed. For extended autonomy, a $240\,W$ solar panel charges the batteries through a MPPT unit.

The localization and orientation of the ASV is determined by fusing the data obtained from an Inertial Measurement Unit (IMU) and a GPS. A second GPS receiver located on the base station provides the information necessary to augment the position accuracy with a Real Time Kinematic (RTK) algorithm. With sufficient satellite coverage, which is typical in most water based environments, this setup results in a position accuracy of about $1cm$. The IMU's accelerometers, gyroscopes and magnetometers complement the GPS data providing orientation. While visual odometry data can be redundant in ideal conditions, GPS signal degrades significantly under bridges or near shores with tall trees. In such cases the extra information is useful.

Communication with the base station is maintained through two redundant channels. A high speed Wi-Fi connection is used as the main channel, while a long range Xbee connection works as a safety precaution for critical data transfer and ASV retrieval.

To sense its surroundings, the Squirtle is fitted with a Sick Laser Range Finder, an Imagenex 852 Ultra-Miniature sonar and a stereo pair of cameras (IDS uEye GigE). Another camera collecting thermal images was added for improved shore recognition. The sonar is set at a 45° angle, pointing forward, so that shallow



(a) The Squirtle ASV hardware diagram. (b) The Squirtle ASV software diagram.

Fig. 2. Diagram representing the software and hardware architecture on the Squirtle ASV

areas can be sensed and avoided. All processing in the Squirtle is performed by an ODROID-X2, equipped with a quad-core ARM processor running at $1.7GHz$.

2.2 Software

The Squirtle software runs on top of Robot Operating System (ROS) framework. ROS provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more [12]. All the frame referentials are managed by the ROS ecosystem. Figure 2(b) shows the Squirtle software diagram.

The stereo pair is used to estimate the visual odometry of the ASV. Localization data (visual odometry, GPS and IMU) is sent to the localization node, responsible for calculating the position, velocity and orientation of the Squirtle. This is achieved using an Extended Kalman Filter (EKF).

The localization information is used by the ROS navigation stack. Commands are issued to the motors and linear actuator by the Squirtle hardware driver node. The Sick LRF and the Imagenex sonar data is used by the navigation stack for obstacle avoidance using the Dynamic Window Approach (DWA) algorithm. The ROS navigation stack was originally designed having in mind omnidirectional and differential drive robots, such as the ones commonly found inside a lab. However due to the manoeuvrability of the Squirtle the navigation stack was configured to control a differential drive robot and fine tuned for the Squirtle.

The overseer node is responsible for all fault-tolerating behaviours, such as recalling the ASV to the base in case communications are interrupted for a predetermined period of time. It is also in charge of re-routing critical data over the XBee connection whenever the Wi-Fi link is broken.

For shore detection, a Bayesian network is trained to recognize the river. The main addition in this field is the use of thermal information as a feature for the network.

2.3 User Interface

User interface was built using the ROS rviz package. A 3D model of the Squirtle and satellite images from Google allow users to configure missions and follow the ASV as it fulfils them, displaying water depth and obstacles in real-time. Figure 3 shows a section of the interface representing the bathymetric data retrieved on a test mission.

Teleoperation is an important part of most robots, and due to the size and weight of the Squirtle ASV, teleoperation becomes paramount. The Squirtle is able to run teleoperation in parallel with the autonomous navigation, giving a user the ability to interrupt the autonomous motion and engage in direct control of the robot at any time. Furthermore the ROS rviz package can also be used to access the cameras in the ASV should the robot become out of sight.

3 Example Mission: Mondego River Monitoring

Individual systems were successfully tested in the Mondego river. The sample mission consisted of autonomously completing two laps of a rectangular path, followed by moving through teleoperation to a target at 300 m from the base to check communication signal strength. The tests were performed with noticeable, south-east oriented, wind and current perturbation. Figure 3 shows the bathymetry obtained. The green rectangle represents the rectangular path reference for the Squirtle. Figure 4 is an overhead satellite image of the same section of the river. The red lines in figures 3 and 4 are equivalent, and shown as reference since the image in the rviz environment has a different aspect ratio. Bathymetry results are consistent with the sand banks observable in figure 4. It can be seen that the discontinuity in depth observed on the right side corresponds to the end of the sand bank.

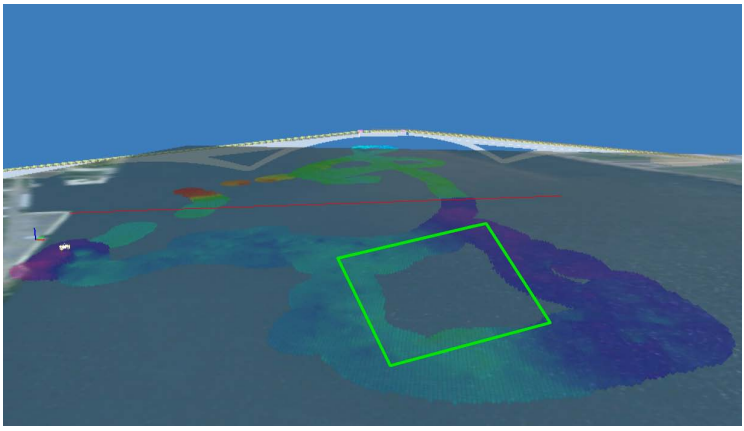


Fig. 3. Real Time Interpolation for Bathymetric Data. Values range from red (3.0 m) to purple (0.2 m).

Fig. 4. Satellite Image View of the Sand Bank

The performance of the path controller itself showed some vulnerability to the perturbations described, as can be seen by the larger patch of covered water near the south-eastern corner of the rectangle. Table 1 shows data relevant to the two laps performed by the Squirtle in autonomous mode.

Table 1. Autonomous Control Test Data

-	First Lap	Second Lap
Time to Completion	5min 56s	7min 24s
Average Absolute Error	4.82 m	4.04 m
Maximum Error	12.50 m	10.28 m
Total Distance Travelled	122.81 m	142.40 m

The signal strength test was successfull. At 300 m the signal was still strong, allowing data transfer at over 1 $Mbps$ through Wi-Fi.

4 Conclusion

Future work will include the update of the current yet sub-optimal navigation controller by performing a series of straight line and zig zag manoeuvres with the goal of identifying the constants of the system model. Once these constants have been identified they can be used along with wind and water current measurements to improve the precision and controllability of the Squirtle ASV. Still, the tests already performed indicate the Squirtle's usefulness and ability to perform as expected, confirming its potential as a scientific tool.

References

1. Dunbabin, M., Marques, L.: Robots for environmental monitoring: Significant advancements and applications. IEEE Robotics and Automation Magazine 19(1), 24–39 (2012)
2. Caccia, M.: Autonomous surface craft: prototypes and basic research issues. In: 14th Mediterranean Conference on Control and Automation, MED 2006, pp. 1–6. IEEE (2006)
3. Curcio, J., Leonard, J., Patrikalakis, A.: SCOUT - a low cost autonomous surface platform for research in cooperative autonomy. In: Proceedings of MTS/IEEE, OCEANS, pp. 725–729. IEEE (2005)
4. Alves, J., Oliveira, P., Oliveira, R., Pascoal, A., Rufino, M., Sebastiao, L., Silvestre, C.: Vehicle and mission control of the delfim autonomous surface craft. In: 14th Mediterranean Conference on Control and Automation, MED 2006, pp. 1–6. IEEE (2006)
5. Caccia, M., Bruzzone, G., Bono, R.: Modelling and identification of the charlie2005 asc. In: 14th Mediterranean Conference on Control and Automation, MED 2006, pp. 1–6. IEEE (2006)
6. Ebken, J., Bruch, M., Lum, J.: Applying unmanned ground vehicle technologies to unmanned surface vehicles, vol. 5804. SPIE (2005)
7. Dunbabin, M., Grinham, A.: Experimental evaluation of an autonomous surface vehicle for water quality and greenhouse gas emission monitoring. In: 2010 IEEE International Conference on Robotics and Automation, ICRA, pp. 5268–5274. IEEE (2010)
8. Tokekar, P., Branson, E., Vander Hook, J., Isler, V.: Coverage and active localization for monitoring invasive fish with an autonomous boat. IEEE Robotics and Automation Magazine (2012)
9. Heidarsson, H.K., Sukhatme, G.S.: Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 3160–3165. IEEE (2011)
10. Wang, H., Wei, Z., Wang, S., Ow, C.S., Ho, K.T., Feng, B.: A vision-based obstacle detection system for unmanned surface vehicle. In: 2011 IEEE Conference on Robotics, Automation and Mechatronics, RAM, pp. 364–369. IEEE (2011)
11. Scherer, S., Rehder, J., Achar, S., Cover, H., Chambers, A., Nuske, S., Singh, S.: River mapping from a flying robot: state estimation, river detection, and obstacle mapping. Autonomous Robots 33(1-2), 189–214 (2012)
12. Quigley, M., Conley, K., Gerkey, B., 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)

Realtime AUV Terrain Based Navigation with Octomap in a Natural Environment

Guillem Vallicrosa, Albert Palomer, David Ribas, and Pere Ridao

Computer Vision and Robotics Institute (VICOROB),
Universitat de Girona, 17071 Girona, Spain
{gvallicrosa,apalomer,dribas,pere}@eia.udg.edu
http://cirs.udg.edu

Abstract. This paper addresses the problems of Terrain Based Navigation (TBN) and Occupancy Grid Mapping for an Autonomous Underwater Vehicle (AUV). The two problems are solved using the same tools to make feasible in future works to implement a Simultaneous Localization and Mapping (SLAM). Realtime Occupancy Grid Mapping on the real vehicle Girona500 AUV is achieved by means of the Octomap library. The resulting map is later used for TBN with the parallelized execution of a Particle Filter making also use of the Octomap library to compare multibeam sonar ranges against the known map. The Occupancy Grid Mapping and the Particle filter are implemented as individual nodes in the vehicle's software architecture in ROS. Tests were carried out in a dataset of a natural environment near the coast. Several parameters involving the Particle Filter (number of particles, number of beams, uncertainty of measurements) are studied. Finally, the results are compared with the dead reckoning obtained by the AUV and the USBL positions obtained from a surface boat.

Keywords: Occupancy Grid Mapping, Terrain Based Navigation, Octomap, AUV, Particle Filter, Realtime.

1 Introduction

Terrain Based Navigation (TBN), is the name used by the marine robotics community to refer to the more general problem of mobile robot localization with a known map. In particular, TBN solves for the robot pose given an a priori known map, fusing information from dead reckoning navigation with map-referenced observations.

TBN has been mainly applied to aerial and underwater vehicles. During the last years the accuracy and extension of the maps has been increased considerably, and TBN has been adopted as a method to complement Inertial Navigation Systems (INS), as an alternative when GPS is not available. During the last two decades, the scientific community working with mobile robots, has pushed forward the boundaries of the knowledge facing an even more challenging problem: the Simultaneous Localization and Mapping[5]. Recently, these techniques have

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_4,

started to be slowly adapted to underwater environments[2,13,11,14,18]. Both SLAM and TBN have a great potential to improve the autonomy of the underwater vehicles, allowing AUVs to freely move abroad the areas of coverage of the acoustic transponder networks.

Navigation sensors provide the robot velocity and attitude measurements, to be used as the input for the dead reckoning equations needed to compute the robot pose. Due to the noisy measurements, the position estimate will grow without bounds. In domains like underwater, GPS is not available to bound the drift. Absolute positioning fixes can be provided underwater by acoustic positioning like the Long Baseline System (LBL), the Short Baseline (SBL), the Ultra Short Baseline (USBL) or the GPS equipped intelligent Buoys (GIB). However, these systems require time for deployment and constrain the vehicle to a certain area of coverage. TBN has the potential to become an alternative to satellite navigation and acoustic transporter networks. TBN takes advantage of existing digital terrain maps from a target area, where the vehicle shall navigate. Conventional dead reckoning navigation methods provide a prior estimate of the robot pose within the map. Then, using exteropceptive sensors, terrain observations are obtained and correlated to the a priori known map in order to compute the robot pose.

1.1 Problem Definition

Let:

- $\{\mathbb{E}\}$ be an inertial earth-fixed frame.
- $\{\mathbb{B}\}$ be the vehicle-fixed frame.
- $x_t = [x\, y\, z\, \phi\, \theta\, \psi]$ be the vehicle pose at time t referenced to $\{\mathbb{E}\}$.
- $\upsilon_t = [u\, v\, w\, p\, q\, r]$ be the vehicle velocity at time t referenced to $\{\mathbb{B}\}$.
- $\mathbb{M}(x_t)$ be a digital elevation map of the environment, assumed to be known a priori.
- z_t be the vector of observations coming from exteroceptive sensors (altimeter, pressure or multibeam sensor).

TBN consist in estimating the vehicle velocity υ_t, referenced to the vehicle $\mathbb{B}$ frame, and/or the vehicle pose η_t, referenced to the inertial $\mathbb{E}$ frame, by matching observations z_t of the measured depth with the ranges from the terrain elevation map stored in $\mathbb{M}$. For AUVs, the linear velocity is provided by a Doppler Velocity Log (DVL) sensor and the attitude by a Attitude Heading Reference System (AHRS).

For an AUV the measurement model is given by:

$$r_{t,i} = \mathbb{M}_i(\eta_t) - d_t + w_t, \qquad \forall 0 \leq i \leq N$$

where w_t is the measurement noise, d_t the vehicle depth, N the number of beams, and $r_{t,i}$ the projection of the range measurements along the vertical axis.

1.2 Bayesian Estimation Techniques for TBN

Let us assume x_t to be a Markovian state and:

- $p(x_{t-1})$ be the probability density function (pdf) describing the probability of the robot of being at a certain pose at time $t-1$,
- $p(x_t|x_{t-1}, u_t)$ be the state transition probability, also known as motion model, which allows to predict the robot pose after certain input u_t,
- $p(z_t|x_t)$ be the measurement probability which given a map $\mathbb{M}()$ provides the probability of observing z_t when being at state x_t,

then, the TBN problem consists on solving for $p(x_t)$, given $p(x_{t-1})$, $p(x_t|x_{t-1}, u_t)$ and $p(z_t|x_t)$. In the context of Bayesian estimation this can be done through the Bayes filter (BF):

$$p(\overline{x}_t) = \int p(x_t|x_{t-1}, u_t)p(x_{t-1})dx_t \tag{1}$$

$$p(x_t) = \eta p(z_t|x_t)p(\overline{x_t}) \tag{2}$$

In the most general case, the BF cannot be implemented because it relies on the close form solution of the integral shown in (1). Under some conditions, the BF can be approximated using different kinds of filters like the Kalman Filter (KF) [3], the Extended Kalman Filter (EKF) [16], the Point Mass Filter (PMF), the Particle Filter (PF) or the Rao-blackwellised Particle Filter (RBPF) [17].

In this paper, the TBN problem is going to be solved with a Particle Filter (PF). Also called sequential Monte-Carlo (SMC) method, it can deal with nonlinear motion and/or measurement models without relying on linearization techniques. PFs use a point mass representation of the density function to approximate the robot pose posterior [1,4]. A set of random samples or "particles", are used to represent the pdf, being able to represent a much broader space of distributions than Gaussians, like multi modal pdfs. A widely used variant of the PF is the *Sampling Importance Resampling* PF (SIR-PF) [6,10,12,17], designed to solve the PF degeneracy problem where after a while all but one particle will have negligible weight, by introducing a resampling step to eliminate particles with small weight while duplicating particles with high weight.

The goal of this paper is to implement a realtime TBN based on a PF in the Girona500 AUV with an AHRS-DVL motion model and corrections provided by a multibeam sonar profiler and a known map.

This paper is organized as follows. On Section 2, the general idea of the used PF is explained. Section 3 introduces Octomap for Occupancy grid Mapping, followed by Section 4 with the detailed implementation of the PF. Finally, Section 5 & 6, present the results and conclusions of the experiments.

2 Particle Filter MCL

The Particle Filter (PF) is a non-parametric solution to the Bayes Filter which discretizes the pdf by a set of samples or "particles". Each one of this particles

can be seen as the possibility of the vehicle being at that position and attitude. PF is divided in three main steps. Prediction or state transition model, where particles position and attitude is predicted according to proprioceptive sensors. Weighting or measurement model, where particles are weighted according to their likelihood with data provided by exteroceptive sensors, and finally, resample, where particles with small weights are discarded.

Algorithm 1. Motion Model

function motion_model (x_{t-1}, u_t) **:**

$$\bar{\phi}_t = \phi_t + \mathcal{N}(0, \sigma_\phi) \qquad \bar{u}_{t-1} = u_{t-1} + \mathcal{N}(0, \sigma_u)$$
$$\bar{\theta}_t = \theta_t + \mathcal{N}(0, \sigma_\theta) \qquad \bar{v}_{t-1} = v_{t-1} + \mathcal{N}(0, \sigma_v)$$
$$\bar{\psi}_t = \psi_t + \mathcal{N}(0, \sigma_\psi) \qquad \bar{w}_{t-1} = w_{t-1} + \mathcal{N}(0, \sigma_w)$$
$$\begin{bmatrix} \bar{x}_t \\ \bar{y}_t \\ \bar{z}_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} + Rot_{\bar{\phi}_t, \bar{\theta}_t, \bar{\psi}_t} \cdot \begin{bmatrix} \bar{u}_{t-1} \\ \bar{v}_{t-1} \\ \bar{w}_{t-1} \end{bmatrix} \cdot t$$

return $\left[\bar{x}_t,\ \bar{y}_t,\ \bar{z}_t,\ \bar{\phi}_t,\ \bar{\theta}_t,\ \bar{\psi}_t\right]$

A constant velocity model with 6 DoF (Alg. 1) is used for the motion model. The inputs are the linear velocities measured by the DVL and the attitude measured by the AHRS, with their corresponding variances. For the measurement model (Alg. 2), the likelihood of the complete multibeam scan can be measured by individual likelihoods of each beam, where each beam range is compared with the range that would see each particle in its current position on the map. The difference between the real measurement and the expected one is weighted by a Gaussian function. Resampling is done by systematically resampling.

Algorithm 2. Measurement Model

function measurement_model (x_t, z_t) **:**

Require: σ_z

$w = 0$

for $n = 1$ **to** N **do**

if $z_t^{[n]} > 0$ **then**

$$w = w + \frac{1}{\sigma_z \sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{z_t^{[n]} - d_{cast}}{\sigma_z}\right)^2\right)$$

end if

end for

return w

3 Occupancy Grid Mapping with Octomap

Octomap [9] is a library for Occupancy Grid Mapping. Building an Occupancy Grid of a 3D environment is a highly consuming task in terms of memory. The

naive approach to construct an Occupancy Grid is to divide the space in equally sized cubes, creating a regular grid. This approach has some drawbacks, like the need to know in advance the extension of the map, and that empty regions in the map are occupying space in memory.

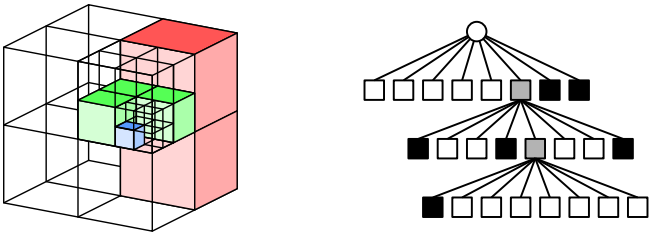


Fig. 1. Space division in octrees

To overcome this problems, Octomap library makes use of an octree representation of the space. An octree is a hierarchical structure which divides the space into cube-shaped cells which can be recursively divided in eight children cells of smaller size (Fig. 1), increasing the resolution by a factor of two at each level. This way to represent space allows to only declare those cubes that have subdivisions/children associated with them, making it much more memory efficient than regular grids.

Algorithm 3. Custom Mapping

function Mapping (x_t, z_t, o_t) **:**
Filter multibeam (z_t)
Project z_t (range, bearing) to (x, y, z).
Transform sensor origin o_t from $\mathbb{B}$ to $\mathbb{E}$ using x_t.
Transform scan z_t from $\mathbb{B}$ to $\mathbb{E}$ using x_t.
for $i = 1$ **to** N **do**
  insertRay(o_t, z_t^i)
end for

Maps are obtained with Girona500 [15], an Autonomous Underwater Vehicle (AUV) equipped with Attitude and Heading Reference System (AHRS), Doppler Velocity Logger (DVL), and a multibeam sonar profiler, which provides the profile of the seabed by means of echosounders and interferometry. Although the Octomap library is already integrated in ROS and provides a general mapping node (Octomap Server) to compute and publish maps, it is too generalized to achieve a good performance with Girona500. A specific node for multibeam sonar mapping was developed (Alg. 3) to increase map creation speed and to apply some basic filtering on multibeam data to avoid invalid measurements. These maps can be used for online path planning as well as for localization. Maps

obtained by the customized node are denser than the ones obtained with Octomap Server at the same resolution.

For each ray inserted on the octree map, the Octomap library will take care of determining all nodes traversed by the ray (ray tracing), lowering their occupancy probability by a specified value, and increasing the occupancy probability of the last node. When big amounts of scans are inserted, the `prune()` method should be called in order to fuse the nodes with same value to save memory.

4 Implementation

Realtime TBN underwater is implemented in a single C++ class (Fig. 2). The main loop which corresponds to the Particle Filter, is triggered when a message from multibeam sonar arrives (red arrow). There are other small callback functions that are triggered when the other sensor messages arrive (green arrows).

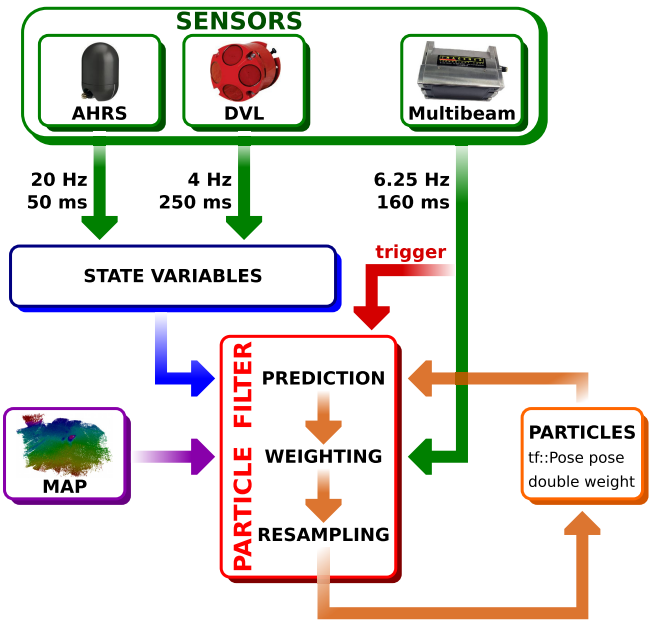


Fig. 2. Block diagram of the *UnderwaterLocalization* class

At the *UnderwaterLocalization* class instantiation, the octree map (known map) is loaded from a file, the ROS parameters are loaded, and particles are initialized at random positions around the starting position. The attitude is initialized by the first reading of the AHRS plus a Gaussian noise. Each particle is composed of a `tf::Pose` which provides position and attitude, and a corresponding weight initialized to zero.

4.1 Sensor Callbacks

When a message arrives to the node, a callback automatically saves its information in a variable if it is one of the navigation sensors (AHRS, DVL). On multibeam sonar, its callback triggers the whole particle filter. One might think that for each message of each sensor that arrives, a prediction of the PF could be run. In fact this should be the way. But taking a look at the frequencies of each sensor messages (Table 1), this simpler solution was chosen.

Table 1. Frequencies and periods of each message

	Freq (Hz)	T (ms)
AHRS	20	50
DVL	4	250
Multibeam	6.25	160

The solution of triggering the whole particle filter with the multibeam message, is feasible because the multibeam operates at a higher frequency than the DVL. The most important information on the prediction step is the velocities of the vehicle. Working at a higher frequency means that no DVL message will be lost. Because AHRS is fast, when the prediction is computed, the newest information of attitude is introduced into the filter. Finally, to achieve a realtime implementation, the whole particle filter must run in less time than the period between multibeam sonar messages (160 ms).

4.2 Particle Filter Loop and Parallelization

A Particle Filter has complexity $O(n)$, being n the number of particles used to represent the pdf. The more particles used, the better represented is the pdf, but it takes more time to compute. Since computations for each particle are independent from the other particles, to speed up the PF some parallelization is introduced with OpenMP library. On the prediction step, a single `for` loop over all the particles runs the motion model. On the weighting step a double `for` loop, one for each particle and one for each beam is used. A small speedup can be obtained by precomputing a part of the exponential weighting outside $\left(\frac{1}{2\sigma_z^2}\right)$, which will be the same value for all particles and beams. For parallelization, only the outermost loop is used, the one over all particles.

Tests were computed on the first 15 seconds of a multibeam dataset (Table 2). The best improvement between both parallelizations is given by parallelizing the update step on the Particle Filter. In the prediction step, the improvement is not appreciable because the operations inside the loop are simple. The results are even worse for a small number of particles. It takes more time to parallelize, than the time saved by using it. In the update step, more complex computations and map queries are performed. The speedup obtained by parallelizing this loop

Table 2. Time to execute one iteration of the PF

	500 particles			1000 particles		
	Mean (ms)	Min (ms)	Max (ms)	Mean (ms)	Min (ms)	Max (ms)
No parallelization	178.2	115.4	243.6	367.1	252.3	518.0
Par. prediction	181.6	119.1	248.9	351.6	244.1	463.5
Par. update	46.3	33.4	75.7	86.9	61.9	114.8
Par. pred. and update	46.3	28.4	64.7	86.6	62.2	118.3

is around 4x on an Intel Core i7-2600 CPU @ 3.40 GHz x 8, 64bit OS and 8 GB of RAM. This means that parallelizing the update step is mandatory, while the prediction step can be useful if the number of particles is big enough. Results of parallelizing PF are almost half of the available time 160 ms (Fig. 3).

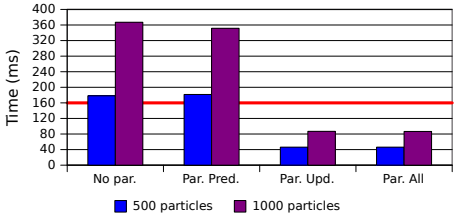


Fig. 3. Comparison of parallelization of different parts of the PF

4.3 Map Queries

The octree map is used on the weighting step of the particle filter. The ranges read from the multibeam sensor have to be compared with the ranges that the particle itself sees from its position on the map. Queries of range are performed by using the `rayCast()` function, where all coordinates have to be given regarding to the map coordinate frame. Origin and endpoints of the multibeam sensor have to be transformed from the sensor frame to the center of the vehicle, and then with each particle pose information, to the frame of the map. The ray casting query returns the octree node that was hit. To obtain the range one must compute the norm of the difference between the origin of the multibeam on the particle o_i and the endpoint of each beam $e_{i,j}$. The range $r_{i,j}$ for the particle i and beam j is:

$$r_{i,j} = ||o_i - e_{i,j}||$$

In the measurement model, instead of the normalized Gaussian the non-normalized Gaussian $\left(\exp\left(-\frac{1}{2}\frac{z^2}{\sigma_z^2}\right)\right)$ is used. This means that when a beam of a particle is in complete agreement ($z = 0$), its weight is one. To normalize the particle weights between zero and one, it is divided by the number of valid beams (non zero range).

5 Results and Discussion

The experiments are carried out on two different datasets obtained at l'Amarrador (41°46.137′ N, 03°01.929′ E), a seamount near Sant Feliu's coast, with Girona500 AUV on two different days. The AUV is brought by means of a small boat near the starting point of the preplanned trajectory and recovered after finishing. The first dataset is a lawnmower survey around the seamount and it is used to obtain the known map for localization by using the custom localization node explained at Section 3. The extension of this map is around $200 \times 250 \times 40\,m$ (Fig. 4).

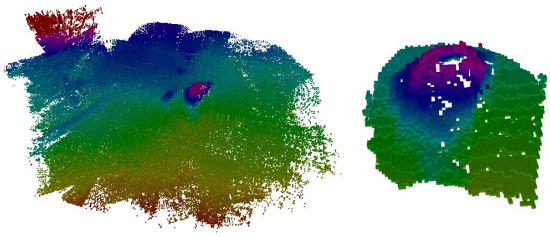


Fig. 4. Octomaps at resolution 0.5 m. *Left:* Map of the seamount obtained from the first dataset (known map). *Right:* Map of the seamount obtained from second dataset.

The second dataset (Fig. 4) is a circular coverage planning around the underwater seamount. In this case, the multibeam sonar was mounted sideways looking instead of downward looking in order to have better coverage of the seamount. This dataset is used next to test the Terrain Based Navigation Particle Filter explained in the previous sections, using the first dataset as an a priori known map. The extension of the dataset is around $40 \times 40 \times 20\,m$.

5.1 Ground Truth

The ground truth is obtained by means of an Ultra Short Baseline (USBL) acoustic positioning system. The USBL system consists of a transceiver mounted on the surface boat and a transponder mounted on the Girona500 AUV. Using the surface boat GPS position conveniently compounded with the bearing and slant ranges as well as with the range measurements from the USBL, the AUV can be located. Nevertheless, only few measurements are available (97 positions for a 10 minute dataset). Hence, the estimation error at a certain timestamp is computed as the difference between the USBL position and the PF weighted mean position of all particles (Eq. 3). The estimation error is computed for the 97 USBL positions.

$$e^t = \sqrt{(x^t_{usbl} - x^t_{pf})^2 + (y^t_{usbl} - y^t_{pf})^2 + (z^t_{usbl} - z^t_{pf})^2} \tag{3}$$

5.2 The Experiments

Different parameters of the PF were evaluated, having a total of $3 \times 4 \times 4 = 48$ experiments, corresponding to three different number of particles (500, 1000, 2000), four different number of beams (240, 48, 24, 12), and four different measurement uncertainties (1.0, 4.0, 7.0, 10.0). Different number of particles and beams can affect significantly to the realtime processing of the particle filter. If less particles are used realtime its achievable, but representation of pdf may not be accurate enough to localize correctly. If less beams are used, from the total of 240 beams provided by the multibeam, computations will also be faster, but since only some readings are going to be used each measurement will be less discriminant. On the other hand, different range uncertainties will imply to be more or less restrictive on positioning errors.

Because PF is stochastic, each experiment was repeated 6 times using different particle initializations. To allow a fair comparison of the experiments using different number of particles, an initialization with the maximum number of particles considered, 2000, was created at the beginning and all the experiments were carried out using a subset of this initial set of particles. This was repeated for each different random initialization. Finally, the runtime of each experiment was of $600s$.

5.3 Results of Terrain Based Navigation

Results were evaluated by taking the mean error through time for each experiment in each iteration. The mean of each iteration was later used to calculate the mean and standard deviation (std) between same experiments with the different initializations. Results can be observed in Table 3 and Figure 5.

Table 3. Localization mean error with standard deviation ($\bar{e} \pm \sigma_e$)

		σ_z			
#part	**#beams**	1.0	4.0	7.0	10.0
500	240	25.923 ± 00.335	26.864 ± 03.338	29.051 ± 00.253	28.236 ± 00.495
	48	35.497 ± 13.510	38.817 ± 08.229	44.661 ± 00.397	23.090 ± 16.313
	24	25.303 ± 02.123	44.751 ± 00.562	38.399 ± 10.617	24.145 ± 15.296
	12	26.666 ± 00.387	42.753 ± 00.586	10.661 ± 00.438	08.974 ± 02.422
1000	240	27.097 ± 16.315	42.960 ± 01.115	45.043 ± 00.831	28.659 ± 01.025
	48	24.158 ± 00.924	23.251 ± 25.075	06.429 ± 00.163	07.433 ± 00.247
	24	19.073 ± 08.261	05.632 ± 00.054	07.400 ± 00.546	07.057 ± 00.473
	12	26.298 ± 00.931	07.248 ± 00.027	07.658 ± 00.219	07.648 ± 00.006
2000	240	24.497 ± 00.073	44.636 ± 02.986	45.409 ± 00.408	06.142 ± 00.416
	48	24.573 ± 01.011	43.710 ± 02.365	05.916 ± 00.471	06.833 ± 00.434
	24	37.077 ± 16.791	06.437 ± 00.354	07.403 ± 00.075	06.652 ± 00.019
	12	20.663 ± 09.686	09.116 ± 00.595	07.899 ± 00.995	07.649 ± 00.758

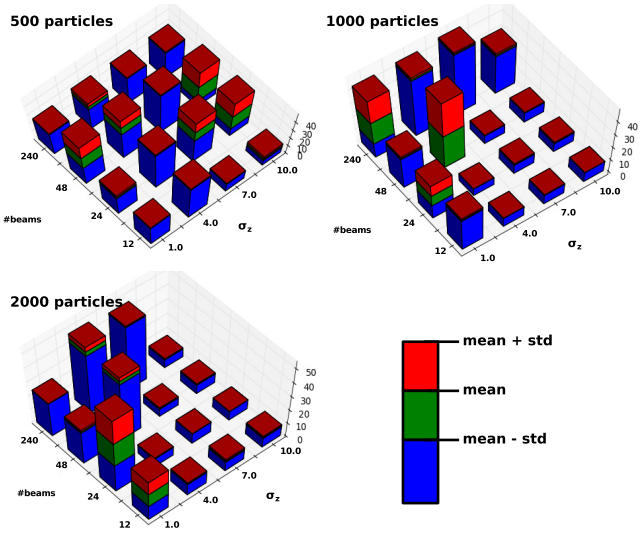


Fig. 5. Localization error and std

As expected, results on different number of particles show a decreasing error for an increasing number of particles, being 500 particles not enough to perform correctly the navigation. Increasing the measurement uncertainty σ_z, provides better results. Taking into account that the map resolution is $0.5m$, the error of range measurements is proportional to multiples of this number, being $1.0m$ too small. Although counter intuitive, decreasing number of beams also improves the error. The reason is that decreasing the number of beams reduces the computational requirements at each iteration, and more particles can be used still achieving realtime performance.

The best results are obtained with 1000 particles, 24 beams and $4.0m$ uncertainty, with $5.632 \pm 0.054m$ error. This error is still significant for a localization system, although is partially due to a bad calibration of the AUV transponder location as well as the location of the boat transceiver and its relative position with respect to the GPS antennas. Other errors may be due to multipath near the seamount. Further work is required to address this problems.

6 Conclusions

In this paper we have proposed and implemented an AUV navigation and mapping method based on the Octomap library. To this aim, several developments were carried out. First, a motion model based on DVL-AHRS navigation and a measurement model for the sonar multibeam were developed. Next, after evaluating the conventional techniques used for terrain based navigation, a PF was selected as the preferred Bayesian estimation technique for solving the TBN

problem in realtime. The system was implemented over the COLA2 software architecture and tested with the Girona500 AUV in a natural environment dataset. Several filter configurations (number of particles, number of beams and measurement uncertainty) were tested. The results of the particle filter were compared against a ground truth obtained by means of USBL positioning.

Acknowledgments. This work was supported by the Spanish project DPI2011-27977-C03-02 (COMAROB) and two European Commission's Seventh Framework Program 2011-2015 projects: FP7-ICT-2011-7-288704 (MORPH) and FP7-INFRASTRUCTURES-2012-312762 (EUROFLEETS2).

References

1. Anonsen, K., Hallingstad, O.: Terrain aided underwater navigation using point mass and particle filters. In: Proceedings of the IEEE/ION Position Location and Navigation Symposium (2006)
2. Barkby, S., Williams, S.B., Pizarro, O., Jakuba, M.: A Featureless Approach to Efficient Bathymetric SLAM Using Distributed Particle Mapping. Journal of Field Robotics 28(1), 19–39 (2011)
3. Bergem, O.: Bathymetric navigation of autonomous underwater vehicles using a multibeam sonar and a Kalman filter with relative measurement covariance matrices. Dr. Scient thesis, University of Trondheim, Norway (1993)
4. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing 10(3), 197–208 (2000)
5. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robotics & Automation Magazine 13(2), 99–110 (2006)
6. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings 140, 107–113 (1993)
7. Gracias, N.: Mosaic–based Visual Navigation for Autonomous Underwater Vehicles. PhD thesis, Instituto Superior T ecnico, Lisbon, Portugal (June 2003)
8. Haralick, R.: Propagating covariance in computer vision. In: Proc. of the Workshop on Performance Characteristics of Vision Algorithms, Cambridge, UK (April 1996)
9. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots (February 2013)
10. Karlsson, R., Gustafsson, F., Karlsson, T.: Particle filtering and Cramer-Rao lower bound for underwater navigation. In: 2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003) (2003)
11. Mallios, A., Ribas, D., Ridao, P.: Localization Advances in the Unstructured Underwater Environment. In: Proceedings of the 9th Hellenic Symposium of Oceanography and Fishery, vol. 1, pp. 111–116 (2009)
12. Maurelli, F., Krupinski, S., Petillot, Y., Salvi, J.: A Particle Filter Approach for AUV Localization. In: OCEANS MTS/IEEE (2008)
13. Fairfield, D.W.N., Kantor, G.: Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels. Journal of Field Robotics 24(1-2), 3–21 (2007)
14. Ribas, D., Ridao, P., Domingo, J.D., Neira, J.: Underwater SLAM in Man-Made Structured Environments. Journal of Field Robotics 25, 898–921 (2008)

15. Ribas, D., Ridao, P., Magi, L., Palomeras, N., Carreras, M.: The Girona 500, a multipurpose autonomous underwater vehicle. In: OCEANS 2011 IEEE-Spain, pp. 1–5. IEEE (June 2011)
16. Sistiaga, M., Opderbecke, J., Aldon, M., Rigaud, V.: Map based underwater navigation using a multibeam echosounder. In: OCEANS 1998 Conference Proceedings, vol. 2 (1998)
17. Thrun, S.: Probabilistic robotics, vol. 45. ACM (2002)
18. Zandara, S., Ridao, P., Mallios, A., Ribas, D.: MBpIC-SLAM: Probabilistic Surface Matching for Bathymetry Based SLAM. In: IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2012) (2012)

Design and Implementation of a Range-Based Formation Controller for Marine Robots

Jorge M. Soares[1,3], A. Pedro Aguiar[1,2],
António M. Pascoal[2], and Alcherio Martinoli[3]

[1] Laboratory of Robotics and Systems in Engineering and Science,
Instituto Superior Técnico, Technical University of Lisbon, Av. Rovisco Pais,
1049-001 Lisboa, Portugal
[2] Department of Electrical and Computer Engineering, Faculty of Engineering,
University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
[3] Distributed Intelligent Systems and Algorithms Laboratory,
School of Architecture, Civil and Environmental Engineering,
École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

Abstract. There is considerable worldwide interest in the use of groups of autonomous marine vehicles to carry our challenging mission scenarios, of which marine habitat mapping of complex, non-structured environments is a representative example. Relative positioning and formation control becomes mandatory in many of the missions envisioned, which require the concerted operation of multiple marine vehicles carrying distinct, yet complementary sensor suites. However, the constraints placed by the underwater medium make it hard to both communicate and localise the vehicles, even in relation to each other, let alone maintain them in a formation. As a contribution to overcoming some of these problems, this paper deals with the problem of keeping an autonomous marine vehicle in a moving triangular formation with respect to two leader vehicles. Simple feedback laws are derived to drive a controlled vehicle to its intended position in the formation using acoustic ranges obtained to the leading vehicles with no knowledge of the formation path. The paper discusses the implementation of this solution in the MEDUSA class of autonomous marine vehicles operated by IST and describes the results of trials with these vehicles exchanging information and ranges over an acoustic network.

1 Introduction

The last two decades have witnessed tremendous progress in the development of marine technologies that are steadily affording scientists and commercial companies advanced equipment and methodologies for ocean exploration and exploitation. Recent advances in robotics, sensors, computers, communications and information systems are being brought to bear on the development of sophisticated technologies to enable safer, better, faster, and more efficient methodologies for ocean exploration. These advances will undoubtedly revolutionise the way the oceans are studied, effectively placing scientists at the threshold of a

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_5,

new and exciting area when science and technology will join efforts to unravel the secrets behind recent and unexpected discoveries: intriguing ecosystems and life forms, thermal vents and cold seeps, and huge accumulations of methane in the form of gas hydrates, to name but a few. New technologies, especially autonomous marine robots capable of roaming the oceans freely, equipped with advanced sensor suites for data collection at an unprecedented scale, will also play a key role in the related fields of marine archaeology, harbour security, and transportation. Advanced marine robotic systems are also expected to afford commercial operators new tools to drastically improve the means available to monitor critical infrastructures and ocean energy production facilities (e.g. wave and wind energy generation plants), assess the size and type of fish stocks, detect and monitor the effect of hydrocarbon spills, assess the extent of mineral, oil, and gas deposits, carry out and monitor the impact of underwater mining activities, and increase the efficiency and safety of gas and oil exploration and exploitation activities. Recent developments in the field of autonomous marine vehicles, with increasingly powerful and affordable vehicles coming on the market, are steadily paving the way for a multitude of novel applications.

Many tasks envisioned to be within the reach of multi-AUV (Autonomous Underwater Vehicle) groups require the vehicles to work cooperatively. That often translates to being able to move in formation, i.e. while maintaining their relative positions. This paper considers the problem of triangular formation keeping under severe communication and localisation constraints, conditions typically found when working with groups of AUVs, and summarises the work that was previously published in [1, 2]. For a reference scenario consisting of two localised leader vehicles on the surface and an underwater follower vehicle, we use acoustic ranging and communications to establish and maintain a moving formation of the three vehicles. Of the multiple real-world applications matching this scenario, a typical one is surface-guided underwater exploration. We make a realistic assumption that the AUV has independent depth control, and focus on formation control in the 2D plane only.

We propose a control strategy that estimates the formation speed and heading from the acoustic ranges obtained to the two leading vehicles, and uses simple feedback laws for speed and heading to drive suitably defined common and differential errors to zero. We then discuss the implementation of this solution in a MEDUSA-class autonomous marine vehicle, describing the challenges posed by the medium and the changes that arise as a consequence, and present the results of real world tests performed with 3 autonomous vehicles.

The paper is organised as follows: the present section provides important background to our work and Section 2 summarizes previous related work; Section 3 describes the specific problem in more detail; Section 4 contains a description of the MEDUSA class of autonomous marine vehicles and their dynamic models; Section 5 describes the error dynamics and outlines the control laws for vehicle heading and linear velocity; Section 6 discusses the necessary adaptations for implementation in a real vehicle and Section 7 summarises the results obtained during real-world trials with 3 autonomous marine vehicles. Finally, Section 8 contains the conclusions and lists directions for future research.

2 Related Work

One interesting work in formation control for mobile robots is described in [3], where the authors discuss approaches for both range-bearing and range-range control, depending on the available sensors, to solve a leader-follower control problem for a formation graph with an arbitrary number of vehicles; in both cases, knowledge of the leader motion is assumed. In [4], and supported by robot experiments, a different graph-based leader-follower solution using range and bearing is proposed. Another strategy is described in [5] for a 4-vehicle station keeping problem, using exclusively range measurements and holonomic vehicles described by simple kinematic points. In [6], a similar scenario is considered, although global convergence is only proved for a triangular formation.

Bearing-only methods are also available for square [7] and triangular [8, 9] formations. In [10], the authors advance algorithms to coordinate a formation of mobile agents when the agents can only measure the ranges to their immediate neighbours. This solution requires that subsets of non-neighbouring agents localise the relative positions of their neighbours while these are stationary, and then move to minimise a cost function.

For the special case of marine vehicles, a solution that decouples the controllers for formation shape, formation motion and vehicle orientation, but requires position information is proposed in [11]. Coordinated path following approaches are presented in [12] and [13], the latter specifically dealing with underwater pipeline inspection. These strategies assume that the path to be followed is known to all vehicles, and generally work by exchanging some along-path synchronisation measure. An example of a real-world AUV operation making use of formation control is documented in [14].

3 Problem Statement

Figure 1 illustrates the control problem discussed in this paper, and shows two leading vehicles (vehicles 2 and 3, represented as x_2 and x_3) moving along a certain unknown path, and a follower (vehicle 1, represented as x), of which we have control. Through the remainder of this paper, and unless otherwise stated, the absence of an index indicates a variable or parameter related to vehicle 1, the controlled or trailing vehicle.

The goal is for the trailing vehicle to follow the leaders in an equilateral triangular formation of side d, i.e. in the figure, x should converge to the desired position x_d. There exists a symmetric solution to the problem, with the desired position x_d mirrored in relation to the segment defined by $x_2\, x_3$. The solution shown in Fig. 1 corresponds to a *following motion* and the mirrored solution to a *leading motion*. We only deal with the case of following motion.

The basic control problem consists, as we have seen, of deriving control laws to drive x to x_d. The challenge stems from working with AUVs with no access to global localisation methods and with slow and unreliable inter-vehicle communication. Here, we make a reasonable assumption that the only localisation

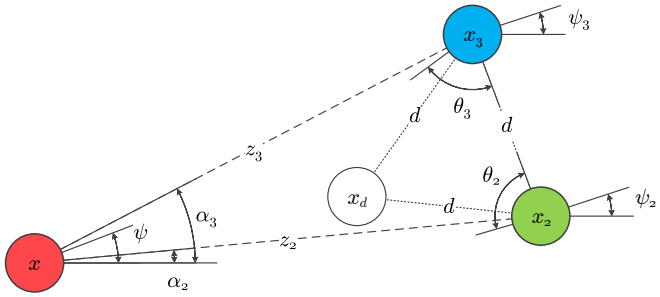


Fig. 1. System of three robots (x, x_2, x_3) and their intended triangular formation (x_d, x_2, x_3). The image shows many of the relevant parameters, including the formation and independent vehicle headings, as well as the relationships (ranges and bearings) between them. The heading and course of the vehicles are only aligned in the absence of current. Note that the colour convention holds throughout the paper.

hardware available on the AUV is a low data rate acoustic modem and ranging device, which is the case for our vehicles, presented in the next section. The complete problem becomes, then, to derive and implement control laws that achieve convergence of x to x_d using only limited communication and information.

4 Vehicle Details

The MEDUSA-class autonomous semi-submersible robotic vehicles, shown in Fig. 2, were developed at the Laboratory of Robotics and Systems in Engineering and Science (LARSyS), Instituto Superior Técnico. The MEDUSAs were originally designed and built as surface vehicles, but a diving capable version is now also operational. Nevertheless, in this paper we use an artificially constrained surface-bound MEDUSA to emulate our AUV. This has some practical advantages, as we can mimic the most relevant characteristics of an AUV while retaining a GPS receiver and a radio communication channel, respectively used for ground truth and remote monitoring (but not for communication with the remaining vehicles).

Each MEDUSA-class vehicle weighs approximately 30 Kg and consists of two longitudinal acrylic housings with a total length of around 1 m. The upper body is partially above the surface and carries an EPIC single-board computer, an RTK-enabled GPS receiver, a full navigation sensor suite and an underwater camera. Most of the lower body is taken up by the batteries. An 802.11 interface is used for surface communications, while a Tritech acoustic modem enables underwater communication. The vehicle is propelled by two side-mounted, forward-facing stern thrusters that directly control surge and yaw motion, and is capable of speeds up to 1.5 m/s.

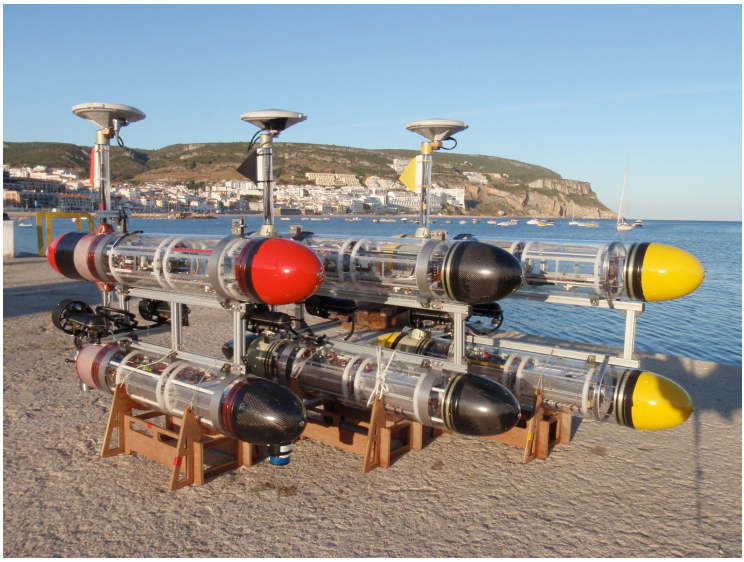


Fig. 2. The MEDUSA AMVs being readied for deployment at an experimental site

As the vehicle moves on the surface, its kinematic equations take the form

$$\begin{aligned}\dot{x} &= u\cos\psi - v\sin\psi \\ \dot{y} &= u\sin\psi + v\cos\psi \\ \dot{\psi} &= r,\end{aligned}$$

where u (surge speed) and v (sway speed) are the body axis components of the velocity of the vehicle, x and y are the Cartesian coordinates of its centre of mass, ψ defines its orientation (heading angle), and r its angular velocity. The motions in heave, roll and pitch can be neglected, due to the large enough metacentric height. The resulting dynamic equations of motion for surge, sway and yaw are

$$\begin{aligned}m_u\dot{u} - m_v vr + d_u u &= \tau_u \\ m_v\dot{v} + m_u ur + d_v v &= 0 \\ m_r\dot{r} - m_{uv}uv + d_r r &= \tau_r,\end{aligned}$$

where τ_u stands for the external force in surge (common mode), τ_r for the external torque (differential mode), and the m and d terms represent vehicle masses, hydrodynamic added masses, and linear and quadratic hydrodynamic damping effects. The complete model for the MEDUSA vehicles is presented in [15].

5 Controller Design

We start by deriving the control strategy using a basic kinematic model for the vehicles (distinct from the realistic model found in the previous section), and under the assumption of continuous communication and control. While the resulting controllers are not guaranteed to apply in the real world, we later show that, with the proper adaptations, they do indeed work on the real vehicles.

We assume that the follower starts from a following position, in order to converge to a following motion, and that the leader vehicles (2 and 3) move at a distance d from each other, according to simple kinematics described by

$$\dot{x}_i = \begin{bmatrix} v_i \cos\psi_i \\ v_i \sin\psi_i \end{bmatrix}, \quad i = 2, 3$$

where $(v_2 + v_3)/2 = v_f$ is the formation speed. The control signals are the linear velocity v and the heading ψ, and the kinematic model of the follower is given by

$$\dot{x} = \begin{bmatrix} v \cos\psi \\ v \sin\psi \end{bmatrix},$$

where $x \in R^2$ denotes its Cartesian position. Here, we accept that both leaders move with a common heading $\psi_f = \psi_2 = \psi_3$, and that the total velocity vector of each leading vehicle is always perpendicular to the line segment that joins them. The heading ψ_f is unknown to vehicle 1.

Separate controllers are designed to stabilise each error measure, with the speed controller stabilising the common mode error and the heading controller stabilising the differential mode error. What follows is an overview of the resulting controllers; intermediate steps in the derivation and proofs of convergence can be found in [1].

5.1 Error Dynamics

Let $z_i = \|x_i - x\|; i = 2, 3$ denote the distances from the trailing vehicle to each of the leaders. From the range measurements, we define the common and differential mode errors

$$\epsilon = \frac{e_2 + e_3}{2} = \frac{z_2 + z_3}{2} - d$$
$$\delta = e_3 - e_2 = z_3 - z_2,$$

respectively with $e_i = z_i - d; i = 2, 3$. From the definition of z_i, it follows that

$$\dot{z}_i = v_i \cos(\alpha_i - \psi_f) - v \cos(\alpha_i - \psi),$$

Although the control strategy can be applied to other types of trajectories, the next sections assume the simpler case of straight line constant-speed motion for the two leading vehicles. This means that $v_2 = v_3 = v_f$ and the simplified error dynamics for ϵ and δ become

$$\dot{\epsilon} = \cos\beta \left(v_f \cos\varphi - v \cos(\varphi + \tilde{\psi}) \right) \tag{1}$$
$$\dot{\delta} = 2 \sin\beta \left(v_f \sin\varphi - v \sin(\varphi + \tilde{\psi}) \right),$$

where

$$\beta = \frac{\theta_2 + \theta_3}{2} - \frac{\pi}{2}$$
$$\varphi = \frac{\theta_2 - \theta_3}{2},$$

and $\tilde{\psi} = \psi_f - \psi$ is the heading error.

5.2 Speed Controller

We propose the following speed controller to regulate the common mode error ϵ to zero:

$$v = K_p^s \epsilon + K_i \int_0^t \epsilon \, d\tau,$$

where $K_p^s > 0$ and $K_i > 0$ are the proportional and integral gains, respectively. The rationale behind the proposed control law is that when the leader vehicles follow a straight-line trajectory with constant speed v_f, $\psi = \psi_f$ and $\delta = 0$ (i.e. x is on the perpendicular bisector of the x_2x_3 line segment), the dynamics of ϵ in (1) reduce to

$$\dot{\epsilon} = \cos\beta(v_f - v),$$

and, since $\cos\beta > 0$, a control law $v = v_f + K_p^s \epsilon$, $K_p^s > 0$ stabilises exponentially the origin $\epsilon = 0$, provided β does not converge to $-\frac{\pi}{2}$. As v_f is unknown, we include an integral term to learn it.

5.3 Heading Controller

For the heading controller we propose the following control law that uses the differential mode error δ:

$$\psi = \hat{\psi}_f + \gamma(K_p^h \delta),$$

where $K_p^h > 0$, $\hat{\psi}_f$ denotes an estimate of the formation heading ψ_f, and γ is any function such that $\sin(\gamma(ay))y > 0, \forall a > 0$. An example is the saturation function $\gamma(y) = \frac{\pi}{2}\,\mathrm{sat}(y)$.

6 Implementation

While the controllers developed show good performance under the assumptions made during their derivation (results in [1]), moving to a real-world implementation requires significant changes.

First and foremost, ranges in an underwater setting are most often measured using acoustic equipment, by registering the time of flight of an echo request and reply. In our case, the ranging is done by the general purpose Tritech acoustic modem that equips the MEDUSA. The low transmission speed makes it so that we can only issue one echo request every few seconds. Since transmissions cannot overlap on the single common channel, time multiplexing must be used to obtain the ranges to each of the leader vehicles. We choose to query each one separately, although other solutions are possible, e.g. emitting a broadcast ping with vehicle-dependent delayed replies. Since both leaders have to be queried, a complete information update only occurs every four seconds. This is in stark contrast with our previous assumption of continuous measurement.

To prevent changes to the algorithms, we have chosen to implement two hybrid Kalman filters that take the discrete samples and output a continuous estimate of the distances. The range information received is never current, and comes with a latency of approximately 0.5 seconds, imposed by transmission times and I/O scheduling on both the sender and receiver. We decided not to implement any mitigation techniques (e.g. back-dating the filter updates), instead retaining the simplicity of the solution.

The measurements taken are inherently noisy. This noise is, for practical purposes, quite low - we did not fully characterise it, but the individual ranging error was predominantly under 0.5 m - but it again must be taken into account. The same way, outliers are inevitable, albeit infrequent. These are mostly caused by floor geometry and non-uniform propagation in the water, leading to the reception of an echo reply through a path other than the shortest one and resulting in an overestimation of the distance. We implemented a simple outlier filter based on a sliding window. Losses are also an inescapable reality, and need to be tolerated within reasonable limits.

The heading for the leaders, despite being used, is also not implicitly available to the follower, and has to be communicated. Seeing as the vehicles use full-featured acoustic modems to measure ranges, it is possible to piggyback data on the ranging reply. This feature must be used with caution, in order not to over-extend communication times (thereby decreasing the sampling rate even further), but adding an integer to the reply is without major consequences. In our implementation, the heading is transmitted as a piggybacked single byte on the echo reply, and fed to another hybrid Kalman filter with the incoming values whenever a new range is received. While each range estimator is, in the absence of losses, updated every four seconds, the heading estimator is updated every two seconds. As the heading of both vehicles should be close to and converge to the formation heading, this allows for a higher quality and more responsive estimate.

An overview of the resulting implementation is presented in Fig. 3. The expressions for the speed and heading controller in our implemented solution remain unchanged, with the required adaptations being handled by earlier stages. While we are using a surface vehicle, the solution is applicable for constant-depth

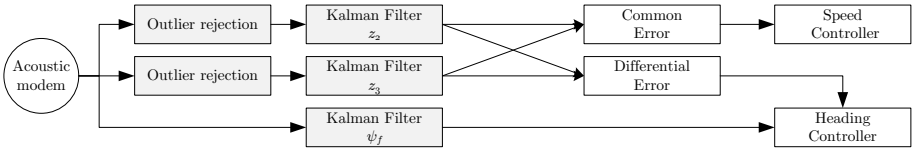


Fig. 3. Structure and data flow in the MEDUSA implementation of the formation control algorithm. Modules in grey perform the conversion of available discrete data to the continuous signals required the algorithm.

underwater operations and, with minor changes, to variable-depth underwater vehicles equipped with a depth sensor and independent depth control.

7 Experimental Evaluation

Real world trials were conducted in June 2012 at Parque das Nações in Lisbon, Portugal. This is a fairly sheltered saltwater bay connected to the Tagus estuary, of which an aerial view is presented in Fig. 4. It provides for ample space for testing, with minimal currents and good conditions for deployment of the control center. Water depth is restricted (generally under 5 m), which limits the performance of the acoustic communication systems. All vehicles were equipped with the full sensor suite, including RTK GPS, but the trailing vehicle only logs the position data for ground truth and does not use it, in any way, for navigation.

The leader vehicles, running the Coordinated Path Following algorithm described in [15], were configured to execute the 3-legged lawnmower manoeuvre presented below, spanning around 120 m x 120 m, at a reference speed $v_f = 0.4$ m/s. The vehicles were set to a triangular formation with $d = 13$ m.

Figure 5 shows a top view of the paths described by the three vehicles, starting in the upper right corner: the leaders are pictured in green and blue, with the red follower trailing behind. A transient can be noticed at the beginning: none of the vehicles start in their designated position or heading and need to adjust. The movement of the leader and their rapidly varying reported headings impact the Kalman filter estimate, causing the controlled vehicle to start in a non-ideal direction. Afterwards, the leaders negotiate the set path while the follower accurately position in the formation.

Minimal packet loss was observed during the trials. When it takes place, it is mostly while turning, presumably due to mis-alignment of the acoustic modems in the three vehicles, and is the leading factor causing the vehicle to stray off path. Nevertheless, Fig. 6 shows that the errors are low: after the initial adjustment period, the common mode error generally remains under 1 m, and the differential mode error remains under 3 m. Two minor peaks in the common error, caused by packet loss, can be seen around 180 s and 400 s, at the beginning of each turn.

Figure 7 shows the speed of the follower (the mean speed of the leaders is, as previously stated, 0.4 m/s), as well as the headings of the follower (red) and leaders (green and blue). It also shows the received heading packets from the

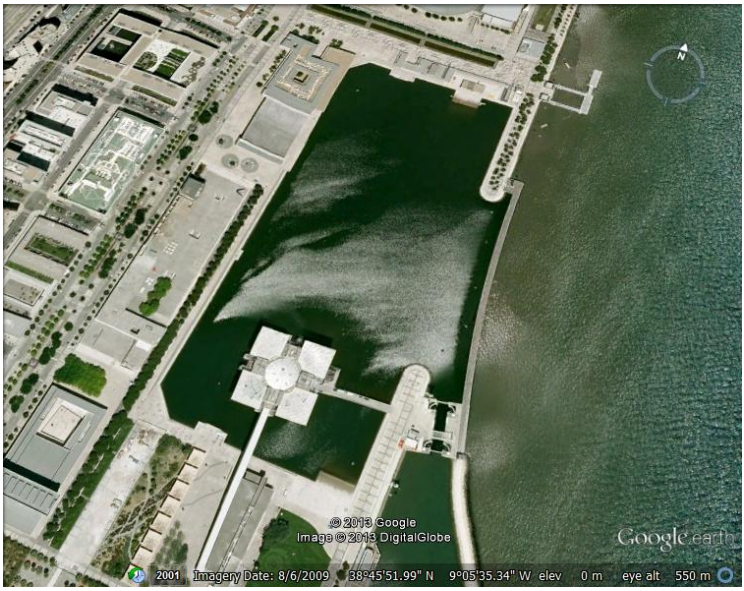


Fig. 4. Aerial view of the sheltered salt-water bay in the Tagus river estuary where the tests were conducted

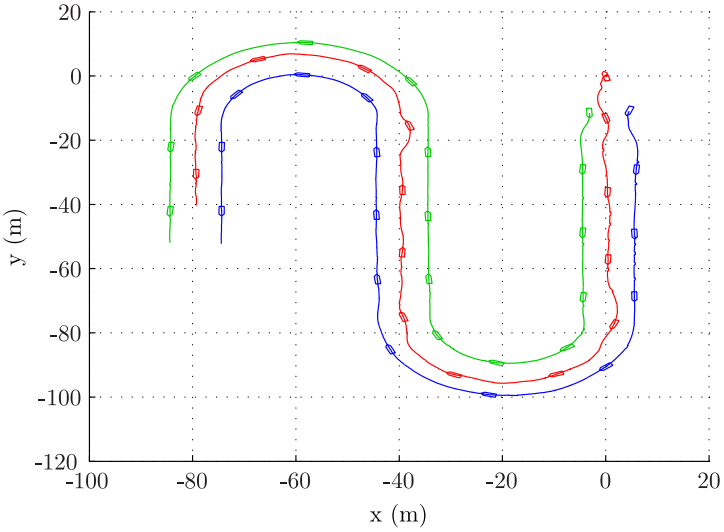


Fig. 5. Path followed by the vehicles during real-world trials. The manoeuvre starts on the top right corner, and the trajectory of the controlled vehicle is shown in red.

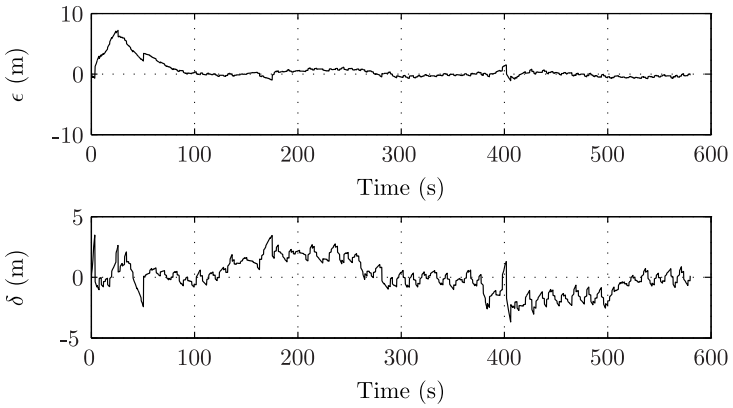


Fig. 6. Time evolution of the common mode and differential mode errors along the lawnmower path

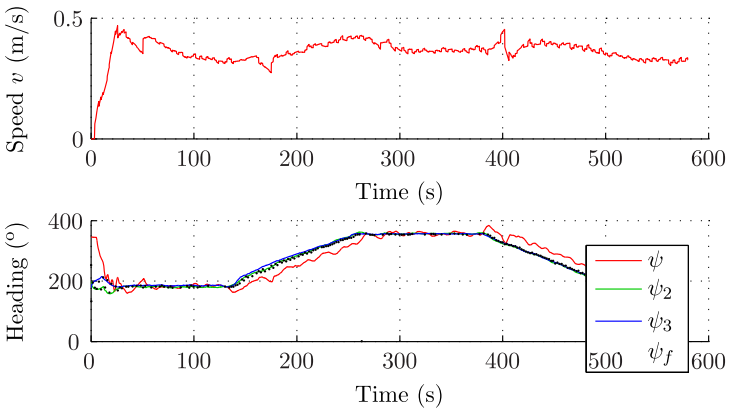


Fig. 7. Time evolution of the follower vehicle speed v, and headings ψ, ψ_2, ψ_3, as well as the discrete references received acoustically, ψ_f. The vehicle speed is estimated from GPS measurements, and has non-negligible associated noise.

leader, represented by the black dots. As expected, the plots closely match the error plots, with a clearly visible peak in speed at 400 s.

8 Conclusions and Outlook

In this paper we reviewed a solution to a three-vehicle formation keeping problem where a follower moves in a triangular formation behind two leading vehicles, using inter-vehicle range measurements with no a priori knowledge of the path taken by the leaders. The algorithm considers a discrete and noisy measurement

model with low sampling rate and uses additional heading information piggybacked on the acoustic echo reply.

The proposed solution was implemented and tested on the MEDUSA class of vehicles. The evaluation results show good performance, with minimal disturbance under straight lines, even in the presence of packet loss, sensor noise and outliers. The hybrid Kalman filters used are able to accurately estimate the distances, despite the low rate of the acoustic ranges, and the piggybacked heading information allows for smooth response to changes in direction.

Work is ongoing regarding the testing of the algorithm on an underwater setting, using a new diving version of the MEDUSA. The algorithm is also being extended to a larger number of vehicles and different formation shapes. Future work should include the pairing of the algorithm with robust methods for initialisation and collision avoidance. Finally, open sea trials will allow us to test and validate the algorithm in the presence of stronger currents, waves and winds.

Acknowledgements. This work was supported by projects CONAV/FCT-PT [PTDC / EEACRO / 113820 / 2009], MORPH [EU FP7 ICT 288704], and FCT [PEst-OE / EEI / LA0009 / 2013]. Partially funded with grant SFRH / BD / 51073 / 2010 from Fundação para a Ciência e Tecnologia.

References

1. Soares, J.M., Aguiar, A.P., Pascoal, A.M.: Triangular formation control using range measurements: an application to marine robotic vehicles. In: IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2012), Porto, Portugal (2012)
2. Soares, J.M., Aguiar, A.P., Pascoal, A., Martinoli, A.: Joint ASV/AUV Range-Based Formation Control: Theory and Experimental Results. In: 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany (2013)
3. Desai, J., Ostrowski, J., Kumar, V.: Modeling and control of formations of nonholonomic mobile robots. IEEE Transactions on Robotics and Automation 17(6), 905–908 (2001)
4. Falconi, R., Gowal, S., Martinoli, A.: Graph-based distributed control of nonholonomic vehicles endowed with local positioning information engaged in escorting missions. In: 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA, pp. 3207–3214. IEEE (May 2010)
5. Cao, M., Morse, A.S.: Station keeping in the plane with range-only measurements. In: 2007 American Control Conference, pp. 5419–5424. IEEE, New York (2007)
6. Oh, K.K., Ahn, H.S.: Formation control of mobile agents based on inter-agent distance dynamics. Automatica 47(10), 2306–2312 (2011)
7. Bishop, A.N.: Distributed bearing-only formation control with four agents and a weak control law. In: 2011 9th IEEE International Conference on Control and Automation (ICCA), pp. 30–35. IEEE, Santiago (2011)
8. Bishop, A.N.: A Very Relaxed Control Law for Bearing-Only Triangular Formation Control. In: Proceedings of the 18th IFAC World Congress, Milano, Italy, pp. 5991–5998 (August 2011)

9. Basiri, M., Bishop, A.N., Jensfelt, P.: Distributed control of triangular formations with angle-only constraints. Systems & Control Letters 59(2), 147–154 (2010)
10. Cao, M., Yu, C., Anderson, B.D.O.: Formation control using range-only measurements. Automatica 47(4), 776–781 (2011)
11. Yang, H., Zhang, F.: Geometric formation control for autonomous underwater vehicles. In: 2010 IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA, pp. 4288–4293. IEEE (May 2010)
12. Ghabcheloo, R., Aguiar, A.P., Pascoal, A.M., Silvestre, C., Kaminer, I., Hespanha, J.: Coordinated path-following in the presence of communication losses and time delays. SIAM Journal on Control and Optimization 48(1), 234 (2009)
13. Xiang, X., Jouvencel, B., Parodi, O.: Coordinated Formation Control of Multiple Autonomous Underwater Vehicles for Pipeline Inspection. International Journal of Advanced Robotic Systems 7(1), 1 (2010)
14. Leonard, N.E., Paley, D.A., Lekien, F., Sepulchre, R., Fratantoni, D.M., Davis, R.E.: Collective Motion, Sensor Networks, and Ocean Sampling. Proceedings of the IEEE 95(1), 48–74 (2007)
15. Ribeiro, J.: Motion Control of Single and Multiple Autonomous Marine Vehicles. Master's thesis, Instituto Superior Técnico - Technical University of Lisbon (2011)

Rotation Estimation for Two-Dimensional Forward-Looking Sonar Mosaicing

Natàlia Hurtós, Xavier Cufí, and Joaquim Salvi

Computer Vision and Robotics Group (ViCOROB), University of Girona,
17071 Girona, Spain
{nhurtos,xcuf,qsalvi}@eia.udg.edu

Abstract. Two-dimensional forward-looking sonars are becoming standard sensors in both remotely operated and autonomous underwater vehicles, increasing the possibility of mapping under low visibility conditions. Due to the inherent nature of sonar image formation, the ideal mapping strategy relies on maintaining the same orientation, so as to minimize intensity alterations due to viewpoint changes. However, this is not always possible and therefore it is necessary to deal with the registration of sonar images under rotational movements. Previous investigations have discouraged the use of feature-based techniques and have suggested the use of global methods that are robust to noise, low-resolution and inhomogeneous insonification and can deal with the decoupled estimation of roto-translations. In this paper we review several candidate methods and assess them by using real data gathered under different conditions. By identifying the best approach for rotation estimation we aim to extend the applicability of sonar mosaicing to more diverse scenarios. Results indicate that applying phase correlation directly to polar frames leads to the highest accuracy under most cases.

Keywords: forward-looking sonar, mosaicing, rotation estimation.

1 Introduction

Mapping of underwater environments under low visibility conditions is a key aspect in many application areas such as harbor and dam inspection or monitoring of rivers and lakes. During the last decade, advances in sonar technology have brought a new generation of two-dimensional Forward-Looking Sonars (2D FLS) that are able to deliver high-resolution acoustic images at near-video frame rate. However, sonar mapping cannot directly borrow from the techniques used on optical images given the challenging characteristics of FLS imagery. Several authors [1] [2] have noticed that the traditional feature-based pipeline (i.e the extraction and matching of local point features to retrieve the transformation that relates two images) is not applicable to FLS imagery. The noise, low-resolution and inhomogeneous intensity variations characteristic of the FLS modality do not allow for the extraction of stable features thus leading to incorrect registrations. In view of that, some works have proposed the extraction of more stable features based on regions computed from image gradients [3][5].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_6,

In earlier work [7], we have proposed the use of Fourier-based registrations to address the alignment of FLS images. Fourier-based techniques have been proved to handle well the challenging characteristics of FLS images. At the same time, they offer an easy way to derive an uncertainty measure of the registration which is later included in a maximum likelihood mapping framework to ensure the consistency of the final mosaic. However, most of the presented results deal with translational displacements, with very little work addressing rotational movements, which the present paper aims to develop. Indeed, since most of the FLS registration techniques existing in the literature are feature-based, rotation estimation has not received explicit treatment as it is intrinsically estimated from the matching stage.

This paper examines different alternatives for estimating the rotation between a pair of FLS images and thus identifying the best estimation approach to integrate in a mosaicing pipeline. The inherent nature of sonar images suggests that mapping an area while maintaining the same orientation view point facilitates the registrations between sonar frames. In this way, a lawn mower pattern where the transition from trackline to trackline is performed by sway displacement instead of rotation is a good mapping strategy since the images do not suffer from large intensity alterations and shadow movements due to different vantage points. However, this approach might not be always possible, either because the vehicle does not allow for the sway degree of freedom, or simply because the area to cover does not follow a rectangular layout and some orientation changes must be carried out in order to cover it efficiently. Moreover, if we think not only of autonomous surveys, but inspections carried out with remotely-operated vehicles (ROVs), the pilot will most likely undertake a large number of rotational movements. Hence, it is worth studying the particular problem of rotation estimation from FLS imagery since a robust solution would enable the applicability of sonar mosaicing in more diverse situations and environments.

The paper is organized as follows: the next section provides a brief background on FLS imaging, describing the geometry model that we take into consideration for the registration process. Section III presents several rotation estimation methods that, due to their characteristics, are suitable candidates for performing rotation estimation in FLS images. Section IV presents the experiments, including data sets of real environments involving rotational movements, either purely rotational or combined with translational displacements. The outcome of the experiments is discussed in section V and finally Section VI points out some conclusions and future work.

2 Background

A Forward-Looking Sonar insonifies the scene with an acoustic wave, spanning its field of view in azimuth (θ) and elevation (ϕ) directions (Fig. 1). The acoustic return is sampled by an array of transducers as a function of range and bearing, resulting in a polar image with the backscattered intensities at each point (r, θ). This polar representation can be then converted to cartesian coordinates yielding

a fan-shaped sonar image of non-uniform resolution. However, given a particular range and bearing it is not possible to disambiguate the elevation angle of the acoustic return since the reflected echo could have originated anywhere along the corresponding elevation arc.

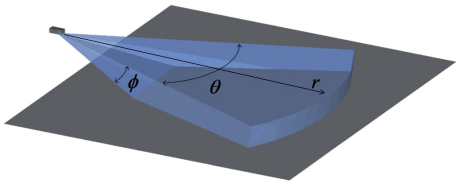


Fig. 1. Imaging sonar geometry (r: range, θ: azimuth, ϕ: elevation)

According to this image formation process, a general sonar motion induces a non-uniform transformation in the image coordinates that can be approximated by an affine transformation that varies across the image depending on the unknown elevation angles [4]. Some works use this model [6][5] which requires the knowledge of the elevation angles, to obtain estimates of the sonar 3D motion including the z direction. The authors make use of shadow information to extract estimates of the elevation angles.

However, given the narrow beam width that typically characterizes FLS devices in the elevation direction (around $7-10°$), several authors consider only the zero-elevation plane, which yields a linear projection model in which the sonar can be seen as an orthographic camera [1]. By using this model, a point on two sonar images can be related through a 2D rigid transformation, comprising only the x-y translation and plane rotation [7]. Although it is only an approximation, it has been shown to hold reasonably well under suitable imaging conditions [3]. For 2D mapping purposes, and under the planar assumption, the estimation of roto-translations is sufficient and enables the applicability of global-area methods that can only estimate up to similarity transformations. The advantage of using a global technique is that instead of using sparse feature information, all the image content is taken into account. Thus, by incorporating more information in the registration process, we are able to handle more changes in the visual appearance of the image and minimize ambiguities in the registration.

3 Global Methodologies for Image Rotation Registration

The estimation of relative orientations between a pair of images has been extensively studied in the context of several image processing tasks such as image registration, pattern recognition or motion compensation to name a few. A large number of existing techniques rely in the extraction of point features and local descriptors which, as pointed out earlier, are not effective in the context of FLS image registration due to the inherent characteristics of this imaging modality.

Following the idea presented in [7], we seek global techniques that involve the use of all the image information without requiring the extraction of features, and at the same time, offer tolerance to noise and inhomogeneous intensity artifacts. In the following sections we will briefly describe five different techniques found in the literature that by their properties seem to be promising in these regards. Notice that the algorithms are focused exclusively in the estimation of rotational angles, as the translation estimation of FLS frames is considered to be solved by using phase-correlation as explained in [7]. It should be noted that in using this scheme, the accuracy of the rotation estimation step is crucial, as the subsequent translation estimation depends on the previous rotation compensation.

3.1 Polar Magnitude of the Fourier Transform

One of the most popular methods to register images that have both translational and rotational misalignments is based on the polar Fourier Transform (FT), often referred to as Fourier-Mellin transform [8][9]. Given two images $f(x, y)$ and $g(x, y)$ related through a roto-translation given by:

$$f(x, y) = g(xcos\theta_0 + ysin\theta_0 + x_0, -xsin\theta_0 + ycos\theta_0 + y_0), \tag{1}$$

where θ_0 and $[x_0, y_0]$ are the relative rotation and translation between f and g respectively, we can express the FT of (1) in polar coordinates as:

$$F(r, \theta) = G(r, \theta + \theta_0)e^{-i(ux_0+vy_0)} \tag{2}$$

From equation 2, we can see that the rotation in the spatial domain is represented by a linear shift in the polar domain. According to the properties of the FT, it can be seen that the translation only affects the phase spectrum, so the rotation can be determined independently from the magnitude of the polar FT:

$$|F(r, \theta)| = |G(r, \theta + \theta_0)| \tag{3}$$

Hence, the rotation estimation problem is converted to a shift estimation where the input images are the polar representations of the FT's magnitude. This shift estimation can be solved by standard-phase correlation, and will lead to two possible solutions (θ and $\theta + \pi$) that can be disambiguated by computing the two phase correlations and choosing the one with highest correlation peak.

To avoid edge effects caused by the sonar fan-shaped footprint, it is necessary to smooth the edges of the the input images as well as to window the polar magnitudes of the FT's. It is common practice [9] to apply some filtering operations to the FTs in order to attenuate unwanted frequencies that can lead to a noisy phase correlation matrix. However, determining these filters can be critical as there is a risk of attenuating not only the unwanted components but also the discriminating phase components. Here, we do not apply phase correlation on the original images, but on the polar magnitude of the FT, which has a low structural nature (particularly in the case of sonar modality) and suffers from inaccuracies introduced by the interpolation process. Therefore, we choose to

apply a filter to emphasize the high frequencies, in order to highlight edges and components that can serve as a reference in the correlation. On the other hand, in the spatial domain, we apply a smoothing filter to reduce the noise on the phase correlation matrix and better identify the location of the peak.

3.2 Gradient Cross-Correlation

Tzimiropoulos et al.[10] presented a method for roto-translation estimation using gradient cross-correlation. The technique is based on the phase-correlation principle but employs as input images a gray level edge map G computed as follows:

$$G = G_x + \jmath G_y, \tag{4}$$

where $G_x = \nabla_x I$ and $G_y = \nabla_y I$ are gradients along the horizontal and vertical directions of image I, computed as the first derivative of a Gaussian. This representation has the advantage that it preserves location, magnitude and orientation of the image's salient regions. The low-spatial frequencies are filtered out (areas without specially distinctive traits), while keeping the most salient regions that reflect the orientation of image content. Then, similar to the previous method, the estimation of the rotation angle is performed as a phase-correlation in the angular direction of the polar gradient magnitude.

The authors claim that the method does not require windowing operations since the gradient map will most likely not contain strong edges near the image's boundary and therefore no information will be lost or attenuated as a consequence of the windowing. However, in our case, due to the fan-shaped footprint of the sonar images it is still required to apply a window before the processing to prevent the correlations from latching on the edges of the image layout.

3.3 Angular Difference Estimation

Some other Fourier-based domain methods are not based on correlation strategies but on a specific property that enables the estimation of rotation working directly in the Cartesian domain, bypassing the need for any polar conversion. Given two images, it has been shown [11] that the difference between the magnitudes of the FT of the first image and the mirrored version of the second has a pair of orthogonal zero-crossing lines. The orientation of these lines with respect to the frequency axis is shown to correspond to half the rotational angle [11]. Several methods have been proposed for the detection of this cross-pattern, including the use of a multiscale Hough transform [11] or the detection through the definition of an Angular Difference Function [12]. Figure 2 shows an example of the Fourier magnitude difference of two sonar images that have been previously masked to avoid edge effects and flipped accordingly. The contour of the difference image around the center reveals an intricate pattern where it is difficult to appreciate the presence of a cross pattern. Further tests, not reproduced here due to lack of space, have attempted to emphasize the cross-pattern by high-pass filtering in the frequency domain without success. FLS images present, in

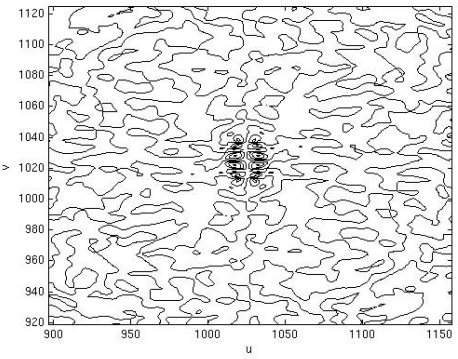


Fig. 2. Contour plot of the FT magnitude difference of two sonar images around the frequency center. Compared to the images reported in [11], it is unfeasible to perform the detection of the zero-crossing orthogonal pattern.

general, a low structural spectral magnitude, while all the reported images in the related literature seem to have strong structural components. Moreover, the traditional speckle noise of the sonar images can also contribute masking the pattern. Therefore, this branch of techniques do not seem suitable for FLS imagery and they are not considered in the experimental section.

3.4 Radon Transform

The Radon Transform (RT) has been used in several works [13],[14] to reduce the dimensionality of the rotation estimation problem by working with 1D projections. The RT of a 2D image $f(x, y)$, noted as $F^R(r, \theta)$, is found by computing line projections along f, where the lines are defined by their perpendicular distance from the origin r and the angle that r makes with the horizontal axis θ. Interpolation is required as the projections do not usually pass through the center of the image's pixels. An image $f(x, y)$ rotated by an angle θ_0 can be represented by its RT as:

$$RT(f(x\cos(\theta_0) + y\sin(\theta_0), -x\sin(\theta_0) + y\cos(\theta_0))) = F^R(r, \theta - \theta_0) \quad (5)$$

Hence, an image rotation corresponds to a translation of the angular variable of the RT. It can also be shown that a translation $[x_0, y_0]$ corresponds to a translation of the spatial variable of the RT by an amount that depends on both the image translation and the angular variable θ_0:

$$RT(f(x + x_0, y + y_0)) = F^R(r - x_0\cos(\theta) + y_0\sin(\theta), \theta) \quad (6)$$

It can be readily seen from Eq.6 that certain choices of θ, particularly $\theta = 0°$ and $\theta = 90°$, lead to a direct estimation of x_0 and y_0 respectively. For instance, translation y_0 can be extracted as the maximum correlation between $RT(f(x, y))$

and $RT(f(x+x_0, y+y_0))$ for $\theta = 90°$. However, when undergoing a rotation, the maximum correlation for the translation will occur in the angle corresponding to $\theta = 90° - \theta_0$. Hence, the location of the correlation maxima over a range of values $\tilde{\theta}$ ranging from $-\theta_{max}$ to θ_{max} gives the estimate for the true rotation:

$$\theta_0 = \arg \max_{\tilde{\theta}} C(\tilde{\theta}) \tag{7}$$

where

$$C(\tilde{\theta}) = max(corr\{F^R(r, \theta = 90°), F^R(r, \theta = \tilde{\theta})\}) \tag{8}$$

Thus, in order to estimate a rotation between two images, it is required to compute 1D normalized correlations between the reference vector $F^R(r, \theta = 90°)$ and each of the different RT vectors corresponding to each of the rotation angles, a total of $\frac{2\theta_{max}}{\delta\theta}$ 1D correlations where $\delta\theta$ is the selected increment of the angle. The range of rotation values must be set to cover all the angles of the field of view of the sonar while the resolution $\delta\theta$ is a trade-off between accuracy and computation time.

3.5 Direct Polar Estimation

As explained in section 3.1, working with the magnitude of the FT isolates the rotation from the translation. However, the Fourier magnitude has an oscillatory nature which together with the conversion to polar coordinates results in the introduction of inaccuracies. Given that in many cases rotations will not be combined with translations, we would like to consider the option of estimating the rotation directly from the polar images.

In the first column of Fig. 3 we can see an example of a pair of polar sonar frames that have undergone a small translation. The second column, shows two frames that have been displaced by a much larger translation. The translations in the cartesian domain introduce a kind of sinusoidal distortion in the polar image. When translational displacements are relatively small compared to the image's size in each direction, the induced distortions in the polar image still allow for the recovery of the rotation by computing the translation in the angular direction.

Although this scheme for rotation estimation is prone to introducing inaccuracies in the presence of large translations, we still want to take it under consideration. The main reason is because the FLS delivers the polar images as raw data and therefore the rotation estimation would be performed on the original data without the need of computing any transformation or undergoing any interpolation. Furthermore, due to the high frame rate of new FL sonars, the translations of consecutive images will be small, allowing for a reasonably accurate angle estimation which can be adjusted later when incorporating the pair-wise registrations inside a global alignment framework. The major drawback would be in non-consecutive registration of frames undergoing large translational movements or in loop-closing situations when attempting to match temporally distant frames that present significant shifts.

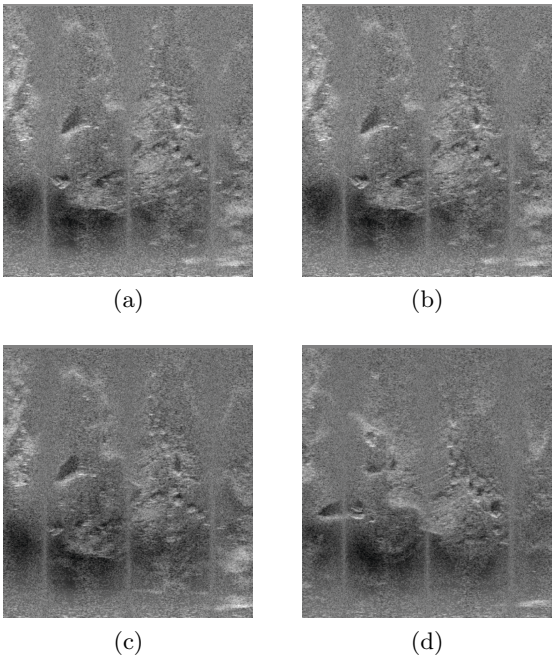


Fig. 3. Comparison of the effect of translations on polar sonar frames. The small translation between the images on the first column (a, c) does not induce severe distortions. The images on the second column (b, d) have been translated by a larger distance and the contents are distorted, making difficult to estimate rotations as a shift in the horizontal axes.

4 Experiments and Results

In this section the presented methodologies are evaluated with real sonar data. Two different situations are analyzed: when the sonar undertakes purely rotational movements, and when there are, simultaneously, rotational and translational displacements. In order to test the algorithms under varied conditions, we employ data sets gathered in different environments and using different sonar models with significant differences in field of view and resolution.

The Cartesian sonar images have been pre-processed to avoid edge effects by applying a mask to taper out the boundaries to zero. In the two methods that deal with images in the polar domain, those have been windowed with a cosine window. For the Radon Transform method, the angle resolution has been set to 0.25°, which is close to the angular resolution of the beams of the used sonars. The average runtime per registration under a MATLAB environment is reported in seconds, using a Intel Core2 Duo 3.4Mhz. From here on, we refer to the different methods as: PMFT (traditional phase-correlation with Polar Magnitude of the FT), GCC (Gradient Cross-Correlation method), RT (Radon Transform approach) and PCP (Phase-Correlation directly on the Polar image).

Table 1. Computation time, mean error and standard deviation of all methods for Dataset 1. Approximate overlap between consecutive registered frames: 90%.

Method	Time(s)	Mean Error(deg)	Std(deg)
PMFT	1.8	0.2183	0.2055
GCC	2.0	0.6449	0.6542
RT	5.4	0.0585	0.0616
PCP	0.09	0.0230	0.0369

4.1 Purely Rotational Movements

The first dataset consists in a 360° scan performed in a dock environment with a DIDSON sonar mounted on a mechanical tripod. A total of 1176 sonar frames were collected in a complete circle at every 0.3° degrees. Due to the high frame rate of acquisition we only use 1 out of every 3 frames, which corresponds to an overlap between consecutive images of around 90%. Then, ground truth rotations along consecutive frames of the data set correspond to a constant value of 0.9°. Table 1 summarizes the mean error and standard deviations for each technique (in degrees) with respect to the ground truth. As can be seen, the PCP method has the lowest error with respect to the ground truth angles. This is expected since under purely rotational movements the estimation of rotations can be directly found as a translation in the polar images coming from the sonar, without the need of altering the data and performing any interpolation. Radon Transform approach, performs also reasonably well and the PMFT and the GCC show slightly higher errors, with GCC arising few clear outliers that would be marked as unsuccessful registrations by the posterior computation of the Peak-to-Noise Ratio of the phase correlation matrix as explained in [7].

We can also analyze the trend of the errors when increasing the step between the images that we take into account. This is equivalent to registering frames that are increasingly more distant from each other, which effectively reduces the overlap. We have decreased the overlap until half of the field view of the sonar, since according to our experience is the maximum angle that we can detect reliably. By construction, the PCP and RT methods do not allow for the estimation of an angle difference higher than the field of view of the sonar. This limit becomes even more restricted if we take into account that a minimum overlap is required in order to establish the correlation and some information is lost with the boundary masking operations, resulting in a practical requirement of an overlap around 50%. Table 2 shows the errors computed for this case, where it can be seen that in general the errors increase, while still staying low for the PCP method. The high accuracy of the rotation estimation from PCP method allows us to generate a consistent mosaic (shown in Fig. 4) just by chaining the consecutive registration results, without the need of any global alignment. Notice though there is a small error alignment at the end as a consequence of the small error accumulation along the sequence.

The second data set used for evaluation was obtained during the ANT'11 sea trial organized by Centre for Maritime Research and Experimentation (CMRE),

Table 2. Computation time, mean error and standard deviation of all methods for Dataset 1. Approximate overlap between consecutive registered frames: 50%.

Method	Mean Error(deg)	Std(deg)
PMFT	2.1393	0.9472
GCC	1.242	1.0323
RT	0.6353	0.5647
PCP	0.351	0.0007

Fig. 4. Mosaic generated by registering with PCP the consecutive frames of Dataset 1

located in La Spezia, Italy, and during which the University of Girona collaborated. The data set was obtained using a BlueView P900-130 FLS mounted on CMRE's Autonomous Surface Vehicle. The employed setup allows us to have precise differential GPS data and heading from 2 GPS units which we use as ground truth. The data set contains 3724 sonar frames (from which we use 1 out of every 3) gathered in a marina environment following a lawn-mover trajectory (Fig. 5) composed by independent translational and rotational movements. Table 3 summarizes the corresponding mean errors and standard deviations.

Again, PCP offers the best results. It is worth highlighting that in the presence of small translations, such as the ones performed during the long straight tracklines of the trajectory, the method is still able to recover correct rotations (around 0°). However, if we pay attention to Fig. 6 it can be seen that the locations with major disagreement (although small) of the PCP with respect the ground truth, are located in the middle of the straight tracklines of the trajectory (around frames 200, 650 or 900), where the presence of slightly longer translations between frames tend to cause distortion in the polar images.

Table 3. Computation time, mean error and standard deviation of all methods for Dataset 2

Method	Time(s)	Mean Error(deg)	Std (deg)
PMFT	5.7	4.0719	5.3603
GCC	6.1	4.2454	5.3801
RT	25.7	2.5847	2.5691
PCP	0.17	1.9885	3.8212

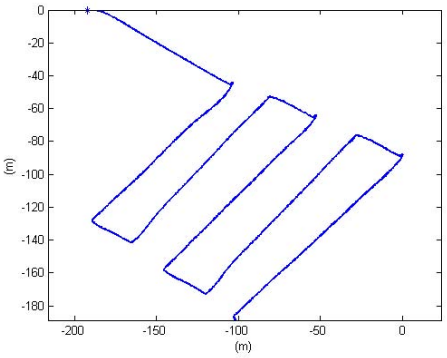


Fig. 5. Lawn-mover trajectory performed in the second dataset, including translations and pure rotations in every trackline change

Although the mean error does not appear to be much higher, PMFT and GCC tend to have an oscillating nature that does not follow the trend of the ground truth rotational angles. Fig. 6 illustrates as an example how the estimates of PCP closely follow the ground truth while PMFT estimates diverge. This fact might be caused by the low resolution and weak content of the images, which make it difficult to estimate small rotations accurately with the methods that work with the FT magnitude images. RT reports better results than PMFT and GCC but the size and the wide field of view of the images (130°) significatively increases the computational time.

We have computed the full registration estimates (translations and rotation with the 4 methods) between all overlapping frames. After discarding the unsuccessful registrations according to the Peak-to-Noise Ratio [7] we introduce them into a pose-base graph framework together with their corresponding uncertainties [7]. The importance of having good rotation estimates can be clearly seen from the resulting mosaics. Fig. 7a shows the mosaic rendered by using the PCP estimates, while Fig. 7b presents the obtained mosaic with PMFT. The errors accumulate along the trajectory and are difficult to compensate using global alignment due to the low presence of loop closures in this data set.

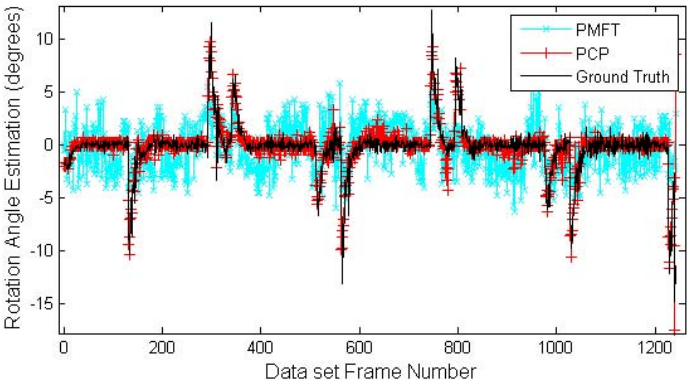
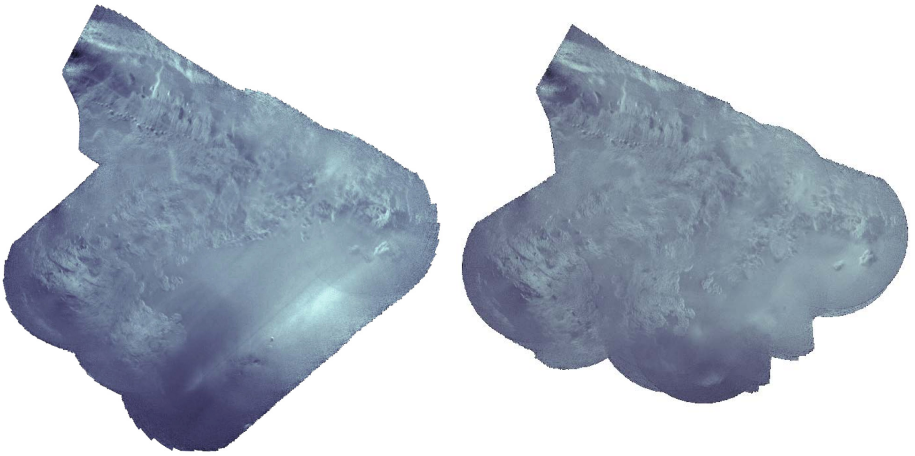


Fig. 6. Comparison between the rotation estimation of consecutive frames with PCP and PMFT



(a) (b)

Fig. 7. Final mosaics of the marina dataset. a) Rotations computed with PCP method. b) Rotations computed with the PMFT method.

4.2 Combined Rotation and Translation

The dataset used in this section was gathered with a BlueView P900-45 FLS installed in a ROV. The trajectory combines rotations and translational movements as can be observed in the navigation data (Fig. 8). Unfortunately ground-truth is not available but the navigation can provide us a good reference of the consecutive angle increments between every position. Table 4 summarizes the corresponding mean error and standard deviations with respect to the navigation data.

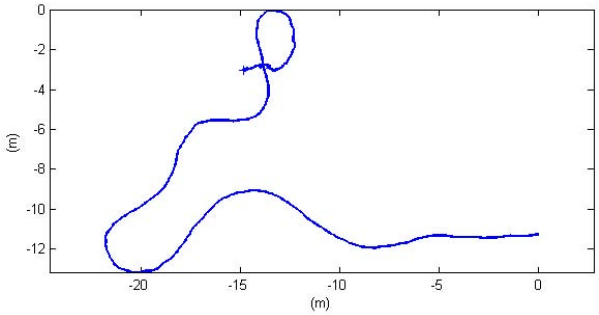


Fig. 8. Navigation trajectory of the third data set, combining rotational and translational movements

Table 4. Computation time, mean error and standard deviation of all methods for Dataset 3

Method	Time(s)	Mean Error(deg)	Std(deg)
PMFT	3.8	5.0199	6.379
GCC	4.2	5.229	6.023
RT	17.1	4.6481	5.7859
PCP	0.19	5.5074	7.1352

In general the errors are higher since the data presents fast rotations which decrease the overlap and reduce the reliability of the registrations. We can see that in this case PCP does not present the best accuracy and exhibits a relatively higher error than in the other two previous data sets. This is coherent with the fact that this data set has an underlying translation that leads to a distortion of the polar images subsequently yielding to a wrong rotation estimation. If we plot the estimates for each method (Fig. 9) we can see that both PCP and RT get close to the ground truth estimate along the trajectory (RT with more outliers), and PMFT and GCC present fluctuations that clearly vary from the ground truth (GCC estimate has not been plotted to avoid overloading the figure, but the trend is similar to the PMFT).

This data set has many potential loop closures as the trajectory and the range of the sonar allow for overlap of temporally distant frames. We consider as a loop closure a successful registration between frames that are not part of the sequential neighborhood of a current frame. According to that, PMFT encountered 2 loop closures, GCC 3, RT 10 and PCP 12.When performing the global alignment, a larger number of loop closures contribute to ensure the consistency of the mosaic. Hence, even that the PCP estimates might not very accurate in this context, a consistent mosaic can be obtained after a graph optimization of a sufficient number of links (Fig. 10).

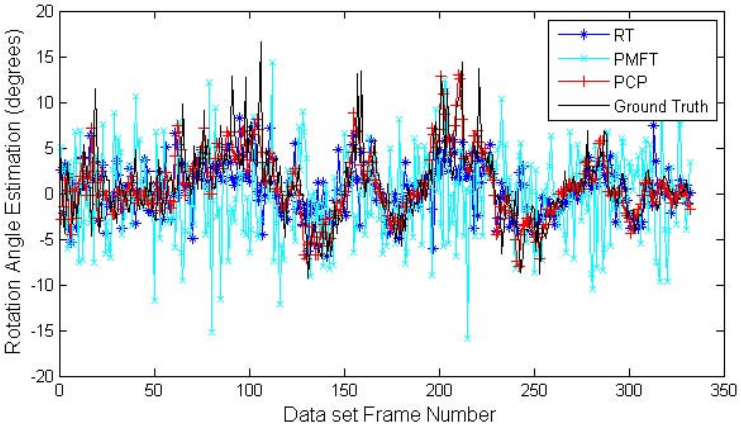


Fig. 9. Comparison between the rotation estimation of consecutive frames with PCP, RT and PMFT

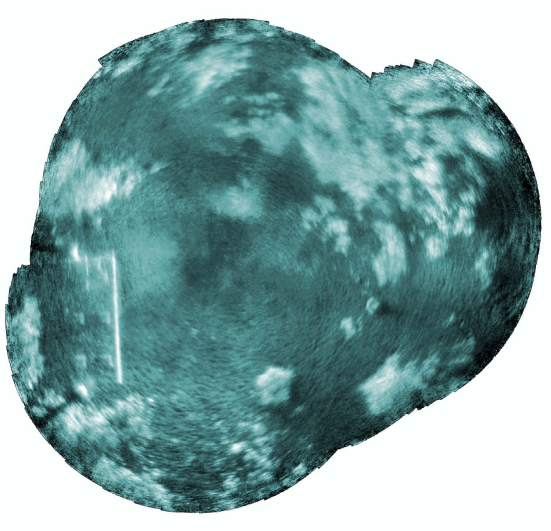


Fig. 10. Mosaic generated with PCP after global alignment

5 Discussion

Direct estimation of the rotation in the polar images, although being prone to inaccuracies in the presence of translations offers the best results, specially for those FLS with a narrow field of view, where the distortions induced by the translations are less noticeable. The estimations are accurate since no transformations and interpolations are involved and the presence of small shifts is handled well thus making the method appropriate in the majority of cases. In the presence of large translations, if the data set trajectory allows for loop closures,

the global alignment stage might constrain these errors, making the PCP a valid option for rotation estimation even in those cases. Otherwise, another method must be selected to deal with these particular situations. The Radon Transform approach offers also good performance in both purely rotational and combined motions but at the expenses of a high computational time. Resolution cannot be sacrificed past a certain point if we want to guarantee a good translation estimation and in general, good registration. Although it might be a good approach in the context of off-line mosaicing, it seems unfeasible for real-time implementations. Regarding the PMFT method, which has been successfully used on optical images, we can conclude that it performs poorly on FLS images. It should be noted that results obtained for the first data set, where the employed sonar had higher resolution compared to the other two, present less error. Also, the preliminary results under rotational movements reported on [7] are successful as they are based on a dataset with strong image components (distinguished and sharp straight lines). However, in general, the noise, low resolution and the lack of strong image features together with the interpolations errors introduced within its computation discourage the usage of PMFT within FLS modality. Finally, we have not noticed any advantage of GCC method over the normal PMFT, and in fact, it appears to give rise to a higher number of unsuccessful registrations. This suggests that the gradient images that are generated as input for the correlation are not stable enough (noise may arises small gradient changes) and although a more robust manner of computing them could be devised, the results obtained with the analogous PMFT method discourage its use.

6 Conclusions

We have assessed the performance of different global registration methods for rotation estimation on FLS images, under different situations and using different sonar models. Results show that rotation estimation performed through phase correlation in the raw sonar polar frames is the best approach. Its performance can decrease when the image rotation is combined with large translations and therefore other techniques may be considered in this situation. Radon Transform is a suitable candidate for those situations, especially when working off-line, and should be studied further to find ways of decreasing its computational burden. On the other hand, if there exists the combination of rotations and large translational movements it is important to plan trajectories that ensure loop closing situations to try to constrain the error. Further improvements could incorporate a selective methodology based on the context: when attempting registrations between frames that were gathered far in time, a special technique could be used, while for the rest of the cases a direct estimation of the rotation in the polar images is the best approach.

Therefore, we have made a step towards generalizing the applicability of sonar mapping into a broader range of situations, involving the generation of mosaics gathered under rotational movements which might be necessary in many environments or in novel applications like real-time sonar mosaicing to support ROV pilots.

Acknowledgments. This work has been supported by the FP7-ICT-2011-7 project PANDORA-Persistent Autonomy through Learning, Adaptation, Observation and Re-planning (Ref. 288273) funded by the European Commission and the Spanish Project ANDREA/RAIMON (Ref. CTM2011-29691-C02-02) funded by the Ministry of Science and Innovation. The authors would like to thank Sound Metrics Inc. and the Center of Maritime Research and Experimentation for providing sonar data.

References

1. Walter, M.R.: Sparse bayesian information filters for localization and mapping. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA (2008)
2. Hurtos, N., Cufi, X., Petillot, Y., Salvi, J.: Evaluation of registration methods on 2D Forward-Looking Sonar Imagery. In: IEEE/MTS OCEANS, Bergen (2013)
3. Johannsson, H., Kaess, M., Englot, B., Hover, F., Leonard, J.: Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4396–4403 (2010)
4. Negahdaripour, S., Firoozfam, P., Sabzmeydani, P.: On processing and registration of forward-scan acoustic video imagery. In: 2nd Canadian Conference Computer and Robot Vision, pp. 452–459 (2005)
5. Aykin, M., Negahdaripour, S.: On feature extraction and region matching for forward scan sonar imaging. In: IEEE OCEANS Hampton Road, pp. 1–9 (2012)
6. Negahdaripour, S.: On 3-D scene interpretation from F-S sonar imagery. In: IEEE OCEANS Hampton Road, pp. 1–9 (2012)
7. Hurtos, N., Cufi, X., Petillot, Y., Salvi, J.: Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5298–5305 (2012)
8. Chen, Q.S., Defrise, M., Deconinck, F.: Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(12), 1156–1168 (1994)
9. Reddy, B.S., Chatterji, B.N.: An FFT-based technique for translation, rotation, and scale-invariant image registration. IEEE Transactions on Image Processing 5(8), 1266–1271 (1996)
10. Tzimiropoulos, G., Argyriou, V., Zafeiriou, S., Stathaki, T.: Robust FFT-based scale-invariant image registration with image gradients. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(10), 1899–1906 (2010)
11. Lucchese, L., Cortelazzo, G.M.: A noise-robust frequency domain technique for estimating planar roto-translations. IEEE Transactions on Signal Processing 48(6), 1769–1786 (2000)
12. Keller, Y., Shkolnisky, Y., Averbuch, A.: The angular difference function and its application to image registration. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 969–976 (2005)
13. Li, L., Qu, Z., Zeng, Q., Meng, F.: A novel approach to image roto-translation estimation. In: IEEE Int. Conf. on Automation and Logistics, pp. 2612–2616 (2007)
14. Costello, C.: Multi-reference frame image registration for rotation,translation, and scale. Technical Report, DTIC Document (2008)

Part II

Control and Planning in Aerial Robotics

Local Heuristics Analysis in the Automatic Computation of Assembly Sequences for Building Structures with Multiple Aerial Robots

Alvaro Sempere, Domingo Llorente, Ivan Maza, and Aníbal Ollero

Grupo de Robótica, Visión y Control, Universidad de Sevilla
{sempere1987,guetell}@gmail.com, {imaza,aollero}@us.es

Abstract. This paper deals with the automatic computation of the assembly sequence for building truss structures from their 3D geometrical analysis. This functionality is part of the autonomous planning architecture of a team of aerial robots equipped with on-board robotic arms. The mission of the team is the construction of a structure in places where the access is difficult by conventional means. The assembly sequence is computed by applying the well known "assembly-by-disassembly" technique to the Non-Directional Blocking Graphs (NDBG) obtained from the geometrical analysis of the structure. In this paper two novel local heuristics are presented to solve the assembly problem: the former is based on the number of free nodes in the graphs and the latter is related to the size of the resulting connected subgraphs when each disconnection is applied to a set of parts. Both techniques are designed to compute the assembly sequence that allows to parallelize the building process of the structure if enough robots are available. Simulation results as well as experimental results with an aerial robot are presented in the paper.

Keywords: local heuristics, automated assembly, aerial manipulation, autonomous planning, NDBG graphs.

1 Introduction

The work described in this paper is part of the ARCAS European Project funded by the European Commission. One of the goals of this project is to build a structure by using a team of aerial robots equipped with on-board manipulators. The practical interest of this system can be found in situations where it is required to build a structure in places with difficult access by conventional means (see Fig. 1).

Other works [2,15,20,9] have addressed cooperation in teams of aerial robots for multi-purpose missions. However, in those papers the dexterous manipulation with aerial robots was not present. The use of aerial robots allows to perform assembly operations in any point of the 3D space, which can represent a relevant advantage compared to ground robots in areas of difficult access.

In previous work of the authors [16], the so-called TA technique was applied to compute assembly sequences using a global search algorithm. It was shown that

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_7,

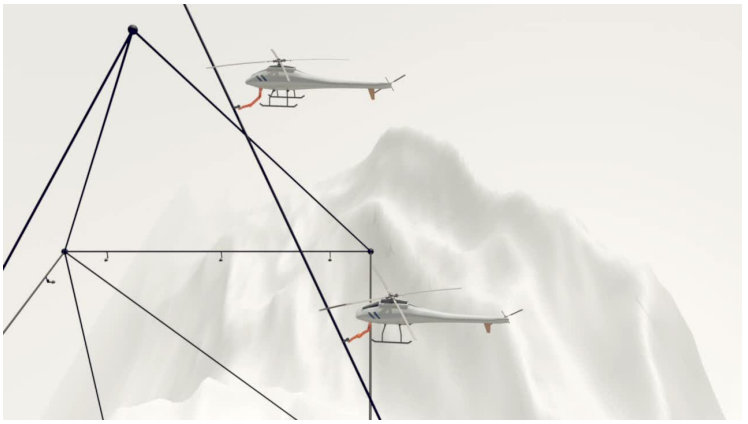


Fig. 1. Two aerial robots equipped with LWR KUKA robotic arms manipulating a bar to build a structure in places with difficult access by conventional means

pre-computing groups of disassembly operations, the depth of the search tree was reduced allowing to decrease the required computational time. In this work, local search is added to the previous approach in order to decrease more the runtime by pruning the tree according to the chosen heuristics in each iteration. These local heuristics are based on those characteristics of the intersection graphs that are related to the potential parallelization of the assembly operations if enough aerial robots are available.

It should be clarified that characteristics such as the number of available robots, the weight of the parts and the obstacles in the environment are not taken into account in this work. Only the geometry of the structure is considered and it is assumed that enough robots are available to exploit the potential parallelization of tasks in the construction process.

The paper is organized as follows. Next section summarizes related work addressing the geometric analysis of the structure, the representation of the assembly sequences and the methods applied for their computation. Section 3 describes the architecture and the algorithms applied to solve the assembly problem, and the local heuristics mentioned above are presented in Sect. 3.2. The simulation results of the different strategies are analyzed in Sect. 4 and the whole approach is illustrated with a real experiment performed with a multi-copter in the facilities of the Center for Advanced Aerospace Technologies (CATEC) located in Spain. Finally, Section 6 closes the paper with the conclusions.

2 Related Work

The particular problem addressed in this work is to compute the sequence of Assembly Operations (AO) required to build a given structure. Reference [11] presents a classification of the structures according to different features: number of hands, monotonicity(whether or not operations of intermediate placement of

subassemblies are required), linearity (whether all assembly operations involve the insertion of a single part in the rest of the assembly or more than one part have to be simultaneously inserted), and coherence (whether or not each part that is inserted will touch some other previously placed part). The structures considered in this paper are sequential (for two robots), monotone, linear and contact-coherent.

Related work is divided according to different areas of research: geometric analysis and problem representation, and the algorithms to compute the assembly sequences.

2.1 Geometric Analysis and Problem Representation

In this area, several authors have introduced ideas such as the *graph of connections*, whose nodes are these individual parts and the links stand for contact connections between them. In [4] a classification of the different types of connections is provided, and in [18,6] the graphs of connections are represented as liaison matrices. There are also more particular representations [21] where predicates are used to specify the relative positions of the different parts (part 1 on top of 2, part 2 at $+x$ with respect to part 5, etc.). Relational graphs have been applied to analyze the distances between parts from their vertices and/or their edges [12] and it has been also used the *configuration space* or *C-space* in [7,8] to define a geometric area which represents the contact-free region for every two parts. In particular, the Minkowski difference $P \ominus Q$ is applied in [7] to determine the free obstacles region between two polygons P and Q.

The use of AND/OR graphs [10] has been widely applied in assembly sequence problems. In this case, the approaches start from a previous analysis of the structure geometry and represents the different possible combinations of assembly operations as an AND/OR graph. The AND branches represent the disassembly of a part (or a subset of parts), whereas the OR branches (different paths from the same node) represent different disassembly options. In a later stage, it is possible to analyze the feasibility of the sequences using knowledge databases [4], geometrical approaches [18], ad-hoc algorithms [23] that differentiate parts and connectors, or building *Non-Directional Blocking Graphs* [22]. Finally, an algorithm based on the so-called *motion space* is presented in [7]. Different sequences of translation paths are computed within the *C-space* in order to build the structure.

2.2 Methods to Compute the Assembly Sequences

A widely used method is called *assembly-by-disassembly*. Starting from the initial structure, the algorithm computes the parts that can be disassembled and updates the structure after removing them. The process is repeated until no remaining parts are left. This technique can be applied with the previously mentioned *Non-Directional Blocking Graphs* representation that includes the blocking directions between the different parts of the structure.

There are other algorithms based on different techniques: ant colonies [21,19,6], neural networks [17], evolutionary algorithms [5,13] and Immune Optimization Approaches(IOA) [3]. Alternatively to perform a graph search, reference [14] proposes to transform the liaisons graph into a table of liaisons in matrix form where a feasible sequence can be found by successively deleting the columns of the parts already included in the assembly and examining their rows for other candidate liaisons to be established.

All these algorithms can be setup just to compute feasible sequences or to optimize the operations according to a given criteria: number of required reorientation of parts [21], required tools, number of parts moved, etc. For instance, in [1] the authors apply Petri Nets to solve the problem minimizing the degree of difficult of assembly sequences.

3 Adopted Approach

The architecture adopted to solve the problem was also applied in [16] and it is shown in Fig. 2. It is based on the following modules: NDBG generator, *assembly-by-disassembly* technique, heuristics based prune, sequence selection based on metrics and final assembly sequence computation.

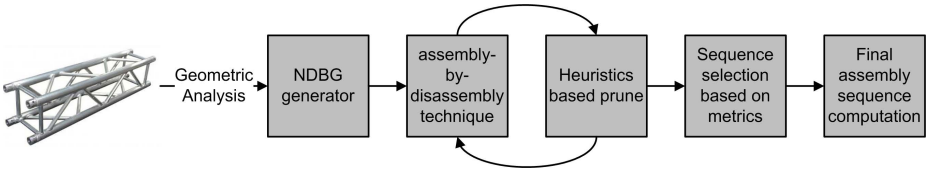


Fig. 2. Architecture adopted to compute the assembly sequence required to build the structure given as input. The different modules are described in this section.

3.1 NDBG Generator

Let us consider a structure S composed by a set of parts P_i. Constant mutual blocking relationships between parts of the structure can be labeled with *Directional Blocking Graphs* (DBG). A DBG $g_S(\boldsymbol{t})$ for the structure S in a direction $\boldsymbol{t}$ is a directed graph $G = (V, E)$ with

$$\begin{aligned} V &\neq 0 \\ E &\subseteq \{(a,b) \in V \times V : a \neq b\} \ , \end{aligned} \tag{1}$$

where V is the set of vertices, (a, b) is an edge with source a and sink b and any node $n_i \in V$ represents a part P_i of the structure S. In the graph $g_S(\boldsymbol{t})$, an arc points from P_i to P_j iff P_j blocks the displacement of P_i along the particular direction given by vector $\boldsymbol{t}$.

Let us define $w = (n_1, n_2, ..., n_k)$ as a path in the graph from node n_1 to n_k. A *strongly connected subset* of $g_S(\boldsymbol{v})$ is the maximal subset of nodes such that

$$\begin{aligned} \forall n_j \in g_S(\boldsymbol{v}) \wedge \forall n_k \in g_S(\boldsymbol{v}) \rightarrow \\ \exists w = (n_j, n_{j+1}, ..., n_{k-1}, n_k) \wedge \\ (n_{j+1}, ..., n_{k-1}) \in g_S(\boldsymbol{v}) \end{aligned} \tag{2}$$

i.e. for every pair of nodes $(n_1, n_2) \in g_S(\boldsymbol{v})$, there is a path from n_1 to n_2 with all the nodes belonging to $g_S(\boldsymbol{v})$. Then, the *strongly connected subsets* of a graph $g_S(\boldsymbol{v})$ represents individual parts (or subsets of parts) that can be disassembled from S.

Let $d_{\boldsymbol{t}}$ be a disassembly operation containing the list of parts that can be disconnected from the structure in a direction $\boldsymbol{t}$, and D the set including all the possible disassembly operations. A set of n parts belongs to D if

$$\{n|(n, p_i)/\forall p_i \in S \wedge (n, p_i) \notin g_S(\boldsymbol{t_j})\} \in d_{\boldsymbol{t_j}} \ . \tag{3}$$

Figure 3 shows the different DBGs $g_S(\boldsymbol{v})$ that belongs to the set G_S for a given 2D structure. Each graph corresponds to a particular bidimensional direction of dissasembly $\boldsymbol{v} \in S^2$. As the structure is 2D, the different angles associated to the directions are represented in a circle S^2. The partition of the circle of directions labeled with the corresponding DBGs is called the Non-Directional Blocking

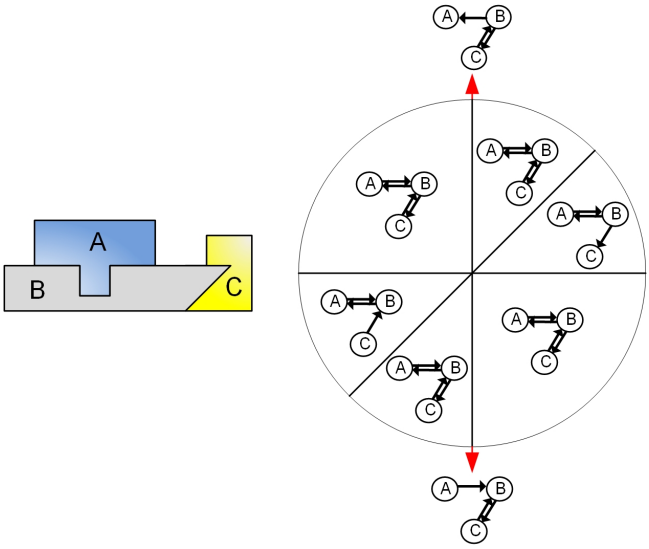


Fig. 3. This is a representation of the full set G_S for the structure shown on the left. On the right, the different graphs $g_S(\boldsymbol{v})$ with $\boldsymbol{v} \in S^2$ are grouped in different circular sectors depending on the translation direction. The graphs on the top and the bottom of the circle correspond to the vertical direction (upwards and downwards respectively).

Graph (NDBG). Then, Figure 3 displays the NDBG for an assembly in a 2D infinitesimal translational case.

As it will be shown later, the disassembly process is iterative until all the parts of the structure are disconnected, i.e. $\forall n_i$ and $\forall g_S(\boldsymbol{t_j})$ with $n_i \in g_S(\boldsymbol{t_j})$, no edges are connected to n_i.

Other authors have considered rotations [22] and multiple translations [8] in the building process. However, in this work in order to follow the graph approach presented above and expressions such as (3), only one-step translations are considered in the process. On the other hand, only the three Cartesian axes have been considered for the translations in the NDBG graphs generation: $\pm x, \pm y, \pm z$. Finally, it should be mentioned that the assembly sequence is computed from the disassembly sequence just reversing the order of the operations.

3.2 Assembly-by-Disassembly Technique

This technique processes the previous G_S in order to find paths of disassembly operations that can disassemble the whole structure. Once a particular sequence of disassembly operations has been chosen, it is just required to perform the operations in reverse order to obtain the assembly plan.

In order to find the different paths of disassembly operations in G_S, Algorithm 1 has been applied. This algorithm generates a list with all the posible paths for a given local heuristic. The different heuristics are described later in Sect. 3.3.

$G_S \longleftarrow$ Build intersections graph ;
$\Sigma \longleftarrow$ Initialize list as empty ;
while *remaining parts connected in* G_S **do**
  $\Delta \longleftarrow$ getDissasemblySet(G_S, H_i);
  update G_S;
  Add Δ to Σ ;
end

Algorithm 1: Core algorithm used to implement the assembly-by-disassembly technique. Σ is a list with all the posible disassemblies Δ applying a given heuristic H_i.

The algorithm stores in Σ all the posible disassemblies Δ applying a given heuristic H_i. The heuristics prune the computation of sequences according to the indices q_{H_i} described in Sect. 3.3. The sequences with the highest values of q_{H_i} are returned to be further processed by the next module of the architecture.

In previous work of the authors [16], it was shown that pre-computing groups of disassembly operations, the depth of the search tree was reduced allowing to decrease the required computational time. That technique has been also used in the method `getDissasemblySet` applied to compute Δ in Algorithm 1.

3.3 Heuristics Based Pruning

Two different heuristics have been applied in the implementation of Algorithm 1: the number of free nodes in the graph (H_F) and the features of the connected subgraphs (H_C). Both heuristics are described in the following.

Number of Free Nodes in the Graph (H_F). Let us define $v(n)$ as the degree or valence of a node n in a graph, i.e. the number of incident edges in a vertex. For a node n in directed graphs, the number of head endpoints adjacent to a node is called the indegree $v^-(n)$ of the node and the number of tail endpoints adjacent to a node is its outdegree $v^+(n)$. In every graph

$$v(n) = v^-(n) + v^+(n) \; . \tag{4}$$

On the other hand, let us denote q_{H_F} as the number of parts free of intersection in the structure, i.e. the nodes n_i such that $g^+(n_i) = 0$ in each graph $g_S(\boldsymbol{v_i})$. It is possible to select those disassemblies leaving the maximum number of free parts for the next iteration in the algorithm previously described. Then, given the j-th node $n_j \in g_S$, the heuristic q_{H_F} is computed in each iteration of Algorithm 1 for a particular disassembly Δ as

$$q_{H_F}(\Delta) = \sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_{ij} \; ; \; \alpha_{ij} = \begin{cases} 1 \text{ if } g^+(n_j) = 0 \text{ where } n_j \in g_S(\boldsymbol{v_i}) \\ 0 \qquad\qquad\qquad\quad \text{otherwise} \end{cases} \; , \tag{5}$$

where $M = |G_S|$ and $N = |g_S(\boldsymbol{v_i})|$. The computation is performed for each g_S, and hence, single parts (nodes n such that $g(n) = 0$) will increase q_{H_F} in a quantity $+|G_S|$, i.e. the total number of g_S graphs. Then, this heuristic prioritizes single parts disassemblies that lead to very linear disassembly sequences. This behaviour can be reduced adding one unit to q_{H_F} if a node is free of intersections in all the graphs g_S.

In order to illustrate the idea behind this heuristic, the structure in Fig. 4 will be used as reference. Figure 5 shows two possible disassemblies (d_1 and d_2): after d_1 there are no additional parts free of intersections, whereas with d_2 there are five free parts. Then, the second disassembly leads to a higher value of q_{H_F}.

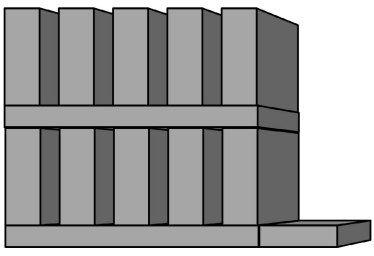


Fig. 4. Basic structure used to illustrate the ideas behind the different local heuristics

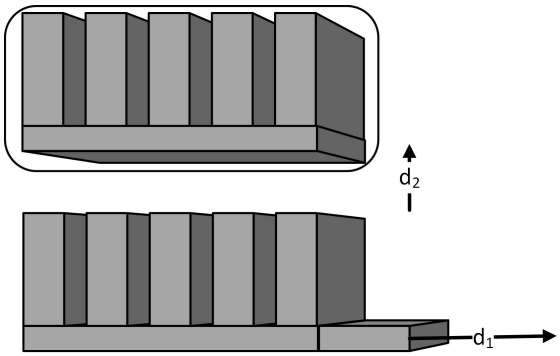


Fig. 5. Two possible disassemblies (d_1 and d_2) for the structure shown in Fig. 4

Connected Subgraphs (H_C). A connected subgraph $c_S(\boldsymbol{v})$ is defined as a subset of connected nodes with a minimum degree equal to one, i.e. $g(n) = 1$. In addition, $c_S(\boldsymbol{v}) \in g_S(\boldsymbol{v})$ and for every two nodes n_1 and n_2 belonging to different $c_S(\boldsymbol{v})$, there is no path connecting both. Each node n with $g(n) = 0$ is a connected subgraph with no edges.

This heuristic is based in the idea of prioritizing disassemblies that divide the graphs g_S in the maximum number of connected subgraphs with a similar number of nodes. The idea is to divide the structure S in many substructures that can be disassembled in parallel. Let us define $\max(c_S(\boldsymbol{v_i}))$ and $\min(c_S(\boldsymbol{v_i}))$ as the connected subgraphs with the maximum and minimum number of nodes from $g_S(\boldsymbol{v_i})$ respectively. The heuristic q_{H_C} is computed in each iteration of Algorithm 1 for a particular disassembly Δ as

$$q_{H_C}(\Delta) = \sum_{i=1}^{M} \frac{1}{\beta_i} \; ; \; \beta_i = \begin{cases} 1 & \text{if } r_{\max} = r_{\min} \\ r_{\max} - r_{\min} & \text{otherwise} \end{cases} , \tag{6}$$

where $M = |G_S|$, $r_{\max} = |\max(c_S(\boldsymbol{v_i}))|$ and $r_{\min} = |\min(c_S(\boldsymbol{v_i}))|$.

The structure in Fig. 4 will be used again to illustrate the ideas behind this heuristic. Figure 6 shows all the possible disassembly operations. It can be seen that the disassemblies from d_3 to d_7 on the top will lead to low values of q_{H_C} because the graphs g_s are divided in subgraphs with a big difference in the number of nodes. The same applies for the disassembly d_1 on the right. But the disassembly d_2 will generate a new graph with two connected subgraphs of 6 (moved parts) and 7 (static parts) nodes.

Heuristics Normalization. In order to combine different heuristics in the Algorithm 1, it is required to normalize them as

$$\text{norm}_{H_F} = \frac{|V_L|}{|V|} \in [0, 1] \tag{7}$$

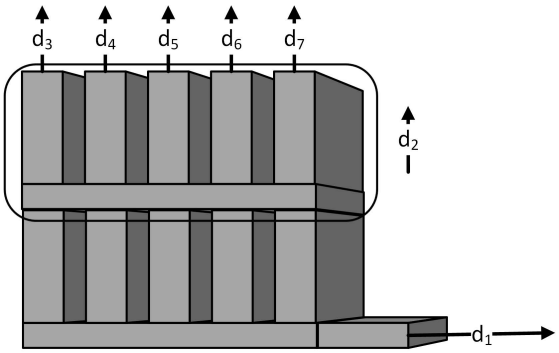


Fig. 6. All the possible disassemblies for the structure shown in Fig. 4.

and

$$\text{norm}_{H_C} = \frac{q_{H_C}(\Delta)}{|G_S|} \in [0, 1] \ , \tag{8}$$

where V_L is the set of free nodes in G_S and V is the set with all the nodes.

3.4 Sequence Selection Based on Metrics

After Algorithm 1 finishes, the structure is fully disassembled and the computed disassembly sequences are ranked according to two metrics: the number of required movements (I_m) and the degree of potential parallelization of the building process (I_p). Both are computed using counters associated to the parts of the structure. After ranking the sequences, it is possible to filter the results of the previous module by using a given threshold.

3.5 Final Assembly Sequence Computation

In the last stage of the process, the assembly sequences are computed just reversing the order of the disassembly operations and are represented using the Planning Domain Definition Language (PDDL). Each assembly operation (AO) has associated different features in this format: preconditions with the required previously assembled parts, effects with the pose of the parts after assembling them, translation vectors, type of connections, etc.

4 Analysis of the Proposed Approach

Apart from the previously presented heuristics, another heuristic H_M has been simulated for comparison purposes. This heuristic is based on the metric I_m presented in Sect. 3.4 and q_{H_M} is computed for a particular disassembly Δ as

$$q_{H_M}(\Delta) = \frac{1}{I_{m_\Delta}} \ . \tag{9}$$

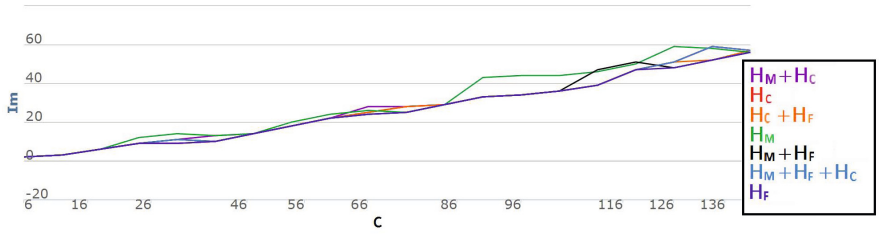


Fig. 7. The lowest values of I_m are shown for the different simulated heuristics H_M, H_F and H_C, which are based respectively on the number of required movements, the number of free nodes and the characteristics of the resulting connected subgraphs. The symbol + represents the combination of different normalized heuristics.

In order to compare the different approaches, a set of structures has been used and ranked with a complexity metric. This complexity C has been defined for a structure as the sum of the number of edges of all its associated graphs $g_S(\boldsymbol{d})$. This metric is shown in the horizontal axis of the plots presented in this section.

Figure 7 is focused on structures with complexity above 85 ($C > 85$) and shows the values of I_m for the different simulated strategies. It can be seen that the lower values of I_m are achieved with the heuristic based on the number of free nodes (H_F).

On the other hand, Figure 8 compares the heuristic H_F with the so-called TA technique presented in [16] that used only global search. As local search was not applied to prune the search tree, the spectrum of assembly sequences is broader and the solutions are better (lower I_m). However, the computational load is higher for the TA algorithm compared to the heuristic H_F. This is an important drawback because the runtime grows exponentially with the complexity of the structure.

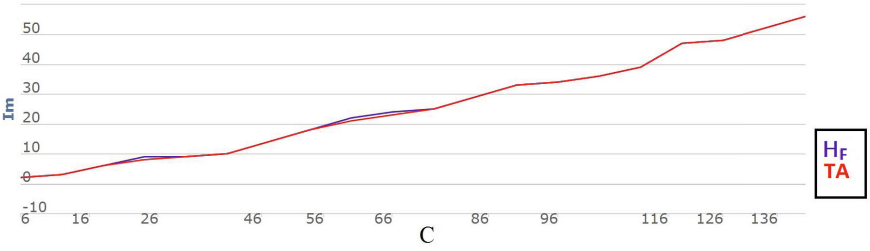


Fig. 8. The lowest values of I_m are shown for the best heuristic in Fig. 7 and the so-called TA technique presented in [16]

5 Experimental Application

The structure shown in Fig. 9 has been used in an experimental setup to test the approach presented in this paper to compute the assembly sequence.

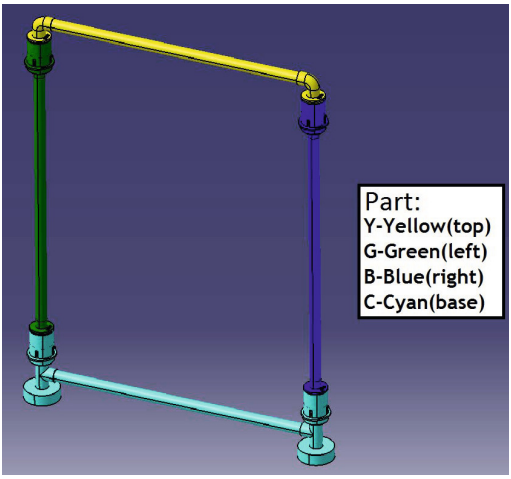


Fig. 9. Structure in the experimental setup used to illustrate the building capabilities of aerial robots equipped with manipulators in the framework of the ARCAS European Project. The CATIA file with the design of the structure is the starting point for the computation of the assembly sequence.

The NDBG generator module computes automatically the set G_S shown in Fig. 10 from the structure designed in CATIA. Then, applying the sequencing Algorithm 1 and the heuristics described in Sect. 3.3, two disassembly sequences are computed and ranked according to the metrics presented in Sect. 3.4.

Table 1 shows the runtime T_c and the values of the metric I_m for the different heuristics and combination of heuristics. It can be seen that for this particular structure the values are very similar in general.

The selected disassembly sequence was composed by the following disassembly operations:

1. Part Y disassembled from G and B towards $+z$.
2. Part G disassembled from C towards $+z$.
3. Part B disassembled from C towards $+z$.
4. Part C as base.

and the final plan for the execution is computed reversing the disassembly operations:

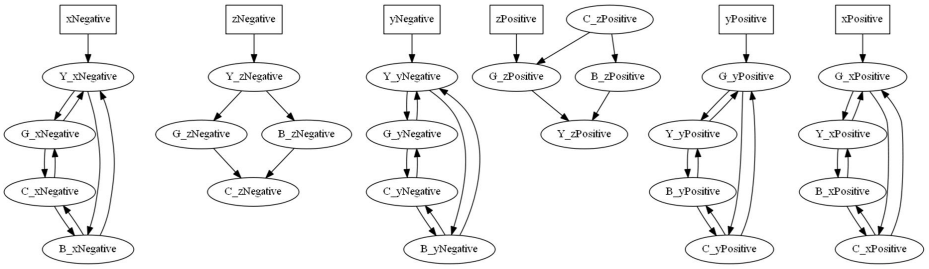


Fig. 10. Set G_S computed for the structure shown in Fig. 9 using the Cartesian directions as translation vectors

Table 1. Runtime T_c and values of the metric I_m for the different heuristics and combination of heuristics applied to the structure shown in Fig. 9

Heuristics	T_c (s)	I_m
H_F	0.42	7
H_C	0.38	7
$H_F + H_C$	0.45	7
H_M	0.46	7
$H_M + H_F$	0.40	7
$H_M + H_C$	0.40	7
$H_M + H_F + H_C$	0.40	7

Fig. 11. Aerial robot executing the assembly operations computed with the approach presented in this paper. A video with the full assembly mission can be downloaded from http://www.youtube.com/watch?v=Id4hwnbPI4U. The test was carried out in the facilities of the Center for Advanced Aerospace Technologies (CATEC) in the framework of the ARCAS European Project.

1. Part C as base.
2. Part B assembled to C towards $-z$.
3. Pieza G assembled to C towards $-z$.
4. Pieza Y assembled to G and B towards $-z$.

This plan was executed with an aerial robot of type multicopter equipped with a manipulator in the facilities of the Center for Advanced Aerospace Technologies (CATEC) located in Spain. The test was carried out in the framework of the ARCAS European Project. Figure 11 shows a photograph of the autonomous execution of the mission.

6 Conclusions

In this paper it has been shown that the local heuristics offer lower computational cost compared to the so-called TA technique presented in [16] that used only global search. However, local search prunes the tree removing in each iteration the disassemblies Δ according to the chosen heuristics. An interesting alternative would be to apply a threshold μ to control the pruning process, so that if μ is equal to zero, the disassemblies are not discarded.

It should be highlighted that the approach presented in the paper has been experimentally validated with an aerial robot equipped with a manipulator building a structure. The robot executed the assembly operations computed by our method in the facilities of the Center for Advanced Aerospace Technologies (CATEC) in the framework of the ARCAS European Project.

Finally, it should be mentioned that the algorithm can be extended adding other heuristics or additional methods oriented to decrease the computational load. Another interesting heuristic could be to prioritize the disassemblies Δ such that in the following iteration all the parts can be disconnected. Let us consider V_L as the set of free nodes, (V, E) as the set of nodes and edges of a graph, and $C_S(\boldsymbol{v_i})$ as the set of connected subgraphs in a particular $g_S(\boldsymbol{v_i})$. Then according to (5)

$$\forall g_S(\boldsymbol{v_i}) = (V, E) \rightarrow |V - V_L| = |C_S(\boldsymbol{v_i})| \ , \tag{10}$$

i.e. removing the nodes free of intersections with $g^+(n) = 0$ from all the graphs $g_S(\boldsymbol{v_i})$, a connected subgraph is generated per each part (all the parts are disconnected). This heuristic could lead to sequences with disassemblies that can be easily disconnected from the structure.

Acknowledgements. This work has been partially funded by the ARCAS European Project (ICT-2011-287617) and the national projects RANCOM (P11-TIC-7066) and CLEAR (DPI2011-28937-C02-01).

References

1. Ben-Arieh, D., Kumar, R.R., Tiwari, M.K.: Analysis of assembly operations' difficulty using enhanced expert high-level colored fuzzy petri net model. Robotics and Computer-Integrated Manufacturing 20(5), 385–403 (2004)
2. Bernard, M., Kondak, K., Maza, I., Ollero, A.: Autonomous transportation and deployment with aerial robots for search and rescue missions. Journal of Field Robotics 28(6), 914–931 (2011)

3. Cao, P.-B., Xiao, R.-B.: Assembly planning using a novel immune approach. International Journal of Advanced Manufacturing Technology 31(7-8), 770–782 (2007)
4. Choi, C.K., Zha, X.F., Ng, T.L., Lau, W.S.: On the automatic generation of product assembly sequences. International Journal of Production Research 36(3), 617–633 (1998)
5. Dao, S.D., Marian, R.M.: Genetic algorithms for integrated optimisation of precedence-constrained production sequencing and scheduling. In: Ao, S.-I., Gelman, L. (eds.) Electrical Engineering and Intelligent Systems. LNEE, vol. 130, pp. 65–80. Springer, Heidelberg (2013)
6. Guo, J., Wang, P., Cui, N.: Adaptive ant colony algorithm for on-orbit assembly planning. In: 2nd IEEE Conference on Industrial Electronics and Applications, ICIEA 2007, pp. 1590–1593 (2007)
7. Halperin, D., Latombe, J.-C., Wilson, R.H.: A general framework for assembly planning: The motion space approach. Algorithmica (New York) 26(3-4), 577–601 (2000)
8. Halperin, D., Wilson, R.H.: Assembly partitioning along simple paths: The case of multiple translations. Advanced Robotics 11(2), 127–145 (1997)
9. Heredia, G., Caballero, F., Maza, I., Merino, L., Viguria, A., Ollero, A.: Multi-unmanned aerial vehicle (UAV) cooperative fault detection employing differential global positioning (DGPS), inertial and vision sensors. Sensors 9(9), 7566–7579 (2009)
10. Homem de Mello, L.S., Sanderson, A.C.: AND/OR graph representation of assembly plans. IEEE Transactions on Robotics and Automation 6(2), 188–199 (1990)
11. Jimenez, P.: Survey on assembly sequencing: a combinatorial and geometrical perspective. Journal of Intelligent Manufacturing, 1–16 (2011)
12. Latombe, J.-C., Wilson, R.H., Cazals, F.: Assembly sequencing with toleranced parts. CAD Computer Aided Design 29(2), 159–174 (1997)
13. Marian, R.M., Luong, L.H.S., Abhary, K.: Assembly sequence planning and optimisation using genetic algorithms part i. automatic generation of feasible assembly sequences. Applied Soft Computing Journal 2(3), 223–253 (2003)
14. Marian, R.M., Luong, L.H.S., Abhary, K.: A genetic algorithm for the optimisation of assembly sequences. Computers and Industrial Engineering 50(4), 503–527 (2006)
15. Maza, I., Caballero, F., Capitan, J., Martinez de Dios, J.R., Ollero, A.: A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities. Journal of Field Robotics 28(3), 303–328 (2011)
16. Sempere, A., Maza, I., Ollero, A.: Cálculo automático de secuencias de ensamblado basado en una técnica de agrupación para la construcción de estructuras mediante equipos de robots. In: Actas de las XXXIV Jornadas de Automatica, Terrassa, Spain, pp. 703–710 (September 2013)
17. Sinanoglu, C., Borklu, H.R.: An assembly sequence-planning system for mechanical parts using neural network. Assembly Automation 25(1), 38–52 (2005)
18. Sinanoglu, C., Borklu, H.R.: An approach to determine geometric feasibility to assembly states by intersection matrices in assembly sequence planning. Journal of Intelligent Manufacturing 15(4), 543–559 (2004)
19. Tseng, H.-E.: An improved ant colony system for assembly sequence planning based on connector concept. In: Hu, W. (ed.) Electronics and Signal Processing. LNEE, vol. 97, pp. 881–888. Springer, Heidelberg (2011)
20. Viguria, A., Maza, I., Ollero, A.: Distributed service-based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions. Advanced Robotics 24(1-2), 1–23 (2010)

21. Wang, J.F., Liu, J.H., Zhong, Y.F.: A novel ant colony algorithm for assembly sequence planning. International Journal of Advanced Manufacturing Technology 25(11-12), 1137–1143 (2005)
22. Wilson, R.H., Latombe, J.-C.: Geometric reasoning about mechanical assembly. Artificial Intelligence 71(2), 371–396 (1994)
23. Yin, Z., Ding, H., Li, H., Xiong, Y.: A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. CAD Computer Aided Design 35(1), 37–56 (2003)

Lift Failure Detection and Management System for Quadrotors

J.J. Roldan, David Sanz, Jaime del Cerro, and Antonio Barrientos

Center for Automation and Robotics UPM-CSIC
C/ Jose Gutierrez Abascal, 2
28006 Madrid, Spain

Abstract. In the last decade, the use of small autonomous unmanned aerial vehicles (UAVs) has been generalized. Specially, multirotor vehicles become really popular and are being employed in several different applications and fields: military missions, agricultural processes, rescue and surveillance operations, audiovisual productions.. and even have arrived to the public at large. This entertainment approach has set some challenges still unresolved: among them, the safety of these systems.

The work presented in this paper is framed in this challenge, focusing on the multirotors systems behaviour when facing a lift problem. It has been tried to detect and manage emergency situations, being the goal to minimize as much as possible the potential damages. It started from a deep study of the main malfunctions or breakdowns affecting to the drone's lift. This study flowed into the design and develop of a model-based algorithm able to detect this events in a fast and robust way. This failure monitor allows to effectively apply control techniques to compensate (if it is possible) the breakdown's effects and to use passive safety methods (e.g. parachutes) to minimize the potential damages derived from a fall.

The system and techniques developed in this work were tested both in simulation and in real experiments, proving they suppose and increment in the operation's safeness (i.e. both the own drone and the third party agents, safety methods).

Keywords: Quadrotor safety, failure detection, degraded flight control system and pasive security methods.

1 Introduction and Objectives

Aerial robotics provides with many advantages when comparing to ground robotics: it allows to access to unreachable areas, to cover large terrains or to easily change their attitude. Nevertheless, all these advantages have also their return. Among them, the safety problems are really remarkable since the consequences are in general more critical than in the ground cases. Besides, the rotatory-wing UAVs has a notable predisposition to suffer accidents. It is not only an statistical fact but also a physical certain: compared to airplanes or other flying devices, rotorcrafts have not own lift ability. It is completely dependent

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_8,

from its propellers' thrust, implying instability and problems in case of failure of one of them. Even more considering the motor distribution in a quadrotor, where a breakdown in any engine will cause a uncontrolled fall.

Apart from the economic looses and legal responsibilities, these events could actually suppose a risk for the humans health (or even life), and this is completely unacceptable. Besides, from the popularization of the drones, the frequency of these incidents has been increased, being more common than may be though.

Without advising against the UAV utilization (due to their great benefits), the regulation of their use is necessary and the implementation of failure tolerant systems is required. Many authors have addressed these challenge in literature, but in general focusing on helicopters: Ducard summarizes most the fault-tolerant methods, basin all of them in statistical reliability [2]. It includes the approach that Rago proposed, where state-space model representations of the failures are used, computing the probability of each failure model [7]. A deeper approach to the lift problem in quadrotors were done by F. Sharifi [8] and A.Freddi [3]: They both use a model and the system's status for generating the "residuals" -as a difference of the predicted and real outputs- that will be use as indicators of the state. Concretely, the first work one uses an state estimator for detecting actuator faults, while the second approach uses model based on a Thau's observer. Nevertheless, they both use complex models that do not allow to apply these approaches in a real time control.

This paper presents the work done in this sense, simplifying the model and continuing the process by proposing a damage-reduction mechanisms that can be really implemented: The section 2 shows the analysis done related to common breakdown causes, leading into the failure detection system. It embraces the quadrotor model and the state machine of the detection system. Section 3 continues with the logic evolution: after the detection, the management system for minimizing the potential damage is presented. Firstly, the control system for recoverable situations is presented in 3.1. It allows to balance the drone and take off in safety conditions. In contrast, the subsection 3.2 presents alternatives to minimize the damages in case of unrecoverable situations. Finally, section 4 presents the results of the experiments and the conclusions extracted.

2 Failure Detection

As described in the previous section, the firs step for a reliable failure managing and control bases on the capacity of detecting the failure before getting an irreversible situation. This analysis is based on a simplified quadrotor model that allows to detect the most common failures affecting the drone's lift by analyzing its dynamic behaviour.

2.1 Quadrotor Model

A quadrotor (also know as quad-rotor) is a rotary-wing UAV that relies in the propulsion provided by four propellers (as the presented in figure 1). It provides

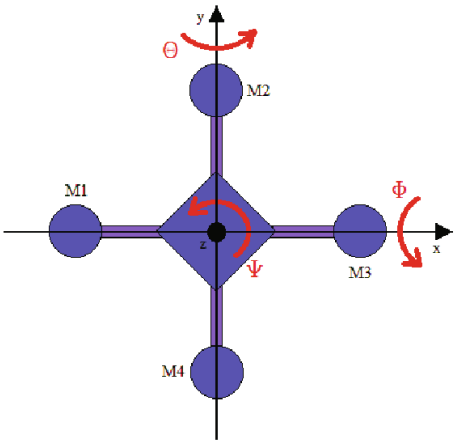


Fig. 1. Quadrotor model

a great dynamic stability and maneuver capacity. Nevertheless, this advantage makes also really unstable the UAV is case of failure.

The kinematic/dynamic model used for analyzing both states is a simplification of the one proposed by F. Sharifi [8]. The inputs considered are the four voltage levels of the motors and the two components of the wind speed (1). As state variables, the position of the fixed reference system and the linear velocity, orientation and angular velocity of the quadrotor are considered. (2). Finally, the output variables defined are the real angular velocity (measured by the gyro), the linear acceleration (extracted from the accelerometer) -both referenced to the dynamic system- and the relative height (barometer) to the fixed frame (3).

$$U = [u_1,\ u_2,\ u_3,\ u_4,\ v_{vx},\ v_{vy}] \tag{1}$$

$$X = [x,\ y,\ z,\ \dot{u},\ \dot{v},\ \dot{w},\ \varphi,\ \theta,\ \psi,\ \dot{\varphi},\ \dot{\theta},\ \dot{\psi}] \tag{2}$$

$$Y = [\dot{\varphi},\ \dot{\theta},\ \dot{\psi},\ \ddot{u},\ \ddot{v},\ \ddot{w},\ z] \tag{3}$$

The individual thrust (4) and torque (5) generated by the motors are a direct function of their supply voltage u_i. Using them, it is possible to calculate the resultant momentum (6) and forces over the axis (7) (being 'l' as the wingspan of the drone).

$$f_i = k_f * u_i \tag{4}$$

$$\tau_i = k_\tau * u_i \tag{5}$$

$$f_m = \sum_{i=0}^{4} f_i \tag{6}$$

$$\tau_\varphi = l * (f_2 - f_4); \;\; \tau_\theta = l * (f_1 - f_3); \;\; \tau_\psi = \tau_1 + \tau_3 - \tau_2 - \tau_4 \tag{7}$$

Operating after considering the quadrotor's mass (m) and the moment of inertia (Ju, Jv and Jw), the gravity (g), the Coriolis forces and the fixed-mobile framework, the state equations can be defined as (8-11):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R * \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} v_{vx} \\ v_{vy} \\ 0 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \ddot{u} \\ \ddot{v} \\ \ddot{w} \end{bmatrix} = \begin{bmatrix} \dot{\psi} * \dot{v} - \dot{\theta} * \dot{w} \\ \dot{\varphi} * \dot{w} - \dot{\psi} * \dot{u} \\ \dot{\theta} * \dot{u} - \dot{\varphi} * \dot{v} \end{bmatrix} + \frac{1}{m} * \begin{bmatrix} F_u \\ F_v \\ F_w \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \dot{\varphi} * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \dot{\theta} * \begin{bmatrix} sen(\varphi) * tan(\theta) \\ cos(\varphi) \\ \frac{sen(\varphi)}{cos(\theta)} \end{bmatrix} + \dot{\psi} * \begin{bmatrix} cos(\varphi) * tan(\theta) \\ cos(\varphi) \\ \frac{cos(\varphi)}{cos(\theta)} \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{J_v - J_w}{J_u} * \dot{\theta} * \dot{\psi} + \frac{1}{J_u} * \tau_\varphi \\ \frac{J_w - J_u}{J_v} * \dot{\varphi} * \dot{\psi} + \frac{1}{J_v} * \tau_\theta \\ \frac{J_u - J_v}{J_w} * \dot{\varphi} * \dot{\theta} + \frac{1}{J_w} * \tau_\psi \end{bmatrix} \tag{11}$$

Finally, as a result, the output equations are obtained from state and input variables (12), where G corresponds to gyro, A to accelerometer and H to altimeter measures.

$$\begin{bmatrix} G_u \\ G_v \\ G_w \end{bmatrix} = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}; \;\; \begin{bmatrix} A_u \\ A_v \\ A_w \end{bmatrix} = \frac{1}{m} * \begin{bmatrix} F_u \\ F_v \\ F_w \end{bmatrix}; \;\; H = z \tag{12}$$

2.2 Failure Analysis: Lift System

Among all the possible failures and breakdowns, this work is centered in the events affecting to the drone's lift. It is due to the potential consequences it would have in the drone control and stability. They can be characterized according to two different parameters: the system affected by the failure and the importance/relevance of the breakdown in the performance of the system.

Firstly, it is required to analyze the failure-susceptible elements. Statistics shown that they are usually found in the propellers, engines, bearings, or electrical connections. All of them can provoke a thrust lost in the broken-down motor, that can be characterized as a linear combination of a translation over the engine's axis and a rotation over the opposite axis. It is the result of abrupt changes in both the linear acceleration and the angular velocity, that can be

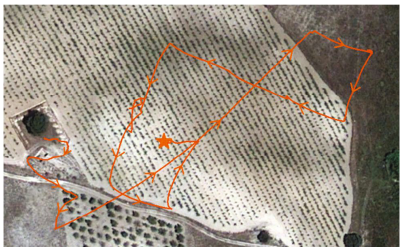


Fig. 2. Accident analysis: top view

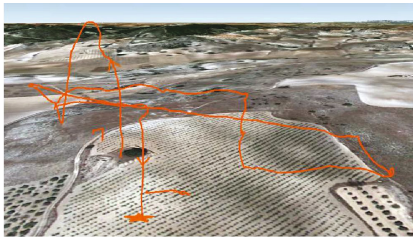


Fig. 3. Accident analysis: 3D view

detected before the drone's position and orientation reach dangerous values. In brief, the sudden lift lost together with the torque originated in the opposite axis makes the drone to fall turning around of its center.

This behaviour was verified in study conducted to analyze an accident happened in the RobCib group: figures 2 and 3 show the event: after the take-off and 2 minutes of normal operation, the drone -an Asctec Pelican- suddenly fell down with no apparent cause. The posterior analysis revealed a failure in one of the motors (the 3, concretely), that altered the lift. Although the real causes of the accident have been not discovered yet (it is though that was due to a problem in the motor power supply), it has been estimated that the failure would had been detected by the system described in the following section. It would had allowed to reduce the damages substantially. Both the detection system and the safety control techniques are presented in the following sections.

2.3 Failure Detection System

Bibliography presents two main different model-based approaches for detecting failures: the first ones measures the speeds in each one of the motors (by using a tachometer) and compares it with the theoretical one. C.Yang proves that this local method provides fast and accurate results [9], but is not able to take into account physical problems in the propeller or damages in the structure. That is why the second approach -based on the inner state- is broadly used.

Several works in the bibliography has addressed this approach: G.Heredia uses an observer based on an analytical (Kalman Filter) identification [4], while C.Berbra [1] employs a multi-observer strategy (every non-linear estimator observers the attitude but using different sensitivities, choosing the output of the one that minimizes a certain criterion). In this work, the Cauchy's Residues Theorem has been selected, simplifying the work done by Rafaralahy [6]: The state of the drone is compared with the theoretical model (presented in subsection 2.1). The state variables taken into account are i) the linear acceleration in X, Y and Z ($\ddot{x}, \ddot{y}$ and $\ddot{z}$, respectively; and ii) the angular velocities in roll, pitch and yaw ($\dot{\varphi}, \dot{\theta}$ and $\dot{\psi}$, respectively) [1]. They are acquired by using a fast high preci-

[1] Linear acceleration in the Z axis and the angular acceleration in yaw are both discarded for the following steps due to their low relevance in the failure detection.

sion IMU (mainly basing on the accelerometer and gyro measures). The 6-to-6 comparison generates the drone model's "residuals".

These features are good indicator of the real/theoretical divergence: when their values are close to zero, the likeliness model-reality is high. Conversely, when discrepancy is relevant, the distance between the model and the actually behaviour is high. Supposing the model good enough, it implies a failure in the operation. Figure 4 presents a simplified diagram of the detector, where the inputs, outputs and residuals (defined as a difference between real and estimated outputs) are the variables defined in 2.2.

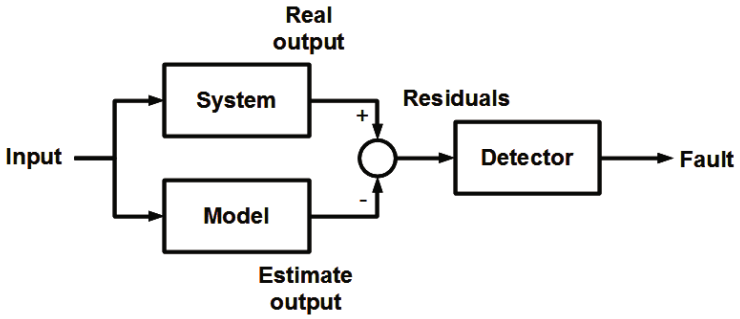


Fig. 4. Diagram of the failure detector

According to the breakdowns effects described in subsection 2.2, breakdown situation can be supposed when the residuals of one of the linear accelerations (X or Y) and one of the angular velocities (roll or pitch) exceed a certain threshold. Even more, a careful study proved that the interrelation among the residuals and their specific values identifies and determines the rotor affected by the failure. Table 1 presents the criteria selected for identifying the motor damaged.

Table 1. Damaged motor identification according to the residuals. The numeration of the motors follows the described in figure 1.

Motor number	Angular velocity	Linear acceleration
1	$\dot{\theta} < -TH_1$	$\ddot{x} < -TH_2$
2	$\dot{\varphi} < -TH_1$	$\ddot{y} > TH_2$
3	$\dot{\theta} > TH_1$	$\ddot{x} > TH_2$
4	$\dot{\varphi} > TH_1$	$\ddot{y} < -TH_2$

In the table TH_1 and TH_2 corresponds to two different thresholds: angular and linear. The have to be defined experimentally, since they depend on the drones characteristics: In this work, the features considered are the ones depicted in table 2, obtaining both thresholds by simulation. These simulation introduced

Table 2. Quadrotor parameters

Parameter	Value
Mass (m)	1 kg
Wingspan (l)	0.55 m
Force/voltage relation (K_f)	5 N/V
Torque/voltage relation (K_t)	0.05 N·m/V
Moment of inertia: U axis (J_u)	100,000 Kg·mm^2
Moment of inertia: V axis (J_v)	100,000 Kg·mm^2
Moment of inertia: W axis (J_w)	200,000 Kg·mm^2
Accelerometer deviation	0.025 m/s^2
Accelerometer noise	0.005 m/s^2
Gyro deviation	0.025 rad/s
Gyro noise	0.005 rad/s
Altimeter noise	0.02 m

Table 3. Results of simulation with mild failure detection system

Failure %	Torque (N·m)	Detection time (s)	α after detection (°)
10	0.125	0,65	31
20	0.251	0,45	29
50	0.625	0,3	32
100	1.25	0,2	28

different failures of diverse consideration, in order to obtain a general rule for the residuals. For example, figure presents a 50% breakdown in motor 1, that was detected after 244 ms. As it is possible to see in the figure 5, the value of the residuals in $\dot{\theta}$ and $\ddot{x}$ is obtained from the detection level determined.

Nevertheless, the simulations showed that for distinguish in fast and reliable way between mild and critical failures -those that implies less or more than 50% lift lost, respectively-, two parallel systems with different thresholds are needed. Tests have proved that 2 m/s^2 and 2 rad/s are the optimum value for detecting critical breakdowns. Conversely, mild failures requires $TH_2 = 0{,}5$ rad/s to perform suitably, considering only the average measures in order to avoid false positive detections due to the sensor's electronic noise.

The results of the simulations are shown in tables 3 and 4. As it is possible to appreciate, critical errors are detected quickly. In contrast, it takes a long time to detect mild errors. However, their effect in the global attitude of the drone is not actually strong, being in any case enough for applying the control methods and techniques described in the following section.

Finally, not only simulations but also real experiments were done in order to validate the hypothesis of the work. Figure 6 depicts the results of an experiment with a 100% failure in motor 2. As it is possible to appreciate, it was detected approximately after 390 ms. Orange, blue, yellow and green lines represent, respectively, the residuals of linear accelerations in X and Y and angular velocities

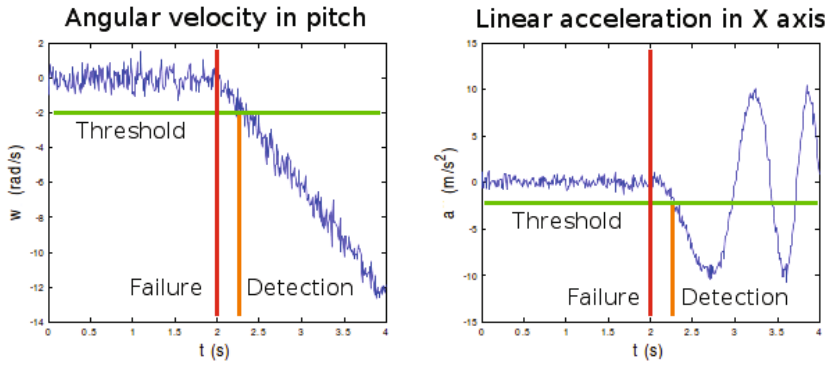


Fig. 5. Simulation of a 50% failure in motor 1: The breakdown is detected using the residuals of angular velocity in pitch and linear acceleration in x axis

in roll and pitch. On the other hand, red lines show the detection thresholds and red points show the time when they were exceeded. The breakdown was detected using the residuals of angular velocity in roll and linear acceleration in Y axis.

3 Failure Management

Once detected a failure in the drone's lift, the main goal is to reduce the risk: Given that a quadrotor is a naturally unstable system -in contrast to hexacopters or octocopters-, a failure in any of the motors makes impossible to continue the operation. Nevertheless, a quick detection of the failure allows a counteraction to stabilize the drone. It makes possible to landing in safe conditions -if the failure is not critical-, or at least to deploy a passive safety method that minimizes the damage.

In both cases, the control techniques to apply have to compensate the failure effects: maintain the drone in the horizontal plane and reduce as much as possible the fallen speed. In the first case, to guarantee that the landing avoids damages for both the own drone and to third part agents. On the other hand, in order to be able to employ the passive guard properly.

Table 4. Results of simulation with critical failure detection system

Failure %	**Torque (N·m)**	**Detection time (s)**	**α after detection** (°)
10	0.125	0,95	65
20	0.251	0,8	92
50	0.625	0,25	23
100	1.25	0,15	16

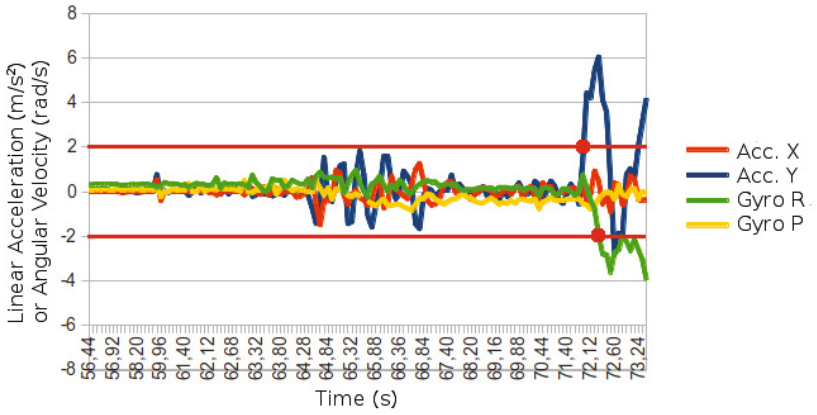


Fig. 6. Results of the detection system carried out in real experiments

3.1 Control Techniques for Degraded Systems

The possibilities of controlling a quadrotor after a failure in their lift system are quite limited and depends largely on the severity of damages. Therefore, the distinction between mild and critical failures raised in the detection of faults is also employed in the management of them.

In the case of serious damage, it has been rejected any possibility of control the damaged system. Instead, it has been proposed some maneuvers to mitigate the effects of the fault: stop the motor opposite the damaged motor, then stop the motor failed and finally increase the thrust of the other two engines. In addition, it has been recommended to deploy as soon as possible a passive safety method to break the fall and reduce damage.

On the other hand, in the case of mild failures it proposes a control over the shaft of the damaged rotor, obtaining the thrust of the opposite rotor of the damaged rotor from the angle and angular velocity of the shaft, as it is possible to see at equation (13).

$$f_a = f_d + k_1\dot{\alpha} + k_2\alpha \tag{13}$$

Where fa is the control thrust, fd is faulty thrust, α is the angular velocity and the angle of the shaft and k_1 and k_2 are constants.

It has been simulated this control system and it has obtained the results of the figure fig:control This figure shows the evolution of the opposite rotor thrust, the angular velocity and the orientation of the axis at the previous and following instants of a 20% failure. As it can see, the control system is able to stabilize the degraded quadrotor.

3.2 Passive Security Methods

As previously said, passive guards complement the underactuated (emergency) control system: these devices are supposed to be deployed in the worst situations,

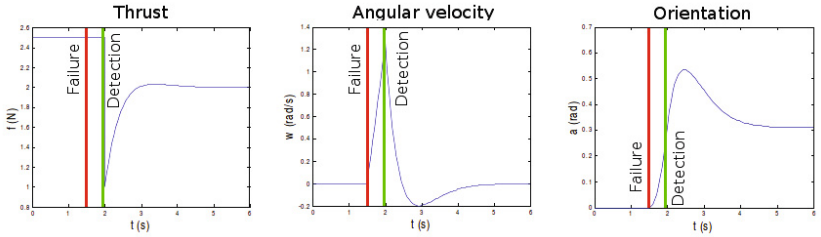


Fig. 7. Results of simulation of emergency control system

when the drone is completely unmanageable. Their goal is to limit and reduce as much as possible the drone damages. In this regard, a qualitative study of different methods (airbag, parachute, static propellers, etc.) was carried out. Due to its simplicity, well know operation, low cost and robustness, the semi-spheric parachute was selected as the most suitable passive guard. However, due to its unfolding complexity, a design where the rotor does not interfere with parachute spread out is required. Furthermore, since the potential deployment methods (e.g.pyrotechnic, elastic, mechanical..) increase the complexity of the system, also specific control techniques are required (some listed above and others such as engine stop).

Despite of the fact the parachute has been more explored in fixed-wing vehicles than in rotary-wing ones, literature and industry have already addressed some of these problems. From skydiving to the aerospace industry, the parachute has proved to be an effective way to reduce the falling speed. Many experiments and tests have been carried out, and the design principles are quite clear. Thus, the objective in this work was to study its viability, providing a quantitative assessment to verify its feasibility.

It has been estimated that a falling speed as low as 5 m/s speed guarantees the structural integrity of this type of UAVs. Thus, that is the aim imposed:

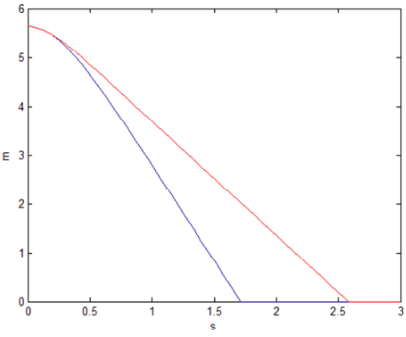


Fig. 8. Parachute analysis: Height evolution

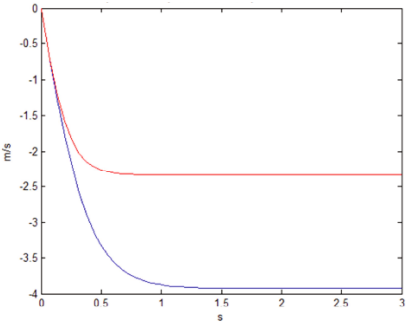


Fig. 9. Parachute analysis: Falling speed evolution

simplifying the model presented by Y.Kim [5], both a numerical and a experimental trial have been done in order to evaluate the requirements of the parachute: for a 3.5 Kg quadrotor, a 1.6m diameter parachute with a 1.2 shape coefficient is required to reach 5 m/s in 0.68 seconds after the full unfolding.

Figures 8 and 9 present an experiment where a dummy scale model felt with and without parachute (red and blue lines, respectively). As it is possible to appreciate, the speed reduction during the fall is relevant, providing as well a longer falling time that can be employed in safety maneuvers.

4 Results and Conclusions

Safety is a critical issue for the development and implantation of the small UAVs in daily tasks. The economic and human costs derived from potential accidents (actually frequent) is really high, so safety methods are necessary. This paper describes the efforts done in this sense, tackling the lift problems in 4-propeller rotorcrafts. A model-base failure detection system has been designed and implemented, proving its effectiveness both in simulation and in real experiments. Furthermore, control system for degraded devices has been also designed and validated combined with the detection system. For unmanageable/unrecoverable situations, passive security methods have been studied, proving that they are really helpful in these situations.

Acknowledgements. This work has been supported by the Robotics and Cybernetics Research Group at Universidad Politecnica de Madrid (Spain), and funded under the projects ROTOS: Multi-Robot system for outdoor infrastructures protection, sponsored by Spain Ministry of Education and Science (DPI2010-17998) and the project ROBOCITY 2030 (S-0505/DPI/ 000235) from the CAM's Regional Plan for the Scientific Research and Technological Innovation (PRICIT).

References

1. Berbra, C., Lesecq, S., Martinez, J.J.: A multi-observer switching strategy for fault-tolerant control of a quadrotor helicopter. In: 16th Mediterranean Conference on Control and Automation, pp. 1094–1099 (2008)
2. Ducard, G.J.: Fault-tolerant flight control and guidance systems: Practical methods for small unmanned aerial vehicles. Springer (2009)
3. Freddi, A., Longhi, S., Monteriu, A.: Actuator fault detection system for a mini-quadrotor. In: 2010 IEEE International Symposium on Industrial Electronics (ISIE), pp. 2055–2060 (July 2010)
4. Heredia, G., Ollero, A.: Detection of sensor faults in small helicopter uavs using observer/kalman filter identification. Mathematical Problems in Engineering (2011)
5. Kim, Y., Peskin, C.S.: 2-d parachute simulation by the immersed boundary method. SIAM Journal on Scientific Computing 28(6), 2294–2312 (2006)

6. Rafaralahy, H., Richard, E., Boutayeb, M., Zasadzinski, M.: Simultaneous observer based sensor diagnosis and speed estimation of unmanned aerial vehicle. In: 47th IEEE Conference on Decision and Control (CDC 2008), pp. 2938–2943. IEEE (2008)
7. Rago, C., Prasanth, R., Mehra, R.K., Fortenbaugh, R.: Failure detection and identification and fault tolerant control using the imm-kf with applications to the eagle-eye uav. In: Proceedings of the 37th IEEE Conference on Decision and Control, vol. 4, pp. 4208–4213. IEEE (1998)
8. Sharifi, F., Mirzaei, M., Gordon, B.W., Zhang, Y.: Fault tolerant control of a quadrotor uav using sliding mode control. In: 2010 Conference on Control and Fault-Tolerant Systems (SysTol), pp. 239–244. IEEE (2010)
9. Yang, C., Yang, Z., Huang, X., Xu, D.: Distributed fault-tolerant control for quadrotor. Journal of Applied Sciences 31(3), 321–330 (2013)

Analysis of Perturbations in Trajectory Control Using Visual Estimation in Multiple Quadrotor Systems

Alejandro Suárez, Guillermo Heredia, and Aníbal Ollero

Robotics, Vision and Control Group,
Universidad de Sevilla,
41092 Sevilla, Spain
alesuafer1@gmail.com,
{guiller,aollero}@us.es

Abstract. This paper describes the trajectory control of a quadrotor using external position estimation obtained from visual tracking in a scenario with multiple quadrotors and a camera mounted in the base of two or more UAVs, with the images being transmitted through a radio link. Applications where visual tracking can be used include fault detection and recovery of internal sensors, formation flying and autonomous aerial refueling. The dynamic model of the quadrotor and its trajectory control scheme is described along with the model of perturbations considered for the external position estimation. Graphical and numerical results are presented in different conditions, commenting separately the effect of each identified perturbation over the trajectory control. This study is done in simulation as previous step before testing quadrotor trajectory control in real conditions due to the high risk of accidents and damages on the vehicle.

Keywords: quadrotor, external position estimation, visual tracking, trajectory control, perturbations.

1 Introduction

Visual tracking in multi-UAV systems can be used in a number of control applications where safety and reliability are important requirements. Examples include cooperative UAV formation flying [1], fault detection and isolation [2] or autonomous aerial refueling [3]. A vision-based external estimation system will eventually provide a measurement that can replace or complement the internal sensors of the UAV being tracked. The paper analyzes the effects of the estimation and communication delays, packet loss and outliers on the control system performance.

In UAVs, Fault Detection, Identification and Recovery (FDIR) is usually done using the aerial vehicle internal sensors [4-10]. In a multiUAV team of cooperating aerial vehicles, the FDIR unit can exploit the team information for detection of faults, using the sensors onboard other UAVs for fault detection in a given UAV. This scheme requires the computation of the relative position/attitude of a UAV from other UAVs.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_9,

There has been a significant interest in the last years in the computation of this relative position between aerial vehicles using vision sensors, especially in autonomous formation flight and the closely related field of autonomous aerial refueling [11]. Uniquely identifiable light markers (beacons) are placed on the leader aircraft and on the refueling drogue to facilitate relative navigation. The beacons are one [12] or several [13] Light Emitting Diodes (LEDs) that emit structured light modulated with a known waveform. The vision sensor onboard the follower aircraft filters the received light energy from the LEDs so that much of the ambient energy is ignored and thus target detection can be achieved in a noisy, ambient environment. Based on the vision data collected, a nonlinear estimation routine is used to estimate the relative position and orientation of the vision sensor relative to the refueling drogue.

Ref. [14] employs the method of active contours in the image processing algorithms and Kalman filtering to track the leader aircraft across several image frames, without uniquely identifiable optical markers on the leader aircraft. Data derived by the use of active contours is used to drive an EKF that produces estimates of range, line-of-sight angle and their rates.

Relative position estimation has been done on UAVs [15], using homography-based techniques to obtain the position of a UAV relative to another UAV from the images that both take from the same scene. In this work, relative position estimation is done using visual tracking of the UAV from several other UAVs of the team.

The application of external estimations (on board other UAVs or on ground) to control systems introduces some problems to be considered carefully when system stability can be affected and the integrity of the robots or persons are in risk. This is the case of multi-UAV systems, especially in such unstable vehicles like quadrotors, where delays, noise and data loss in the external estimation may have catastrophic consequences for their control.

In this work it is considered the trajectory control of a quadrotor whose position is given by a vision-based external estimation system affected by a number of perturbations, including delays, noise, packet loss and outliers. Several experiments were conducted prior to this study, where the position of a quadrotor was obtained from a set of images and measurements of the position and rotation of the cameras that tracked it, as well as the real position of the tracked quadrotor obtained by means of a Vicon Motion Capture System used as ground truth, what allowed the identification of the perturbations mentioned.

The paper is organized as follows. Section 2 describes the external position estimation problem for quadrotor trajectory control. Section 3 covers the dynamic model and the control scheme. The perturbations under study are presented in Section 4. Graphical and numerical results of the simulations in different conditions can be found in Section 5, while Section 6 contains the conclusions.

2 Relative Position Estimation in multiUAV Systems

Consider a situation with three quadrotors A, B and C. Two of them, A and B, have cameras mounted in their base with known position and orientation referred to a

global frame. Images taken from cameras are sent along with their position and orientation to a ground station. Both cameras will try to stay focused on the third quadrotor, C, so a tracking algorithm will be applied to obtain the centroid of the object on every received image. An external position estimator executed in the ground station will use this data to obtain an estimation of quadrotor C position that can be used for position or trajectory control in the case C does not have this kind of sensors, they are damaged, or they are temporarily unavailable. The situation described above has been illustrated in figure 1. Here the cones represent the field of view of the cameras, the orange quadrotor is the one being tracked and the green ball corresponds to its position estimation.

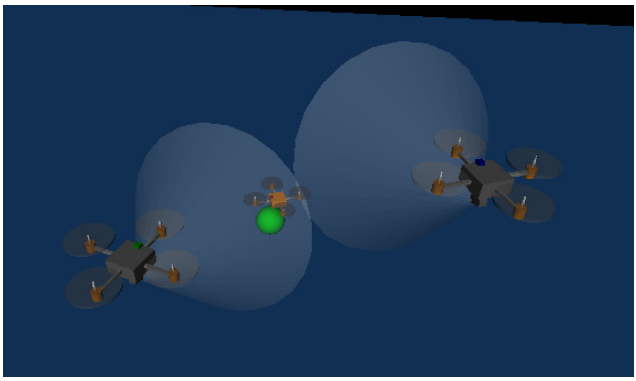


Fig. 1. Two quadrotors with cameras in their base tracking a third quadrotor whose position want to be estimated, represented by the green ball

One of the main issues in vision-based external position estimation applied to position control is the presence of delays in the control loop, which should not be too high to prevent the system of becoming unstable. The following sources of delay can be identified:

- Image acquisition delay
- Image transmission through radio link
- Image processing for tracking algorithm
- Position estimation and its transmission

The first two are imposed by hardware and available bandwidth. The last one is negligible in comparison with the others. On the other hand, image processing is very dependent on the computation cost required by the tracking algorithm. Before this simulation study, the external position estimation system was developed and tested with real data, obtaining position and orientation of cameras and tracked quadrotor from a Vicon Motion Capture System in the CATEC testbed. The visual tracking algorithm used was a modified version of CAMShift algorithm [16], [17]. This color-based tracking algorithm uses Hue channel in the HSV image representation for building a model of the object and detecting it, applying Mean-Shift for computing

the centroid of the probability distribution. As this algorithm is based only in color information, a small orange ball was disposed at the top of the tracked quadrotor, in contrast with the blue floor of the testbed. Figure 2 shows two images captured by the cameras during data acquisition phase.

Fig. 2. Images taken during data acquisition experiments at the same time from both cameras, with two orange balls at the top of a Hummingbird quadrotor

With the captured data, the estimation of the quadrotor position was obtained offline with an extended Kalman filter. Graphical results showed three types of errors in the estimation: an offset error about 15 cm between the real position (taken from Vicon) and the estimated one due to bad calibration of the cameras, a small amplitude noise typical in any estimation process, and the presence of outliers derived from faulty data in orientation measurement caused by occlusions. These perturbations will be simulated so their effects over the trajectory control of quadrotor can be analyzed.

3 Quadrotor Trajectory Control

Since this study is carried out by simulation, a dynamic model of the quadrotor and its control system is required. Otherwise, using a real system for testing perturbation effects over trajectory control would imply high risk of accidents and damages on the UAV.

This section presents the dynamic equations of the quadrotor, relating the three-dimensional position and orientation variables with the control signals, as well as the control scheme that allows the quadrotor to follow a path defined by a set of way-points.

3.1 Dynamic Model

As any other mechanical system, quadrotor dynamics can be derived from two ways [18]: using Newton-Euler formulation or Euler-Lagrange formulation, although the first one requires less computations and adding dynamic terms to the model is easier. In this case, the relationship between translational and rotational accelerations of the rigid body and external forces and torques is given by:

$$\begin{bmatrix} m\,I_{3\times3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix}. \tag{1}$$

where m is the total mass of the quadrotor, I is the inertia matrix, assumed to be diagonal, V and ω are the translational and rotational velocities, and F and τ are the external forces and torques applied to the system. Only two kind of external forces are considered here: gravity and the thrust of the motors denoted by T_i, with $i = 1, 2, 3, 4$. Then, the rotational acceleration due to motor forces and gyroscopic terms is:

$$\begin{aligned} I_{xx}\ddot{\phi} &= \dot{\theta}\dot{\psi}\left(I_{yy} - I_{zz}\right) + l(-T_2 + T_4) \\ I_{yy}\ddot{\theta} &= \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + l(T_1 - T_3) \\ I_{zz}\ddot{\psi} &= \dot{\phi}\dot{\theta}\left(I_{xx} - I_{yy}\right) + \sum_{i=1}^{4}(-1)^i Q_i \end{aligned} \quad . \tag{2}$$

where ϕ, θ, and ψ are roll, pitch and yaw angles, I_{xx}, I_{yy} and I_{zz} are the inertia moments along the three axes (the inertia products are zero if the mass distribution is symmetric), l is the length of the arms and Q_i is the counter-torque generated by the i-th motor. On the other hand, the translational accelerations can be computed as follows:

$$\begin{aligned} m\,\ddot{z} &= mg - (c\phi\; c\theta)\sum_{i=1}^{4} T_i \\ m\,\ddot{x} &= (s\psi\; s\phi \;+\; c\psi\; s\theta\; c\phi)\sum_{i=1}^{4} T_i \\ m\,\ddot{y} &= (-c\psi\; s\phi \;+\; s\psi\; s\theta\; c\phi)\sum_{i=1}^{4} T_i \end{aligned} \quad . \tag{3}$$

There are three important considerations here: 1) quadrotor rotation is independent from position, while position depends on rotation, 2) quadrotor moves due to projection of total thrust of the motors over the horizontal plane, and 3) quadrotor is an under-actuated system with 6 DOF and only four control signals (the thrust of each motor).

Finally, in ref. [18] it is assumed that motor thrust T_i and counter-torque Q_i are proportional to the square of its speed:

$$T_i = b \cdot \Omega_i^2 \quad ; \quad Q_i = d \cdot \Omega_i^2 \; . \tag{4}$$

3.2 Attitude and Height Control

Although in practice controllers actuate over motors speed, four control signals are defined in the following way:

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b \cdot (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ b \cdot (-\Omega_2^2 + \Omega_4^2) \\ b \cdot (\Omega_1^2 - \Omega_3^2) \\ d \cdot (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix}. \tag{5}$$

These signals allow to control quadrotor height, roll, pitch and yaw angles, respectively, in terms of rotor speeds, decoupling control problem despite de gyroscopic term in (2).

Looking at (2), if these terms are rejected, orientation dynamic can be approximated by a double integrator. The same assumption can be done in (3) for small angles. Gyroscopes and accelerometers are combined for having a good estimation of rotational position, but they usually also provide rate information in roll,

pith and yaw angles. That is why a position and velocity feedback scheme [19], [20] can be applied, so references in angular speed and their corresponding control signals will be given by:

$$\begin{bmatrix}\dot{\phi}_d\\ \dot{\theta}_d\\ \dot{\psi}_d\end{bmatrix}=\begin{bmatrix}K_p^{\phi}\cdot e_{\phi}\\ K_p^{\theta}\cdot e_{\theta}\\ K_p^{\psi}\cdot e_{\psi}\end{bmatrix}\quad;\quad\begin{bmatrix}U_2\\ U_3\\ U_4\end{bmatrix}=\begin{bmatrix}K_v^{\phi}\cdot(\dot{\phi}_d-\dot{\phi})\\ K_v^{\theta}\cdot(\dot{\theta}_d-\dot{\theta})\\ K_v^{\psi}\cdot(\dot{\psi}_d-\dot{\psi})\end{bmatrix}. \tag{6}$$

The errors in roll, pitch and yaw angles are defined as the difference between the reference and the orientation measurement, with the references in roll and pitch given by the velocity controller (see Section 3.3):

$$\begin{aligned} e_{\phi} &= \phi_d - \phi\\ e_{\theta} &= \theta_d - \theta\\ e_{\psi} &= \psi_d - \psi \end{aligned}. \tag{7}$$

This technique however cannot be applied for height control because in general, for small UAVs, speed on Z axis is not an available signal. Instead, height control signal can be decomposed in the sum of a constant term for compensating quadrotor weight, and a correction term that considers height error:

$$U_1 = mg + u_h(t)\,. \tag{8}$$

It was found that designing a discrete time controller able to handle discontinuities in height references, like steps, was a hard task. Moreover, continuous time controllers that provided stable response became unstable when they were discretized. Ref. [21] proposes an optimal digital controller for double integrator system with a sample time of 0.1 seconds, suitable for height control, which will be used in this work. This controller was tested in simulation with good results.

3.3 Velocity Control

The velocity control in the XY plane is now considered, maintaining a null reference for yaw angle. Let (x_d, y_d) be the desired point and (x, y) the current position of the quadrotor. Then, the direction of the velocity in the global frame can be computed as follows:

$$\Psi = \tan^{-1}\left(\frac{y_d - y}{x_d - x}\right). \tag{9}$$

The quadrotor will try to maintain a constant speed V, although accepting speed transitions when the direction changes between way points. Then the velocity error on XY axes is computed as follows:

$$\begin{aligned} e_{Vx} &= V\cdot\cos(\Psi) - \dot{x}\\ e_{Vy} &= V\cdot\sin(\Psi) - \dot{y} \end{aligned}. \tag{10}$$

Assuming a null reference for yaw angle, then speed on X axis will depend on pitch angle, while speed on Y axis will do in roll angle. Simulation experiments show

that a simple proportional controller provides good results, so roll and pitch references are obtained in the following way:

$$\phi_d = K_p^y \cdot e_{Vy} \quad ; \quad \theta_d = K_p^x \cdot e_{Vx} \,. \tag{11}$$

3.4 Trajectory Generation

A trajectory Γ will be defined as a sequence of N way-points, where each way-point specifies the XYZ desired coordinates that the quadrotor should reach:

$$\Gamma = \{wp_1, wp_2, \ldots, wp_N\} \quad ; \quad wp_n = [x_n^d, y_n^d, z_n^d]^T \tag{12}$$

The criterion used for jumping from current way point to the next one was to consider a threshold distance d_{th}, so the way-point counter will be incremented if the following condition is satisfied:

$$\sqrt{(x_n^d - x)^2 + (y_n^d - y)^2 + (z_n^d - z)^2} < d_{th} \,. \tag{13}$$

No interpolation polynomials were used in this work, so quadrotor will follow segments of linear trajectories.

4 Model of Perturbations

This section presents the five perturbations considered in the external position estimation that is sent to the quadrotor for its trajectory control.

4.1 Sampling Rate

As any digital system, external estimator will receive external observations at discrete time instants. Two working modes can be considered: periodic sample rate or variable sampling time. In general, quadrotor position control system will work at higher frequencies that the external estimator, particularly if this functional block makes use of computer vision techniques.

Because quadrotor control system requires position data at fixed rate, a sampling rate conversion process is then needed. The simplest solution is the use of Zero Order Holder, so last update obtained from estimator will be provided until the next update is received. Previous experiments to this work show that the sampling rate for visual-based external position estimation is around 5-20 Hz, depending on image resolution and available bandwidth for its transmission, while quadrotor sample rate for position control is typically between 1-10 Hz for GPS and 100-200 Hz for a Vicon Motion Capture System.

4.2 Delay

Here the delay is defined as the elapsed time between the reception in the quadrotor of the estimation of its position and the moment when data used for this estimation was captured. In the case of image based estimation, the delay includes image acquisition

and transmission, visual tracking algorithm application, estimation computation and its transmission to the UAV.

Height is a critical magnitude in quadrotor control, particularly sensitive to delays. In many cases, small size UAVs have ultrasonic devices mounted in their base, so it is not too hard assumption to consider its internal measurement. In practice, ultrasonic devices used with quadrotors usually have a delay less than 100 ms. In most cases, delay can be considered to be equal to the sampling time.

4.3 Noise

Position estimation process will be affected by a number of perturbations that cause the estimation to be slightly different from real value. When the position estimation is computed using a pair of images and the position and orientation data of the cameras themselves, the main contributions to the error are distortion in non-well calibrated lenses, noise in the measurement of position and orientation of the cameras and errors in the centroid of the tracked object. On the other hand, an offset error in the measurement of the orientation of the cameras causes an estimation error in the position proportional to the distance between the cameras and the tracked object.

In this work, noise will be simulated as an additive random variable, with normal or uniform distribution function, characterized by its mean and variance or maximum and minimum. The offset error in the position estimation obtained after the experiments was around 0,15 m, with a mean distance of 4 m between the cameras and the quadrotor. The standard deviation of the noise will strongly depend on the number of consecutive frames with tracking loss in one or both cameras. In normal conditions, a standard deviation of 0,05 m can be assumed.

4.4 Outliers

An outlier can be easily identified as a point that is located far away from the rest of estimation points. It has short duration in time, and it will appear at each sample period with a certain probability. Therefore two parameters can be associated: the amplitude of the outlier, and its probability. This kind of perturbation was identified during analysis of the results of image based position estimation, after the application of extended Kalman filter, and it was caused by outliers in cameras orientation measurement and inconsistencies in centroids in the pair of images. As an extended Kalman filter was used for estimation, the outliers will be partially filtered, but amplitudes around 25 m can be considered, while an acceptable value for the outlier probability is 0,05.

4.5 Packet Loss

Position estimation is externally estimated by an agent that receives images from the cameras as well as the position and orientation of the cameras, and will send its estimation via radio link to the tracked object. In practical, it should be considered the possibility of the loss of one or more packets containing this estimation. For this

simulation, we consider a probability of packet loss at each sample time, and a uniform random variable that will represent the number of packets lost. Values of 0,01 for packet loss probability and a uniformly distributed random number between 1 and 10 in the number of packets lost can be used for reference.

5 Simulation Results

Graphical and numerical results of simulations in different conditions are presented here, showing separately the effects of perturbations indicated in previous section so it will be easier to interpret their effects on the performance of trajectory control. These results are only for reference. Effects of perturbations will obviously depend on the control structure being used, although the one presented in Section 3 is a conventional control scheme.

5.1 Different Speeds and Delays

Delay in position measurement and quadrotor speed will be increased until instability is reached, without considering any other perturbation. The sample time for quadrotor control is 20 ms, except in height control, where it was used 100 ms period. Figure 3 shows the reference path and the trajectories followed for different values of delay in XY position measurement and fixed delay of 100 ms in height when the speed of the quadrotor is 0,5 $m \cdot s^{-1}$. A threshold distance of d_{th} = 0,25 m was specified for jumping between way-points.

Table 1 relates different speeds of the quadrotor with maximum delays allowed in XY position and in height measurements. In the first case, instability is associated with the existence of a loop that catch the quadrotor, preventing it to follow the

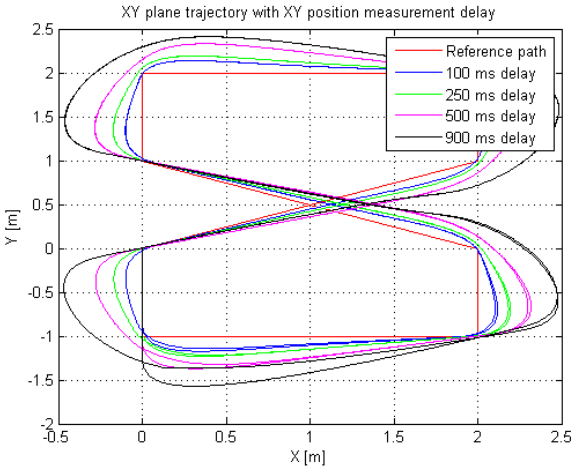


Fig. 3. Reference path and trajectories followed in XY plane with different values of delay in XY position measurement, fixed delay of 100 ms in height measurement and V = 0,5 $m \cdot s^{-1}$

trajectory, while the criterion used for height was an overshoot greater than 50%. These results were obtained for the reference path shown in figure 3. The effect of delay in height control is independent from quadrotor speed, but in the first three cases, height limits the maximum delay allowed when a full XYZ external position estimation is considered.

Table 1. Maximum values of delays in XY position and height measurement for different references of speed

Speed [m·s^{-1}]	XY plane max. delay [s]	Height max. delay [s]
0,25	2,4	0,2
0,5	0,94	0,2
0,75	0,44	0,2
1	0,18	0,2

The threshold distance d_{th} between the current quadrotor position and the next way-point can be used to compensate delay. Then, the higher the threshold is, the lower the effect of the delay over the trajectory will be, although this value is limited by the minimum distance between way-points. Quadrotor speed will also be limited by the maximum error allowed in the trajectory tracking or in the position control for a certain application where reducing delay is not possible. However, the speed of the UAV might be adjusted depending on the distance to the next way-point, specifying higher speeds when the distance is large enough, and decreasing it as the quadrotor approaches to the way-point.

5.2 Noise and Outliers with Fixed Delay

In this simulation it will be applied an additive Gaussian noise with 0,2 m^2 variance and outliers with uniformly distributed amplitude in the range ±10 m and a probability of 0,1 over position measured in XY plane. It will be maintained a delay of 0,25 s in this measurement, and 0,1 s on height. Figure 4 shows the external XY position estimation used by the trajectory controller. The reference path was the same.

In figure 5 it can be seen the effects of noise and outliers separately over trajectory control in the XY plane. It has also been represented the trajectory followed without this two perturbations, considering only the delay. There is an important change on the behavior when noise is applied over estimation. Outliers instead are filtered by the low-pass dynamics of the quadrotor, so their effect is not very significant in comparison due in part to the lack of a derivative term in the velocity controller, as seen in (11). Trajectory control can be therefore enhanced if position estimation is filtered before being used by controller.

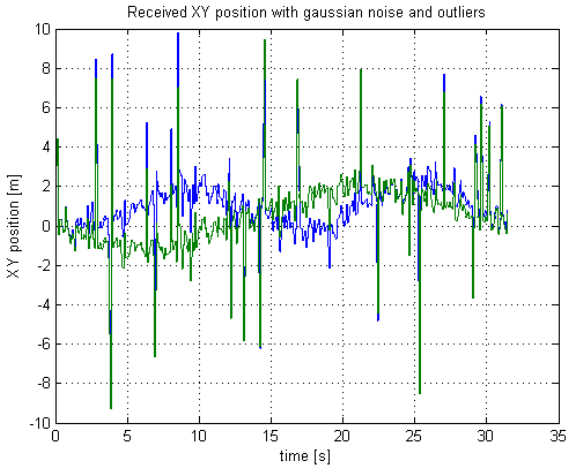


Fig. 4. External estimation of XY position with Gaussian noise, outliers and a reference speed of V = 0,5 $m \cdot s^{-1}$

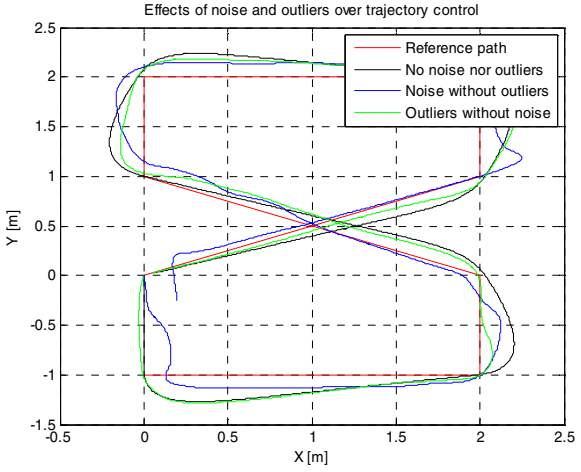


Fig. 5. Trajectories followed by the quadrotor with noise and outliers (blue) and without them (black)

5.3 Delay, Noise, Outliers and Packet Loss

Finally, all perturbations presented in Section 4 will be considered together, introducing now a packet loss probability of 0,02 with a uniform random number of packets lost between 1 and 10. These perturbations are only applied over XY position estimation because, as mentioned above, the height is assumed to be measured with internal sensors. Figure 6 represents the trajectory followed by the quadrotor for speeds of 0,5 and 0,75 $m \cdot s^{-1}$. As quadrotor trajectory controller needs position

measurements every sample period, when a packet is lost, the system will use the last measurement received. This implies serious problems if the last measurement was an outlier and the number of packets lost are large enough, causing strange behaviors The sudden change in the direction of the quadrotor in figure 6 for V = 0,5 m·s^{-1} around (x, y) = (1,7, -1,1) may be due to the coincidence of both perturbations.

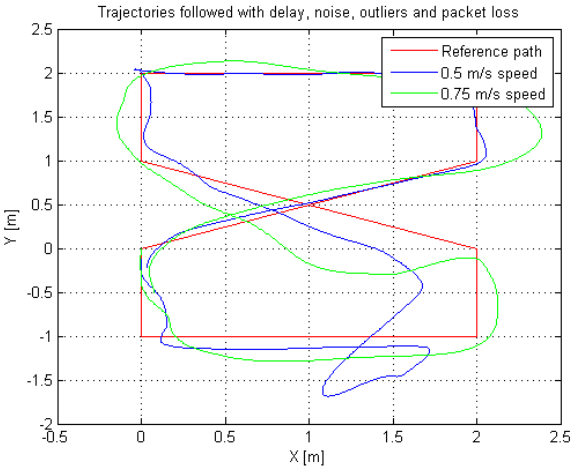


Fig. 6. Quadrotor trajectories with simultaneous application of noise, delay, outliers and packet loss for V = 0,5 m·s^{-1} (blue) and V = 0,75 m·s^{-1} (green)

On the other hand, height control is not affected by these perturbations despite the coupling between XY position and height through the control signals, as motors thrust is used for controlling these three variables. In figure 7 the Z-axis reference and the real height are shown in the conditions described above and V = 0,75 m·s^{-1}.

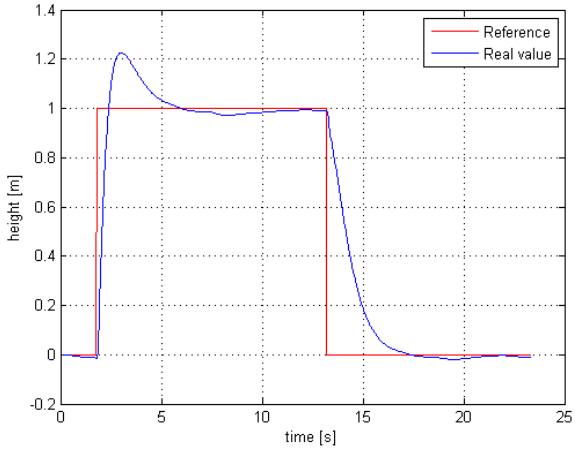


Fig. 7. Reference and real value for height when XY position is affected by noise, delay, outliers and packet loss with a reference speed of V = 0,75 m·s^{-1}

The simultaneous application of all perturbations considered in this study has an unacceptable effect over the trajectory control. Delay and packet loss have a similar influence as the second one can be considered as a random delay. Outliers, if they are not consecutive, are filtered by the low-pass quadrotor dynamics, and they can be easily detected too. On the other hand, noise is the main cause of the bad results in the trajectory tracking shown in figure 6. There are two ways to decrease its effect: improving the position estimation, or considering control methods more robust to noise.

In general, the accuracy of the visual position estimation will depend on the number of available cameras without tracking loss, their relative position, and the accuracy in the measurement of their position and orientation. As the number of cameras increases, the uncertainty in the estimation and the effect of outliers are reduced. On the other hand, it could be possible to obtain an estimation at higher rates if the image processing is done onboard each of the UAVs instead of sending and processing images in a ground station. A global time reference would also be needed for computing the elapsed time since the last frame was captured, so cameras do not need to be synchronized.

6 Conclusions

Visual estimation in multi-UAV systems can be used for trajectory or position control, replacing or complementing the measurements from the internal sensors of a certain UAV. However, a number of perturbations are introduced by the external estimation system in the control loop, whose effect should be studied. In this paper it has been analyzed by simulation the effects of delay, noise, outliers and packet loss in the trajectory control of a quadrotor. These four perturbations are typical in an external position estimation system that computes the position of a tracked object from images captured by two cameras whose position and orientation are known, and that transmit this estimation through a radio link. This simulation is required before testing trajectory control with external position estimation due to the risk of accidents and damages on real quadrotor if the system becomes unstable. Graphical and numerical results have been presented in different conditions, identifying separately the effects of delay, noise and outliers, as well as considering the four perturbations together. Delay can be partially compensated by increasing the threshold distance to the next way-point, changing the direction of the quadrotor before crossing it. Noise will strongly depend on the tracking algorithm and in the position and orientation measurements, although it can be partially filtered or rejected using more robust control methods. On the other hand, outliers can be easily identified and therefore removed from estimation. However, the presence of noise or outliers in the external estimation before a packet loss may have unacceptable consequences in applications where safety and reliability are important.

Acknowledgements. This work was partially funded by the European Commission under the FP7 Integrated Project EC-SAFEMOBIL (FP7-288082) and the CLEAR Project (DPI2011-28937-C02-01) funded by the Ministerio de Ciencia e Innovacion of the Spanish Government.

References

1. Wang, X., Yadav, V., Balakrishnan, S.N.: Cooperative UAV formation flying with obstacle/collision avoidance. IEEE Transactions on Control Systems Technology 15, 672–679 (2007)
2. Patton, R.J., Chen, J.J.: Observer-based fault detection and isolation: robustness and applications. Control Engineering Practice 5, 671–682 (1997)
3. Fravolini, M.L., Ficola, A., Campa, G., Napolitano, M.R., Seanor, B.: Modeling and control issues for autonomous aerial refueling for UAVs using a probe-drogue refueling system. Aerospace Science and Technology 8, 611–618 (2004)
4. Drozeski, G., Saha, B., Vachtsevanos, G.: A fault detection and reconfigurable control architecture for unmanned aerial vehicles. In: Proc. of the IEEE Aerospace Conference, Big-Sky, USA, March 5-12 (2005)
5. Heredia, G., Ollero, A.: Detection of Sensor Faults in Small Helicopter UAVs using Observer/Kalman Filter Identification. Mathematical Problems in Engineering, vol 2011, article id. 174618 (2011)
6. Falcoz, A., Henry, D., Zolghadri, A.: Robust Fault Diagnosis for Atmospheric Reentry Vehicles: A Case Study. IEEE Transactions on Systems, Man and Cybernetics, Part A 40(5), 886–899 (2010)
7. Heredia, G., Ollero, A.: Virtual Sensor for Failure Detection, Identification and Recovery in the Transition Phase of a Morphing Aircraft. Sensors 10(3), 2188–2201 (2010)
8. Enns, R., Si, J.: Helicopter Flight-Control Reconfiguration for Main Rotor Actuator Failures. Journal of Guidance, Control and Dynamics 26(4), 572–584 (2003)
9. Bateman, F., Noura, H., Ouladsine, F.: Fault Diagnosis and Fault-Tolerant Control Strategy for the Aerosonde UAV. IEEE Transactions on Aerospace and Electronic Systems 47(3), 2119–2137 (2011)
10. Heredia, G., Caballero, F., Maza, I., Merino, L., Viguria, A., Ollero, A.: Multi-Unmanned Aerial Vehicle (UAV) Cooperative Fault Detection Employing Differential Global Poisitioning (DGPS), Inertial and Vision Sensors. Sensors 9(9), 7566–7579 (2009)
11. Sattigeri, R.: Adaptive Estimation and Control with Application to Vision-Based Autonomous Formation Flight, PhD Thesis, Georgia Institute of Technology, (2007)
12. Tandale, M.D., Bowers, R., Valasek, J.: Robust Trajectory Tracking Controller for Vision based Probe and Drogue Autonomous Refueling. In: AIAA Guidance, Navigation and Control Conference, San Francisco, CA (2005)
13. Pollini, L., Mati, R., Innocenti, M.: Experimental Evaluation of Vision Algorithms for Formation Flight and Aerial Refueling. In: AIAA Modeling and Simulation Technologies Conference and Exhibit, Providence, RI (2004)
14. Ha, J., Alvino, C., Pryor, G., Niethammer, M., Johnson, E., Tannenbaum, A.: Active Contours and Optical Flow for Automatic Tracking of Flying Vehicles. In: Proc. of the American Control Conference, Boston, MA (2004)
15. Merino, L., Wiklund, J., Caballero, F., Moe, A., De Dios, J.R.M., Forssen, P.-E., Nordberg, K., Ollero, A.: Vision-based multi-UAV position estimation, pp. 53–62. IEEE Computer Society Press (2006)
16. Bradski, G.: Computer Vision Face Tracking for Use in a Perceptual User Inter-face. Intel Technical Articles (1998)
17. Exner, D., Bruns, E., Kurz, D., Grundhöfer, A.: Fast and Robust CAMShift Tracking. In: IEEE Computer Society Conference Computer Vision and Pattern Recognition Workshop, pp. 9–16 (2010)

18. Bouabdallah, S.: Design and Control of Quadrotors with Application to Autonomous Flyin, PhD Thesis, École Polytechnique Fédérale de Lausanne (2007)
19. Rao, V.G., Bernstein, D.S.: Naive Control of the Double Integrator. IEEE Control Systems 21(5), 86–97 (2001)
20. Yang, T., Stoorvogel, A.A., Saberi, A.: Global Stabilization of the Discrete-Time Double Integrator Using Saturated Linear State Feedback Controller. In: American Control Conference, pp. 4440–4445 (2011)
21. Polyakov, K.Y., Rosenwasser, E.N.: Optimal Digital Controllers for Double Integrator: Comparison of Four Methods. In: IEEE International Symposium on Computer Aided Control System Design, pp. 199–204 (2002)

RUAV System Identification and Verification Using a Frequency-Domain Methodology

I. Sánchez[2], D. Santamaría[1], A. Viguria[1], Aníbal Ollero[2], and Guillermo Heredia[2]

[1] Center for Advanced Aerospace Technologies (CATEC),
Parque Tecnológico y Aeronáutico de Andalucía,
C/ Wilbur y Orville Wright 19 41309 - La Rinconada Seville, Spain
{dsantamaria,aviguria}@catec.aero
[2] Robotics, Vision and Control Group (GRVC),
Universidad de Sevilla, Sevilla, Spain
{mivansm,aollero,guiller}@us.es

Abstract. The aim of this paper is to show a methodology to obtain a model of a rotary wing UAV (Unmanned Aerial Vehicle) employing a frequency-domain System Identification (SYSID) methodology using CIFER®. The methodology is applied to the CB-5000 RUAV and discuss several identification issues, from the telemetry acquisition process, parametric model to be identified and identification technique, to finally validate and implement the model. The UAV's real autopilot software is integrated with the CIFER® model showing a good behaviour without any change on the tuning of the real autopilot gains. In order to validate and compare the results, an alternative two rigid body kinematic model is presented. Finally, the models integrated with the autopilot are compared by using the experimental data of the real RUAV (Rotorcraft UAV) platform following the same flight plan.

Keywords: UAV modelling and control, Identification, CIFER®, UAV, rotorcraft, aircraft modelling and control.

1 Introduction

In the next years, unmanned helicopters are going to be used for numerous commercial, civil, and military applications. This type of platform has a VTOL (Vertical Take-off and Landing) capability that enables UAS (Unmanned Aerial System) to operate in uneven terrains, or even operate from mobile platform such as the deck of a ship. Easy to transport and deploy, these flying autonomous vehicles are safe and cost-effective platforms to carry cameras and other surveillance equipment or even to transport and deploy loads [1].

SYSID (SYStem IDentification) is a technique for dynamic response determination by means of a state measurement series of a physical system. It is used to develop or improve the mathematical representation of a system using experimental data. Among the different types of SYSID, this paper is concerned about the identification type supported by CIFER® (Comprehensive Identification from FrEquency Responses):

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_10,

frequency-domain identification of a linear parametric model. This type has been widely proved to be the best option in SYSID of rotary wing aircraft (see e.g. [2] and the references given there). Results here presented derive from [3].

It is intended to follow all identification steps – in natural order – throughout this document. Therefore this paper begins with a brief description of the identification flight maneuvers for the data acquisition. Section 3 shows the parametric model to identify and some issues about parameter estimation and initial conditions needed for identification. Section 4 provides some guidelines and tasks for data conditioning before CIFER® usage. Section 5 deals with the main issues that should be taken into account in the identification and verification process with this software. In section 6 an alternative modelling technique is presented. Section 7 introduces the guidance and control UAV algorithms. In Section 8, the controller previously described is simulated with the different models. The results are compared with real data obtained on experiments by following the same flight plan. Finally, conclusions of this work and future extensions are presented in Section 9.

2 Flight Data Collection

Identification flight tests were performed on a modified RC helicopter (see Fig. 1) that was converted into a UAV with the integration of several sensors and a processing unit. The RUAV navigation algorithm is based on an Extended Kalman Filter estimation (see [4] and [5]), using a loosely coupled INS/GPS (Inertial Navigation System/Global Position System) integration combined with a magnetometer unit. A real-time operating system embedded in a PC-104 sends all measurements via Wi-Fi to a ground station. There they are logged and supervised by flight test engineers.

Main Rotor	
Radius	0.925 m
Blade Chord	0.070 m
Solidity Ratio	0.045
Nominal Angular Speed	1300 rpm
Tail Rotor	
Radius	0.355 m
Blade Chord	0.030 m
Tail Arm Distance	1.400 m
Nominal Take Off Weight	11.360 kg
Power Plant	Two-Stroke Engine
Flight Autonomy	~30 min

Fig. 1. (Right) RUAV platform with all the avionics to collect the frequency sweep data. (Left) CB-5000 main characteristics.

These flight tests are the input of the identification process; hence, the model accuracy is directly related with the dataset. The main goal is to record a set of data good enough to identify the model parameters. Therefore they should excite all dynamics modes of interest in the range of frequencies where the model is going to fit the real system. This range will also depend on the purposes and applications of the model.

The tests that a pilot should perform in open loop are the following:

- Trimming the RUAV in a particular flight condition (e.g. hover flight at a given altitude). Since the parametric model is linearized around a trim condition (see Section 3), the dynamic response should consist in perturbations around that condition.
- Doing sinusoidal sweeps for each control variable reducing progressively the oscillation period (see Fig. 2). Ideally, each test should be accomplished using just the control variable of interest. However, RUAV are very unstable (even more than manned helicopters) and to maintain the trim condition is really challenging. Because of this, small pulse corrections need to be done by the secondary control variables. These data logs are employed for identification tasks.
- Doing doublet-steps maneuvers (see Fig. 3) for each control variable of approximately 10s of duration for CIFER® verification tasks.

Hence, once the helicopter is trimmed in a flight condition, the pilot sweeps using just one control variable (e.g. lateral cyclic control, as can be shown in Fig. 2) and introducing small an instantaneous corrections with the rest of control signals to maintain the helicopter as stable as possible during the sweep. The process is repeated for each control signal (four times in this case). Helicopter status and maneuvers are supervised using the commented ground station.

It is important that all maneuvers start and end in the same trim condition. Also the data set will only be valid for identification in calm atmosphere with winds up to $\sim 2.5 m/s$. More guidelines and suggestions to accomplish identification flight test can be found in [2], [3], [6] and [7]. Nevertheless these recommendations are hard to execute in practice. In fact, the accomplishment of RUAV identification tests in cruise condition is very challenging; the pilot has to control safely a very unstable aircraft maintaining the cruise condition, sweeping one control input and reducing the actuation of the secondary ones while the RUAV is getting far away. Hence, an adequate flight tests design and an expert pilot familiarised with these experiments are required.

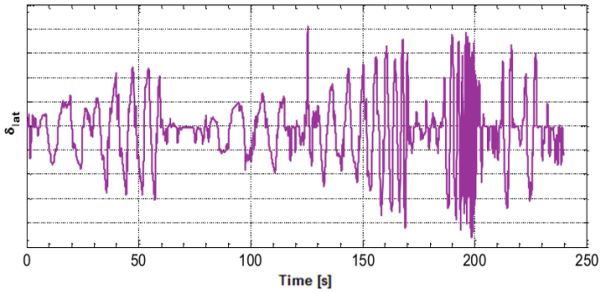


Fig. 2. Typical lateral cyclic control signal sweep maneuver

Alternatively, a close loop identification test could be done but this approach has the drawback that the behaviour being identified includes the system composed of the platform and the control system, and then the results are not useful for control design purposes.

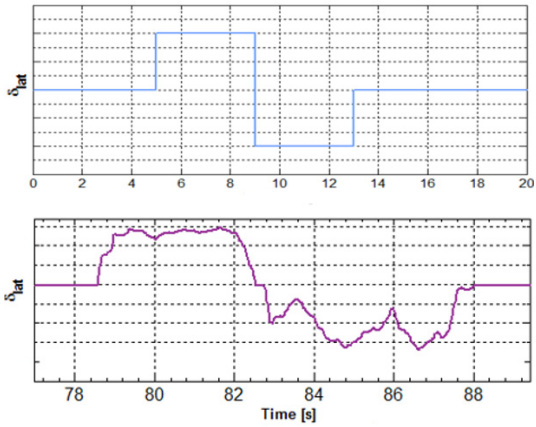


Fig. 3. (Up) Ideal Doublet-Step verification manoeuvre for lateral cyclic. (Down) Real Dublet-Step manoeuvre.

In order to collect the required data for identification, a detailed maneuver design should be done. This process is about determining the maximum and minimum frequencies of interest in system dynamics, length and number of sweeps for each control variable and sample rate required. Windowing parameters in CIFER® (see Section 5) are connected with these parameters. References [2] and [8] offers practical guidelines for these purposes. As is explained in [3], a spreadsheet can implement these parameters for easy maneuver design tasks (see Table 1).

Then, for each control signal, as many sweeps of length T_{b_i} as necessary should be performed to obtain an approximate sweep flight time of T_{b_T}. These sweeps should never have a duration of less than $T_b]_{min}$. It is advisable to execute each sweep to adjust a specific frequency range (LF, MF, and HF).

In order to facilitate the sweep maneuvers, sweeps for each control can be divided by frequency range (see Fig. 2):

- Low Frequency (LF): ~0.05Hz - ~0.75Hz
- Medium Frequency (MF): ~0.75Hz - ~1.50Hz
- High Frequency (HF): ~1.50Hz - ~<5.0Hz

Experience demonstrates than over the 40% flight test time is employed on LF sweeps because their low rate excitation brings out easily the RUAV from trim condition. However, this LF information is a key aspect to capture the RUAV translational dynamics.

RUAV dynamics are conservatively collected in a range of frequencies between 0.05Hz and 5Hz [6], but it can be estimated from a normal scale helicopter (see [7]) with a similar configuration using the rule given by

$$f_{RUAV} = \frac{1}{\sqrt{s}} f_{real} \tag{1}$$

where s is the model scale.

Table 1. Spreadsheet structure (Input Parameters & Computed Parameters) for detailed maneuver parameter design

INPUT PARAMETERS		
Damping	ξ	0.05
Frequencies		**Hz**
Minimum of Interest	f_{min}	0.05
Maximum of Interest	f_{max}	5.00
Fixed Sample Rate	$f_{s_{fix}}$	50.00
OUTPUT PARAMETERS		
Frequencies		**Hz**
Recommended Sample Rate	f_s	50.00
Low Pass Filter Frequency	f_{filt}	10.00
Global Maximum Frequency	$f_{win_{max}}$	10.00
Periods & Times		**T [s]**
Mininum Period	T_{min}	0.20
Maximum Period	T_{max}	20.00
Sweep Length	T_{b_i}	100.00
Minimum Sweep Length	$T_b]_{min}$	80.00
Total Sweep Length	T_{b_T}	200.00
Maximum Window Size	$T_{win_{max}}$	40.00
Min. Win. Size (Ligth Damping)	$T_{win_{min}}$	3.18
Min. Win.Size (Heavy Damping)		6.00

3 Parametric Model and Parameter Estimation

The identification process determines numerical values of a parametric model that will fit the real system. So, it is necessary to develop a parametric model. The aim here is to obtain a model that predicts the real system dynamics response with enough accuracy for control strategies purposes but simple enough to easily identify and manipulate it for control system design. In addition, a simple model contributes to evaluate the identified parameters physical meaning.

In order to achieve an appropriate accuracy-simplicity balance, and based on other successful aircraft identifications (see [9]), a high order hybrid formulation model (see [2] and the references given there) can be developed (see [3]). This model is very

similar to the one developed by Mettler in [6] and can be used as reference for RUAV identifications.

The modelling approach consists in formulating a non-linear ODEs system based on physical principles (mechanics, aerodynamics, momentum theory, blade element theory, etc.) and the convenient hypothesis. After that, this model is linearized around a generic flight condition and is expressed in terms of a matrix state-space system in the form,

$$\boldsymbol{M}\dot{x}(t) = \boldsymbol{F}x(t) + \boldsymbol{G}u(t - \boldsymbol{\tau}) \tag{2}$$

$$y(t) = \boldsymbol{H_0}x(t) + \boldsymbol{H_1}\dot{x}(t) \tag{3}$$

which is a formulation used by CIFER® equivalent to the standard space-state system formulation (see [**9**] and [**8**] for more details.

Matrices $\boldsymbol{M}, \boldsymbol{F}, \boldsymbol{G}$ and $\boldsymbol{\tau}$ are shown in Fig.4, where τ_r, τ_f are Rotor and Fly-bar time constants. Stability Derivatives written as $\widetilde{\#}$ embed flight conditions (e.g. $\tilde{X}_q \doteq X_q - mw_0$) just for compact purposes. Index $\#_0$ represents trim flight condition. Matrices $\boldsymbol{H_0}$ and $\boldsymbol{H_1}$ are,

$$\boldsymbol{H_0} = \begin{bmatrix} \mathbf{0_{3x3}} & \mathbf{0_{3x3}} & \mathbf{0_{3x2}} & \mathbf{0_{3x4}} & \boldsymbol{I_{3x3}} \\ \mathbf{0_{3x3}} & \boldsymbol{I_{3x3}} & \mathbf{0_{3x2}} & \mathbf{0_{3x4}} & \mathbf{0_{3x3}} \\ \mathbf{0_{3x3}} & \mathbf{0_{3x3}} & \begin{matrix} 0 & g\cos\theta_0 \\ -g\cos\theta_0 & 0 \end{matrix} & \mathbf{0_{3x4}} & \mathbf{0_{3x3}} \end{bmatrix} \tag{4}$$

$$\boldsymbol{H_1} = \begin{bmatrix} \mathbf{0_{6x3}} & \mathbf{0_{6x12}} \\ \boldsymbol{I_{3x3}} & \mathbf{0_{3x12}} \end{bmatrix} \tag{5}$$

State vector, measurements vector and control signals (x, y and u respectively) have the structure.

$$x = [u_i \quad v_i \quad w_i \quad p \quad q \quad r \quad \phi \quad \theta \quad a \quad b \quad c \quad d \quad u \quad v \quad w]^T \tag{6}$$

$$y = [u \quad v \quad w \quad p \quad q \quad r \quad a_x \quad a_y \quad a_z]^T \tag{7}$$

$$u = [\delta_{lon} \quad \delta_{lat} \quad \delta_{col} \quad \delta_{ped}]^T \tag{8}$$

where u, v, w are body velocities at estimated Center of Gravity (CG) and u_i, v_i, w_i are body velocities at real CG. p, q, r are the angular rates, ϕ, θ are the roll and pitch Euler angles, a_x, a_y, a_z are the body accelerations, a, b are Rotor Tip Path Plane longitudinal and lateral inclination angles, c, d are Fly-bar Tip Path Plane longitudinal and lateral inclination angles. The control signals are $\delta_{lon}, \delta_{lat}$ longitudinal and lateral cyclic control signals, and δ_{col} δ_{ped} are collective and tail rotor control signals. Note that yaw Euler angle (ψ) does not appear in the state vector since it is full decoupled of the system and determined by its kinematical equation.

Helicopters hover and axial flight dynamic response change significantly for cruise flight. However, air density variations can be neglected in RUAVs flight envelope, which implies the flight condition is independent of altitude. Therefore, at least two models have to be identified: one for hover condition and another for a particular cruise condition. The method is valid to identify both flight conditions. The difference

will be the different parameters remaining at the end of the identification (see section 5) and its values. Nevertheless, the hover model is valid up to speeds about $\sim 7-8\ m/s$ (see [6]).

Here, taking into account the verification experiments presented in section 8, only a hover model is presented.

Note that a simple rigid body formulation is not enough to give a complete characterization of the RUAV dynamics. Rotor is actually a dynamic system that responds both to control signals and fuselage state. Hence the rotor forces and moments are not instantaneous and a dynamic coupling will exist between these systems. This coupling is emphasized when gyroscopic rotor inertia is of the same order as fuselage inertias – it is the case of RUAVs – because the rotor effect becomes more relevant. So, in addition, a simplified set of rotor dynamics equations are needed. In practice, if roll and pitch frequency responses clearly show characteristics of a second order system, they constitute a trademark of such coupled dynamics.

The physical effects modelled (see [3]) are listed below. Each one has associated a colour that matches each effect with their correlative parameters in the system structure shown in Fig.4.

- Fuselage rigid body dynamic equations. It can be clearly differentiated rotational – roll, pitch and yaw – dynamics (yellow) and translational – horizontal and axial – dynamics (magenta).
- Fuselage roll and pitch kinematic equations. Yaw kinematics is decoupled and they do not need to be included in the identification (blue).
- Rotor flapping dynamics coupled to fuselage (red).
- Fly-bar flapping dynamics coupled to rotor (green).
- Corrections associated to uncertainty of the CG, as considered in [**6**] (grey).

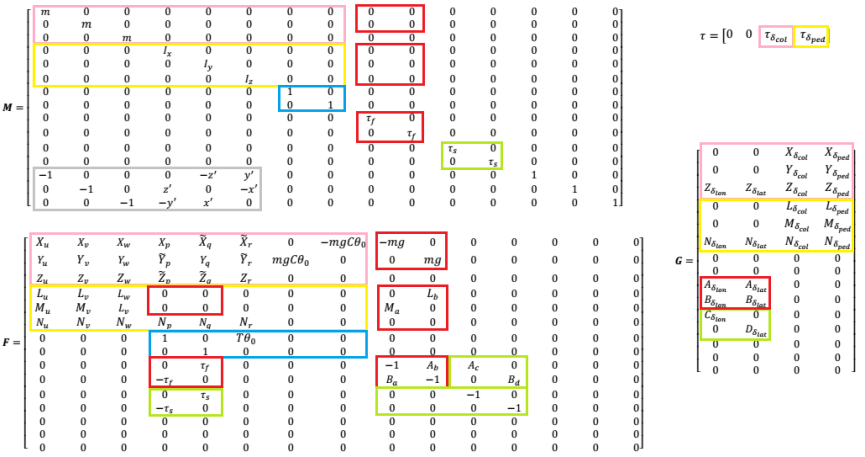


Fig. 4. Model matrices. Each colour corresponds to a modelled physical effect.

CIFER® system identification consists in an iterative process. Therefore a set of initial conditions are needed for the different parameters involved. The aircraft model in the identification problem can be shown as an equation system where there are more unknowns – that is, the parameters to identify because vectors x, u and y are now inputs from flight data – than equations. Then, it is critical to begin identification from a good starting point.

The initial conditions are usually fixed to zero if nothing is known about it, or fixed to a value close to cero if a parameter sign is known. Sometimes an estimation can be obtained using other measurable data (e.g. rotor and fly-bar time constants, τ_r and τ_f respectively, can be estimated from aerodynamic blade foil analysis) or employing similar aircraft parameters (e.g. non-dimensional stability derivatives).

Table 2. Estimated parameters

Inertias [kg·m^2]				**Rotor**	**Fly-bar**
I_{xx}	0.12919	Blade Lift Slope	C_{l_α}	7.491	7.549
I_{yy}	0.72918	Inertia [kg·m^2]	I_b	0.2776	0.0126
I_{zz}	0.60011	Lock Number	γ	1.5732	0.7584
I_{xy}	-0.00436	Time Constant	τ	0.0747	0.1550
I_{xz}	-0.02609				
I_{yz}	-0.00050				

Moreover, some parameters should be known a priori. Some of them can be easily and directly measured (e.g. masses), while others are difficult to determine accurately. This is the case of the inertia terms. If a CAD model of the RUAV is available, they can be accurately calculated. Another alternative is to determine them using pendulum test for inertia determination (see [10]) but the RUAV can be damaged in the process and it needs specific tools and equipment. Usually in literature this problem is solved condensing these inertias into the forces terms (e.g. defining stability derivatives such as $M_u^* \doteq M_u/I_y$) but model parameters physical values and meaning are lost.

For CB-5000 identification purposes, a CAD model was not available and the RUAV configuration was continuously changing (e.g. components on avionics tray) so the inertia terms where estimated using a concentrated masses model. Main RUAV parts and its position were weighted and measured.

On the other hand, some system dynamics equations can be oversimplified in order to obtain a SISO transfer function. For instance, pure axial dynamics (ascent-descent movements, associated to collective control signal) are part of the MIMO system, but can be reduced to a SISO model considering only on-axis terms (neglecting any coupling dynamics) as can be shown equation (9),

$$\frac{w}{\delta_{col}} = \frac{Z_{\delta_{col}}/m}{s - Z_w/m} e^{-\tau_{col}s} \tag{9}$$

where w/δ_{col} is the output-input pair (w is the upward velocity and δ_{col} is the collective control signal), m is the RUAV total mass, $Z_{\delta_{col}}$ and Z_w are the on-axis Stability Derivatives, and τ_{col} is the total aerodynamic delay since signal control is commanded and a RUAV upward velocity response is observed.

A first approximation – for initial conditions in MIMO system purposes – of the above last three terms can be obtained using SISO system identification. CIFER® NAVFIT module performs rapid frequency identification tasks for these SISO systems. This last technique for initial condition estimation was used here for estimating on-axis Stability Derivatives and delays for axial and yaw dynamics.

4 Telemetry Postprocessing

Once flight data are obtained, a post processing is needed before they can be treated by CIFER®. Otherwise it is necessary to waste time trying to identify an inconsistent data set. Typically the licensed software SMACK (SMoothing for AirCraft Kinematics, see [11], [12], [13]) , developed by NASA for flight data reconstruction purposes, is used. Alternatively, it is possible to use the Matlab application presented in [3]. The fundamentals and techniques for flight data treatment implemented in this application are gathered in [7].

The tasks performed to the rough data logs are the following: Units, axes and signs convention homogenization; Magnitudes translation from IMU to the assumed CG; Outliers and dropouts suppressing; Selection of data of interest (frequency sweeps) from rough data logs; Analysis of Kinematic Consistency; Trim condition determination and perturbation magnitudes construction; Control signals processing, depending on actuators and the particular RUAV; High frequency filtering, and data conversion to a format supported by CIFER®.

5 Identification and Verification Process

CIFER® is a tool to solve the aircraft system identification problem based on the frequency response approach. The Army/NASA Rotorcraft Division at Ames Research Center in United States jointly developed it in 80's. Mainly, this software is composed by a set of six core programs that carry out the basic computations of the frequency response identification method. They are FRESPID, MISOSA, COMPOSITE, NAVFIT, DERIVID, and VERIFY. In general, these programs are employed in this. More information about the software can be consulted by the reader in [8] and [2].

Once flight data are suitably conditioned, frequency response databases can be extracted from them employing FRESPID. This module computes this database by using a FFT (Fast Fourier Transform) variant. In terms of computational calculations, the quality of this database does not only depend on the dynamic information inside these flight data, but also is depends on the windowing configuration parameters. It is important to configure those variables in FRESPID as closely as possible to the optimal window size (see Table 3) computed using parameters obtained in flight maneuver design (Table 2). Coherence function ($\hat{\gamma}_{xy}^2$ see [9], [3]) offers a good

reference of computed dataset quality for each frequency considered. The frequency range where this function is good enough (typically $\hat{\gamma}_{xy}^2 \geq 0.6$) basically determines the frequency range where model can be fitted.

Table 3. Optimal FRESPID Windowing

FRESPID Window	TWIN $[s]$	$\omega_{máx}$ $[rad/s]$
A	40.000	62.832
B	20.000	62.832
C	23.000	62.832
D	14.500	62.832
E	6.000	62.832

Now, as it is already known, the objective is to fit the parametric model to the frequency response database. This fitting procedure is supervised checking Both bode diagrams for flight data and identified model (see Fig. 5). It is recommended to select conservative frequency ranges for each frequency-response database pair (e.g. w/δ_{col}), that is, those ranges where coherence function is at its highest level. These ranges will be increased or reduced in future model refinements.

Moreover, as was commented in Section 3, the identification process can be seen as solving a mathematic system where there are more unknowns than equations. That was the reason why a good initial condition set for parameters is needed but is also the reason why the CIFER® user has to employ a determined algorithm in the identification tasks. That is, the user should include or exclude some system equations, parameters, constraints among these parameters and frequency response pairs to be identified while identification procedure is taking place.

This algorithm will depend on the dynamic system and model equations to be identified and essentially, it lies in breaking down the model in small subsystems. As the identification is taking place, the initial model is growing little by little to include new terms and couplings until it identifies the whole model. Therefore, to build a model based on physical principles simplified using reasonable hypotheses is a key aspect in order to understand each model parameter meaning and create this identification algorithm.

For example, attending to the model structure in Fig. 4 in RUAV modelling can be shown that whole helicopter rotational dynamics are controlled via control signals δ_{lat} and δ_{lon} through rotor flapping dynamics and a fuselage-rotor coupling terms associated to the rotor physical linkage (red). In addition, there are easy good initial conditions for the parameters involving this subsystem – τ_r (rotor time constant), A_b and B_a (lateral and longitudinal rotor flapping coupling effect), M_a and L_b (fuselage-rotor linkage). This coupling corresponds basically to the I/O on-axis pairs p/δ_{lat} and q/δ_{lon}, and I/O off-axis pairs q/δ_{lat} and p/δ_{lon}, where p and q corresponds to roll and pitch angular rates.

After that, the equations for the fly-bar effect (green) can be included, and sequentially translational horizontal dynamics, CG uncertainty, axial dynamics, and yaw dynamics. Finally the rest of coupling terms (such as L_v, M_u) are added in a model refinement process.

However, it is unknown a priori if some parameters, such as some aerodynamic Stability Derivatives, should appear in the model or they should be dropped.

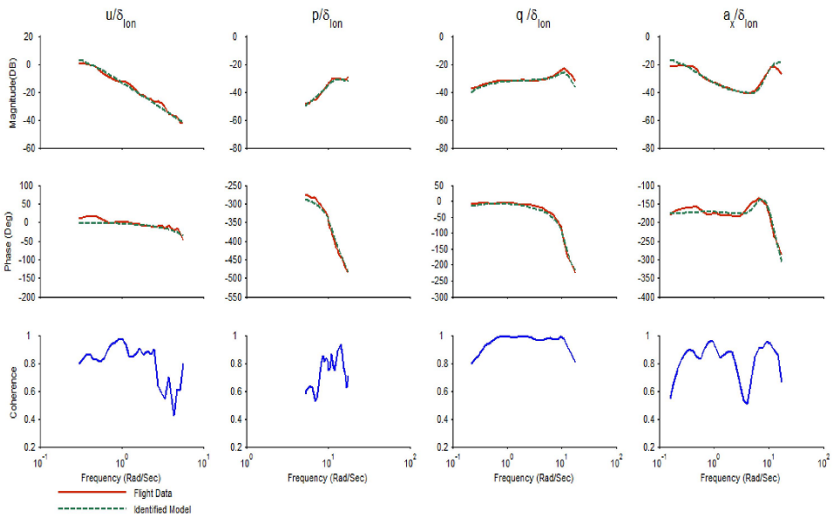


Fig. 5. Typical DERIVID output bode diagrams. It will appear as many diagrams as I/O pairs identified.

For this purpose it is important the use of CIFER®'s statistical parameters and cost functions derived during each identification step. According to [2], these variables can determine which model parameters are representative or not for the frequency response database adjustment (insensitivity analysis), and which parameters seem to adjust the same effects or are correlated (Cramer-Rao Bounds, Ellipsoid of Confidence) and the level of fitness between the frequency response database and the identified mathematical model. Once this point is reached, it is remarkable to analyse model parameters obtained – their sign, magnitude and physical meaning – and check model eigenvalues – number and level of stable or unstable modes and compare, if it is possible, with similar aircraft.

Assume that the flight data of a dynamic mode associated to a parameter in the mathematical model (e.g. X_u or Y_v) has not been excited enough (e.g. longitudinal translational dynamics, associated to low frequencies that in practice are complex to excite correctly). In that case, CIFER®'s statistical parameters will discard these terms during identification because there is not good enough information about that dynamics in flight data. This is the reason why flight test should collect as much good data as possible.

Also as many I/O pairs as possible should be included for RUAV dynamic characterization. Including less pairs – typically off-axis effects – can make the model unable to predict with enough accuracy the real aircraft dynamics. However,

Table 4. I/O pairs used in the hover identification. Colour scheme represents the main relationship with the corresponding physical effect associated.

I/O Pair	$f_{min}\,[rad/s]$	$f_{max}\,[rad/s]$	Colour
u/δ_{lon}	0.303	5.605	
p/δ_{lon}	1.360	17.520	
q/δ_{lon}	0.219	17.520	
a_x/δ_{lon}	0.157	17.520	
v/δ_{lat}	0.219	5.605	
p/δ_{lat}	0.350	17.970	
q/δ_{lat}	5.863	29.690	
r/δ_{lat}	0.750	1.601	
a_y/δ_{lat}	0.184	17.520	
w/δ_{col}	0.192	22.000	
r/δ_{col}	0.295	15.080	
a_z/δ_{col}	0.750	17.520	
r/δ_{ped}	0.661	18.500	

including too many pairs can have the same effect because it is possible that some model equations might not catch some non-modelled dynamics (and, in consequence, they are not relevant for aircraft dynamics characterization).

Table 5 presents final model parameters identified for a hover flight condition and their respective insensitivities (I) and Cramer-Rao bounds (CR). Parameters not appearing in that table were dropped from model structure in model refinements.

Table 5. Identified parameters value and their statistical parameters

PARAM	VALUE	CR [%]	I [%]	PARAM	VALUE	CR [%]	I [%]
X_u	-0.594	11.24	4.579	τ_r	0.039	8.41	1.764
X_v	3.167	35.34	6.768	$A_b = B_a$	0.389	5.50	0.814
X_p	-55.430	5.82	1.542	A_c	1.297	12.21	0.816
X_r	4.042	34.18	2.238	B_d	1.172	9.47	0.592
Y_u	1.157	18.85	6.185	τ_f	0.099	7.03	0.900
Y_v	-3.097	12.65	1.992	$Z_{\delta_{col}}$	-3.068	2.77	1.327
Y_q	56.850	10.22	2.328	$N_{\delta_{lon}}$	-0.742	33.05	2.290
Y_r	30.030	26.10	2.309	$N_{\delta_{col}}$	-0.225	5.04	2.515
Z_w	-6.054	11.93	5.701	$N_{\delta_{ped}}$	0.905	3.89	1.945
L_v	0.206	13.23	3.152	$A_{\delta_{lon}}$	$1.183{\cdot}10^{-3}$	21.01	3.492
L_b	44.640	7.29	0.789	$A_{\delta_{lat}}$	$-2.900{\cdot}10^{-4}$	12.68	2.334
M_u	-0.102	18.15	5.939	$B_{\delta_{lat}}$	$-3.162{\cdot}10^{-3}$	9.05	1.217
M_a	-73.150	8.11	0.809	$C_{\delta_{lon}}$	$-4.491{\cdot}10^{-3}$	7.81	0.829
N_v	0.468	13.02	3.257	$D_{\delta_{lat}}$	$6.644{\cdot}10^{-3}$	6.50	0.311
N_p	-2.009	21.92	2.746				

Once the model has been identified, the next step is the model verification using VERIFY module. Time domain verification flight data test (doublets steps maneuvers) are compared in time-domain with time-domain identified math model prediction, using as inputs the same fight tests control signals (Fig. 6).

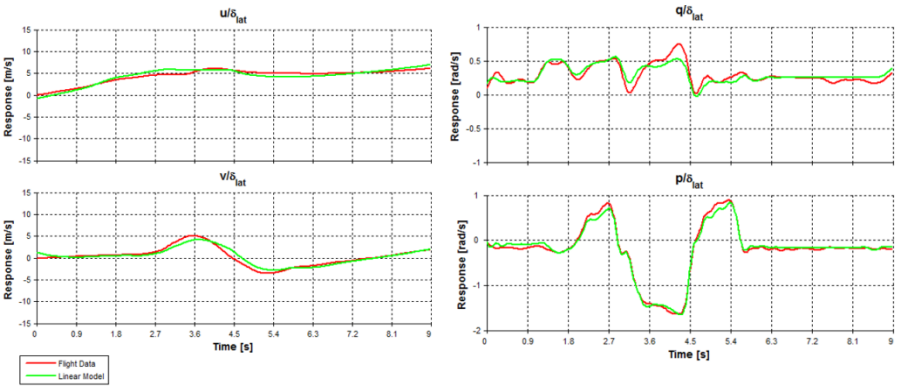


Fig. 6. Example of model verification using VERIFY

6 Alternative Two Rigid Bodies UAV Kinematic Model

An alternative modelling approach for the same RUAV is presented in order to compare its behaviour with the model obtained using CIFER®. This model, derived from basic mechanical principles, consists of the dynamic and kinematic equations of two rigid bodies: the main rotor and the fuselage. The controller and simulation model are based on the work developed in [15]. The main hypothesis behind this approach is that the ratio between the main rotor fuselage masses to the main rotor rotational speed is higher than in the full size helicopter case. All these characteristics imply that the inertial effects of the main rotor cannot be neglected.

The full mathematical model of the UAV is composed of two main parts: the mechanical model, and the torques and forces generated by the aerodynamic model. The aerodynamic of a helicopter is very complex, but by using experimental data we concluded that a linear algebraic approximation is good enough to be used on the design process of the control system. An additional drag force source was added using a simple equivalent sphere drag model. In this case, drag module and direction depends on the velocity vector of the helicopter.

In the model the following generalized coordinates are used: $q_{1,2,3}$ that define the position of the CM in the inertial frame GS, $u_{1,2,3}$ are the velocities in GS, $q_{4,5,6}$ are the Euler angles in the helicopter frame, $u_{4,5,6}$ are the angular velocities (see [15]). The inputs of the system are the main rotor force f_3^H and the torques $t_{1,2,3}^H$. The scheme of all the components that are used in the mathematical simulation model is shown in Fig. 7 (see [14] for more details).

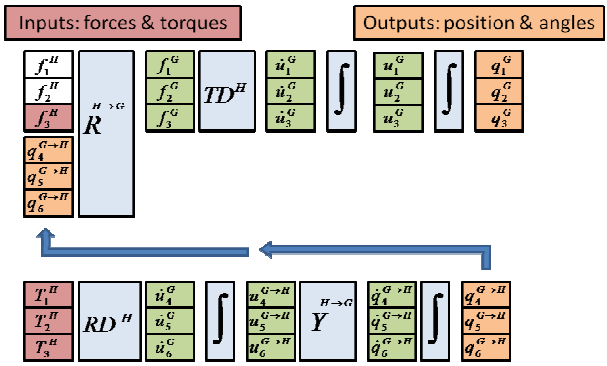


Fig. 7. Two rigid bodies' mathematical structure, taken from [16]

The model is broken down into two main parts:

- Translational dynamics are first characterized by a rotating block $R^{H\to G}$ that transforms forces in HS into the GS according to Euler angles q_4, q_5, q_6. The Newton's Second Law is applied in an inertial frame by means of block TD^H. Since previous operation results in linear accelerations $\dot{u}_1, \dot{u}_2, \dot{u}_3$, the subsequent two columns of integrators are required to eventually obtain linear positions q_1, q_2, q_3.
- The law of angular momentum conservation is applied for rotational dynamics in block RD^H, whose inputs are the torques T^H in HS. In order to simplify the equations, two assumptions are accounted for. The first is that the CM of the fuselage is located below the rotation axis of the main rotor. The other assumption is that the angular velocity around the third axis u_6 is constant. This is true since the control of the rotation along the third axis is realized by a separated control loop implemented by a commercial gyroscope. Finally, $Y^{H\to G}$ block implement the kinematics equations that are necessary to transform angular velocities into derivatives is Euler angles.

7 Description of Guidance and Control Algorithms

The control system for the autonomous helicopter is based on the inverse relationships of the translational, kinematical and dynamic equations (described in the previous section) combined with linear PID controllers [16] and [5]. The tuning of most of the control constants is made by a pole placement method, using the cancellation between the inverse relationships of the control and the mathematical model. The few remaining control constants are tuned by using an experimental methodology that consists on introducing perturbations in the control variables.

The controller is composed of two nested loops (see Fig. 8). In the outer loop, the position and velocity errors are used to calculate the desired accelerations that are needed to reduce those errors. Controller references are denoted by variables with asterisk. Position and angle variables are denoted by q_i with $i = 1, 2, \ldots, 5$

corresponding to x, y, z, ϕ (roll) and θ (pitch) respectively. Velocity and rate variables are accordingly denoted by u_i. This control loop is implemented by means of three PID controllers, one for each degree of freedom. Next, the resulting accelerations are converted into the desired orientation of the main rotor plane and the needed lifting force ($F^*_{MR,3}$).

The inner control loop computes the main rotor torques ($T^*_{MR,1}$ and $T^*_{MR,2}$) from the desired and current orientation. The orientation around the third axis (yaw dynamics, given by the yaw angle q_6) is controlled by a separated control loop.

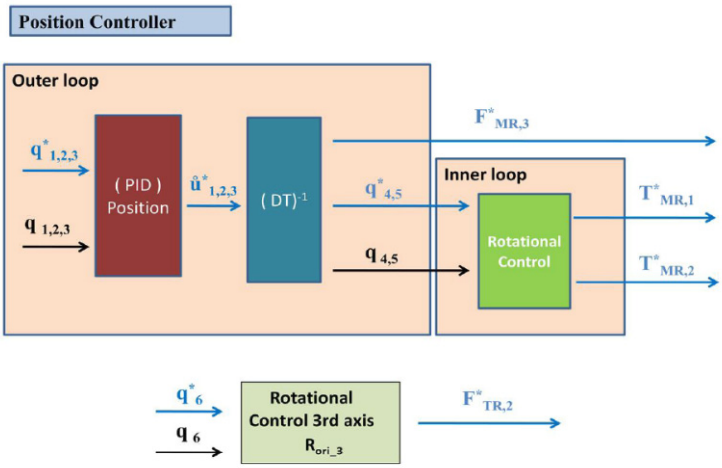


Fig. 8. Controller flow-chart and structure taken from [16]

8 Control and Model Simulation and Comparison with Experimental Data

In order to completely verify, compare and validate the presented method, both the CIFER® model and the alternative model are integrated via Simulink with the UAV's real autopilot software. A flight test was performed where the real RUAV with the autopilot was following specific position references (flight plan). Those recorded references were used as input for the models. Fig. 9 shows a very simple schematic of this idea.

The flight plan is focused on testing the lateral and longitudinal model dynamics to compare them with the real helicopter behaviour when they are linked to the autopilot. Firstly, a lateral movement of ±50m is commanded (time 145s to 190s approximately) as can be shown in Fig. 10. The associated roll angle (ϕ) and roll rate (p) can be found on Fig. 11 while the corresponding lateral control signals are shown on Fig. 12.

The same procedure has been performed with a longitudinal movement of ±50m around time 210s to 270s (Fig. 10) with the associated pitch angle (θ) and pitch rate (q) states variables (Fig. 11), and the longitudinal control signals (Fig. 12).

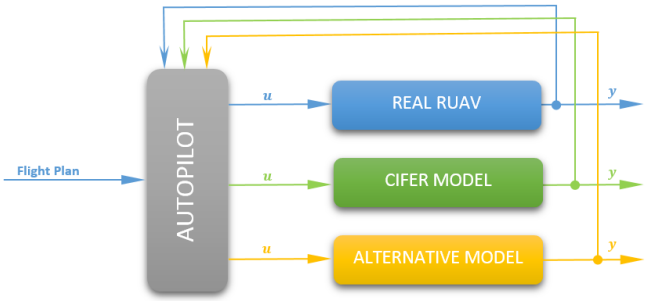


Fig. 9. Verification experiments schematic

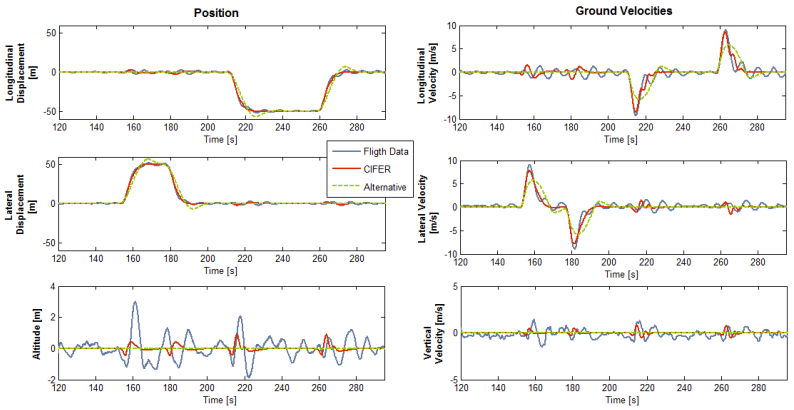


Fig. 10. Position references and ground velocities

As can be shown on the graphs, both models are suitable for general simulation purposes. Nevertheless, the following differences can be found between the different models and the experimental data.

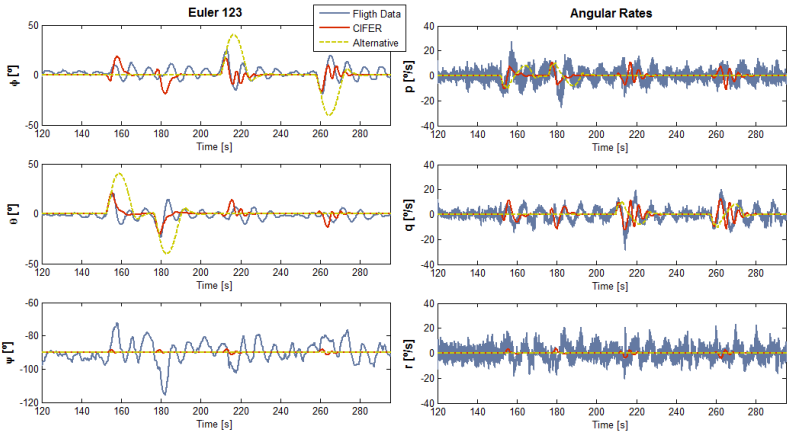


Fig. 11. Attitude and angular rates

On the one hand, if it is compared the response of both models with the real telemetry, it can be concluded that a CIFER® simulation in terms of position and velocities is closer to the real data (Fig. 10). The controller used in both simulations is the same to the one that was tuned previously for the real aircraft. This means that CIFER® model captures some effects that the two bodies' model cannot reproduce. On the other hand, Fig. 10 shows that real position and velocities are continuously fluctuating with approximate amplitude of two meters while models simulations do not exhibit this behaviour. This can be explained in terms of sensor states accuracy. That is, real RUAV system employs sensor measurements to estimate its state variables but these measurements have a position error about two meters because the INS/GPS does not have a DGPS (Differential GPS). Moreover, the real helicopter is also affected by non-modelled gusts, winds and other perturbations.

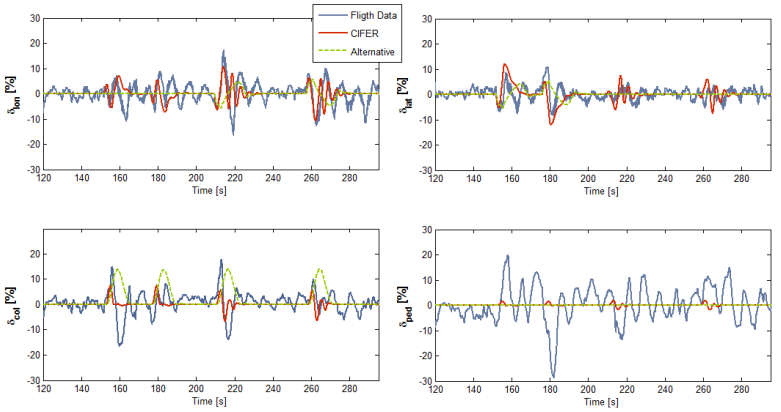


Fig. 12. Control signals

Fig 12. shows that, in spite of the fluctuating control corrections due to the lack of sensor accuracy mentioned before, the control signal of the controller using CIFER® model is more similar to the real signals in shape and magnitude that the signal generated by the controller using the two rigid bodies model. Another difference between CIFER® and two rigid bodies' model is that CIFER® capture better off-axis dynamics. An example is the grade of coupling between translational and collective dynamics. This effect can be shown by analysing the collective correction associated with the longitudinal movement of the helicopter around time 210s to 270s. At the beginning of the translational movement the lifting force should be tilt into the movement direction. However, this tilting implies that its vertical projection now cannot make up for the aircraft weight. In order to compensate it, thrust vector module needs to be increased, and this is done by increasing collective control signal. Once the aircraft acquires forward velocity it needs less collective input to maintain the same altitude because of the new air inflow thought main rotor. This value is even lower that the required for hover flight condition. When the helicopter slows down, the lifting force should be tilted into the opposite direction of the movement direction, the thrust vector module needs to be increased again in order to compensate it. The reader can extract similar and other considerations from those figures.

Taking into account all the facts mentioned before, CIFER® modelling seems to be more suitable for control and more realistic simulation purposes. Nevertheless both modelling techniques can be widely used for sense and avoid, guidance, trajectory generation and other applications.

9 Conclusions and Future Work

In this work, we have presented the entire modelling process of a RUAV from frequency sweeps capture to the verification and validation process. This modelling technique has been compared with an alternative two rigid bodies' model and the real flight data using the same flight plan and control system.

Furthermore, we have described several key aspects that should be taken into account on small UAVs frequency system identification. In addition, an RUAV control system architecture has been presented.

CIFER® modelling approach has the advantage of being based on direct experimental data but it has the drawback that it requires an elaborated flight tests. These tests are usually difficult to accomplish and sometimes requires risky manoeuvres.

The presented simulation results are fairly similar to the real flights data and it can be concluded that the degree of complexity versus accuracy of these models is good enough for general UAV control and guidance applications.

Identification results could be improved by using computer aided flight tests (see [3]) and increasing position accuracy using a DGPS sensor. An alternative non-linear model can be obtained using several linear models identified in different flight conditions (see [18]).

Acknowledgements. This work was partially funded by the FP7 European Commission under the Integrated Project ARCAS (FP7- 287617) and the Project CLEAR (DPI2011-28937-C02-01) funded by the Ministerio de Ciencia e Innovacion of the Spanish Government.

References

1. Kondak, K., Maza, I., Ollero, A., Bernard, M.: Autonomous Transportation and Deployment with Aerial Robots for Search and Rescue Missions. Wiley Blackwell 28(6), 914–931 (2011)
2. Tischler, M.B., Remple, R.K.: Aircraft and Rotorcraft System Identification. Engineering Methods with Fligth Test Examples. AIAA (2006)
3. Sánchez Montaño, I.: Indentificación en frecuencia de UAVs de ala giratoria usando CIFER. Aplicación al CB-5000. Internal Report. University of Seville (2013)
4. Kalman, R.E.: A New Approach to Linear Filtering and prediction problems. Research Institute for Advanced Study. Baltimore Md (1960)
5. Alarcón, F., Jimenez, A., Viguria, A., Bejar, M., Ollero, A., Santamaría, D.: Model-Based design development and validation for UAS critical software. Journal of Intelligent and Robotic Systems 65(1-4), 103–114 (2012)

6. Mettler, B.: Identification Modeling and Characteristics of Miniature Rotorcraft. Kluwer Academic Publishers (2003)
7. Klein, V., Morelli, E.A.: Aircraft System Identification. Th. and Practice, AIAA (2006)
8. CIFER user's guide
9. Theodore, C.R., Tischler, M.B.: Rapid Frequency Domain Methods for UAV Flight Control Applications. In: AIAA Atmospheric Flight Mechanics Conf., Austin, Texas (2003)
10. Miller, M.P.: An Accurate Method of measuring the moments of inertia of Airplanes, vol. Naca-tn-351 (October 1930)
11. Wingrove, R.: Applications of a Technique for Estimating Aircraft States from Recorded Flight Test Data. AIAA (1972)
12. Bach, R.E.: State Estimation Applications in Aircraft Flight-Data Analysis: A User's Manual for SMACK, NASA (1991)
13. Fletcher, J.W.: Obtaining Consistent Model of Helicopter Flight-Data Measurement Errors Using Kinematic-Compatibility and State-Reconstruction Methods. American Helicopter Society Annual Forum (1990)
14. Béjar, M.: Methodology and Techniques for designing control systems of autonomous helicopters, PhDThesis, University of Seville (2009)
15. García, M., Viguria, A., Ollero, A., Santamaría, D.: Position-based velocity control system for rotary-wing UAVs. In: Proceedings of the 1st Workshop on Research, Development and Education on Unmanned Aerial Systems, RED-UAS (December 2011)
16. Kondak, K., Bernard, M., Losse, N., Hommel, G.: Elaborated modeling and control for autonomous small size helicopters. In: ISR/ROBOTIK 2006 Joint Conference on Robotics (2006)
17. La Civita, M.: Integrated Modeling and Robust Control for Full-Envelope Flight of Robotic Helicopters, Pittsburgh: Ph D. Carnegie Mellon University (2002)

Fault Estimation and Control for a Quad-Rotor MAV Using a Polynomial Observer. Part I: Fault Detection*

Gerardo R. Flores-Colunga[1], H. Aguilar-Sierra[2], R. Lozano[3], and S. Salazar[2]

[1] Heudiasyc UMR 6599 Laboratory,
University of Technology of Compiègne, France
[2] LAFMIA UMI 3175, Cinvestav, México
[3] Heudiasyc UMR 6599 Laboratory,
UTC CNRS France and LAFMIA UMI 3175, Cinvestav, México
{gfloresc,rlozano}@hds.utc.fr, haguilar@ctrl.cinvestav.mx,
sergio.salazar.cruz@gmail.com
http://www.hds.utc.fr

Abstract. This work addresses the problem of fault detection and diagnosis (FDD) for a quad-rotor mini aerial vehicle (MAV). Actuator faults are considered on this paper. The basic idea behind the proposed method is to estimate the faults signals using the extended state observers theory. To estimate the faults, a polynomial observer is presented by using the available measurements and know inputs of the system. In order to investigate the observability and diagnosability properties of the system, a differential algebra approach is proposed. Furthermore, an evaluation function depending on the system states is developed, in order to be used in a controller, which will compensate the failures. The effectiveness of the methodology is illustrated by means of numerical simulations and some experimental tests.

Keywords: Quad-rotor, polynomial observer, diagnosability, fault detection and diagnosis.

1 Introduction

The growing development in research on MAVs and the consequent improvement of technologies like microcomputers, vision systems and other sensor devices, have increased the performance requirements of such kind of systems. Problems related to trajectory tracking, flight-formation, vision-based localization and lately MAV equipped with manipulators, have been widely researched in the last few years. Therefore, a good performance in the inner-loop of such flight envelopes is needed.

* This work was partially supported by the Institute for Science & Technology of Mexico City (ICyTDF) and the Mexican National Council for Science and Technology (CONACYT).

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_11,

A wide range of nonlinear control techniques like backstepping [1], [2], singular perturbation techniques [3], sliding modes and switching control [4], [5], have been treated to deal with the complex dynamics of the quad-rotor.

Due to the high cost of the MAV equipment, it is imperative to provide such systems with a fault-control loop, responsible for the identification of possible faults presented at any time of the flight envelope.

Motivated by the *fault diagnosis problem*, which is the problem of observing fault signals, and the necessity of developing sufficiently robust controllers to cope the presence of likely faults, this research work deals not only with the MAV stabilization problem, but also with the identification of actuator faults. Few works dealing with the fault diagnosis problem applied on quad-rotors are presented in the literature [6], [7], [8], [9], [10].

Taking the attitude, position, angular and translational velocities of the quad-rotor MAV as available measurements, we develop a solution for the fault diagnosis problem by means of the differential algebraic approach. With this approach, it is possible to construct a bank of observers in order to implement a scheme of residual generation for fault diagnosis [11], or implement a control law based on state estimation [12]. Thus, it is possible to combine different schemes of nonlinear observers. In [13], the authors present a reduced order and a sliding mode observer, to reconstruct faults in an experimental task, for the case when only one output is available. A reduced order observer and an algebraic observer is presented in [14]. The approach given in [15] is used for fault detection and fault estimation of a wound-rotor induction motor (WRIM). In [16] a polynomial observer, a reduced order observer and a sliding mode observer are used in order to estimate an reconstruct the system states and faults for the case of multiple available outputs. In [17], the polynomial observer is used for the synchronization of chaotic systems.

The paper is organized as follows. The fault diagnosis problem is formulated in Section 2. In order to estimate not only the system states but also the faults dynamics, an extended Luenberger observer called polynomial observer is developed in Section 3. Next, in Section 4, the results previously obtained is applied to the Quad-rotor MAV. Section 6 presents some simulation results for the fault reconstruction problem. Finally, some conclusions and future works are presented in section 7.

2 Fault Diagnosis Problem

The Fault Detection and Diagnosis (FDD) task has the goal of detecting the presence of a fault and construct an estimate of the unknown fault dynamics. Such faults can affect directly the performance of the system components. Therefore, a FDD scheme provides all the necessary information about faults, such as presence (time), type (actuator/sensor) and dynamics (magnitude and form). Thus, based on this information, it is possible to design a system reconfiguration to minimize the fault effects. We begin by defining the terms *fault* and *failure* as follows:

Fault: An undesired change in a system parameter or variable that reduces the performance/magnitude of one component of its nominal value. In summary, a fault is an *unacceptable tolerable malfunction.*

Failure: A complete breakdown of the system, caused by a catastrophic malfunction of one or more components of the system. In summary, a failure is an *intolerable malfunction.*

Throughout this work, we describe a class of nonlinear systems with faults as follows

$$\begin{aligned} \dot{x}(t) &= g(x, u, f) \\ y(t) &= h(x, u) \end{aligned} \tag{1}$$

where

$x \in \mathbb{R}^n$ is the state vector
$u \in \mathbb{R}^m$ is the vector of known inputs
$f \in \mathbb{R}^\mu$ is the faults vector (unknown inputs)
$y \in \mathbb{R}^p$ is the outputs vector

In this paper, we consider only the case of faults in the actuators. So, we introduce the concept of observability and diagnosability in the field of the differential algebra.

2.1 Observability and Diagnosability Condition

The observability and diagnosability notion of a system, linear or nonlinear in the differential algebra approach need a basic definition. Further details can be found in [13].

Definition 1. *For the system described by (1) a state x is said to be observable if it is possible to estimate the state by means of the available measurements of the system, so we say that x is observable if it is algebraically observable, i.e., the state x satisfies a polynomial equation in terms of u and y and some of their time derivatives:*

$$P\left(x, y, \dot{y}, \ddot{y}, ..., y^{(n)}, u, \dot{u}, \ddot{u}, ..., u^{(n)}\right) = 0 \tag{2}$$

Definition 2. *A fault f is said to be diagnosable if it is possible to estimate the fault from the available measurements of the system, i.e., f is diagnosable if it is algebraically observable if it satisfies a polynomial equation in terms of u and y and some of their time derivatives:*

$$P\left(f, y, \dot{y}, \ddot{y}, ..., y^{(n)}, u, \dot{u}, \ddot{u}, ..., u^{(n)}\right) = 0 \tag{3}$$

Remark 1. The diagnosability condition is independent of the observability of the system.

Referring to system (1), the vector f contains the unknown inputs. In order to estimate its uncertain dynamics, the state vector is extended to deal with the fault vector. So, we can rewritten the system in an extended form as follows

$$\begin{cases} \dot{x}\,(t) = g\,(x, u, f) \\ \dot{f}_k(t) = \Omega_k\,(x, u, f) \qquad 1 \leq k \leq \mu \\ y\,(t) = h\,(x, u) \end{cases} \tag{4}$$

The following results from the theory of differential algebraic are on useful tool to determine whether a fault can be reconstructed from the know inputs and available outputs.

Theorem 1. *Assume that the system (1) is diagnosable, then the number of faults is less or equal to the number of available measurements (outputs), i.e.*

$$\mu \leq p$$

The proof of Theorem 1 can be seen in [18].

3 Polynomial Observer

The polynomial observer, is a scheme that combines two kinds of observers. The first one is like an extended Luenberger observer which is used to reconstructing and estimate the system states. While the second one is a free mode observer, which has the function of reconstruct an estimate of the faults dynamics. The polynomial observer can be seen as a Taylor series, where the first-order term is the observed state, thus improving the performance and speed of convergence including terms of high-order correction in the structure. It is worth mentioning that this scheme is considered for the case of multiple outputs available, where the terms of higher order correction are odd powers and are a linear combination of the observation errors of each output available and the order of the polynomial compensations is a determining factor for the parameter "q".

Consider the system with presence of faults, given in (4), the observation problem for the unknown vector of faults can be estimated using a polynomial observer. Therefore the system (4) can be rewritten as

$$\begin{cases} \dot{x}\,(t) = Ax + \Psi\,(x, \bar{u}) \\ \dot{f}_k(t) = \Omega_k\,(x, \bar{u}) \qquad 1 \leq k \leq \mu \\ y\,(t) = Cx \end{cases} \tag{5}$$

where $\|\Omega_k(x, \bar{u})\| \leq N, N \in \mathbb{R}^+$ and $\Psi(x, \bar{u})$ is a nonlinear function that satisfies the Lipschitz condition, with $\bar{u} = (u, f)$ uniformly bounded.

$$\|\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})\| \leq L\|x - \hat{x}\| \tag{6}$$

3.1 Observer Design

Now, consider the system with faults (5), the following lemma describes the construction of the polynomial observer.

Lemma 1. *Let the system (5) be algebraically diagnosable, then, the following nonlinear system is a full order state observer for the given system*

$$\begin{cases} \dot{\hat{x}}\,(t) = A\hat{x} + \Psi\left(\hat{x}, u, \hat{f}\right) + \\ \qquad\qquad + \sum_{i=1}^{p}\sum_{j=1}^{q} K_{ij}(y_i - C_i\hat{x})^{2j-1} \\ \dot{\hat{f}}_k(t) = \sum_{l=1}^{q} \bar{K}_{kl}(f_k - \hat{f}_k)^{2l-1} \end{cases} \tag{7}$$

Where
$A \in \mathbb{R}^{n\times n}$
$\hat{x} \in \mathbb{R}^{n\times 1}$ is the estimate of the state x
$\hat{f}_k \in \mathbb{R}^{\mu}$ is the estimate of faults vector f
$q \in \mathbb{R}^{+}$
$\Psi(\hat{x}, u, \hat{k}) \in \mathbb{R}^{n\times 1}$
$[K_{ij}]_{\substack{1\le i\le p\\ 1\le j\le q}}$, $[\bar{K}_{kl}]_{\substack{1\le k\le \mu\\ 1\le l\le q}}$ are positive gains

where $\hat{x}_0 = \hat{x}(t_0)$ and $\hat{f}_{k0} = \hat{f}_k(t_0)$ are arbitrary initial conditions, the parameter q determines the order of the polynomial compensation. To ensure the observer convergence, the following assumptions are considered:

A1: $f_k(t)$ is algebraically observable
A2: The gains $[K_{il}]_{1\le i\le p}$ can be chosen such that the following algebraic Riccati equation has a symmetric and positive definite solution P for some $\epsilon > 0$

$$(A - \sum_{i=1}^{p} K_{i1}C_i)^T P + P(A - \sum_{i=1}^{p} K_{i1}C_i) + L^2 PP + I + \epsilon I = 0$$

A3: The gains $[K_{ij}]_{\substack{1\le i\le p\\ 2\le j\le q}}$ are chosen such that

$$\lambda_{min}((PK_{ij}C_i)^T + (PK_{ij}C_i)) \ge 0$$

We define the estimation error vector as $e = [e_x, e_k]^T$, whit $e_x = x - \hat{x}$ and $e_k = f_k - \hat{f}_k$. So from the systems (5) and (7), we determine the dynamics for the corresponding error estimation

$$\begin{aligned} \dot{e}_x &= (A - \sum_{i=1}^{p} K_{i1}C_i)e_x - \\ &\quad - \sum_{i=1}^{p}\sum_{j=2}^{q} K_{ij}(C_i e_x)^{2j-1} + \\ &\quad + [\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})] \\ \dot{e}_k &= \Omega_k - \bar{K}_{k1}e_k - \sum_{j=2}^{q} \bar{K}_{kj}(e_k)^{2l-1} \end{aligned} \tag{8}$$

3.2 Convergence Analysis

In order to ensure the convergence to zero of the estimation error, we establish the following theorem.

Theorem 2. *For the system (5), suppose that* $x(t)\ \exists\ \forall\ t \geq 0$*, the function* $\Psi(x, \bar{u})$ *satisfies the Lipschitz condition given in (6), and* $x(t), f(t)$ *are algebraically observable. Thus, if there exists a positive definite matrix* P *and positive observer gains* $K_{ij}, \bar{K}_{kl}$ *such that the system (7) is an observer for system (5), then the estimation error converges to zero asymptotically.*

Proof. Consider the following Lyapunov function candidate

$$\begin{aligned} V &= V_1 + V_2 \\ V_1 &= e_x^T P e_x; V_2 = \tfrac{1}{2} e_k^2 \end{aligned} \tag{9}$$

where the matrix P satisfies the assumption A2.
The proof of theorem 2, is developed in two parts as follows:

i) The time derivative of V_1 is given as

$$\begin{aligned} \dot{V}_1 &= \dot{e}_x^T P e_x + e_x^T P \dot{e}_x \\ &= e_x^T ((A - \sum_{i=1}^{p} K_{i1} C_i)^T P + P(A - \sum_{i=1}^{p} K_{i1} C_i)) e_x + \\ &+ 2 e x^T P[\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})] - \\ &- 2 \sum_{i=1}^{p} \sum_{j=2}^{q} K_{ij} (C_i e_x)^{2j-2} e_x^T ((P K_{ij} C_i)^T + (P K_{ij} C_i)) e_x \end{aligned}$$

From the follow inequality based on the Lipschitz condition

$$2 e x^T P[\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})] \leq L^2 e_x^T P P e_x + e_x^T e_x \tag{10}$$

and using the Rayleigh's inequality together with assumption A3. it follows that

$$-e_x^T P K_{ij} C_i e_x \leq -\lambda_{min}(P K_{ij} C_i + (P K_{ij} C_i)^T) \|e_x\|^2 \tag{11}$$

Therefore, applying inequalities (10) and (11) we have

$$\begin{aligned} \dot{V}_1 &\leq e_x^T [(A - \sum_{i=1}^{p} K_{i1} C_i)^T P + P(A - \sum_{i=1}^{p} K_{i1} C_i) + \\ &+ L^2 P P + I] e_x - \\ &- 2 \sum_{i=}^{p} \sum_{j=2}^{q} K_{ij} (C_i e_x)^{2j-2} \lambda_{min}(P K_{ijC_i} + (P K_{ijC_i})^T) \|e_x\|^2 \\ &\leq e_x^T [(A - \sum_{i=1}^{p} K_{i1} C_i)^T P + P(A - \sum_{i=1}^{p} K_{i1} C_i) + \\ &+ L^2 P P + I] e_x \\ &= -\epsilon \|e_x\|^2 \end{aligned}$$

ii) In the same way, for the second term in the Lyapunov function candidate, we obtain the time derivative of V_2 as

$$\begin{aligned}\dot{V}_2 &= e_k \dot{e}_k \\ &= e_k(\Omega_k - \bar{K}_{k1} e_k - \sum_{l=2}^{q} \bar{K}_{kl} e_k^{2l-1}) \\ &= e_k \Omega_k - \bar{K}_{k1} e_k^2 - \sum_{l=2}^{q} \bar{K}_{kl} e_k^{2l} \\ &\leq |e_k||\Omega_k| - \bar{K}_{k1} e_k^2 \\ &\leq |e_k| N - \bar{K}_{k1} |e_k|^2 \\ &= -[\bar{K}_{k1}|e_k| - N]|e_k|\end{aligned}$$

$\dot{V}_2$ is negative inside the set $\{|e_k| > N/\bar{K}_{k1}\}$, i.e., exists $\bar{\epsilon} > 0$ such that $\bar{K}_{k1}|e_k| - N = \bar{\epsilon} > 0$.
We prove that $|e_k|$ is upper bounded. Now let constants α, β upper bounds of $V_2(e_k)$. With $\beta > \frac{N^2}{2K_{k1}^2}$, the solution that initiates in the set $\{V_2(e_k) \leq \beta\}$ will remain inside that set for all $t \geq 0$, because $\dot{V}_2$ is negative in $V_2 = \beta$. Therefore the solution of $\dot{e}_k$ is uniformly bounded [19]. Furthermore, if $\frac{N^2}{2K_{k1}^2} < \alpha < \beta$, then $\dot{V}_2$ will be negative in the set $\{\alpha \leq V_2 \leq \beta\}$. In this set V_2 will decrease monotonously until the solution is in the set $\{V_2 \leq \alpha\}$. According to [19] the solution is uniformly ultimately bounded with ultimate bound $|e_k| \leq \sqrt{2\alpha}$. For example, if we define $\alpha = \frac{N^2}{2K_{k1}^2}$ and $\beta = \frac{N^2}{K_{k1}^2}$, the ultimate bound is

$$|e_k| \leq \frac{N}{\bar{K}_{k1}}$$

Hence

$$\dot{V}_2 \leq -\bar{\epsilon}|e_k|$$

Finally, from (i) and (ii), we conclude that

$$\dot{V} \leq -\epsilon \|e_x\|^2 - \bar{\epsilon}|e_k| < 0$$

4 Application to Quad-Rotor MAV

In this section, the polynomial observer approach developed in Section 3 is applied to a Quad-rotor MAV. We will state the mathematical model of the MAV and some notations. Then, the diagnosability analysis of such vehicle dynamics is developed.

4.1 Modeling

The Quad-rotor mathematical model using the corresponding coordinate system shown in Figure (1) is given as follows

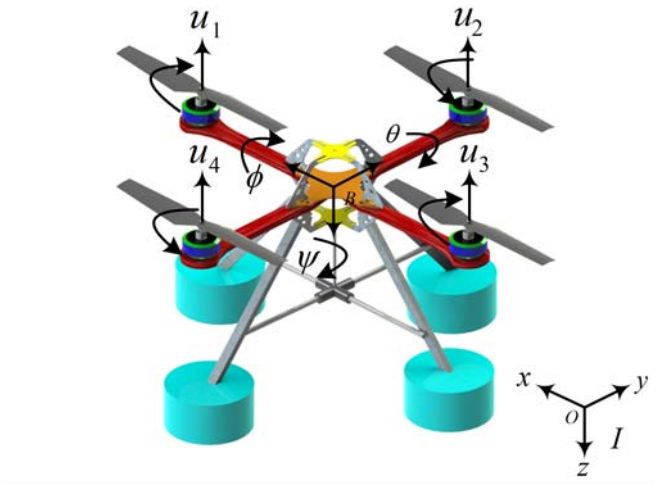


Fig. 1. The three-dimensional quad-rotor model

$$\begin{aligned} m\ddot{x} &= (u_1+u_2+u_3+u_4)(S_\psi S_\phi + C_\psi S_\theta C_\phi) \\ m\ddot{y} &= (u_1+u_2+u_3+u_4)(S_\psi S_\theta C_\phi - C_\psi S_\phi) \\ m\ddot{z} &= mg-(u_1+u_2+u_3+u_4)(C_\phi C_\theta) \\ \ddot{\theta} &= u_3-u_1 \\ \ddot{\phi} &= u_2-u_4 \\ \ddot{\psi} &= u_1-u_2+u_3-u_4 \end{aligned} \tag{12}$$

where u_i are the control inputs, (x, y, z) the position states and (θ, ϕ, ψ) the orientation angles. Using the model (12), the following changes of coordinates is made in order to formulate the problem

$$\begin{array}{llllll} x_1 = x & x_3 = y & x_5 = z & x_7 = \theta & x_9 = \phi & x_{11} = \psi \\ x_2 = \dot{x} & x_4 = \dot{y} & x_6 = \dot{z} & x_8 = \dot{\theta} & x_{10} = \dot{\phi} & x_{12} = \dot{\psi} \end{array}$$

With this notation, the state vector is given by $x = [x_1, x_2, ..., x_{12}]^T$ and the input vector as $u = [u_1, u_2, u_3, u_4]$. For a Quad-rotor MAV, we consider a fault as a reduction of the performance of one or two actuator, with the constraint that if the failure affects an actuator, the second failure will affect the opposite actuator by the symmetry of the structure, i.e. the failure effects can only be minimized when these occurs in the following form:

1. The fault effects only one actuator
2. The fault affects the pairs (u_1, u_3) or (u_2, u_4)

Other wise we would have a catastrophic malfunction and it would be impossible to minimize the fault effects.

We now consider the presents of a fault on four of the engines, so we define the input with presents of fault as $\bar{u}_k = u_k + f_k$. Therefore the system with faults is given by:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(S_{x_{11}} S_{x_9} + C_{x_{11}} S_{x_7} C_{x_9}) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(S_{x_{11}} S_{x_7} C_{x_9} - C_{x_{11}} S_{x_9}) \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= g - \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(C_{x_9} C_{x_7}) \\
\dot{x}_7 &= x_8 \\
\dot{x}_8 &= \bar{u}_3 - \bar{u}_1 \\
\dot{x}_9 &= x_{10} \\
\dot{x}_{10} &= \bar{u}_2 - \bar{u}_4 \\
\dot{x}_{11} &= x_{12} \\
\dot{x}_{12} &= \bar{u_1} - \bar{u}_2 + \bar{u}_3 - \bar{u}_4
\end{aligned} \tag{13}$$

where f_k are additive faults that which affect directly the performance of the engines that produce the thrust inputs u_k.

4.2 Control Strategy

The proposed control strategy is based on the idea that the global system (12) is constituted of two subsystems, the attitude dynamics and the position dynamics, each one with a time-scale separation between them [20]. From this fact, it is possible to propose a hierarchical control scheme where the position controller provides desired attitude angles ϕ_d, θ_d which are the angles to be tracked by the orientation controllers. We have implemented a nonlinear control strategy based on this principle. In this paper we don't present the control strategy, due this is not the purpose of the present work, however the interested lector can see more details about the control strategy applied on this paper in [3].

4.3 Diagnosability Analysis

From Theorem 1 it is required that the number of faults ($\mu = 4$) be less or equal to available measurements. For this case, we consider the output vector as $y = [y_1, y_3, y_5, y_7, y_9, y_{11}] = [x_1, x_3, x_5, x_7, x_9, x_{11}]$.i Taking into account the above mentioned considerations, the condition from theorem 1 is hold with $4 = \mu < p = 6$. To determine the diagnosability of the system (13), we evaluate the algebraic diagnosability condition given in definition 2. For the considered faults, inputs and outputs, the system (13) results in

$$
\begin{aligned}
\ddot{y}_2 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(S_{y_{11}} S_{y_9} + C_{y_{11}} S_{y_7} C_{y_9}) \\
\ddot{y}_3 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(S_{y_{11}} S_{y_7} C_{y_9} - C_{y_{11}} S_{y_9}) \\
\ddot{y}_5 &= g - \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(C_{y_9} C_{y_7}) \\
\ddot{y}_7 &= \bar{u}_3 - \bar{u}_1 \\
\ddot{y}_9 &= \bar{u}_2 - \bar{u}_4 \\
\ddot{y_{11}} &= \bar{u}_1 - \bar{u}_2 + \bar{u}_3 - \bar{u}_4
\end{aligned}
\tag{14}
$$

From system (14), we have that

$$\frac{m\,(g - \ddot{y}_6)}{C_{y_9} C_{y_7}} = u_1 + f_1 + u_2 + f_2 + u_3 + f_3 + u_4 + f_4 \tag{15a}$$

$$\ddot{y}_7 = u_3 + f_3 - u_1 - f_1 \tag{15b}$$

$$\ddot{y}_9 = u_2 + f_2 - u_4 - f_4 \tag{15c}$$

$$\ddot{y}_{11} = u_1 + f_1 - u_2 - f_2 + u_3 + f_3 - u_4 - f_4 \tag{15d}$$

Adding (15a) and (15d)

$$\frac{m(g - \ddot{y}_5)}{C_{y_9} C_{y_7}} + \ddot{y}_{11} = u_1 + f_1 + 2u_3 + 2f_3 \tag{16}$$

Adding 2(15b) and (16)

$$f_3 = \frac{m(g - \ddot{y}_5)}{4C_{y_9} C_{y_7}} + \frac{1}{2}\ddot{y}_7 + \frac{1}{4}\ddot{y}_{11} - u_3 \tag{17}$$

Replacing (17) into (15b)

$$f_1 = \frac{m(g - \ddot{y}_5)}{4C_{y_9} C_{y_7}} - \frac{1}{2}\ddot{y}_7 + \frac{1}{4}\ddot{y}_{11} - u_1 \tag{18}$$

Adding, (15a) and (15c)

$$\frac{m(g - \ddot{y}_5)}{C_{y_9} C_{y_7}} + \ddot{y}_9 = u_1 + f_1 + 2u_2 + 2f_2 + u_3 + f_3 \tag{19}$$

Replacing (18) and (17) into (19)

$$f_2 = \frac{m(g - \ddot{y}_5)}{4C_{y_9} C_{y_7}} + \frac{1}{2}\ddot{y}_9 + \frac{1}{4}\ddot{y}_{11} - u_2 \tag{20}$$

Finally, replacing (20) into (15c) it follows that

$$f_4 = \frac{m(g - \ddot{y}_5)}{4C_{y_9} C_{y_7}} - \frac{1}{2}\ddot{y}_9 + \frac{1}{4}\ddot{y}_{11} - u_4 \tag{21}$$

Therefore, from equations (17), (18), (20) and (21) we conclude that the system (13) is diagnosable, with the considered inputs and outputs.

4.4 Polynomial Observer

The system (13), can be expressed in a similar way as in (5) with: $A \in \mathbb{R}^{12\times 12}$ where the elements of the matrix are given as follows: $a_{1,2} = a_{3,4} = a_{5,6} = a_{7,8} = a_{9,10} = a_{11,12} = 1$ and zero other wise.

The nonlinear function $\Psi(x, \bar{u}) = [\psi_1, psi_2, ..., \psi_{12}]$ is given by:

$$\begin{array}{l}
\psi_1 = \psi_{=}3 = \psi_5 = \psi_7 = \psi_9 = \psi_{11} = 0 \\
\psi_2 = \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(S_{x_{11}} S_{x_9} + C_{x_{11}} S_{x_7} C_{x_9}) \\
\psi_4 = \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(S_{x_{11}} S_{x_7} C_{x_9} - C_{x_{11}} S_{x_9}) \\
\psi_6 g = -\frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(C_{x_{11}} C_{x_9}) \\
\psi_8 = \bar{u}_3 - \bar{u}_1 \\
\psi_{10} = \bar{u}_2 - \bar{u}_4 \\
\psi_{12} = \bar{u}_1 - \bar{u}_2 + \bar{u}_3 - \bar{u}_4
\end{array}$$

So, the following system is a polynomial observer for the given system

$$\begin{cases}
\dot{\hat{x}}\,(t) = A\hat{x} + \Psi\left(\hat{x}, u, \hat{f}\right) + \\
\qquad\qquad + \sum_{i=1}^{6} \sum_{j=1}^{3} K_{ij}(y_i - C_i\hat{x})^{2j-1} \\
\dot{\hat{f}}_k(t) = \sum_{l=1}^{3} \bar{K}_{kl}(f_k - \hat{f}_k)^{2l-1}
\end{cases} \tag{22}$$

Where we fixed the parameter $q = 3$ and the $f_{1\leq k\leq 4}$ are given by (18), (20), (17), (21).

5 Evaluation Function

In this section, an evaluation function is presented in order to use the estimate of the failure with the methodology developed in the previous section. Such evaluation function is presented in a way that could be used in a controller to compensate the failure.

Without loss of generality, we present the evaluation function for the roll dynamics, but the same procedure could be developed for the pitch and yaw dynamics.

5.1 Fault Detection

In order to detect the fault and at the same time differentiate it from perturbations inherent in the system, we define the evaluation function e_f as a function of system states as follows

$$e_f = \frac{1}{(1 + e^{-c_b(\phi - b)})(1 + e^{-c_a(\dot{\phi} - a)})} \tag{23}$$

where parameters c_a and c_b are positive real numbers which define the fault slope for ϕ and $\dot{\phi}$ respectively. The parameters a and b represent the position

Table 1. The parameters of the Quad-rotor MAV considered for the experimental tests.

Parameter	Value
c_b	1
c_a	1
b	20 deg
a	10 deg/sec

and velocity in roll dynamics for which there will be a fault. Accordingly, they are the boundaries between a disturbance and a fault.

So, the basic idea is to estimate the disturbances and then use them in the evaluation function (23). In this way, one should predefine the parameter values a, b. So, by means of an observation of the system dynamics, one knows the existence of a fault. This approach allows to determine the existence of a fault without any measurement of engine speed, eliminating the use of additional sensors on the platform.

In addition, a control strategy could be implemented by using (23), due such evaluation function is continuously differentiable and furthermore, it depends on the system states $(\phi, \dot{\phi})$.

5.2 Experimental Results by Using the Evaluation Function

In this subsection, some experimental results are presented to visualize the performance of the evaluation function at real-time experiments. The experimental results have been tested on the Quad-rotor experimental platform developed at the HEUDIASYC Laboratory. More details about this platform can be seen in [21]. We have simulated the failures by disturbing the Quad-rotor platform in the roll dynamics. For these tests, we have used the parameters shown in table 1.

Two motors are involved in the pitch dynamics (Fig. 1). Thus, if one of this motors fails, its velocity will be reduced or augmented, causing an aggressive rolling moment due to the difference of thrust between the faulty motor and the operating motor. Therefore, we can evaluate such moment by inspecting the pitching dynamics, i.e. the behavior of the states $(\theta, \dot{\theta})$. In order to visualize the behavior of the evaluation function on the real-time experiments, we have perturbed the Quad-rotor platform on its roll axis. Fig. 2 shows the performance of the evaluation function (23). As we see in this figure, the disturbances lower than 20 deg have been omitted by the evaluation function, while the disturbances greater or equal to 20 deg have been taken as failures. With this approach, we can tune the parameters given in table 1, in order to choose a desired performance to tell the system how to distinguish a perturbation of a fault.

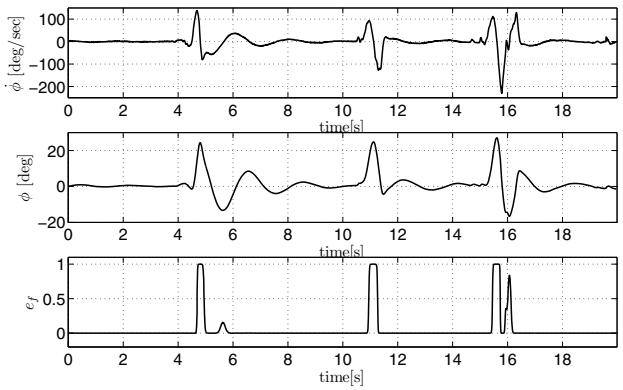


Fig. 2. Evaluation function (23) and states of the roll dynamics.

6 Fault Reconstruction Results

In this section, we present some simulation results of the procedure developed in Section 3. The dynamics of the Quad-rotor MAV and the fault dynamics have been simulated using MATLAB Simulink. For the simulation procedure, we have consider the following conditions: The desired values for the position dynamics are $x_d = y_d = 0$m and $z_d = 0.75$m and for the attitude dynamics are $\theta_d = \phi_d = 0$ and $\psi_d = 45$ degrees. The objective is that the Quad-rotor take off and reaches the desired height and remain stable in that position, in other words, we want that the desired values for the linear and angular velocities are equal to zero. To simplify the calculations we assume that $m = g = 1$. A simulation time of 300s and a step of 0.001s has been chosen.

For all simulation results we have considered that the fault affects the performance of each engine, i.e. the actuators for the Quad-rotor MAV. Four faults were artificially generated as follows

$$
\begin{aligned}
f_1 &= 0.226(1 + sin(0.5te^{-0.1t}))\mathcal{U}(t-50) \\
f_2 &= 0.045(1 + sin(0.076e^{(t-0.3)}))\mathcal{U}(t-20) \\
f_3 &= 0.055e^{-0.01(t-0.3)}\mathcal{U}(t-10)+ \\
&\quad +0.068e^{-0.005(t-1)}\mathcal{U}(t-80)+ \\
&\quad +0.159e^{-0.07(t-1.3)}\mathcal{U}(t-140) \\
f_4 &= 0.718e^{-0.01(t-2)}\mathcal{U}(t-30)
\end{aligned}
$$

where $\mathcal{U}(t)$ is the unit step function. The magnitude of the faults were selected very close to the magnitude of the generated thrusts inputs for the case without faults, to obtain better results.

The results of the implementation of the polynomial observer proposed in (22) for the fault diagnosis task, for the considered available outputs and inputs. As shown in the Figure 3, for the fault f_1. The gain values for the proposed observer

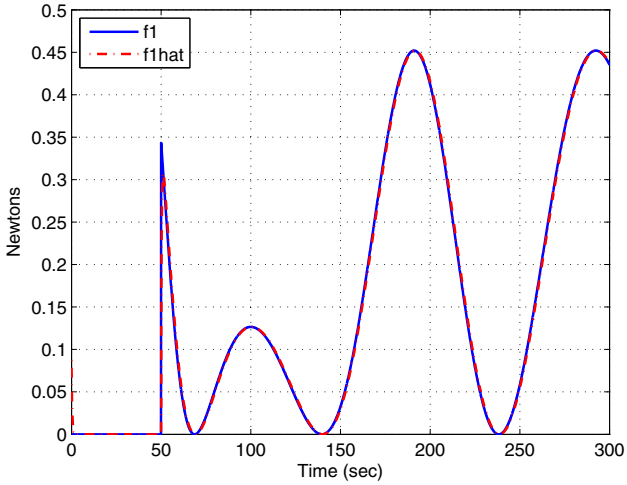


Fig. 3. Estimation result for the fault f_1

were $K_{11} = 2.5, K_{12} = 34$ and $K_{13} = 66$. In the same way in figures 4, 5 and 6, we show the estimation result for the faults f_2, f_3 and f_4, where the gain values for each observer are $K_{21} = 5.05, K_{22} = 2, K_{23} = 1.6$, $K_{31} = 1.5, K_{32} = 27, K_{33} = 56$ and $K_{41} = 4.23, K_{42} = 7, K_{43} = 3$ respectively.

In order to evaluate the effectiveness of the proposed polynomial observer, we use initial conditions different from zero, to see how long it takes to converge to the actual value of the fault. The initial conditions were $f_{1,c.i} = 0.087$, $f_{2,c.i} = 0.065$, $f_{3,c.i} = 0.055$ and $f_{4,c.i} = 0.073$.

For all faults we obtained good estimation results. As can be seen, the proposed observer converge quickly to the actual values of the faults, and although the approach only considers the case of fault with differentiable dynamics, it is noted that the approach has the capacity to reconstruct abrupt faults as shown in figures 5 and 6.

The attitude dynamic under the effect of the faults is shown in figure 7. The direct consequence of the presence of faults on actuators, is that, the controller tries to stabilize the system and bring the dynamics of roll, pitch and yaw to the desired values. However due to thrust limitation, the objective is not fully accomplished .Noticed that the faults affects more the dynamics in yaw.

Figure 8, shows the corresponding angular velocities in roll, pitch and yaw for the case of presence of faults. As can be seen, between in the first 100 seconds the angular velocities change abruptly, because in this time interval, all the faults appear.

The position dynamics affected for the faults are shown in Figure 9. Note that the difference between the dynamic without faults and with fault is very significant, especially for the dynamics in the y-axis, which is very large. The

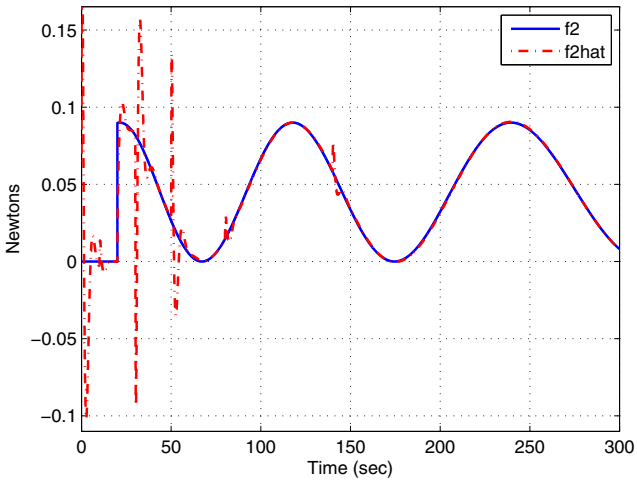


Fig. 4. Estimation result for the fault f_2

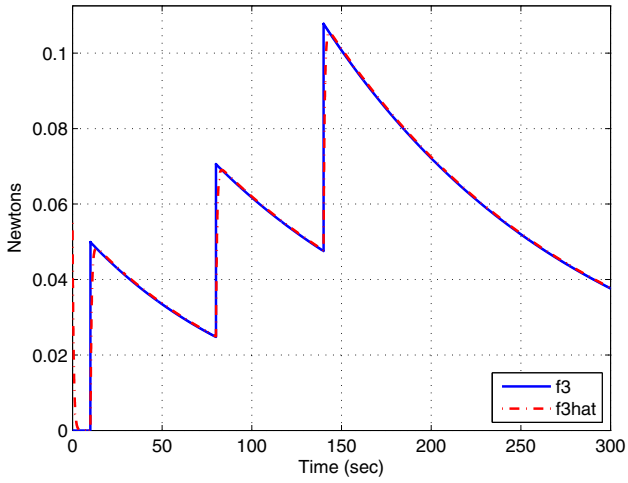


Fig. 5. Estimation result for the fault f_3

Position controller generates large inputs to try reach the desired position values. However due to the faults the errors grow and the controller is unable to compensate such errors.

Finally figures 10 and 11 show the control inputs and corresponding thrusts generated by the control strategy. As can be seen in Figure 10 the difference between both cases is very significant, because for the case without faults the

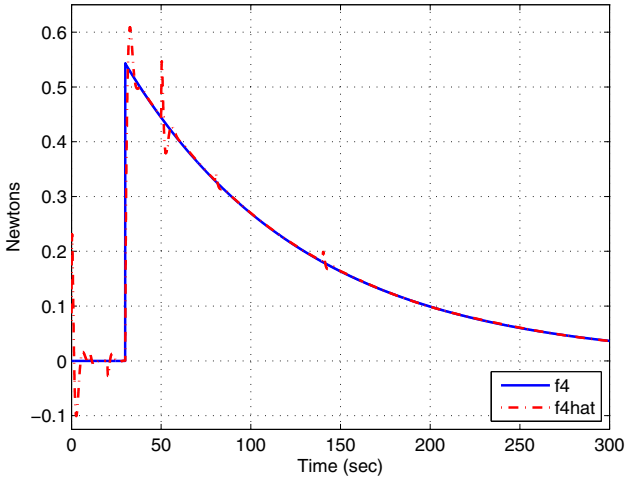


Fig. 6. Estimation result for the fault f_4

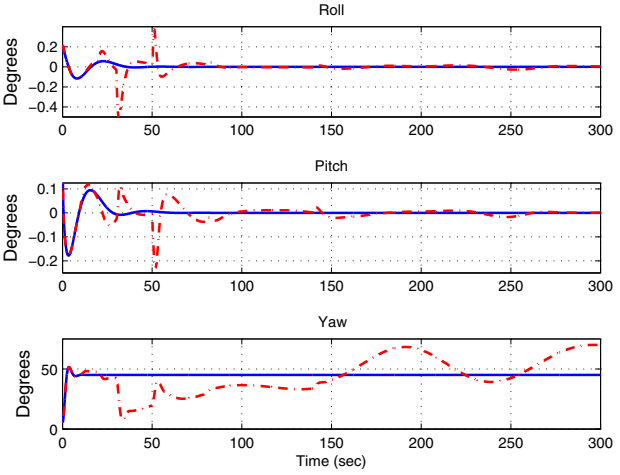


Fig. 7. Comparison for the attitude dynamics for the case without faults (blue line) and with faults (red dash-dot line)

control inputs are constant while for the case with faults, the thrusts are non-constant and larger. The controller tries to compensate the error generated by the presence of the faults, but it shows clearly in Figure 11, where we see that the corresponding thrust forces inputs are very similar to the dynamics of the faults but with opposite sign. Notice that the faults cause the controller does

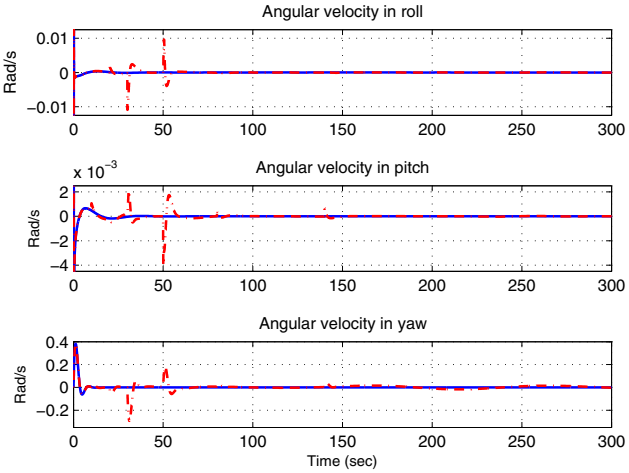


Fig. 8. Comparison for the angular velocity dynamics for the case without faults (blue line) and with faults (red dash-dot line)

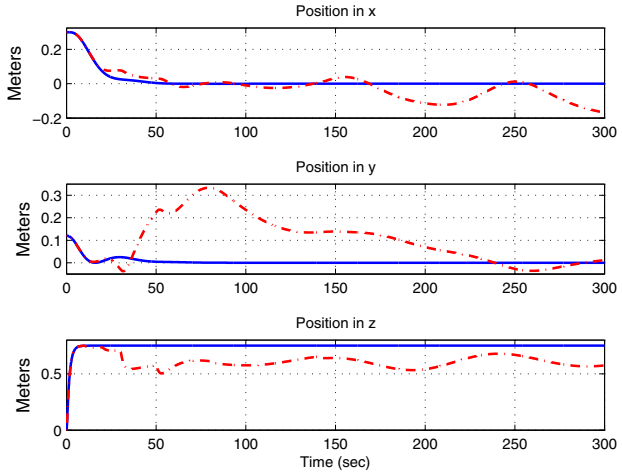


Fig. 9. Comparison for the position dynamics for the case without faults (blue line) and with faults (red dash-dot line)

not function properly, as we observed for the thrusts 1 and 4 (Figure 11) and the control inputs 2, 3 and 4 (Figure 10), there is a time instant when its becomes negative and this is impossible, because it would mean that the thrust force is opposite, i.e, when the thrust force becomes negative, the engine does not have

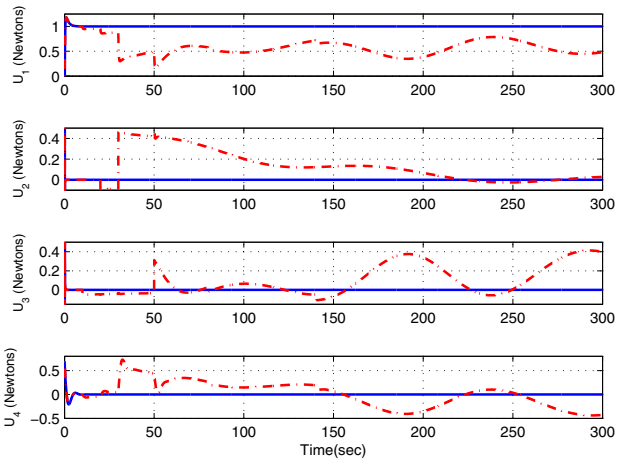


Fig. 10. Comparison for the inputs generated by the control strategy for the case without faults (blue line) and with faults (red dash-dot line)

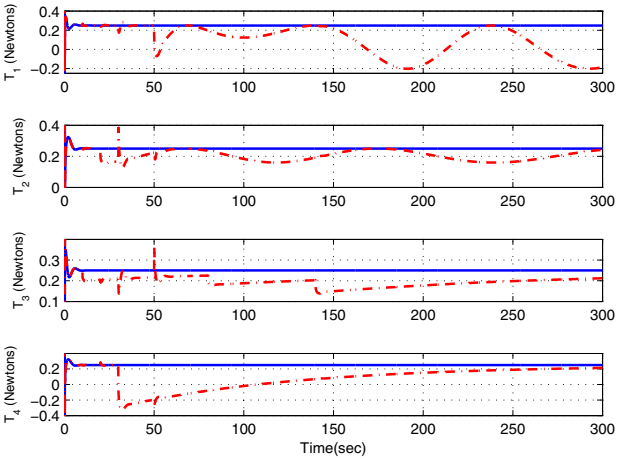


Fig. 11. Comparison for the corresponding thrust inputs for the case without faults (blue line) and with faults (red dash-dot line)

the ability to change the direction of rotation, and therefore stops completely (turned off) and in this case we are not dealing with a fault, we would have a failure.

7 Conclusion and Future Work

This work deals with the problem of fault detection and diagnosis task for a Quad-rotor mini air vehicle (MAV) using the differential algebra approach. This approach consider the unknown faults like an augmented state of the system, the strategy is proposes a bank of observers in order to estimate the fault dynamics, in this case we are only use the available measurements and known inputs. A polynomial observer was proposed to deal with the fault estimation problem for the case of multiple faults. This approach detects and identifies multiple faults of relative small magnitudes. In this work the FDD task for a system stabilized in the closed-loop using a control strategy is presented.

The second part of this paper concentrates on the study of the controllability of the system with a failure. For that purpose, we have presented an evaluation function depending on system states. By means of this evaluation function, we can develop a control methodology which compensates the effect of the failure under certain considerations. That will be the subject of the second part of the paper.

References

1. Flores, G., Escareno, J., Lozano, R., Salazar, S.: Quad-tilting rotor convertible mav: Modeling and real-time hover flight control. Journal of Intelligent and Robotic Systems 65(1-4), 457–471 (2012)
2. Bouabdallah, S., Siegwart, R.: Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In: In Proceedings of IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, pp. 2247–2252 (April 2005)
3. Flores, G., Lozano, R.: Lyapunov-based controller using singular perturbation theory: An application on a mini-uav. In: Proc. IEEE American Control Conference (ACC 2013), Washington, DC (to appear, June 2013)
4. Bertrand, S., Hamel, T., Piet-Lahanier, H.: Stability analysis of an uav controller using singular perturbation theory. In: Proceedings of the 17th World Congress The International Federation of Automatic Control, Seoul, Korea, pp. 5706–5711 (July 2008)
5. Flores, G., Garcia, L., Sanahuja, G., Lozano, R.: Pid switching control for a highway estimation and tracking applied on a convertible mini-uav. In: Proc. 51st IEEE Conference on Decision and Control (CDC 2012), Maui, HI, USA (December 2012) (to be published)
6. Khebbache, H.: Robust control algorithm considering the actuator faults for attitude tracking of an uav quadrotor aircraft. International Journal of Control and Automation 5(4), 55–66 (2012)
7. Freddi, A., Longhi, S., Monteriù, A.: A model-based fault diagnosis system for a mini-quadrotor. In: 7th Workshop on Advanced Control and Diagnosis (ACD 2009), Zielona Góra, Poland (November 2009)
8. Freddi, A., Longhi, S., Monteriù, A.: Actuator fault detection system for a mini-quadrotor. In: 2010 IEEE International Symposium on Industrial Electronics (ISIE), Bari, Italy, pp. 2055–2060 (July 2010)
9. Freddi, A., Lanzon, A., Longhi, S.: A feedback linearization approach to fault tolerance in quadrotor vehicles. In: Proceedings of The 2011 IFAC World Congress, Milan, Italy (2011)

10. Ranjbaran, M., Khorasani, K.: Fault recovery of an under-actuated quadrotor aerial vehicle. In: 2010 49th IEEE Conference on Decision and Control (CDC), pp. 4385–4392 (December 2010)
11. Fliess, M., Join, C., Sira-Ramirez, H.: Robust residual generation for linear fault diagnosis: an algebraic setting with examples. International Journal of Control 77(14), 1223–1242 (2004)
12. Fliess, M., Hebertt, Sira-Ramirez, o.: Control via state estimations of some nonlinear systems. In: IFAC Symposium on Nonlinear Control Systtems, NOLCOS 2004 (2004)
13. Rincon-Pasaye, J.J., Martinez-Guerra, R., Soria-Lopez, A.: Fault diagnosis in nonlinear systems: An application to a three-tank system. In: American Control Conference, pp. 2136–2141 (2008)
14. Martinez-Guerra, R., Luviano-Juarez, A.: Fault diagnosis of nonlinear systems using reduced-order observers and algebraic observers. In: 2006 45th IEEE Conference on Decision and Control, pp. 544–549 (2006)
15. Garcia-Beltran, C.D., Rodriguez-Palacios, A., Guerrero-Ramirez, G.V., Silva-Carranza, F.M., Sorcia-Vazquez, F.: Fault diagnosis of an induction motor based on differential algebra reconstructors. In: World Automation Congress (WAC), pp. 1–6 (June 2012)
16. Sierra, H.A., Martinez-Guerra, R., Mata-Machuca, J.L.: Fault diagnosis via a polynomial observer. In: 2011 8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE), pp. 1–6 (October 2011)
17. Mata-Machuca, J.L., Martinez-Guerra, R., Aguilar-López, R.: An exponential polynomial observer for synchronization of chaotic systems. Communications in Nonlinear Science and Numerical Simulation 15(12), 4114–4130 (2010)
18. Cruz-Victoria, J.C., Martínez-Guerra, R., Rincón-Pasaye, J.J.: On nonlinear systems diagnosis using differential and algebraic methods. Journal of the Franklin Institute 345(2), 102–118 (2008)
19. Khalil, H.K.: Nonlinear Systems. Prentice Hall, New York (2002)
20. Kokotović, P., Khalil, H.K., O'Reilly, J.: Singular Perturbation Methods in Control: Analysis and Design. Academic Press, London (1999)
21. Carrillo, L.R.G., Flores, G., Sanahuja, G., Lozano, R.: Quad-rotor switching control: An application for the task of path following. In: Proc. IEEE American Control Conference (ACC 2012), Montreall, Canada, pp. 4637–4642 (June 2012)

Part III
Assistive Robotics

Assistive Robotics as Alternative Treatment for Tremor

E. Rocon[1], J.A. Gallego[1], J.M. Belda-Lois[2], and J.L. Pons[1]

[1] Bioengineering Group, CAR-CSIC, Madrid, Spain
e.rocon@csic.es
http://www.car.upm-csic.es/bioingenieria/
[2] Instituto de Biomecánica de Valencia, Valencia, Spain

Abstract. Tremor is the most common movement disorder and strongly increases in incidence and prevalence with ageing. Although not life threatening, upper limb tremors hamper independent life of 65% of those suffering from them, greatly impacting on their quality of life. Current treatments of tremor include drugs and surgery. However, tremor is not effectively managed in 25% of patients. Therefore, further research and new therapeutic options are required for an effective management of pathological tremor. This paper introduces some rehabilitation robots developed for tremor suppression based on biomechanical loading, their evaluation and the identification of their limitations. At the end, authors aim to provide a view of the potential of this novel approach for tremor management and the plans for commercialization.

Keywords: tremor, assistive robotics, biomechanical loading.

1 Introduction

Tremor is defined as a rhythmical, involuntary oscillatory movement of a body part, [1] . Under certain circumstances, like the performance of precise tasks, or prolonged maintenance of a posture, we all exhibit a certain degree of tremor. This is called physiological tremor [2]. When tremor arises from a neurological condition, becoming cause of disability, it is referred to as pathological tremor, [3]. Pathological tremor is the most prevalent movement disorder, [4], and projection studies foresee that prevalence will double by 2050, [5]. Importantly, pathological tremor, referred to as tremor in the remainder in the document, does not constitute a monolithic entity, and appears caused by ten different so-called syndromes. Among them, Parkinson's disease (PD), and essential tremor (ET) are the most relevant in terms of prevalence, [6].

Tremors are currently managed through pharmacotherapy or surgery, consisting either in stereotactic thalamotomy, or more commonly nowadays, in Deep Brain Stimulation (DBS). Unfortunately, both alternatives have significant drawbacks associated: drugs often induce side effects, and show decreased effectiveness over years of use, while DBS is related to increased risk of intracranial haemorrhage (4 % of patients), and psychiatric manifestations, and the percentage of

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_12,

eligible patients is extremely low; for instance, only 1.6 to 4.5 % of those with Parkinsons Disease, [2]. As for the tremors themselves, the mechanisms accounting for the alleviation of their symptoms by drugs, thalamotomy or DBS are unknown, hampering the refinement of the existing treatment forms, and the development of novel ones. As a result, tremor is not effectively managedin a significant proportion of patients, up to 25 % according to some estimates [1], and is a major cause of dependance and loss in quality of life. In addition, tremor carries important social and psychological burden associated, which further affects both patients' and relatives' lives. This motivates that the development of novel therapies for tremor is matter of paramount importance.

It has been established in the literature that most of the different types of tremor respond to biomechanical loading. In particular, it has been clinically tested that the shunt increase of damping and/or inertia in the upper limb leads to a reduction of the tremorous motion, i.e. the change in impedance characteristics of the upper limb has a direct effect on the tremor characteristics, [7].

This paper describes two research projects in which two different wearable robots based on force loading were developed and validated for tremor management. The first consisted of a robotic exoskeleton that applied forces to different joints of the upper limb and consistently attenuated moderate and severe tremors. The second apporach takes the form of a neuroprosthesis based on transcutaneous neurostimulation.This system is based on the same approach of biomechanical loading but using human muscles as actuators. It was evaluated with patients and successfully alleviated mild tremors, although to a lesser extent than moderate or severe ones, but it moves toward the implementation of a textile-based device that better fulfills patient expectations. This paper is organized as follows. The robotic exoskeleton is briefly described in the next section, which is followed by a description of the development and validation of the neuroprosthesis. Section 4 discuss the major findings of both apporachs. The paper concludes by outlining current and future research in the field of biomechanical loading.

2 Wearable Orthosis for Tremor Suppression

The active orthosis (exoskeleton) WOTAS was developed under the framework of the European project DRIFTS [8]. The concept of WOTAS is to develop an active upper limb exoskeleton based on robotics technologies capable of applying forces to cancel tremor and retrieve kinematic information from the tremorous upper limb. The overall aim of this project was to develop a powered orthosis to provide means of testing non-grounded tremor reduction strategies in three joints of the upper limb. This robotic orthosis platform is able to monitor, diagnose and control tremor in subjects. This robotic exoskeleton is equipped with kinematics (angular position, velocity and acceleration) and kinetic (interaction force between limb and orthosis) sensors. Moreover, it could also apply dynamic force to the articulations of the upper limb by means of a set of flat DC motors + pancake gears [8]. Innovations of the WOTAS exoskeleton are its portability, it

is a no invasive system, and provides direct information from each joint of the upper limb, allowing the estimation of the contribution of each articulation to the overall tremorous motion in the kinematic chain of the upper limb. This platform also allows the implementation of different control strategies to tremor suppression for each joint, in such a way that independent control strategies could be applied to each joint. This robotic device spans the elbow and wrist joints, being able to apply independent tremor suppression strategies to elbow flexo-extension, wrist flexo-extension and wrist prono-supination, see Figure 1. The total weight of the final system is roughly 850g.

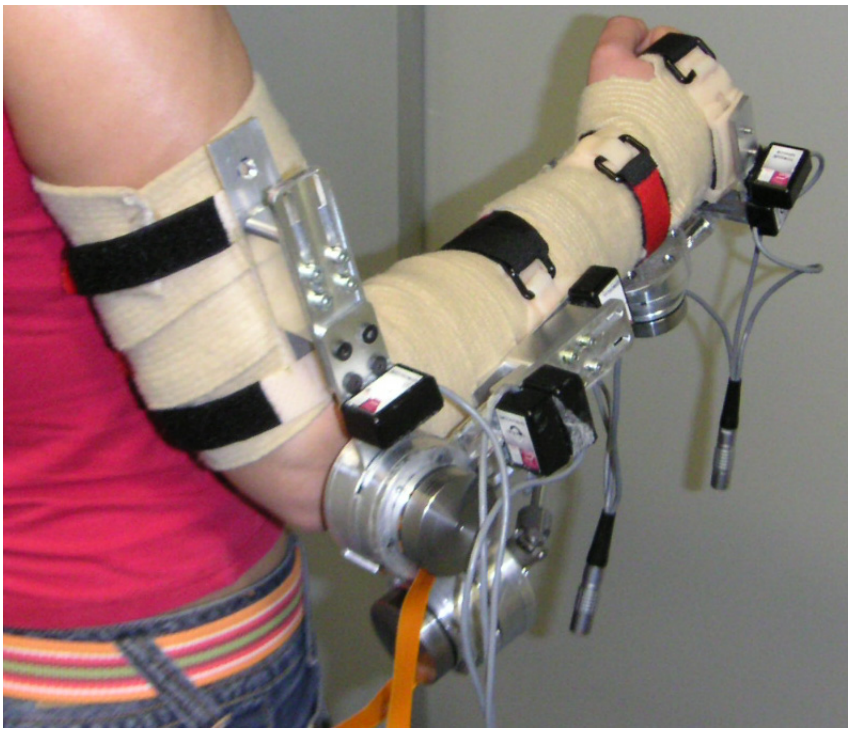


Fig. 1. Assistive Robotic device for tremor reduction (WOTAS, Wearable Orthosis for Tremor Assessment and Suppression)

The device was clinically validated with tremor patients (from different pathologies: essential tremor, Parkinson, Multiple sclerosis, Post-traumatic tremor and mixed tremor) with successful results: it was able to achieve a consistent 40 % of tremor power reduction for all patients, and to attain a reduction ratio in the order of 80 % tremor power in specific joints of patients with severe tremor. Moreover, patients related that the exoskeleton did not affect their concomitant voluntary motion, which is a common drawback of tremor and very

important for users acceptance. The device achieved this performance by means of, [8]:

1. an algorithm able to distinguish in real-time the voluntary from the involuntary movement of the patient at each upper limb joint;
2. the modification of inertial and damping characteristics of the corresponding musculo-skeletal system;
3. the application of out-of-phase tremor cancellation forces.

The approach to mechanical suppression of tremor by means of exoskletons presents limitations mainly due to the physical interaction between the exoskeleton and the human limb:

- The transmission of forces through soft tissues plays an important role in the efficiency of tremor suppression. There is a physical limitation for tremor suppression through wearable devices due to force generation (size and power consumption of the actuators) and transmission through soft tissues.
- Emerging actuators technologies, i.e. Magneto-Rheological Fluid actuators (MRFs), Electro-Active Polymer actuators (EAP) and Ultrasonic motors, were evaluated for an orthotic implementation. It was concluded that, despite the success of the approach, there is no suitable actuator technology in terms of cosmetic and aesthetic (low weight, compact to be worn beneath the clothes) as well as functional requirements (torque, bandwidth).
- Patients related that these bulky exoskeletons could not be considered as a solution to their problem since it is considered that the use of such device should cause social exclusion.

In summary, robotics based solutions have shown clinical evidence of the approach based on human limb impedance control. However it results in bulky and non cosmetic solutions for which patients are especially reluctant.

3 Neuroprosthesis for Tremor Suppression

In the framework of the TREMOR project (EU-ICT-2007-224051),a neuroprosthesis was developed to circumvent the major limitations identified in the WOTAS exoskeleton described in the previous section. We elected to use Functional Electrical Stimulation (FES) as a means of generating biomechanical loading to the tremulous limb. The neuroprosthesis developed takes the form of a textile that may be easily worn underneath the clothes, being therefore closer to a final product. This is enabled by the fact that the neuroprosthesis does not need external actuators, since it utilizes the patient's muscles to load the tremulous limb. Furthermore, most of the existing systems apply constant forces to the tremulous limb, while this neuroprosthesis adapts the force applied to the ongoing tremor characteristics, and has the ability of actuate only when needed, i.e. when tremor poses a functional problem. Obviously, this is an important improvement regarding user acceptance and the performance of the system itself.

The neuroprosthesis developed utilizes closed-loop transcutaneous neurostimulation to apply mechanical loads in order to alleviate upper-limb tremor. It drives wrist flexion-extension and elbow flexion-extension; pronation-supination is not targeted because of the difficulty of using transcutaneous neurostimulation on muscles that elicit supination due to from the above-mentioned selectivity issues. Therefore, neurostimulation was delivered at the following sites: flexor carpi radialis, extensor carpi ulnaris, biceps brachii and triceps brachii (lateral head). An independent multichannel unipolar neurostimulator (Una Systems, Belgrade, Serbia) controlled each pair of antagonists; common electrodes were placed at the distal third of the forearm and close to the olecranon process, [9].

Neurostimulation was modulated based on instantaneous tremor characteristics, which are estimated from solid-state gyroscopes based on a two-stage algorithm. Each targeted movement was measured with a pair of gyroscopes using a differential configuration. The two-stage algorithm, built upon that implemented in WOTAS, separates the volitional and tremor components of movement based on their different frequency content, and given that they are additive. Figure 2 illustrates the operation of the controller.

The evaluation of the neuroprosthesis with patients indicated that tremor was attenuated irrespectively from their etiology (mean reduction Ratt $= 48.1 \pm 26.3$ %, $p < 0.001$), and, for the six patients evaluated, its amplitude and frequency did not have an effect on the performance of the NP. However, there could be a subtle trend towards larger attenuation of more severe tremors, although it was not statistically significant. Remarkably, no patient found the sensation induced by NP-driven muscle co-contraction unbearable, and a number of them spontaneously reported that felt an improvement in their condition. All patients

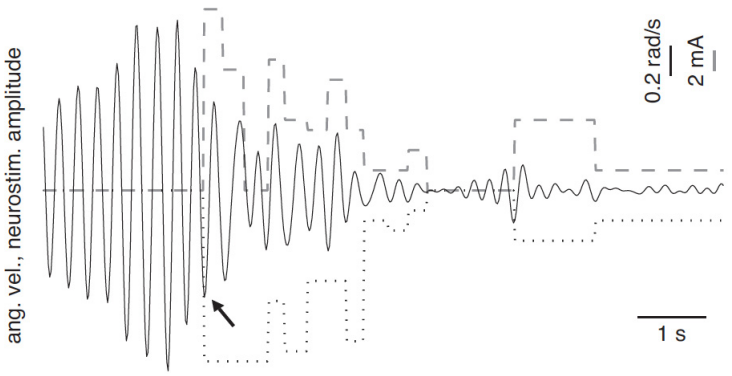


Fig. 2. An example of the controller that modulates muscle co-contraction to attenuate tremor. The plot shows the estimation of tremor (solid line) and the amplitude of the current applied at the extensors (dashed line) and flexors (dotted line). The instant at which the NP was triggered is signaled with an arrow. Notice the different scales employed for the representation of neurostimulation amplitude and angular velocity. A positive value corresponds to wrist extension.

exhibited a positive, yet variable, response to the approach here presented, which is of great interest for ET patients, given that only 50 % of them benefits from the drugs that are currently prescribed to manage their disorder, [9].

In summary, the results of the evaluation demonstrate that irrespectively from its etiology, severity, and frequency, tremor is significantly attenuated by NP-driven muscle co-contraction. Furthermore, the positive outcome of these experiments encourages the large scale validation of the approach, either alone, or as a complement to pharmacotherapy.

4 Conclusions

This paper presents two alternative treatments for tremor suppression based on biomechanical loading. In summary, biomechanical loading is non-invasive and painless. It may be effective in patients who are insufficiently responsive (or have adverse reactions) to drugs or in whom surgery is contraindicated. Moreover, this treatment avoids the potential side effects of drugs and the risks of surgical procedure.

The work presented here is based on clinical evidence with a limited number of patients (n=10 for WOTAS and n=12 for the neuroprosthesis), and most of them showed a positive response to the approach. Although the number of patients is small, it is considered sufficient to provide proof of concept of the feasibility and interest of using biomechanical loading as an alternative treatment for tremor. Owing to the reduced number of patients, the results of this study cannot be extrapolated to the general population and cannot be considered as clinical validation. Nevertheless, we are encouraged by the results and are considering the possibility of performing a large-scale multicenter validation of the neuroprosthetic concept. We aim to include 500 patients (suffering from different pathologies that cause tremor) from 30−50 different hospitals around the world. The expected outcome is that biomechanical loading could either substitute or complement the pharmacotherapeutic management of tremors.

References

1. Rocon, E., Belda-Lois, J., Sanchez-Lacuesta, J., Pons, J.L.: Pathological tremor management: Modelling, compensatory technology and evaluation. Technology and Disability 16, 3–18 (2004)
2. Elble, R.J.: Tremor: clinical features, pathophysiology, and treatment. Neurologic Clinics 27, 679–695 (2009)
3. McAuley, J.H., Marsden, C.D.: Subharmonic solutions with prescribed minimal period for nonautonomous Hamiltonian systems. Brain: A Journal of Neurology 8, 1545–1567 (2000)
4. Wenning, G.K., Kiechl, S., Seppi, K., Müller, J., Högl, B., Saletu, M., Rungger, G., Gasperi, A., Willeit, J., Poewe, W.: Prevalence of movement disorders in men and women aged 50-89 years (Bruneck Study cohort): a population-based study. Lancet Neurology 4(12), 815–820 (2005)

5. Bach, J.-P., Ziegler, U., Deuschl, G., Dodel, R., Doblhammer-Reiter, G.: Projected numbers of people with movement disorders in the years 2030 and 2050. Movement Disorders 26(12), 2286–2290 (2011)
6. Deuschl, G., Bain, P., Brin, M.: Consensus statement of the movement disorder society on tremor. Movement Disorders 13 (suppl. 3(2)), 2–23 (1998)
7. Rocon, E., Gallego, J., Belda-Lois, J.-M., Benito-León, J., Pons, J.L.: Biomechanical loading as an alternative treatment for tremor a review of two approaches. Tremor and Other Hyperkinetic Movements, 1–13 (2012)
8. Rocon, E., Belda-Lois, J.M., Ruiz, A.F., Manto, M., Moreno, J.C., Pons, J.L.: Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression. IEEE Transactions on Neural Systems and Rehabilitation Engineering 15(3), 367–378 (2007)
9. Gallego, J.Á., Rocon, E., Belda-Lois, J.M., Pons, J.L.: A neuroprosthesis for tremor management through the control of muscle co-contraction. Journal of Neuro Engineering and Rehabilitation 10(1), 36 (2013)

Ultrasonic Motor Based Actuator for Elbow Joint Functional Compensation

Dorin Sabin Copaci, Antonio Flores Caballero, Alejandro Martin Clemente, Dolores Blanco Rojas, and Luis Moreno Lorente

RoboticsLab, Carlos III University of Madrid, Avda. de la Universidad 30, 28911, Leganés, Madrid, Spain
{dcopaci,afcaball,aimartin,dblanco,moreno}@ing.uc3m.es
http://roboticslab.uc3m.es/

Abstract. Robotic rehabilitation has become very popular in recent years. Nevertheless it still faces various problems related to the system actuation devices. Some classic actuator features, like the noise, weight, size-force relationship, and the efficiency, make them little suitable for such applications. For this reason, piezoelectrically driven ultrasonic motors have become the alternative actuators to the conventional electromagnetic motors. This is due to their light weight, compact size and soundless performance characteristics. The aim of this article is to present a new actuator based on an ultrasonic motor (USM) and a magnetorheological clutch is integrated into a device used in robotic rehabilitation for the elbow joint. At the end, a control strategy algorithm for the USM and its first preliminary results will be presented. The work presented was carried at the Systems Engineering and Automation Department of Carlos III University of Madrid, under the HYPER CONSOLIDER-INGENIO 2010 Spanish project.

Keywords: ultrasonic motor, magnetorheological clutch, rehabilitation robotics, MR fluid.

1 Introduction

Apart from the conventional engines and within human centered actuators there are the bio-inspired actuators, like the pneumatic muscles, piezoelectric materials, and Shape Memory Alloys (SMA) [1]. Ultrasonic motors (USMs) are a kind of actuators that uses ultrasonic level mechanical vibration as their driving source. In comparison to conventional electromagnetic motors, they have several important advantages: the high holding torque, high response characteristics, high torque density at a low speed (no gears necessary), silent operation, no electromagnetic noise, and compact size [2]. However, they have some disadvantages that limit their use today such as their reduced life span and slight oscillation of its controller speed.

In robotic rehabilitation, an important issue is the interaction between the human body with the rehabilitation device, its portability, and the response

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_13,

time of the actuator. Other criteria that must be taken into account are the aesthetic and comfort of the device. These requirements hugely depend on the actuator, which should be: light, small, and soundless. Standard devices based on conventional engines are noisy, while ultrasonic motors are silent enough for portable applications and daily use.

On the other hand, in rehabilitation systems focused on human limbs, the quick response of the device is very important. The actuators should be able to free the joints and shafts instantly to avoid muscle injuries in situations such as involuntary muscles contractions or spasms. To deal with this requirement, the work presented uses a magnetorheological clutch in conjunction with USM. For spasm detection a new sensor will be integrated in the system. The proposed sensor in this case is a torque sensor. This is capable to detect the involuntary muscle contraction with a good precision.

The proposed actuator uses an ultrasonic motor (USR60-E3NT), a planetary gear (with a ratio of 4.3:1), and a magnetorheological clutch as a conceptual design for the elbow joint rehabilitation.

To test the feasibility of the proposed system a test bench and its control architecture which simulates the upper limb have been developed, based on a full system software simulation previously validated.

This work is organized as follows. Section 2 describes the prototype of the proposed device. Section 3 presents the software simulation of the system. Section 4 describes the control algorithm of the actuator. Section 5 gives the experimental results and discussions. At last, Section 6 give the conlusions of the presented work.

2 Prototype of the Proposed Device

The ultrasonic motor offers a torque of $1Nm$ at $150rpm$, using a planetary gear the assembling from the two devices forms an actuator with a $4Nm$ maximum torque at $37.5rpm$. A cross section of the actuator and the magnetorheological clutch is shown in Figure 1.

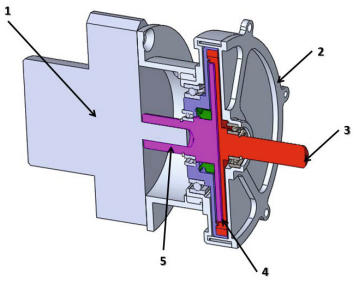


Fig. 1. Crossection of the proposed actuator. 1. Motor and planetary gear 2. Clutch housing 3. The output shaft 4. Free space for magnetorheological fluid 5. Input shaft.

The total weight of the actuator is represented of the sum of the three devices: the weight of the ultrasonic motor ($250g$), the weight of the planetary gear ($200g$) and the weight of the electromagnetic clutch ($500g$). The maximum dimensions of the actuator are given by the diameter of the clutch ($89mm$) with a total length (motor, planetary gear and clutch) of $132.2mm$.

2.1 Test Bench to Simulate the Elbow Joint

For the first validation of the rehabilitation device of the elbow joint, it was necessary to design and to develop a new test bench capable of simulating the biomechanics of this joint. Due to differences between the angles of each person correlated to the weight differences and muscle structure of the arm, it is not possible to have an accurate simulation. Based on recent studies, in the daily life, the elbow joint shows a functional arc of $100degrees$ of flexion and $50degrees$ of supination. The angles of the two degrees of freedom can reach up to 150, and $85degrees$ respectively. The proposed design of the test bench allows only the simulation of the flexor angle which is considered the main movement of the elbow.

The weight of the arm and the forearm were calculated according to a person of $70kg$ of weight and $1,80m$ of height. Knowing that the arm segment is the 2.6% of the total mass, forearm is the 1.6% and the hand is the 0.7% [6], the arm has a weight of $1.82kg$, and the forearm with the hand, has a weight of $1.61Kg$. The sizes were calculated according to individual height, representing the 14.6% for the forearm, the 18.6% for the arm and the 10.8% for the hand. Using these parameters the test bench (Figure 2) is able to simulate the inertia of the segments.

According to the characteristics of the test bench, the torque in the elbow joint can be calculated depending on the center of gravity of the forearm segment:

$$M = w * g * r = 1.6 * 9.81 * 0.2 = 3.158Nm \tag{1}$$

Where M is the force moment for elbow joint, w the weight, g the acceleration of gravity, and r the radius (from the joint to the center of gravity of the forearm segment).

For rehabilitation exercises, with not load applied it has been considerated enough an actuator with $4Nm$ torque.

2.2 Actuator Hardware Components

Magnetorheological Clutch

A. Proposed Design

The clutch design consists of a cylinder representing the housing part of the clutch, which is connected to the input shaft. Inside, there is a central disc close to the wall part (housing part), and connected to the output shaft. The space

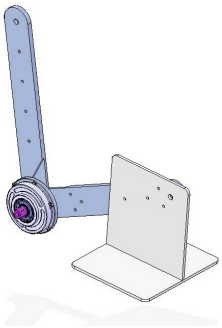


Fig. 2. Test bench for the elbow joint of a person with $70kg$: arm of $1.82kg$, forearm and hand of $1.61kg$

between the disk and the housing is filled with magnetorheological fluid (MR), formed by many micrometer-sized particles, which may be polarized in order to change the viscosity of the whole fluid depending on the intensity of the magnetic field they are exposed to. The proposed design of the clutch is shown in Figure 3. The torque on the output shaft is obtained from the movement of the input shaft and transmitted by the fluid.

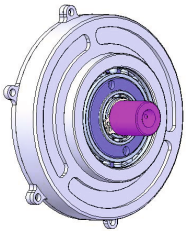


Fig. 3. MR clutch proposed design

B. Transmitted Torque

The transmitted torque to the output shaft, for a variable number of inner disks (n-disk) is calculated using the equations (2)-(4). This n-disk property is the main difference with the one presented in [3].

In order to simplify the calculations, some modifications are done considering that approximately the 95% of the inner disk area is effective in terms of the transmitted torque to the output shaft.

$$\frac{T_d}{n-1} = \frac{\pi\eta \cdot |\Delta\omega|}{h_d}(R_2^4 - R_1^4) + \frac{4}{3}\pi(R_2^3 - R_1^3)\tau_B \tag{2}$$

$$T_h = \frac{\pi\eta \cdot |\Delta\omega|}{h_h} \cdot R_2^4 + \frac{4}{3}\pi \cdot R_2^3 \cdot \tau_B \tag{3}$$

$$T_{tot} = T_d + T_h \tag{4}$$

In equations (2)-(4), the calculation is divided in two parts.

1. T_d is the one corresponding to the transmitted torque due to the inner disks, and T_h is referred to the housing part. The total sum depends also on the number of inner disks n.
   In the expressions above, η is the fluid viscosity with no magnetic field applied, and it is assumed to be a constant value of $0.3Pa \cdot s$.
2. $|\Delta\omega|$ is the relative rotational speed between both, input and output shafts.
3. h_d is the separation between two consecutive inner disks, and h_h is the separation between the closer inner disk to the housing part.
4. R_2 is the inner disk radio, while R_1 is the shaft radio that connects them.
5. τ_B is the yield stress of the fluid, for the applied magnetic field. This variable parameter in function of the magnetic field offers the possibility to control the transmitted torque or the fully release of the elbow joint in the necessary case. From now, it is assumed that the fluid, when magnetized, is always at the same level. Therefore this value is approximated to a constant, that for the rest of the calculations is assumed to be $5 \cdot 10^4 Pa$ (Figure 4).

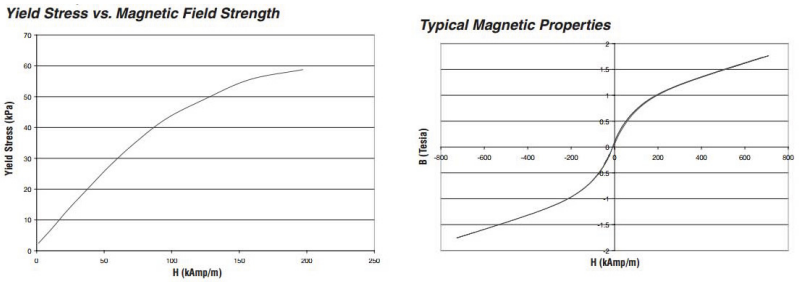


Fig. 4. MRF-140CG Magneto-Rheological Fluid

For the proposed clutch, the maximum torque for transmission was calculated at $8Nm$.

Ultrasonic Motor

The characteristics of these kind of actuators, compared to conventional electromagnetic motors, is that they are suitable for robotic, automotive, aerospace, or prosthesis applications where the requirements are good positioning precision and low speed. Some examples of these robotic applications, where ultrasonic motors were used, are a robotic hand [7], a prosthetic hand and the trembling reduction device [8].

Impaired upper limb function is one of the most common sequels in spinal cord injury (SCI) patients. Upper limb strength is impaired to some extent in people who have suffered cervical SCI, making them difficult to perform many activities of daily living (ADL) such as drinking, eating and personal hygiene. For this reason, they may require technical assistance [9].

A methodology to analyze the kinematic data of the upper limb when performing a functional activity like drinking from a glass is reported in [10]. The forces were reported such as the movement's momentums and also expressed on the proximal reference system of the joint. An additional small external load of $0.300kg$ was simulated in the hand segment in order to simulate light lifting. At the elbow, the maximum flexion moment was of $3.1Nm$ in the C6 group (subjects with metameric level C6 tetraplegia). This value was greater in the C6 group than in control group and C7 group (subjects with metameric level C7 tetraplegia).

According to the characteristics of the test bench (1) and [9], [10], in order to raise the forearm segment it is necessary more than $3.1Nm$ of torque. For this reason, it is used one ultrasonic motor as an actuator device: USR60-E3NT with a speed of $150rpm$ and a maximum torque of $1Nm$ together with the planetary gear with the ratio of 4.3:1. Using a planetary gear, the speed is reduced while increasing the torque to the $4Nm$. This ensemble is able to raise the forearm segment, with a maximum velocity of the $37.5rpm$. The actuator torque and the maximum velocity of this device, is classified between actuators for rehabilitation of elbow joint in flexion/extension.

The USM is endued with one relative encoder with resolutions between 96 and 1024 counts per revolution. With a relative encoder, it is not possible to find the initial position of the forearm, so it is introduced a new absolute encoder in the system. This encoder is based on Hall effect and he have a resolution of 1024 counts per revolution.

2.3 Control Hardware

According to the requirements, the electronic hardware is selected according to the electric power requirements and the characteristics of the input-output ports interface. The architecture is based on an autonomous embedded control system with no need of a PC style computer.

The selected microcontroller is based on 32 bits ARM CPU, clocked at 24 MHz, and clocked at 72 MHz when a LCD color touch screen is being used. This electronic architecture provides enough computational resource for the whole system.

The driver of the USM is the original given by the manufactured, named D6060/24V.

3 Software Simulation

Various software packages have been used for simulating the performance of the proposed device in the test bench, which was designed using Solidworks®.

Mechanical structure of the test bench is developed using one toolbox for Matlab®, Simmechanics®, which offers a comprehensive simulation environment correlated to the mass and the inertia of the arm segments. This environment allowed a co-simulation between the proposed device, (developed in Simulink®) and the test bench. The co-simulation design in Matlab is shown in Figure 5.

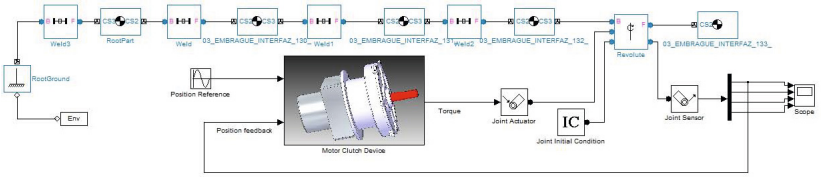


Fig. 5. Design of the control algorithm of the simulated device

The co-simulated model consists of: the USM, the clutch MR, and the test bench. The USM receives as reference signal the angular position of the arm and a PID control algorithm feeds the clutch with the speed and the torque signals. The MR clutch block, using the input parameters (the number of disks and dimensions), calculates the torque output depending on the fluid viscosity (variable depending on the magnetic field). The output signal of the clutch gives the torque for the elbow joint, and a sensor in the joint measures the angular position (which closes the control loop), the angular velocity, the angular acceleration, and the torque (which is identical to the torque generated by the clutch). The first results of the co-simulation can be seen in Figure 6.

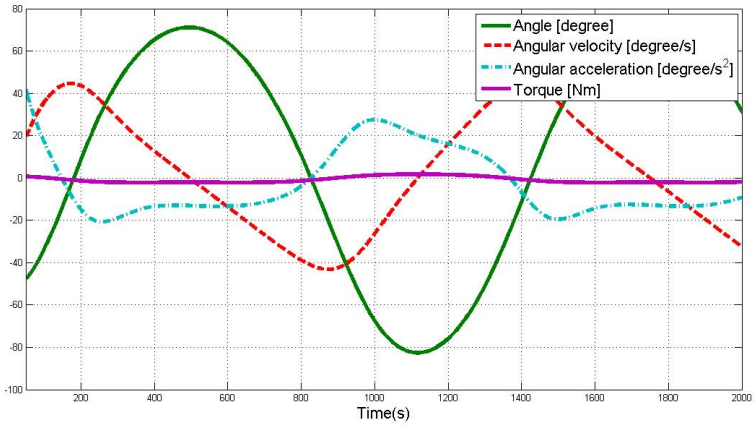


Fig. 6. Simulate result

This software simulation environment provides a useful and accurate tool for the development and tuning the device control algorithms that will be tested in test bench.

The results show that the proposed ultrasonic motor model fits to the parameters of speed, time force, and response time necessary to rehabilitate a human arm (for a 70kg person) without needing any help force of the muscles.

When the speed is $0.5rad/s$ the input torque to the elbow joint is between $-2.2Nm$ and $1.5Nm$. The asymmetry of the torque is due to the angular position, velocity and the inertia of the moving arm.

The results are verified with a specialized software in simulation of musculoskeletal structure, named MSMS®[5]. Using this software, a simulation environment for human arm with muscles has been developed. The model uses the size and the characteristics of a person of $70kg$ of mass (Figure 7).

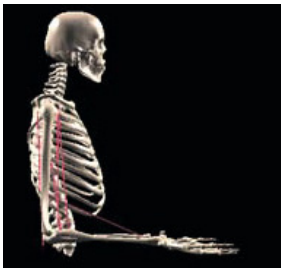


Fig. 7. MSMS®environment

The results (Figure 8) highlight that the torque of the elbow joint, following a sinusoidal movement, does not exceed $2Nm$. The difference with the results giving in Fig. 6 is due to the arm position and the velocity.

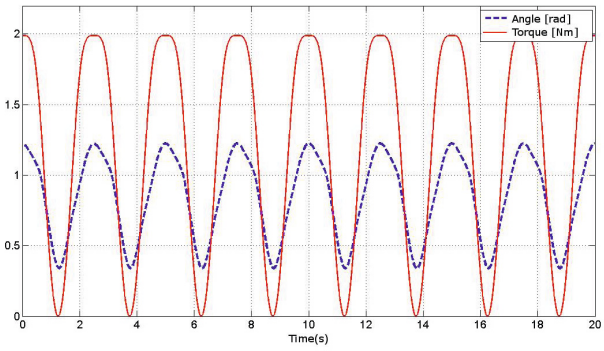


Fig. 8. Simulate result with MSMS®

In the both cases the proposed actuator respects the parameters of the response time, of the torque, and speed and the medical requirements.

4 Control Algorithms

4.1 Control Strategy for the Magnetorheological Clutch

Several control strategies based on the models like: Navier-Stokes, Bouc-Wen, Prandtl-Ishlinskii are presented and analyzed in previous work [4]. The first results showed that Prandtl-Ishlinskii model is the most appropriate for a good response time (short response time). The main advantage is that the use of a hysteresis compensator, as seen in Fig. 9, is fast and precise enough to control the exerted torque, and it is only possible using this model (the analytical inverse exists). The identification procedure of finding the operator values is accomplished using a genetic type algorithm.

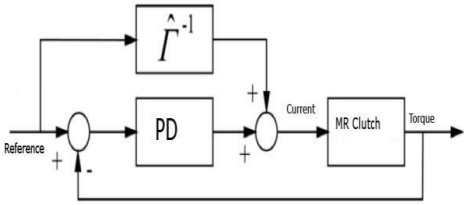


Fig. 9. Hybrid control strategy using a Prandtl-Ishlinskii compensator

4.2 Control Algorithm of USM

In the literature, there are various models of control for USMs: PI control models of speed, position or torque [11] [12], fuzzy models [13], [14], models based on neural networks, or a combination thereof. Because of the complexity of the mathematical model of these types of motors, most control designs are made based on input/output data acquisition (in most cases black-box model).

In this work it is designed a new cascade control model with two control loops for angular velocity, and angular position, each one with its own PI controller. The proposed control scheme consists of a relatively simple algorithm, that meets the objectives, with an error in the position angle of 0.08 degrees and a low computational power use. The control scheme is shown in Figure 10.

The control scheme uses two encoders: a relative encoder, positioned in the shaft of the motor, to calculate the speed and one absolute encoder positioned in the shaft of the gear, to operate at lower speed and to calculate the real position of the arm.

Matlab®and Simulink®are used to develop the control algorithm and regulating the controllers. The program is developed in signed fixed-point 32-bit

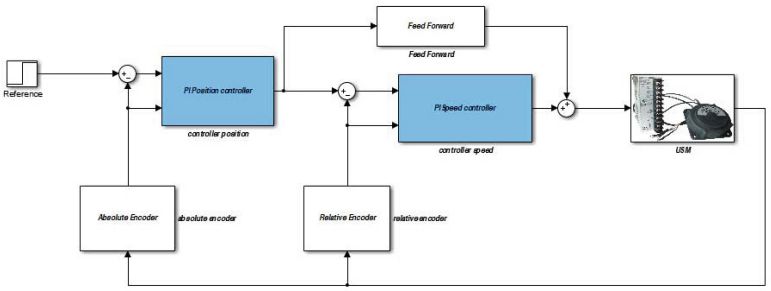


Fig. 10. Design of control algorithm for USM

word length with 10 bits of them for the decimal part, on real-time kernel architecture. Data acquisition was done in real time in Matlab®using port Universal Serial Bus®(USB).

5 Experimental Results

This section shows some of the experimental results obtaining with the test bench. The operation of the USM is analyzed with several experiments to check the correct performance of the prototype. The real and the simulation model of the system can be seen in Figure 11.

The USM answer in the test bench can be seen in Figure 12. The structure of the first device actuator tested is composed of the USM and the planetary gear. Due to the planetary gear tolerance, the error of control in position increases

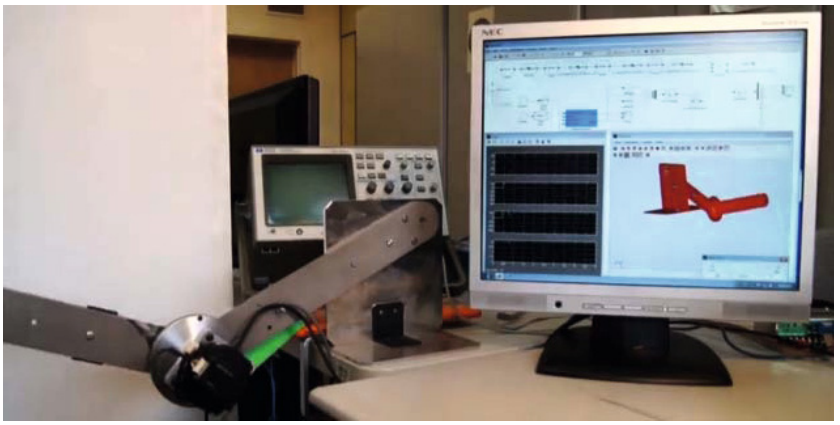


Fig. 11. The real model and the simulation model

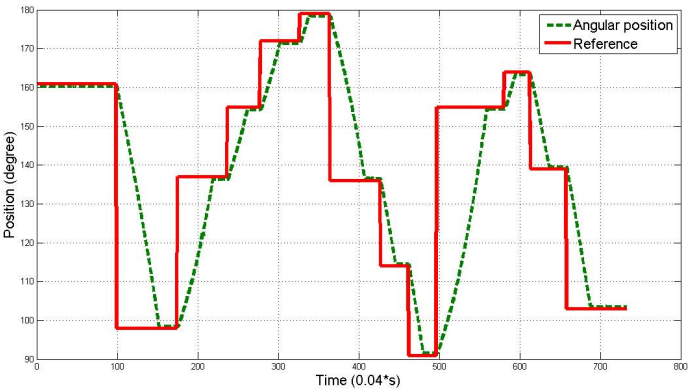


Fig. 12. USM response to step reference with constant speed in test bench

up to 0.7 degrees. In this case, the engine must follow the reference signal in position with a constant speed.

The exoskeleton movement (flexion and extension) need to have a good response to the input signal, in position with one adequate speed. In this case, the proposed algorithm is able to maintain a constant speed while despite of inertia and mass of the arm. Figure 12 highlights the output signal, from cascade control with the reference signal in position.

The first tests of the proposed actuator in the test bench were made by magnetization of the magnetorhelogical clutch with Neodymium magnet, allowing transmitting the necessary torque 1:1 (the torque of output shaft clutch is equal with the torque of the motor). In this case, the clutch is subjected to a magnetic induction of approximately $0.8T$. In the future, the necessary magnetic field will be produced by an electromagnet wich will be able to provide independent control for the torque transmitted by the clutch just by changing the viscosity of the fluid.

In Figure 13 shows the response of the proposed actuator controlled in position when the magnetorhelogical fluid is magnetized, giving the possibility to transmit more than $5Nm$. In this case, the USM have a constant speed of $200degrees/second$.

As can be observed in Fig.13, the clutch not change the performance of the control algorithm, the skating effect not appear for the inner disk.

Figure 14 shows the actuators response to a step input reference where the clutch is not subjected to a magnetic field. In this case, the inner disk of clutch skates leaving the output shaft of actuator free. Then, the torque transmitted is close to $0Nm$. This test highlights the capacity of the actuator to leave free the forearm when the clutch is not subjected to any magnetic field.

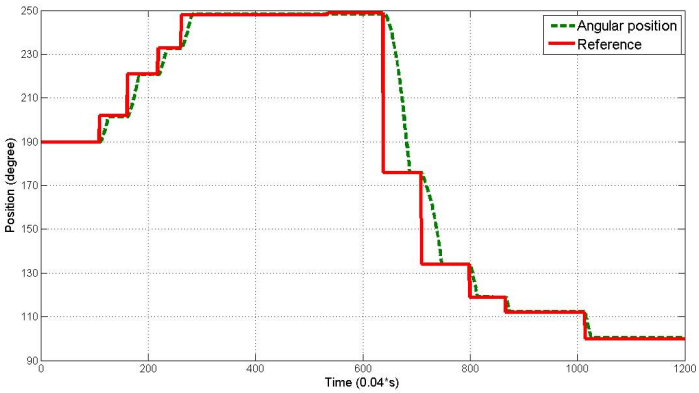


Fig. 13. Actuator response to step reference with constant speed in test bench, when the clutch is magnetized

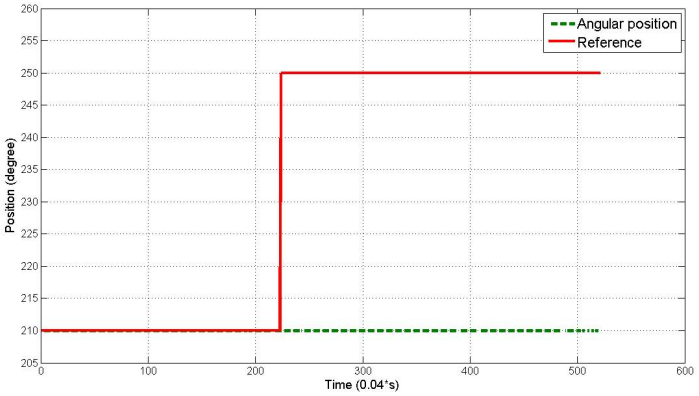


Fig. 14. The actuator response to step reference, when the clutch is not magnetized

6 Conclusions

This paper presents a new type of actuator for rehabilitation device, based on USM and magnetorelogical clutch. This actuator is proposed to be used in exoskeleton for rehabilitation of the elbow joint.

The designed actuator is able to work under real conditions of torque, weight, size, noise and response, according to medical requirements.

The new cascade control scheme, in position control and speed control, has been presented and implemented for USM. The presented results of the control algorithm highlight the precision of the position control.

On the other hand, the preliminary results of the actuator, based on magnetorhelogical clutch, highlight the capacity of the actuator to transmit the torque of the motor. In critical response, when the involuntary contractions occurs (when used in human joints) (such as spasm), the actuator is capable to leaving free the human joint in milliseconds. In the future, the magnetic field will be provided by one electromagnet, giving the possibility to control the torque transmitted by the clutch independently, by changing the viscosity of the fluid with the magnetic field.

Acknowledgment. The authors would like to thank the Spanish Government for the funds provided by the HYPER CONSOLIDER-INGENIO 2010 project. Dorin Copaci, Antonio Flores, Alejandro Martn, Dolores Blanco and Luis Moreno are with Robotics Lab within the Automation and Systems Engineering Department of Carlos III University of Madrid.

References

1. Flores, A., Copaci, D., Martin, A., Blanco, D., Moreno, L.: Smooth and Accurate control of multiple Shape Memory Alloys based actuators via low cost embedded hardware IROS ATBIO (2012)
2. Sashida, T., Kenjo, T.: An Introduction to Ultrasonic Motors, pp. 17–23. Clarendon Press (1993)
3. Hongsheng, H., Juan, W., Liang, C., Jiong, W., Xuezheng, J.: Design, control and test of a magnetorheological fluid fan clutch. In: IEEE International Conference on Automation and Logistics, ICAL 2009, IEEE (2009)
4. Martin, A., Flores, A., Blanco, D., Copaci, D., Moreno, L.: Lightweight Magnetorheological Based Clutch Actuator for quick response times Actuator (2012)
5. Khachani, M., Davoodi, R., Loeb, G.: Musculo-Skeletal Modeling Software (MSMS) for Biomechanics and Virtual Rehabilitation. IEEE Trans. Biomedical Eng (2000)
6. Greene, D., Roberts, S.: Kinesiology: Movement in the Context of Activity, pp. 28–32. Elsevier Mosby (2005)
7. Yamano, I., Maeno, T.: Five-fingered Robot Hand using Ultrasonic Motors and Elastic Elements. In: International Conference on Robotics and Automation (2005)
8. Pons, J.: Wearable robots: biomechatronic exoskeletons, pp. 37–40, 235-242. Wiley (2008)
9. Gil-Agudo, A., Ama-Espinosa, A., Reyes-Guzmn, A., Bernal-Sahun, A., Rocn, E.: Applications of Upper Limb Biomechanical Models in Spinal Cord Injury Patients. In: InTech 2011 (2011)
10. Reyes-Guzmn, A., Gil-Agudo, A., Ama-Espinosa, A., Penasco-Martin, B., Sols-Mozo, M., del Ama-Espinosa, A., Perez-Rizo, E.: Kinematic analysis of the daily activity of drinking from a glass in a population with cervical spinal cord injury. Journal of Neuro Engineering and Rehabilitation (2010)
11. Senjyu, T., Miyazato, H., Uezato, K.: Performance comparison of PI and adaptive controller for adjustable speed drives of ultrasonic motors. In: Proceedings of the IEEE International Conference Industrial Technology (1994)

12. Rodriguez, H., Pons, J.L., Ceres, R.: A ZPET-repetitive speed controller for ultrasonic motors. In: Proceedings of the IEEE International Conference Robotics and Automation, ICRA (2000)
13. Huafeng, L., Chunsheng, Z., Chenglin, G.: Precise position control of ultrasonic motor using fuzzy control. IEEE Ultrasonics Symposium (2004)
14. Izuno, Y., Nakaoka, M.: High performance and high precision ultrasonic motor-actuated positioning servo drive system using improved fuzzy-reasoning controller. In: 25th Annual IEEE Power Electronics Specialists Conference, PESC 1994 Record (1994)

Skeletal Modeling, Analysis and Simulation of Upper Limb of Human Shoulder under Brachial Plexus Injury

Cecilia E. García Cena[1,2], Roque Saltarén Pazmiño[1], Marie André Destarac[1], Edgar Loranca Vega[2], Ricardo Espinosa Gomez[1], and Rafael Aracil Santonja[1]

[1] Centre for Automation and Robotics, José Gutiérrez Abascal, 2 28006, Madrid
[2] Escuela Técnica Superior de Ingeniería y Diseño Industrial,
Ronda de Valencia, 3 28012, Madrid
cecilia.garcia@upm.es

Abstract. This article presents the skeletal-mathematical model and a measurement system for providing real-time information related to position, angular velocity, and orientation of different movements related to the upper limb of human shoulder using Inertial Measurement Units (IMU). The main goal of this system is to improve the rehabilitation tasks of unilateral brachial plexus injury, therefore a complete kinematic analysis of the shoulder skeletal system, considering only the essential variables to simulate the movements used in rehabilitation, is detailed. Finally, are presented some experimental tests based on joint physiology and biomechanics of human shoulder for children less than 10 years, according to desired restriction of medical staff belong to Hospital Infanta Sofía of Madrid.

Keywords: Brachial Plexus Injury, Kinematic Model, Kinematic Analysis, Inertial Sensor Unit, Experimental test.

1 Introduction

The unilateral brachial plexus injury is the lesion o tear of a group of nerves located in the neck region. These nerves travel thru the axial pass under the neck bone and ramify to give place to most of the nerves that allow the movement and give sensitivity to the arm and forearm [1], [2].

According to the National Statistic Institute in Spain, the current birth rate is 400,000 children per year; this allows estimating a frequency of 400 to 800 obstetric paralyses per year. Although most of the babies heal on their own, still a 10 to 20 percent remain with permanent weakness. According to [2], it is stipulated that the brachial plexus paralysis is a prevalent lesion found in new born children, this lesion may be seen in births of a traumatic nature. This type of paralysis can cause long term functional disability. Furthermore, obstetric brachial lesions may present certain peculiarities: when is produced in the moment of birth, they have higher rate of recuperation, than those seen in adult patients [3].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_14,

The mechanic forces acting on an infant during birth, especially the traction and compression can produce a variety of lesions to the brachial plexus. For example, [2] explains a shoulder dystocia (it is produced when the anterior region of the infants shoulder is impacted on to the mothers pubic simphysis as seen on Figure 1a) during vaginal birth with a cephalic presentation, this may produce an over extension of the neck and therefore generating a lesion to the plexus. Nevertheless this type of lesion is also seen after a c-section and vaginal births with an abnormal presentation; for example a pelvic presentation shown in Figure 1b).

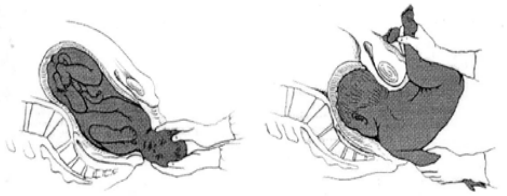


Fig. 1. Production mechanisms of obstetric paralysis. a) Shoulder dystocia b) Delivery in pelvic presentation.

It is also explains that risk factors may be maternal, fetal or those associated with a difficult birth. Within the risks on the maternal side, it can highlight an excessive weight, maternal diabetes, uterine abnormalities and history of brachial plexus paralysis. Risks associated with the infant it is only mentioned fetal macrosomy finally risks associated with birth include shoulder dystocia, prolonged labor, assisted delivery with forceps or vacuum extractors and podalic presentation.

The location of the lesion is determined through a clinical analysis; however the type of lesion is established in correlation with the recovery time. For example, it is considered that if the superior roots are dissected an Erb palsy would occur, better known as “bad shoulder-good hand”. In this type of palsy there is a shoulder function deficiency and limited hand function as shown in fig. 2a. According to [2] in this lesion the muscle group affected is the external rotators, the shoulder abductors, elbow flexors, forearm supinators and frequently the wrist extensors. In other words, the superior musculature is affected. On the other hand, if the inferior nerve roots are dissected a Klumpke plasy would occur. This palsy is known as “good shoulder –bad hand”. It can observe a lesion were the muscles affected are the wrist flexors, finger flexors and the hand intrinsic muscles, as seen on fig. 2b. A complete lesion would affect both the shoulder movements as well as the hand movement. In this type of lesion all the roots within the plexus are affected [2].

The superior paralysis is the most frequent. It is estimated to represent almost 80% of the cases, more than 90% of this lesion resolves within a year of the infants life. A complete paralysis appears in 20%, the inferior lesion has the lower incidence a 2 %.

According to [2], thorough studies show that during the systematic exploration of the brachial plexus after a lesion only 20-30% showed functional sequels, this concludes that the rate of recovery is elevated. Classification of the type of lesion may help to determine the long term recovery prognosis. However the impact of the lesion

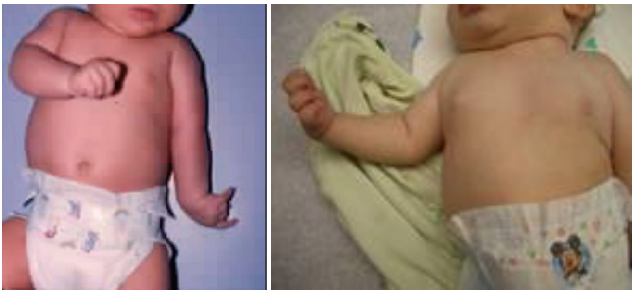


Fig. 2. Brachial plexus injuries. a) Upper left limb injury. b) Lower right arm injury type [4].

over arm and hand function during lactancy could generate significant sequels over motor and sensitive functions in the development of the infant.

According to [2] - [5] the treatment for brachial plexus paralysis is initiated within the neonatology unit were a professional specialized in passive exercises initiates therapy and gives support and guidelines to the family. It is of special importance to reduce the stress over the plexus both from the gravity force and traction force. The exercise will reduce the risk of atrophy and muscle contractures on the long term; this allows having a better ground to work in case of a reconstructive surgery. Although the majority of cases show a rapid recovery, it is important to keep the public informed about the lesion its evolution and treatment.

The persistence of symptoms beyond the first month of live suggests that the lesion could require specialized treatment. The surgical procedures for infants according to [2], [3] consist in a meticulous exploration of the plexus to determine the location and extension of the lesion. The lesion may require nerve graft. Self graft implantation both from the sural region of the leg, or coetaneous arm nerves. The objective of the surgery is to restore muscle equilibrium and free possible contractures or deformities. The recovery time after surgery is usually between 7 to 9 months, rehabilitation and nerve recognition therapy is necessary. Both therapies are crucial to improve strength sensitivity and movement of the intervened extremity.

The development of new technology has allowed a better cinematic analysis of body movement, in hand helping with the development of new tools for the treatment. Both biomedics and biomechanics represent a viable alternative to help physicians to discover the degree of damage and assist in better patient's rehabilitation.

In this article a direct kinematic model is presented in order to get the biomechanical parameters that describe the behavior of rigid bodies that form the human shoulder. These parameters are related to both the position and the orientation of shoulder bone structure. Authors also present the development of software and hardware architecture developed for monitoring the rehabilitation process. The system is made of an inertial measurement unit that is fixed in the shoulder of patient. Data are collected in user interface where it is possible to observe the range of shoulder's movement.

Section 2 presents a description of the shoulder anatomy and the Denavit-Hartenberg parameters are showed. The kinematic analysis is also presented taking into consideration the manipulability measure. Section 3 describes the hardware and

software architecture developed to collect data of the shoulder rehabilitation. Section 4 presents some experimental results performed with a healthy child under 10 years old. Finally, conclusions and futures steps in this research are commented.

2 Skeletal Model of Human Shoulder under the Pathology

The human shoulder is an extremely sophisticated and interrelated system that can produce a wide variety of complex movements in space. Since a geometrical point of view, the shoulder has between 13 and 16 degree of freedom (DoF) and the range of movement are increased by muscles, ligaments and tendons. All of these elements are connected between them; therefore the model is complex but not impossible.

In [6] it is stated that, given the complexity of the human shoulder, it is convenient to model the shoulder girdle (clavicle and scapula) as a closed kinematic chain (the simplest of parallel mechanisms) while the glenohumeral joint should be modeled as serial chain.

The foregoing is based on the girdle is responsible to support the load and the inertial forces generated by the arm to manipulate objects. Furthermore, the glenohumeral joint has the same range of motion but lower load capacity.

In [7], girdle is modeled as an element with 5 DoF: 3 for the glenohumeral joint and 2 to the sternoclavicular joint. Authors used Denavit-Harbtenberg (D-H) methodology to determine the parameters of mobility.

In [8] it is stated that there are two ways to model the kinematic chain of the human shoulder, open chain and closed chain. The open chain corresponds to three rigid bodies (clavicle, scapula and humerus bone) connected by three joints (sternoclavicular joint, glenohumeral joint and claviescapular join) which result in a 9 DoF mechanism.

The closed chain is defined with the interpretation of "false scapular thoracic", which is a kinematic connection that allows modeling the shoulder girdle. Also it provides the possibility of sliding on the anterior part of thorax with possible rotation on its axis perpendicular.

On the other hand [9] [10] modeled the human arm as a rigid body kinematic chain with three joints (shoulder, elbow, elbow, wrist, wrist-hand) with six degrees of freedom, three in the shoulder joint (flexion / extension, abduction / adduction and rotation), two in the joint of the elbow (flexion / extension), and one in the junction of the wrist (flexion / extension).

Figure 3 shows the elements used to simulate the movement of the shoulder, the sternoclavicular and glenohumeral joint and collarbone as one rigid link. Figure 4 shows the coordinate systems of each joint according to the D-H convention.

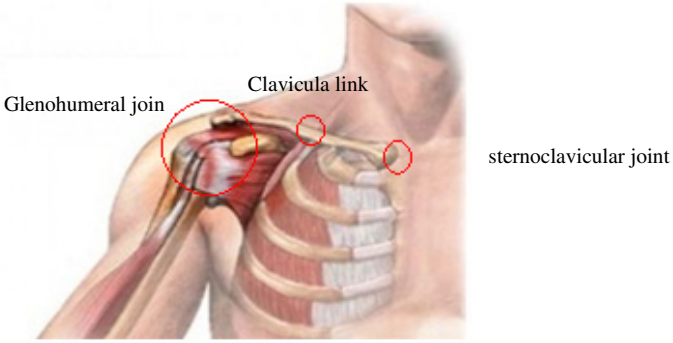


Fig. 3. Bone model used for implementation of kinematic chain: glenohumeral and sternoclavicular union, and clavicle. [11].

2.1 Shoulder's Denavit-Hartenberg Parameters and Kinematic Analysis

As can be seen, each reference system is associated with the kinematic chain described: 2 DoF for the sternoclavicular joint (SC), and 3 DoF for the glenohumeral joint (GH).

It is noteworthy that there are two offsets with respect to the axes modeled:

- Longitudinal displacement along x_1 axis (distance A in Figure 4). This corresponds to the distance between the sternoclavicular joint and the glenohumeral joint region (clavicle length).
- Displacement along z_2 axis that represents the distance between the glenohumeral joint and scapula (distance D in Figure 4).

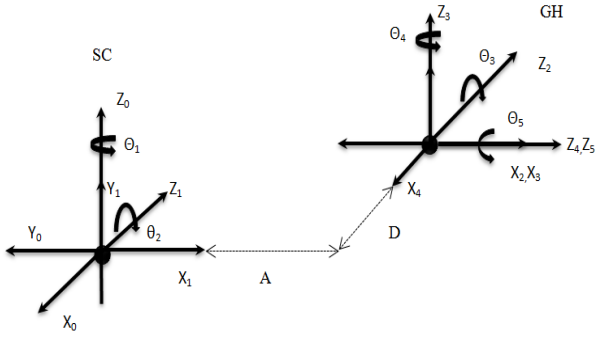


Fig. 4. Reference systems according to D-H convention

The values considered for constant A is 8 cm and 3 cm for constant D. These values are according to a child under 10 years old.

Table 1 summarizes the DH parameters considered in the shoulder model.

Table 1. D-H Parameters

DoF	θ	d	a	α
1	$q_1 + \pi/2$	0	0	$\pi/2$
2	q_2	D	A	0
3	q_3	0	0	$-\pi/2$
4	$q_4 + 3\pi/2$	0	0	$\pi/2$
5	q_5	0	0	0

Figure 5 shows an interface development under Matlab® to simulate the kinematics parameters given in Table 1.

Clearly it can be observed the 5 DoF modeled: 2 DoF for the sternoclavicular joint and 3 DoF for the glenohumeral joint. Thereby, it validates that the kinematic behavior of the model is according to the basis movement of human's shoulder. After this the Jacobian matrix was computed and the manipulability measure (ec. (1)) was evaluated [12].

$$w = \sqrt{\det J(q) J^T(q)} \tag{1}$$

The interface was especially useful to get the limit of movements of sternoclavicular and glenohumeral joints. After many tests presented in section 4 it was found that the range of movement for both is between -80 and -45°. The same results were reported in medical literature like [2], [5], [13].

The main conclusion of this study is that for muscle, tendons and ligaments provide greater range of motion to the human arm.

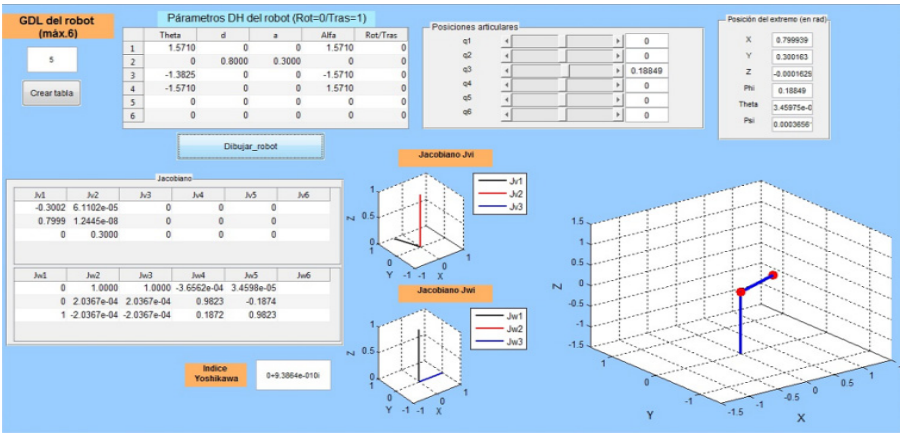


Fig. 5. Kinematic user interface

Table 2 summarizes the results obtained by varying the first degree of freedom of the sternoclavicular joint from -80 ° to -45 °. The remaining joints were fixed at 0°. Figure 6 shows the manipulability ellipsoid. It could be observed that the Jacobian matrix is well-defined when q_1 is close to -70º.

Table 2. Manipulability analysis for -45º<q_1< -80º

q_1	Minor axis of ellipsoid $\sigma_1 u_1$	Major axis of ellipsoid $\sigma_2 u_2$	**Manipulability Measure** w
-80º	0.98174	0.1126	1.4029e-006
-70º	0.9812	0.076	1.8583e-006
-60º	0.9808	0.040	7.3367e-007
-50º	0.9803	0.025	1.3949e-006
-45º	0.9730	0.020	8.9465e-007

Table 3 shows that the manipulability index is not altered when the second degree of freedom of sternoclavicular joint moves. Therefore, it was fixed in -70° and it was concluded that the first DoF of sternoclavicular joint is the most relevant for the movement of the shoulder if the bone model is only taken into consideration.

In thirst test, q_3 joint was moved while the rest joint remained constant in 0º. The dependence between glenohumeral and sternoclavicular joints was demonstrated. Table 4 summarized the results. It can be observed that the manipulability index is null for each movement considered to q_3 joint.

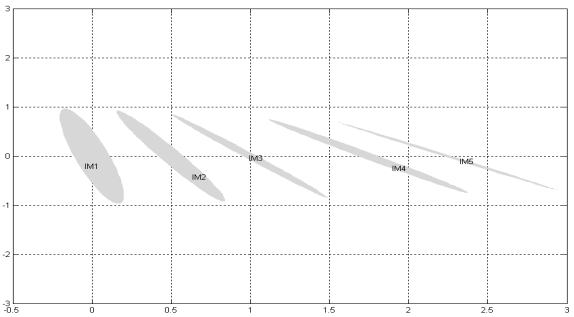


Fig. 6. Manipulability ellipsoid for -45º<q_1< -80º

Table 3. Manipulability analysis for -45º<q_1< -80º and q_2=70º

q_1	q_2	Minor axis of ellipsoid $\sigma_1 u_1$	Major axis of ellipsoid $\sigma_2 u_2$	**Manipulability** w
-80º	-70º	0.9812	0.1126	1.4029e-006
-70º	-70º	0.9812	0.076	1.8583e-006
-60º	-70º	0.9808	0.048	7.3367e-007
-50º	-70º	0.9803	0.028	1.3949e-006
-45º	-70º	0.980	0.028	8.9465e-007

Table 4. Manipulability analysis for -45º<q_1< -80º and q_2=70º

q_3	Minor axis of ellipsoid $\sigma_1 u_1$	Major axis of ellipsoid $\sigma_2 u_2$	**Manipulability** w
-80º	0.9863	0	0
-70º	0.9794	0	0
-60º	0.9792	0	0
-50º	0.9785	0	0
-45º	0.9860	0	0

Table 5. Manipulabity measurment for SC and GH joints

q_1	q_2	q_3	Minor axis $\sigma_1 u_1$	Major axis $\sigma_2 u_2$	**Manipulability** w
-70º	-70º	-80º	0.9812	0.076	1.8583e-006
-70º	-70º	-70º	0.9812	0.076	1.8583e-006
-70º	-70º	-60º	0.9812	0.076	1.8583e-006
-70º	-70º	-50º	0.9812	0.076	1.8583e-006
-70º	-70º	-45º	0.9812	0.076	1.8583e-006

In the last test it was settled the values for sternoclavicular joint at -70º while q_3 was settled as it is shown in Table 5. The results obtained for manipulability measure show that the range of shoulder's motion is conditioned primarily by sternoclavicular joint while glenohumeral provides stability and it does not affect to range of motion.

3 Hardware and Software Architecture

For getting data from the movement of human shoulder, a commercial (Razor 9DoF AHRS) inertial measurement unit (I.M.U.) sensor was used. Figure 7 shows a view of the user interface developed to visualized data from IMU sensor. In the user interface it is possible to see the data's graphics and details of communication between sensor and PC.

The IMU has implemented an Atmel microcontroller through which programs can dump the data to control 3 sensors. Also, IMU has already implemented Firmware Test Program, which provides the data in different ways (floating point, ASCII data or binary). The IMU is connected to PC through USB port.

The IMU is provided with a 3-axis ADXL345 accelerometer, a 2-axis LPR530AL gyroscope that is able to take measure in X and Y axis and, a LY530ALH gyroscope for measuring along Z axis and finally a magnetometer HMC5843 model.

After calibration, it is able to provide data from Euler angles (yaw, pitch and roll). Calibration of the sensors, and programming the data fusion algorithm was performed under Arduino®.

For processing data from IMU the direction cosine matrix was used. The matrix is composed of all possible combinations of the vectors resulting from the information measured by the sensors.

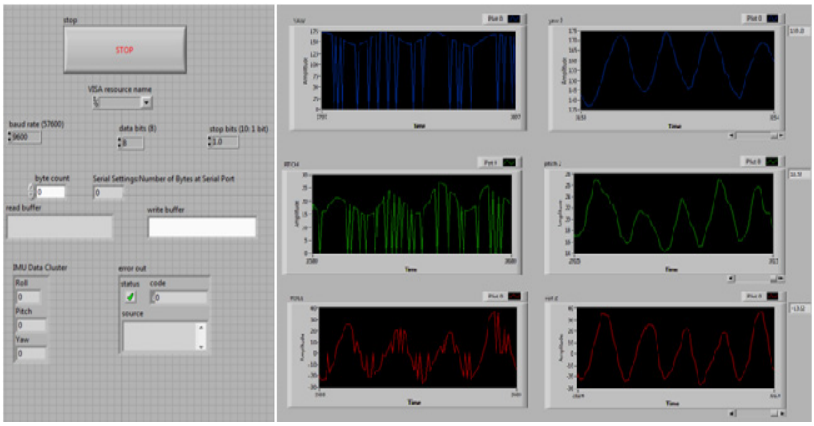


Fig. 7. I.M.U Sensor and User Interface designed for getting experimental data

If the matrix is given by ec. (2), the pitch, roll and yaw angles can be computed as ec.(3)-(5) show respectively.

$$DCMR=\begin{bmatrix} \cos\theta\cos\phi & sen\psi sen\theta\cos\phi-\cos\psi sen\phi & \cos\psi sen\theta\cos\phi+sen\psi sen\phi \\ \cos\theta sen\phi & sen\psi sen\theta sen\phi+\cos\psi\cos\phi & \cos\psi sen\theta sen\phi-sen\psi sen\phi \\ -sen\theta & sen\psi\cos\theta & \cos\psi\cos\theta \end{bmatrix} \quad (2)$$

$$R_{31}=RK_i=-sen(\theta) \;\rightarrow\; \theta=-sen^{-1}(Rk_i) \quad (3)$$

$$\frac{R_{32}}{R_{33}}=\frac{sen(\psi)\cos(\theta)}{\cos(\psi)\cos(\theta)}=\tan(\psi) \;\rightarrow\; \psi=a\tan 2\left(\frac{R_{32}}{R_{33}}\right) \quad (4)$$

$$\frac{R_{21}}{R_{11}}=\frac{sen(\psi)sen(\theta)\cos(\phi)-\cos(\psi)sen(\phi)}{\cos(\theta)sen(\phi)}=\left(\frac{R_{21}}{\cos(\theta)},\frac{R_{11}}{\cos(\theta)}\right)\rightarrow\phi=a\tan 2\left(\frac{R_{21}}{\cos(\theta)},\frac{R_{11}}{\cos(\theta)}\right) \quad (5)$$

4 Experimental Results about Articular Physiology

Experimental tests were performed with a healthy child under 10 year-old. The movements are performed according to the joint physiology [14] and the shoulder biomechanics [15]. These movements are important in the process of rehabilitation of a child with the disease.

The tests were performed from a neutral anatomical position, i.e. with elbows bent at 90º so that the forearm was in the sagittal plane. The IMU is placed on the child glenohumeral joint as shown in Figure 8.

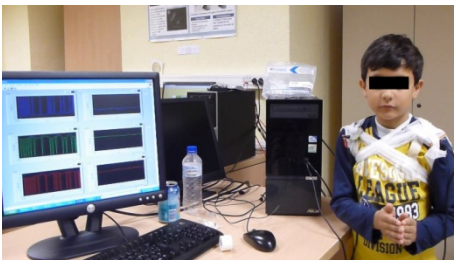


Fig. 8. Neutral anatomical position

Figure 9a) shows the results obtained when the child makes a move by the contralateral anterior way. Child was asked to perform 3 or 4 slow movements and prolonged manner like a child with the disease.

Figure 9b) shows the results obtained when child performs a homolateral movement.

The Three Phases Deflection [15] movement is reported in Figure 9c). This movement is the one with greater range and the sensor could capture all the data when the shoulder moves up and down an average of 4 times.

Finally, Figure 9d) shows recording data when the child makes the movement known as "Measure Hippocratic of Abduction". This starts from the rest position (Figure 8) and an angle of 120° to the dorsal trunk side. This exercise represents the action cleaning the head.

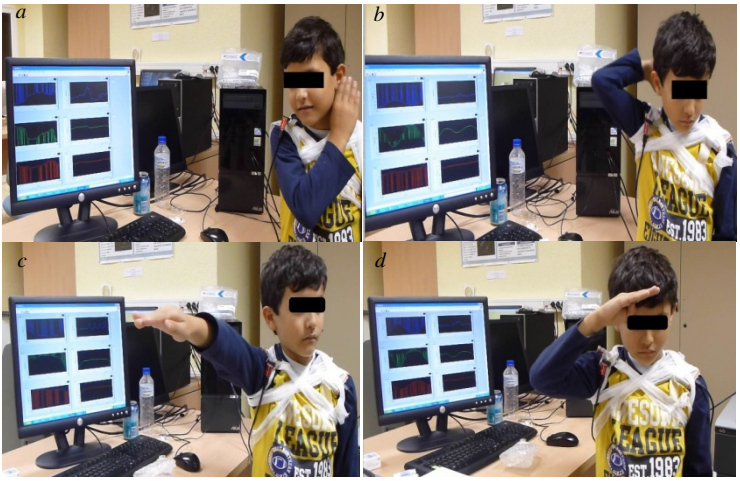


Fig. 9. Experimental Results

In [16] a complete description about software implementation and other tests including healthy adults can be follow.

These experimental tests were performed following the hospital' rehabilitation protocol. Despite they were performed with a healthy child it was possible to get a feedback from medical staff. Some of them are sketched below:

- The information provided by the sensor is useful for evaluating the progress of rehabilitation tasks.
- The presentation of the information is adequate and easy to understand.
- The user interface can provide extra information about patient's medical history.
- A flexible mechanism must be designed in order to fix the sensor to patient's shoulder.
- It could be very useful to check the sensor in rehabilitation of other members like legs, arms, knee, hit, etc.
- The user interface's appearance must be improved in order to be easily accepted (for example, the black color in the bottom of figure is not recommendable).

5 Conclusions

In this paper authors presented two results needed in the development of robotic orthosis for treatment of unilateral brachial plexus pathology.

A rigorous study of the skeletal system that forms human shoulder was presented. This system has between 13 and 16 DoF, but in order to model the plexus brachial injure, 5 DoF are enough. Tree of these DoF can be located in the glenohumeral joint while the other two in the sternoclavicular join.

Injury is considered as *prevalent* because during the last three decades has not been registered a decrease in the rate of its occurrence. Therefore, this research area must be explored.

This paper shows a kinematic model and analysis of the skeletal shoulder system under plexus brachial injury.

From the model obtained, it was able to make a study on the range of shoulder motion and it has proven the joint interdependence even if the nerve is damage. Furthermore it has been observed that glenohumeral joint is one of the responsible of the shoulder movement while the sternoclavicular provides stability to the arm.

Using classical techniques of robot kinematic modeling it was possible to obtain the range of sholder's motion and analyze its behavior when the joint references are modified. A specific user interface was implemented in Matlab. The interface shows results in an easy and fast way.

The kinematic model and its analysis is a very useful result in order to design an exoskeleton for improving the rehabilitation tasks.

In this article authors also present the hardware and software architecture development for getting data from real movements. A commercial inertial measurement unit was programmed and calibrated. Data is showed in other user interface designed under LabView. The IMU was mounted in a healthy kid's shoulder and he was asked to perform typical movement in rehabilitation protocol. The feedback provided by medical staff has been very positive and the system can be used in the next future.

5.1 Future Steps in This Research

The bone model is not enough to model the brachial plexus injury. Therefore authors are working in order to complete the model using muscles, tendons and ligaments. These elements are responsible of the forces used to move the arm and then it will be possible to get dynamical results.

Using kinematics and dynamical results, an exoskeleton can be developed to help medical staff in the hard tasks of rehabilitation. The information from inertial sensor will be used to control the mechanism at the same time that it provides rehabilitation data.

It is important to research about how many sensors must be used and where they must be fixed.

Acknowledgements. The authors would like to thank the financial support of the Spanish Government through the CICYT Project Ref. DPI2009-08778 and also to Madrid Government who supports the project ROBOCITY2030-II Ref. P2009/DPI-1559.

Authors want to thank the technical support of Dr. Javier López López from Infanta Sofía Hospital and his rehabilitation staff.

References

1. Palastanga, N., Field, D., Soames, R.: Anatomía y Movimiento Humano Estructura y Movimiento. Paidotribo, Barcelona (2000)
2. Sutcliff, T.: Brachial plexus injury in the newborn. Neo Reviews: An official Journal of the American Academy of Pediatrics 8(6), 239–246 (2007)
3. Marrero, L.O., Cabrera, N., Rodríguez-Triana, J.A., Navarro, A., et al.: Diagnóstico y Tratamiento de la Parálisis Braquial Obstétrica. Complejo Científico Ortopédico Frank País, La Habana Cuba (2010)
4. Debora, G.: Plexus braquial Injury: Diagnosis and treatment. medigraphic.org.mx, México City (2012)
5. Bradley, W., Daro, R., Fenichel, G., Jankovic, J.: Neurology in Clinical Practice, cap. 79. Editorial Butterworth-Heinemann Elsevier, Philadelphia (2008)
6. Jadran, L., Stanisic, M.: A humanoid Shoulder Complex and The Humeral Pointing Kinematics. IEEE Transactions on Robotics and Automation 19(3) (2003)
7. André, S., Helm Frans, V.D.C.: Kinematic Design to Improve Ergonomics. IEEE Transactions on Neural System and Rehabilitation Engineering 14(4) (2006)
8. Tondu, B.: Modelling of The Shoulder Complex and Application to the Design of Upper Extremities for Humanoid Robots. In: IEEE-RAS International Conference on Humanoid Robots (2010)
9. Yaquin, T., Huosheng, H.: A novel sensing Data Fusion System for 3D Arm Motion Tracking in Telerehabilitation. IEEE Transactions on Instrumentation And Measurement 57(5) (2008)
10. Perez, R., Costa, Ú., Torrent, M.: Upper Limb Portable Motion Analysis System Based on Inertial Technology for Neurorehabilitation Purposes, Sensors (2010)

11. Torné, J.D.: Traumatology and Orthopaedic Surgery (2012), http://www.delgadotrauma.com, http://www.delgadotrauma.com/Miembro-Superior.html (Último acceso: 30 Enero 2013)
12. Yoshikawa, T.: Analysis and Control. Massachusetts Institute of Technology, Massachusetts (1990)
13. Loureiro, R., Harwin, W., Nagai, K., Johnson, M.: Advances in upper limb stroke rehabilitation: a technology push. Medical and Biological Engineering and Computing 49(10), 1103–1118 (2011) ISSN: 1741-0444
14. Kapandji, A.: Fisiología Articular. Editorial Médica Panamericana, Madrid (2007)
15. Hall, B.: Basic Biomechanics. Mc-Graw Hill (2007)
16. Loranca Vega, E.: Model and Kinematic Analysis of the human shoulder for rehabilitation treatments of Brachial Plexus Injury. Master Thesis. Universidad Politécnica de Madrid (March 2013)

Assistive Robot Multi-modal Interaction with Augmented 3D Vision and Dialogue

Juan G. Victores, Félix R. Cañadillas, Santiago Morante, Alberto Jardón, and Carlos Balaguer

Robotics Lab Research Group within the Department of Systems Engineering and Automation, Universidad Carlos III de Madrid (UC3M),
jcgvicto@ing.uc3m.es

Abstract. This paper presents a multi-modal interface for interaction between people with physical disabilities and an assistive robot. This interaction is performed through a dialogue mechanism and augmented 3D vision glasses to provide visual assistance to an end user commanding an assistive robot to perform Daily Life Activities (DLAs). The augmented 3D vision glasses may provide augmented reality vision of menus and information dialogues over the view of the real world, or in a simulator environment for laboratory tests and user evaluation. The actual dialogue is implemented as a finite state machine, and includes possibilities of Automatic Speech Recognition (ASR), and a Text-to-Speech (TTS) converter. The final study focuses on studying the effectiveness of these visual and auditory aids for enabling the end user to command the assistive robot ASIBOT to perform a given task.

Keywords: assistive robotics, end-user development, human-robot interaction, multi-modal interaction, augmented reality, speech recognition.

1 Introduction

During the past years, and due to the complexity of systems and robotic platforms to control, the importance of the developments in the field of Human-Robot Interaction (HRI) has been greatly increasing. This is most noticeable in systems developed for people with disabilities, such as assistive robotic systems, where HRI's must be designed taking the type of users who will use the system as well as their disabilities into account. In recent years, innovative breakthroughs have been developed in this field thanks to the development of multi-modal interfaces that may adapt to the needs of the users of these systems.

This paper presents the latest developments in multi-modal interfaces with the ASIBOT assistive robot [1]. The main components are an augmented reality 3D vision glasses system with inclinometer, and an interactive dialogue mechanism which has been implemented as a finite state machine. Figure 1 depicts an actual screenshot of the user's view of the developed augmented reality interface at run-time. The system as a whole is capable of guiding the user through the different options within the interactive dialogue, while presenting synchronized

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_15,

Fig. 1. Assistive Robot Multi-Modal Interaction Augmented 3D Vision screenshot

information to the user through the augmented reality 3D vision glasses interface. Through the interactive dialogue, the user is capable of commanding the robot to perform a set of actions, as well as controlling several different visual and functional aspects of the interface.

Section 2 of this paper describes the State of the Art and several works related to the presented developments. Section 3 presents the Open System Architecture in its current implementation. Section 4 describes the experiments performed allowing users to test the system. Finally, Section 5 outlines several conclusions.

2 State of the Art

Augmented reality systems are being widely used for the development of multimodal interfaces, as they provide the possibility of multiple configurations. One of the main articles dealing with this kind of device is [2], which defines the types of configurations that augmented reality systems may have. In the article, systems called Head-Mounted Displays (HMD) are defined, a concept that refers to screens located near the eyes of a user's head, in addition to showing the advantages and disadvantages of using this kind of system. Subsequently, this study was updated in [3], introducing new concepts such as the called Head-Worn Displays (HWD), based on the use of small projectors that project onto a semi-transparent mirror to display information over the real world. Another relevant article is [4], where a survey is performed, which shows the main features of the displays used in augmented reality systems. In addition, in the article, the different techniques and positioning of the augmented reality systems based on the new technologies used in this field is described.

Before studying the different fields that rely on augmented reality technologies, it is important to review how the collaboration between humans and robots

is when using augmented reality interfaces. A review on how HRI must be in the context of augmented reality interfaces can be found in [5]. This proposal is later evaluated in [6] by the same authors.

As is shown in [7], technologies based on augmented reality are being used in many fields thanks to several possibilities that these system allow. In the industrial robotic field, augmented reality systems are used for the tele-operation of industrial robots. The use of an augmented reality interface for the control of a manipulator robot in unstructured environments is described in [8]. Another example is the system proposed in [9], where the positioning of a robot and generation of its trajectories is obtained through the use of augmented reality. A different area where augmented reality systems are being used is in the field of robotics oriented medical applications. In [10], the authors suggest the use of augmented reality system for controlling a robot as support in surgery tasks. Another interesting application in this field is proposed in [11], where they use augmented reality as support for performing laparoscopic surgery. In rehabilitation robotics, studies are underway, also based on augmented reality with the aim to help patients with mobility disabilities. In [12], the authors use an augmented reality system to support the rehabilitation of the hand following a cerebrovascular condition. A more limited number of studies can be found in the field of assistive robotics. Among them is the system proposed in [13], which uses an augmented reality setup for control and interaction between a wheelchair and the user.

The most common interfaces that are being used in assistive and social robotics are based on voice recognition, where a person can interact with a robot through voice commands, sending orders or requesting information. An example of interactive dialogue mechanism between a human and a social robot can be found in [14]. Examples of robots that may perform tasks, actions, or exchange information with a user can be found in [15] or [16]. An example of a voice recognition system to control a wireless assistive environment can be found in [17].

3 Open System Architecture

The ASIBOT Open System Architecture is provided through the use of the YARP [18] robotics platform. It acts as glue between the components, which are simultaneously decoupled and asynchronously updated with the flow of user and environmental information. Figure 2 depicts a basic connection diagram between the different components that compose the architecture. Two types of connections are used: streaming data flow connections for information that should be updated quickly, and remote procedure port connections that return acknowledgements of reception that may also be used for information on the degree of accomplishment of a certain task.

The following subsections will describe some of the main characteristics of the different components that compose the system and are depicted in the diagram.

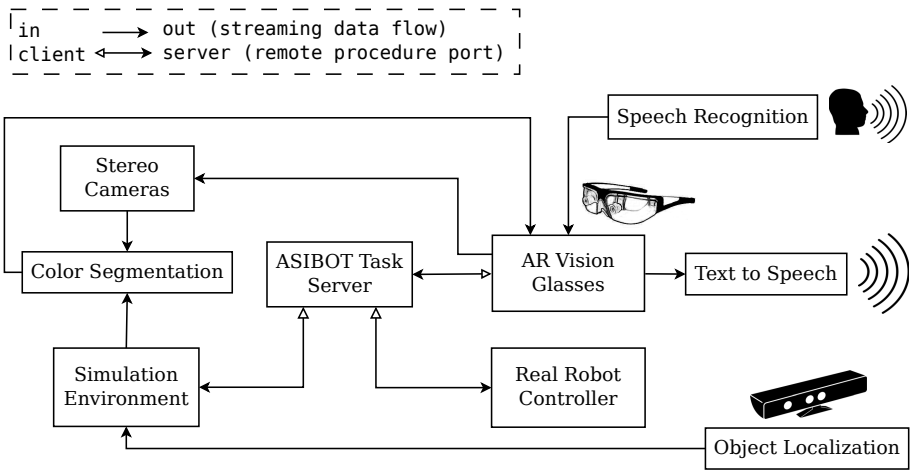


Fig. 2. Assistive Robot Multi-Modal Interaction Augmented 3D Vision scheme

3.1 Augmented Reality 3D Vision Glasses

The Augmented Reality 3D Vision Glasses used in this system are the Vuzix Wrap 920AR Glasses, which are shown in Figure 3. This device provides a Head-Mounted Display with two frontal cameras for the stereo video capture. In addition, it has a 6-degree of freedom head inclination tracker, and a high fidelity stereo audio output.

Fig. 3. Vuxiz Wrap 920AR Augmented Reality System

The development of the interface that is shown on the display is based on the Open Source OpenGL libraries. To develop the augmented reality graphical interface, text and geometrical elements with textures are overlapped upon the real world or simulator images that are incoming from the cameras.

3.2 Simulation Environment

Our basic setup usually involves a three-layered structure: the simulator class (which uses the OpenRAVE-core libraries for graphical and physical aspects),

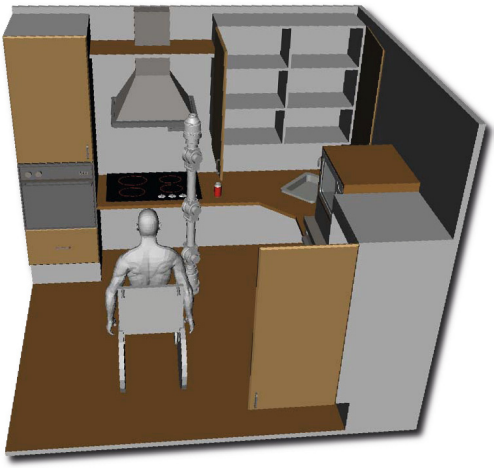


Fig. 4. ASIBOT assistive kitchen simulation environment

a robot kinematic solver class, and a robot kinematic controller class. Figure 4 depicts the default loaded simulated environment.

As a new feature for this application, an OpenRAVE plugin called externObj has been developed. This plugin enables a Simulation Environment input port that receives streaming data from the Object Localization module, synchronizing the position of an object in the simulated environment with the actual real position of an object provided by the Object Localization module.

3.3 Real Robot Controller

The Real Robot Controller is also three-layered, with the real robot controller class at the motor-sensor level. This class manages the movements of the real robot using CAN-bus messaging with the robot's drivers. The bus is internally treated as a shared resource protected by software semaphores to avoid the possibility of different threads attempting to access the bus simultaneously.

3.4 ASIBOT Task Server

A Task Server has been developed, which allows tasks to be implemented as classes that inherit from a same Task base class and be instanced through remote procedure port calls. The specific tasks used for this application have been those implemented in the TaskGrabCan and the TaskDrink classes. TaskGrabCan is capable of making the robot to grab a can in the environment, taking the current position of the robot and the located object into account. TaskDrink moves a grabbed object near the user's lips to allow the user to drink.

3.5 Color Segmentation

An Open Source library that wraps around OpenCV has been released, namely Travis. Travis (which stands for Tracking and Vision library) is a small library for computer vision in robots [19]. The parametric image segmentation Travis provides has been used within the Color Segmentation module, which processes an incoming stream of images within a periodical thread. The Color Segmentation module is used as a bypass module that outputs to the Augmented Reality 3D Vision Glasses display. Its input may be switched between the 3D Vision Glasses' cameras or their simulated environment analogous, providing object segmentation information to the user in either of the two cases. Figure 5 depicts the Color Segmentation of the red can of the simulated environment. The visual output includes the object contour (pink contour), centroid (blue dot), and rotated bounding box (blue rectangle).

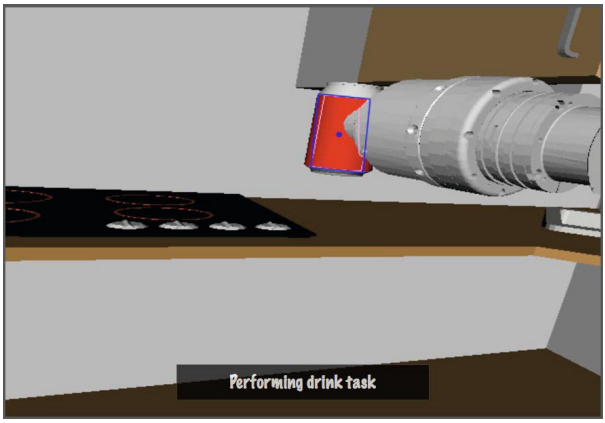


Fig. 5. User's view of the Color Segmentation module bypassing the simulator image

3.6 Object Localization

The Object Localization module uses data from a Microsoft Kinect sensor placed in the environment. The color segmentation of the object is based on the Travis library, described in the previous subsection. It obtains object centroids, that are then matched with the depth image provided by the official YARP wrapper of the Kinect OpenNI drivers. The real world coordinates of the object are computed through homogeneous transformation matrices provided by the developed ASIBOT TinyMath library.

3.7 Automatic Speech Recognition

The Automatic Speech Recognition module uses the CMU Pocketsphinx and GStreamer packages in a Python environment. Natural language parsing is not

used, and instead a small vocabulary corpus of specific words is compiled for our applications.

3.8 Text-to-Speech

The Text-to-Speech module used is the iSpeak module that can be directly found within the iCub Software repository. It is actually a wrapper around the Festival and eSpeak packages for speech synthesis.

4 Experiments

The experiments where performed with ten healthy robotics-related people using the system in the ASIBOT kitchen environment[1] in conjunction with its simulated representation. The range of ages of the participants was between 25 and 35 years old. After a brief description of the system and its components (approximately 5 minutes), they were allowed to use the robotic system freely.

The following is an actual transcription of a dialogue performed between one of the human users and the ASIBOT assistive robot developed system.

ASIBOT→ I'm ready. Ask me to show tasks, or to perform an action.
User→ *Show tasks.*
ASIBOT→ I understood SHOW TASKS. Is that correct?
User→ *Yes.*
ASIBOT→ How do you want to see the tasks? Text, speech, icons?
User→ *Text.*
ASIBOT→ I understood TEXT. Is that correct?
User→ *Yes.*
-Text with task names appear in the interface-
ASIBOT→ Okay, perfect. I'm showing the text. I'm ready. Ask me to show tasks, or to perform an action.
User→ *Grab.*
ASIBOT→ Performing grab task.
-The robot performs the task of grabbing a red can object-
ASIBOT→ Finished grab task.

In order to measure the satisfaction of the users with the robotic system, we provided them with SUS tests (System Usability Scale). As a summary of the results:

- The average punctuation was 84 ± 10.88 over 100 (where 100 is the best score). This is higher than the 70.5 ± 9.5 score achieved using a web-browsable multi-modal system, recently published by the authors [20].

[1] It is important to notice, however, that the system is ubiquitous and has been designed to work in any part of a living environment.

- The best results were obtained in the items "I though the system was easy to use" and "I would imagine that most people would learn to use this system very quickly", both with an average of 4.7 ± 0.48 (where 5 is the best score).
- On the other hand, the worst results were obtained in: "I think that I would need the support of a technical person to be able to use this system", with an average of 2.3 ± 1.16 (where 5 is the best score).

5 Conclusions

In this paper, the creation of an Assistive Robot Multi-Modal Interface system based on augmented 3D vision and interactive dialogue has been proposed. To this end, different systems based on augmented reality and interactive dialogue mechanisms have been studied. A complete system has been developed under our Open System Architecture, which has been tested with users in the form of a closed user-ready system.

As a result, we have understood the capabilities and functionalities of our system. The limitations of our current developments detected by the users in the tests and our own subjective and objective appreciations will lead to future developments with increased accessibility, usability, and end user satisfaction.

Acknowledgments. The research leading to these results has received funding from the ARCADIA project DPI2010-21047-C02-01, funded by CICYT project grant on behalf of Spanish Ministry of Economy and Competitiveness, and from the RoboCity2030-II-CM project (S2009/DPI-1559), funded by Programas de Actividades I+D en la Comunidad de Madrid and co-funded by Structural Funds of the EU.

References

1. Huete, A.J., Victores, J.G., Martinez, S., Giménez, A., Balaguer, C.: Personal autonomy rehabilitation in home environments by a portable assistive robot. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42(4), 561–570 (2012)
2. Azuma, R.T.: A survey of augmented reality. Presence 6(4), 355–385 (1997)
3. Ronald, T.: Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. IEEE Computer Graphics and Applications 21(6), 34–47 (2001)
4. Van Krevelen, D.W.F., Poelman, R.: A survey of augmented reality technologies, applications and limitations. International Journal of Virtual Reality 9(2), 1 (2010)
5. Green, S.A., Billinghurst, M., Chen, X., Chase, G.: Human-robot collaboration: A literature review and augmented reality approach in design. International Journal of Advanced Robotic Systems 5(1), 1–18 (2008)
6. Green, S.A., Chase, G., Chen, X., Billinghurst, M.: Evaluating the augmented reality human-robot collaboration system. International Journal of Intelligent Systems Technologies and Applications 8(1), 130–143 (2010)

7. Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., Ivkovic, M.: Augmented reality technologies, systems and applications. Multimedia Tools and Applications 51(1), 341–377 (2011)
8. Milgram, P., Yin, S., Grodski, J.J.: An augmented reality based teleoperation interface for unstructured environments. In: ANS 7th Meeting on Robotics and Remote Systems, Augusta, pp. 101–123 (1997)
9. Fang, H.C., Ong, S.K., Nee, A.Y.C.: Interactive robot trajectory planning and simulation using augmented reality. Robotics and Computer-Integrated Manufacturing 28(2), 227–237 (2012)
10. Tang, S.-L., Kwoh, C.-K., Teo, M.-Y., Sing, N.W., Ling, K.-V.: Augmented reality systems for medical applications. IEEE Engineering in Medicine and Biology Magazine 17(3), 49–58 (1998)
11. Su, L.-M., Vagvolgyi, B.P., Agarwal, R., Reiley, C.E., Taylor, R.H., Hager, G.D.: Augmented reality during robot-assisted laparoscopic partial nephrectomy: toward real-time 3d-ct to stereoscopic video registration. Urology 73(4), 896–900 (2009)
12. Luo, X., Kline, T., Fischer, H.C., Stubblefield, K.A., Kenyon, R.V., Kamper, D.G.: Integration of augmented reality and assistive devices for post-stroke hand opening rehabilitation. In: 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS 2005, pp. 6855–6858. IEEE (2005)
13. Braga, R.A., Petry, M., Moreira, A.P., Reis, L.P.: A development platform for intelligent wheelchairs for disabled people. In: International Conference on Informatics in Control, Automation and Robotics, pp. 115–121 (2008)
14. Alonso-Martin, F., Salichs, M.A.: Integration of a voice recognition system in a social robot. Cybernetics and Systems: An International Journal 42(4), 215–245 (2011)
15. Foster, M.E., By, T., Rickert, M., Knoll, A.: Human-robot dialogue for joint construction tasks. In: Proceedings of the 8th International Conference on Multimodal Interfaces, pp. 68–71. ACM (2006)
16. Seabra Lopes, L., Teixeira, A.: Human-robot interaction through spoken language dialogue. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2000, pp. 528–534. IEEE (2000)
17. Becker, E., Le, Z., Park, K., Lin, Y., Makedon, F.: Event-based experiments in an assistive environment using wireless sensor networks and voice recognition. In: Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments, p. 17. ACM (2009)
18. Fitzpatrick, P., Metta, G., Natale, L.: Towards long-lived robot genes. Robotics and Autonomous Systems 56(1), 29–45 (2008)
19. Morante, S.: Interfaz y librera para visión artificial, navegación y seguimiento en robótica. Undergraduate's honors thesis, Universidad Carlos III de Madrid, Dpto. Ing. Sistemas y Automática (June 2012)
20. Victores, J.G., Morante, S., Jardón, A., Balaguer, C.: Give me the red can: Assistive robot task creation through multi-modal interaction. In: V Congreso Internacional de Diseño, Redes de Investigación y Tecnología para todos, DRT4ALL (September 2013)

Development of a Lower-Limb Active Orthosis and a Walker for Gait Assistance

D. Sanz-Merodio, M. Cestari, J.C. Arevalo, X. Carrillo, and E. Garcia

Centre for Automation and Robotics, CSIC-UPM,
La Poveda, 28500 Madrid, Spain
{daniel.sanz,juan.arevalo,manuel.cestari,
xavier.carrillo,elena.garcia}@csic.es
http://www.car.upm-csic.es/fsr/egarcia/ATLAS.html

Abstract. Lower-limb exoskeletons and powered orthoses in gait assistance applications for patients with locomotive disorders possess the potential to significantly affect society in the near future. This paper presents the primary features of a lower-limb exoskeleton to enable paralysed children to walk. Because these patients are unable to move their limbs, the device generates their basic motions in everyday life, e.g., standing up, sitting down, and stable ambulation. A walker provides stability in the lateral plane, while the active orthosis provides stability in the sagittal plane while walking. The walker has been devised with a two degree of freedom mechanism to allow the user to sit down and stand up in a stable and comfortable way without the movement of the walker itself. The gait of the orthosis parameters such as step height, body height or step length are modified online, based on an impedance control approach, providing a safe and smooth gait pattern. Two shoe insole pressure measurement systems provide ground reaction force and center of pressure to adapt these gait parameters online. An adjustable compliance actuator has been designed and incorporated to the knee joint of the active orthosis. This Actuator with Adjustable Rigidity and Embedded Sensor (ARES) fulfills the demanding characteristics required in an active orthosis's joint, namely, intrinsic compliance to allow human-machine interaction, high power-to-weight ratio, high peak torque, small size and low weight. Exploiting the characteristics of ARES actuator a control scheme has been designed and implemented to achieved a reduction in the energy expenditure while keeping compliant to accommodate unexpected disturbances. The final ATLAS exoskeleton has been successfully tested in a healthy user, in a quadriplegic child, and in a patient with neuromuscular disease.

Keywords: Powered active orthosis, Gait trajectory generation, Exoskeleton, Walker, Energy consumption.

1 Introduction

Active orthoses are devised to increase the locomotive ability of an individual who is suffering from leg pathology. In the field of exoskeleton devices a classification

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_16,

can be made depending on the disease and potential for improvement of the user abilities:

- *Rehabilitation exoskeletons.* In a case of patients with neurological injury or with chronic incomplete spinal cord injury (SCI), a gait rehabilitation exoskeleton can help the user to relearn and recover the motion of their limbs. These rehabilitation exoskeletons [1] reproduce the motion of the user limbs in a body-weight supported treadmill [2]. This motion helps in the formation of user new neural path-ways to relearn to walk [3].
- *Exoskeletons for partial assistance.* In the case that user has lost strength in some of his or her limbs, such as an aged user, this portable exoskeleton device can detect the user intention by the use of electro-miographical signal (EMG) and augment it [4], like in the hybrid assistive limb (HAL) [5, 6].
- *Exoskeletons for full support assistance.* If the patient suffers from a complete spinal cord injury (SCI) which has resulted in paraplegia or quadriplegia, an assistive exoskeleton that helps to walk a person with lower-limb pathology can replace the function of a wheelchair. These types of exoskeletons are called active or powered orthoses. An active orthosis besides offering much more mobility than a wheelchair, significantly improves the circulation, bone density and excretory system of user. The use of active orthoses, either rehabilitation or assistive, favors the exercise of the user's muscles, preventing rapid deterioration associated with wheelchairs and providing the user with psychological benefits of self-esteem.

However, it is not straightforward to exchange a wheelchair for an exoskeleton. An individual who has spent 20 years in a wheelchair loses articular range and experiences excessive rigidity (espasticity), hip dislocation and scoliosis, which impede normal locomotion even when assisted by an exoskeleton. Therefore, the design of an active orthosis for children is urgent because the sooner that movement can be provided to paralysed limbs, the better they evolve and resist functional degeneration.

Currently, there are commercially available autonomous lower-limb exoskeletons. However, the development of fully functional active orthoses for paralysed patients requires further research. Specifically, the development of active orthoses for completely paralysed patients, such as quadriplegics, has not been considered because of the added difficulty of stability control and acquiring the user's intention. In the development of ATLAS project these objectives are tackled through the development of a walker and an active orthosis, providing the user the ability to stand, sit and walk. The walker has been designed to provide stability in lateral and frontal planes. This is a height adjustable walker, with smart features to assist the user allowing them to sit down and stand up while still attached to the walker. It is intended for the development of daily activities and ensuring adequate stability for users to submit greater mobility limitations.

A full lower-limb active orthosis has been developed in which the locomotion cycle is configurable based on a new parameterised trajectory generation approach. Additionally, the exoskeleton adapts reactively to unexpected events.

For the robot to be robust against perturbations in the sagittal plane and to control dynamic stability, an impedance control approach is proposed.

In a lower-limb exoskeleton, one of the most important features is safety in human-robot interaction. Thus, changing the stiffness of the joints to adapt to the movements of the user is a need. For this reason, a specific actuator has been designed and developed for the knee joint of the ATLAS active orthosis with adjustable compliance. The compliant actuator extends along the leg so that it does not protrude. This actuator not only provides elasticity to the joint, it also acts as a joint torque sensor. Its design is based on a stiff conventional actuator in which adjustable compliance has been included by the pivot displacement technique [7]. Using this variable stiffness actuator, a control scheme is applied to reduce energy consumption by exploiting the natural dynamics of the limbs and the energy stored in the springs.

In this paper, Section 3 presents the details of the walker design. Section 2 describes the full active orthosis concept, featuring the hardware specifications for the robotic structure, the actuators and the sensorial system. Subsection 3.1 details the control of locomotion and Subsection 3.2 describes the safe gait pattern that is followed by the joints. Section 4 introduces the adjustable-stiffness actuator (ARES) in the knee joint of the orthosis. Making use of this actuator, an energy-efficient control scheme of locomotion is described in Section 5. Finally Section 6 presents the primary conclusions.

2 ATLAS Orthosis Concept and Design

In orthotic terminology ATLAS orthosis (see Fig. 1) can be considered as an active THKAFO (Trunk-Hip-Knee-Ankle-Foot Orthosis). ATLAS is intended to support a 25-kg girl affected by leg impairment and help her to walk at a moderate speed ($\leq 1m/s$). It is a 6 degree of freedom (DOF) mechanism, having 3 DOF per leg; hip, knee and ankle to allow the user to move in the sagittal plane. The flexion and extension motion of the hip, knee and ankle joints is driven by electrical brushless Maxon motors in combination with harmonic drive units. The setup provides repeated peak torque up to 57 Nm, and average torque of 32 Nm at speed higher than 20 rpm for each actuated joint. The orthosis is adjustable for height, allowing a range of user heights from 140 cm to 165 cm and a maximum weight of 50 kg. The sensorial system is composed of goniometers at hip, knee and ankle to measure user joint angles and an in-shoe plantar pressure measurement system at each foot (see Fig. 2) that provides the total ground reaction force (GRF) and the center of pressure (COP). This insoles are based on a matrix of 85 conductive pressure sensors distributed along the sole area measuring a pressure range of 15-600 kPa with a resolution of 2.5 kPa. This amount of sensors allow very precise location of the center of pressure and a correct measurement of the ground reaction force.

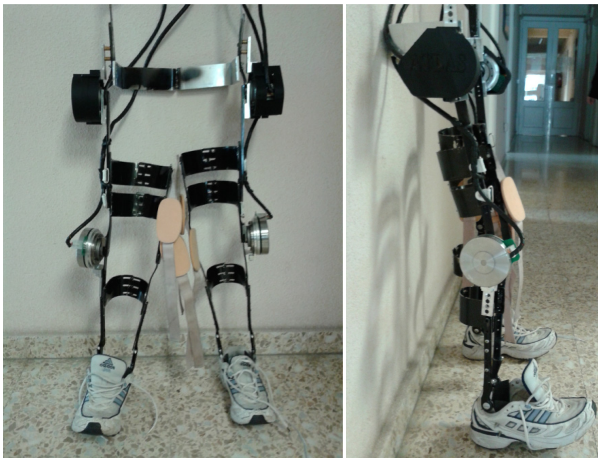
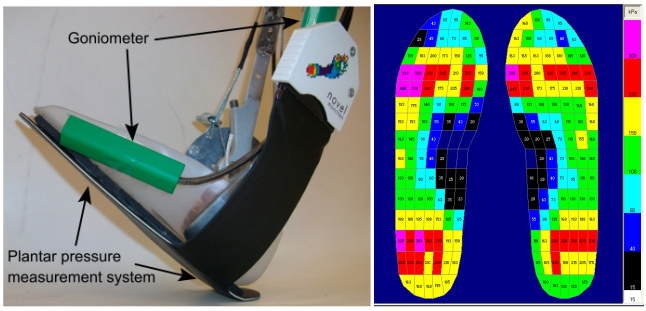


Fig. 1. Front and lateral view of ATLAS exoskeleton



(a) (b)

Fig. 2. Goniometer and flexible in-shoe plantar pressure measurement system: (a) Insole and 2D goniometer attached; (b) Pressure analysis

3 Walker Concept and Design

The walkers used for rehabilitative function are not designed for its incorporation and use in an user's everyday activities. Those devices which incorporate mechanisms to enable sitting, standing or exercise muscle toning are characterized by large structures or rigid configuration systems that do not allow users' requirements in a regular social environment. The patent document [8] presents a practical device that provides the ability to sit and stand, but this system requires a movement of the wheels of the walker on the horizontal plane to compensate for the natural movement when sitting. Similarly, in case of locking the wheels of the device, this restriction of movement causes an unnatural movement of the user's body. For these reasons, we have designed an adjustable height walker with an integrated intelligent control system that provides greater security. The kinematic chain which is derived from the connections to the user,

provides two degrees of freedom to the sitting mode sufficient to fit the positions described in the sitting and standing user. Figure 3 shows how the device adapts to the user's movements while sitting. Figure 4 presents snapshots of a user in a test while sitting down, see how the walker allow to follow the body trajectory without displacement.

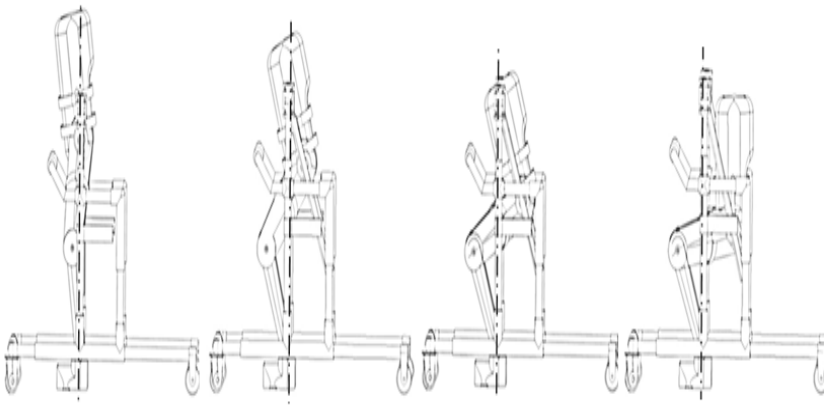


Fig. 3. The walker adapts to the user's movements while sitting

Fig. 4. Snapshots of a user in a test while sitting down

3.1 Compliance Controller

The control of an active orthosis for quadriplegic users has to be thought of in a similar way as an autonomous biped robot. Therefore we have implemented a control scheme based on a compliance controller, adapted from [9], which basically uses parameterized joint trajectories as equilibrium points while small forces are allowed to separate slightly from those equilibrium points in order to allow for some compliance. The control scheme is shown in Fig. 5. The desired foot forces are calculated to support the user weight and maintain the center of

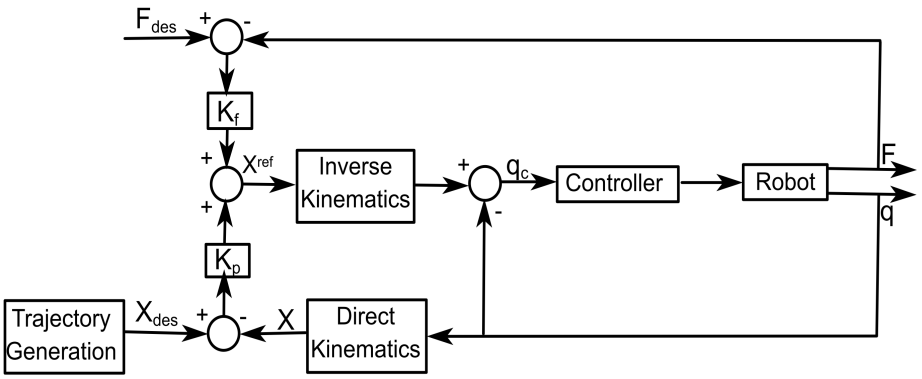


Fig. 5. Block diagram of the active-compliance controller

pressure within the desired path. Actual cartesian force vector is calculated from the values obtained with the insoles. Therefore, the active compliance equation for each leg has the form:

$$\mathbf{F_{des}} - \mathbf{F} = -\mathbf{K_p}(\mathbf{X_{des}} - \mathbf{X})\mathbf{K_f}^{-1} + \mathbf{X^{ref}}\mathbf{K_f}^{-1} \tag{1}$$

where $\mathbf{X}$ is the vector of actual cartesian position of the foot, $\mathbf{X_{des}}$ is the vector of reference positions obtained from parameterized trajectories, $\mathbf{F}$ is actual force sensed from insoles in stance and $\mathbf{F_{des}}$ is the vector of desired Cartesian foot forces, obtained from the force-distribution algorithm [10]. $\mathbf{K_p}$ and $\mathbf{K_f}$ are diagonal matrices of gains. $\mathbf{X^{ref}}$ are cartesian foot position references as inputs to joint controllers after inverse kinematics transform.

These final position reference, $\mathbf{X^{ref}}$ and its derivative are traced by the controller through a PD current control scheme.

$$\mathbf{I} = \mathbf{K_{ip}}(\mathbf{X} - \mathbf{X^{ref}}) + \mathbf{K_{id}}(\dot{\mathbf{X}} - \dot{\mathbf{X}}^{\mathbf{ref}}) \tag{2}$$

where $\mathbf{X}$ is again the actual position, $\dot{\mathbf{X}}$ is the actual velocity and $\mathbf{I}$ is the current vector.

3.2 Gait Pattern Generation

Traditional orthoses base their movement on the tracking of clinical gait analyses (CGAs) patterns, which are de-normalized and adjusted to the user. This gait trajectory ensures an ergonomic and natural gait. However, they cannot adapt to the environment characteristics. Biped robots, on the other side, use adaptable gaits. To go further, the gait parameters in an active orthosis should be modifiable like biped robots do.

ATLAS project uses CGA based gait trajectories. However, the parameters of the gait are changed as a function of the ground reaction force and the centre

of pressure sensed by the insoles. ATLAS orthosis can modify the ground clearance, the body height and the step length. Fig. 6 shows how the foot trajectory changes when these parameters change. These new reference trajectories have to be defined without abrupt movement, this is, the final path that the foot must follow must be differentiable.

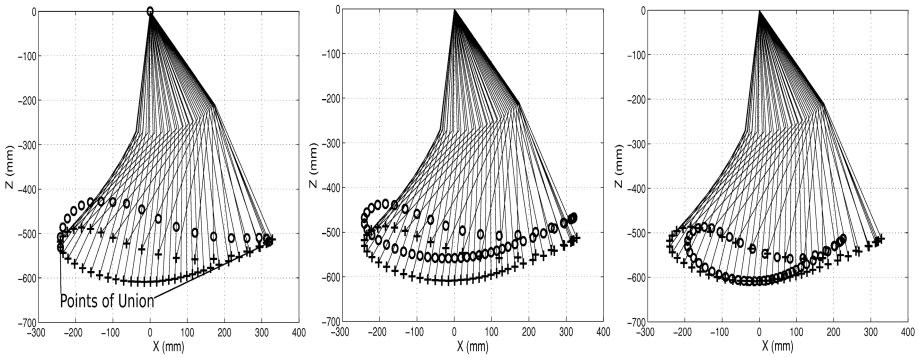


Fig. 6. Original foot path obtained by CGA *(marker +)* and modified *(marker o)* to (a) provide more ground clearance (b) reduce body height (c) reduce step length

The stability of the proposed orthosis with compliance controller has been successfully tested with a healthy user, with a quadriplegic child and with a patient with neuromuscular disease using the walker described in Section 3. Figure 7 shows snapshots of a user in a test while walking with the walker and the orthosis. Videos of the tests can be watched online at (`http://www.car.upm-csic.es/fsr/egarcia/ATLAS.html`).

Fig. 7. Snapshots of a user in a test while walking

4 Adjustable Compliant Knee Joint

The actuation of the joints in an exoskeleton requires peaks of high torque, about 50Nm at the knee [11], while allows compliance to the human wearer. Pneumatic actuators with antagonist configuration offer high power output and are able to change stiffness in a biomimetic way. Some rehabilitation devices have been developed with pneumatic actuators [12], but active orthoses must be portable and nowadays pneumatic and hydraulic actuator needs equipment that is too large and weighted to be feasible. So an efficient electric actuator is needed to drive the knee joint. Also it has to be able to change its stiffness fast enough to accomplish a steady gait at approx. 1 m/s. This kind of actuators must be small enough for not to bother the user and have an aesthetic appearance. Also they have to be as light as possible, for not to increase the inertia moment of the limbs and for being easy to handle, put on and off. For this reason some design prototypes that meet these specifications are being proposed. Some new electric actuators with variable compliance have been designed, as AMASC designed at Carnegie Mellon University [13], MACCEPA at Vrije Universiteit Brussels [14], VS-joint and QA-joint mechanisms at the German Aerospace Centre (DLR) [15], and AwAS at the Italian Institute of Technology (IIT) [16], but most of these actuators have been designed for robotic arms, and none of them has been implemented and demonstrated in a portable and wearable active orthosis.

4.1 Working Principle of the Variable Stiffness Actuator

In the first prototype of ATLAS active orthosis, the flexion and extension motion of the hip and the knee joints was driven by electrical brushless flat Maxon motors in combination with Harmonic Drive units. The resulting motor-gearbox set provided enough torque to move the user and showed a large power-to-weight ratio but was too much rigid for a human-machine interaction task. Therefore, based on this configuration for being flat and powerful enough, we have designed our actuator prototype ARES (Actuator with Adjustable Rigidity and Embedded Sensor) that incorporates variable compliance and provides an accurate torque measurement. Figure 8 represents a CAD scheme of our design and the real prototype. The stiffness adjustment is based on the pivot displacement principle in which the actuator position and stiffness control are decoupled achieving more energy efficiency [7]. The torque provided by the motor-gearbox set is transmitted by a slotted bar and a pivot to an elastic device that finally moves the joint. These elastics devices are composed by parallel springs that provide intrinsic compliance to the joint.

Through the slotted bar, the pivot allows vertical displacement driven by a motor which provides stiffness variation. The higher the arm between the center of the motor and the pivot point, the greater rigidity.

A linear encoder has been attached to the pivot point to measure the spring compression, this provides a measure of the torque applied by the relationship:

$$\tau = \frac{2\Delta x K_{equiv} L_{arm}}{cos(\phi)} \tag{3}$$

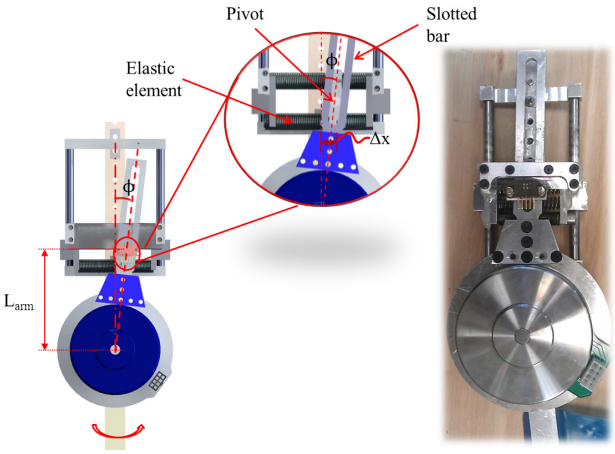


Fig. 8. CAD design of the variable stiffness actuator, principle of operation and real prototype

where, Δx corresponds to the elastic elements compression, sensed by the encoder, K_{equiv} is the equivalent rigidity of the spring set, angle ϕ is the deflection between the actuator and the joint, and L_{arm} is the length of arm.

Thus, this actuator allows the modification of the stiffness while provides information about the torque applied, allowing compliant force control. Table 1 presents the main specifications of the variable impedance actuator with embedded sensor (ARES).

Table 1. ARES variable impedance actuator general specifications

Compliant joint properties	
Peak torque	Up to 76 Nm
Max deflection	±8°
Stiffness adjusting time	0.6 sec
Weight	900 gr
Length	235 mm
Width	50 mm
Power	90W

5 Energy-Efficient Control Scheme of Locomotion

In our previous work [17], we proposed some methods to increase energy efficiency, based on the study of high-efficiency passive biped robots. In the design of an energy-efficient control scheme we have focused on three points, due to the characteristics of the actuator:

1. Utilize the elastic elements of the actuator to store and release energy.
2. Exploit passive dynamics during swing.
3. Reduce costs at heel strike collision exploiting the intrinsic impedance.

This new actuator provides intrinsic impedance that the rigid one doesn't, then, to take full advantage of the strategies proposed and achieve a reduction in energy consumption, two parallel strategies are proposed:

a. Modification of the gait pattern to follow based on a force control approach.
b. Design of a state machine controller with different values of stiffness at each state.

The motion control of active orthoses has traditionally been based on the rigid tracking of clinical gait analysis (CGA) reference patterns, typically resulting in high power consumption. The position pattern obtained from the CGA, incorporates the intrinsic compliance that the human joints have. Therefore, instead of following accurately the CGA pattern, we have exploited the inherent compliance of the ARES joint and followed a simplified commanded joint pattern, allowing the joint to adapt to the ground while walking. Figure 9 shows the CGA position pattern followed by the knee and the modified pattern.

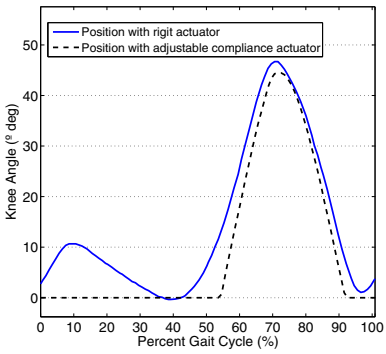


Fig. 9. CGA pattern utilized with the rigid actuator and the pattern implemented in the variable compliant actuator ARES

During locomotion, a state machine switches between stiffness values for the knee joint. The state machine, see Figure 10, that we have incorporated at the knee joint is based on the study that we have presented in our previous work [17]. Four discrete knee stiffness states are required:

State 1. After toe-off, at the beginning of swing phase, a large stiffness is needed to let the motor lift up the shank.
State 2. When it has reached the highest foot height during swing, zero stiffness allows exploiting the intrinsic dynamics of the limb for swinging it forward and the potential energy stored when the motor changes its direction.

State 3. At the end of swing, the stiffness is increased to prepare for the collision at heel strike but kept low enough to allow the actuator springs to absorb the impact energy, enabling an adequate load response to begin the stance phase.

State 4. At the support phase the motor of the joint is kept blocked and a high stiffness is enforced to support the user weight but still providing some compliance.

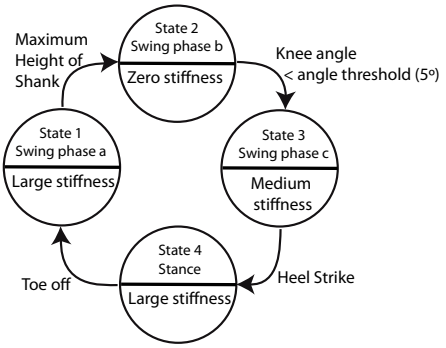


Fig. 10. State machine controller. In each state a range of stiffness and the transition condition are shown.

It is important to remark that due to the characteristics of the variable compliance joint design, the force required to change the stiffness of the joint increases as the torque does. Therefore, to reduce the energy consumed by the motor which changes the stiffness, every modification in stiffness is performed when a reduced torque is being done.

5.1 ARES Experimental Results

The actuator prototype has been incorporated in the knee joint of our ATLAS active orthosis, see Figure 11. The state machine scheme proposed in Section 5 has been implemented but with a variation, due to the fact that with our actuator we cannot achieve zero stiffness. As zero impedance is not achieved, it is necessary to use a control scheme that reduces energy expenditure. Zero torque control reduces the impedance of the joint via software control.

Figure 12 (a) presents the commanded position to the actuator and the real joint position achieved due to the compliance, controlled with the state machine. Notice that even without having set a path that follows the pattern of hump in the heel strike, it is observed that the load response follows that pattern due to the low impedance implemented that allows the knee to adapt to the ground during the support phase. A comparison with a CGA knee trajectory is shown. The controlled knee stiffness reproduces the natural knee behavior, without commanding a stiff CGA pattern.

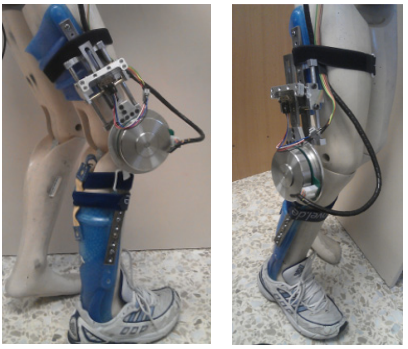
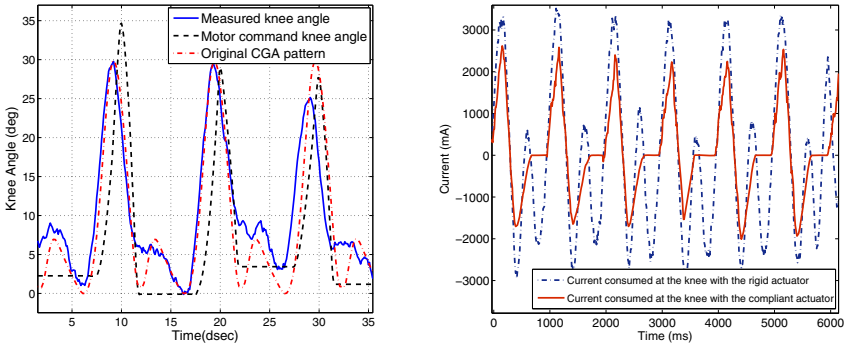


Fig. 11. Implementation of the adjustable compliant actuator



(a) (b)

Fig. 12. (a) CGA pattern utilized with the rigid actuator and the pattern implemented in the variable compliant actuator. (b) Commanded position and the real position reached with the compliant actuator. (c) Current consumption with the rigid actuator and with the variable stiffness actuator.

A comparison of the current consumption in a steady gait with a CGA pattern and rigid actuator and with the new pattern and the compliant actuator is shown in Figure 12 (b). Notice the energy expenditure reduction in each current peak, which corresponds to changes of direction of the trajectory in the joint. This is because the springs accumulate and release energy at each change of direction. During the support phase, no energy is required to block the knee due to the large stiffness, the high reduction ratio in the motor-gear kit and the position pattern proposed. Even so, in Figure 12 (a) it is shown how the knee allows adaptation thanks to the inherent compliance of the joint.

6 Conclusions

This paper presents results on the development of a full lower-limb active orthosis and a walker for a paralyzed child.

A walker provides stability in the frontal and lateral planes while walking, in addiction, it allows the user to sit down and stand up keeping the walker fix to the ground and adapting to the pattern described by the point of attachment for these actions characteristics of the user.

The ATLAS active orthosis concept consists of a lightweight, strong 6-DOF mechanical structure driven by DC motors at the hip and the knee and a passive synergic ankle. The user's joint angles, foot plantar pressure and trunk inclination are measured to provide motion control, which is based on a compliance controller that follows parameterised trajectory data. A parameterised trajectory make the orthosis motion adaptable and provide safety and a more natural gait while the compliance controller increases posture performance.

The 70% of potential users of an active orthosis suffer spasmodic movements, making absolutely necessary intrinsic compliance in their joints. Besides, the active orthosis actuators must occupy little space and be lightweight to be comfortable, aesthetically pleasing and portable. Therefore, our prototype actuator -ARES- features the demanding characteristics that an active orthosis requires, namely, intrinsic compliance to allow human-machine interaction, small size and low weight. This variable stiffness actuator has been implemented in the knee joint of the lower-limb exoskeleton ATLAS.

For the command of the actuator motion in the lower-limb active orthosis a new pattern has been defined and a state machine that switches between different stiffness has been defined to take advantage of the characteristics of the actuator and the dynamics of the legs. Although we have implemented a simpler pattern than that obtained from CGA, the response of the knee joint resembles the CGA data, due to the intrinsic compliance of the actuator, imitating the behavior of a biological joint. Compared to the rigid actuator following a pattern of CGA, the compliant actuator achieves a reduction in energy consumption of 39%. It is observed a reduction in the peak of the current when the direction of the joint changes. An important reduction is achieved keeping the knee motor locked in the support phase, while providing compliance enough to accommodate the body weight.

Experiments have been successfully accomplished in a healthy user, in a quadriplegic child and in a patient with neuromuscular disease.

Acknowledgments. This work has been partially funded by the Spanish National Plan for Research, Development and Innovation through grant DPI2010-18702.

References

1. Mayr, A., Kofler, M., Quirbach, E., Matzak, H., Frhlich, K., Saltuari, L.: Prospective, blinded, randomized crossover study of gait rehabilitation in stroke patients using the lokomat gait orthosis. Neurorehabilitation and Neural Repair 21(4), 307–314 (2007), http://nnr.sagepub.com/content/21/4/307.short
2. Vallery, H., van Asseldonk, E.H.F., Buss, M., van der Kooij, H.: Reference trajectory generation for rehabilitation robots: complementary limb motion estimation. IEEE Transactions on Neural Systems and Rehabilitation Engineering 17(1), 23–30 (2009), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4668434
3. PhD, D.P.F., Msme, G.S.S., Mpt, A.R.D.: Powered lower limb orthoses for gait rehabilitation. Topics in Spinal Cord Injury Rehabilitation 11(2), 34–49 (2005), http://dx.doi.org/10.1310/6GL4-UM7X-519H-9JYD
4. Fleischer, C.: Application of EMG signals for controlling exoskeleton robots. Biomed Tech 51, 314–319 (2006)
5. Suzuki, K., Mito, G., Kawamoto, H., Hasegawa, Y., Sankai, Y.: Intention-based walking support for paraplegia patients with robot suit HAL. Advanced Robotics 21(12), 1441–1469 (2007), http://www.tandfonline.com/doi/abs/10.1163/156855307781746061
6. Lee, S., Sankai, Y.: Power assist control for walking aid with HAL-3 based on EMG and impedance adjustment around knee joint. In: Intl. Conference on Intelligent Robots and Systems (2002)
7. Visser, L.C., Carloni, R., Stramigioli, S.: Variable stiffness actuators: A port-based analysis and a comparison of energy efficiency. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 3279–3284 (2010), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509127
8. Craig, W.: Walker with lifting arms (August 2005), http://global.soopat.com/Patent/Patent/US7363931B2
9. Garcia, E., Gonzalez de Santos, P.: On the improvement of walking performance in natural environments by a compliant adaptive gait. IEEE Transactions on Robotics 22(6), 1240–1253 (2006)
10. Jiang, W., Liu, A., Howard, D.: Optimization of legged robot locomotion by control of foot-force distribution. Transactions of the Institute of Measurement and Control 26(4), 311–323 (2004)
11. Dollar, A.M., Herr, H.: Lower extremity exoskeletons and active orthoses: Challenges and state-of-the-art. IEEE Transactions on Robotics 24(1), 144–158 (2008)
12. Ferris, D.P., Sawicki, G.S., Domingo, A.R.: Powered lower limb orthoses for gait rehabilitation. Topics in spinal cord injury rehabilitation 11(2), 34 (2005), http://thomasland.metapress.com/index/6gl4um7x519h9jyd.pdf
13. Hurst, J.W., Chestnutt, J.E., Rizzi, A.A.: The actuator with mechanically adjustable series compliance. IEEE Transactions on Robotics 99, 1–10 (2010)
14. Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B., Lefeber, D.: MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. Robotics and Autonomous Systems 55(10), 761–768 (2007)
15. Wolf, S., Hirzinger, G.: A new variable stiffness design: Matching requirements of the next robot generation. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 1741–1746 (2008)

16. Jafari, A., Tsagarakis, N.G., Vanderborght, B., Caldwell, D.G.: A novel actuator with adjustable stiffness (AwAS). In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4201–4206 (2010)
17. Sanz-Merodio, D., Cestari, M., Carlos, J., Garcia, E.: Control motion approach of a lower limb orthosis to reduce energy consumption. International Journal of Advanced Robotic Systems, 1 (2012), http://www.intechopen.com/journals/international_journal_of_advanced_robotic_systems/control-motion-approach-of-a-lower-limb-orthosis-to-reduce-energy-consumption

Part IV
Surgical Robotics

On Genetic Algorithms Optimization for Heart Motion Compensation

Angelica I. Aviles[1] and Alicia Casals[1,2]

[1] The Polytechnic University of Catalonia-BarcelonaTech, Intelligent Robotics and Systems Group, Jordi Girona 1-3 K2M Building, 08034, Barcelona, Spain
{angelica.ivone.aviles,alicia.casals}@upc.edu
[2] Institute for Bioengineering of Catalonia, Robotics Lab, Baldiri Reixac 4-6, 08028 Barcelona, Spain

Abstract. Heart motion compensation is a challenging problem within medical robotics and it is still considered an open research area due to the lack of robustness. As it can be formulated as an energy minimization problem, an optimization technique is needed. The selection of an adequate method has a significant impact over the global solution. For this reason, a new methodology is presented here for solving heart motion compensation in which the central topic is oriented to increase robustness with the goal of achieving a balance between efficiency and efficacy. Particularly, genetic algorithms are used as optimization technique since they can be adapted to any real application, complex and oriented to work in real-time problems.

Keywords: Genetic Algorithms, Deformation, Stochastic Optimization, Beating Heart Surgery, Robotic Assisted Surgery.

1 Introduction

Research in minimally invasive beating heart surgery has gained interest because of the advantages it presents over traditional cardiac procedures [1, 2]: shorter rehabilitation, quicker recovery, reduction of both risk for neurological injury and blood transfusions or even improve cosmetics etc. Therefore, this research line has become a key issue for improving the quality of human life because according to World Health Organization (WHO) cardiovascular diseases are the first cause of death throughout the world [3].

In spite of the benefits of beating heart surgery, two sources of disturbance affect this surgical procedure: heart motion and breathing. In consequence, heart stabilization becomes a challenge since the surgeon has to deal with a constant moving target. Even though small devices positioned over the heart based on vacuum pressure have been proposed in order to reduce motion in the area of interest, some residual motion remains [4]. That is, human tracking is reliable up to frequencies of 1Hz and the heart rate is about 1.18Hz ($\approx 70\ beats/min$) [5,6]. Therefore, the research goal is to compensate these disturbances to offer the surgeon the sensation of being operating on a static area. Current heart motion compensation relies on the estimation of movement using computer vision

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_17,

techniques. In this regard, the first proposal in this research line was presented by Nakamura [7] in which the introduction of heartbeat synchronization was established. However, the main drawback of this approach is the use of artificial markers which are impractical due to the complexity of fixing them on the heart surface during a real procedure. Subsequently, a solution based on natural markers was proposed in [8], which has given rise to realistic solutions.

Despite the fact that different solutions have been proposed such as those presented in [9–11] in which texture, shape-from-shading and TPS are the center topic, there is not yet a suitable solution working on real conditions due to the lack of robustness. The reason is the complex nature of the problem that is due to various factors: small workspace, variable lighting conditions, glossy area, deformable tissue and hardware limitation that is restricted to the use of the endoscope. Therefore, due to the characteristics of the problem it is necessary to carefully choose the optimization technique that offer the optimal results in a short time. In this regard, Genetic Algorithms (GAs) can deal with previous requirements and at the same time offer an elegant solution. Thus, GAs is a perfect match to this application.

This paper is organized as follows. In Section 2 the mathematical formulation of the problem is presented as well as the set of problems in which it is divided. In Section 3, the optimization based on genetic algorithms oriented to the cardiac context is explained while in Section 4 the experimental part is presented based on a realistic data set. Finally, in Section 5 some conclusions are presented as well as the future work.

2 Heart Motion Compensation

Heart motion compensation is based on the perception of movement of the heart and the generation of the same actions, as can be seen in the scheme presented in Fig. 1. The motion compensation's module is divided into three modules: i) deformation model, ii) the objective function, and iii) optimization technique. That is, the solution can be seen as the performance of the three modules acting in an iterative loop. In particular, the selection of the deformation model has to offer both valuable information and good computational time [13]. The selection of each module is crucial in the sense that global performance depends on them. In particular, the deformation model has to present characteristics such as: optimal mathematical properties, low computational cost and the ability of describing the displacement field even in complex deformations. The second module that allows managing the available information has to be easy to compute and to evaluate mathematically. Likewise, the third module that is related to choosing the best value from different alternatives has to deal with crucial factors such as: complex problems, large amount of data, flexibility, and a good response time.

2.1 Formulation of the Problem

From a mathematical point of view, heart motion compensation can be formulated as an energy minimization problem. That is, given two images the former

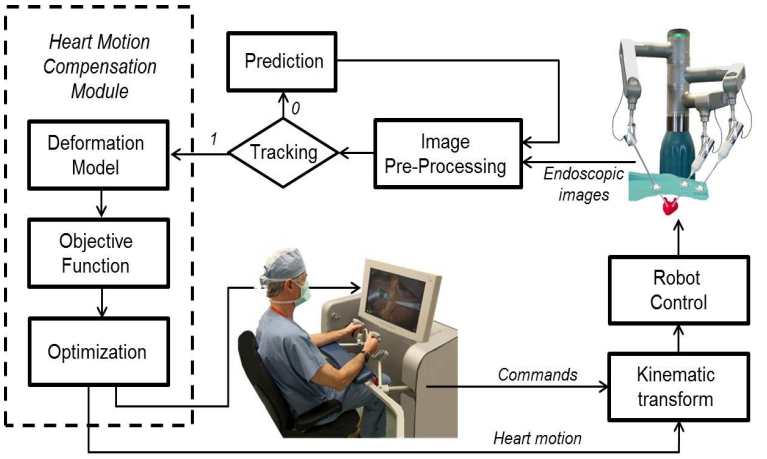


Fig. 1. Scheme that shows the integration of heart motion compensation to a robotic surgical system

represents the fixed or reference image $F : \Omega_F \subset \Re^d$ and the latter the acquired image (i.e. from laparoscope) $A : \Omega_A \subset \Re^d$, where the image domain is denoted by Ω. Then, the goal is to find the optimal transformation T for:

$$\arg\min_T E_t(T) = \arg\min_T D(F, T(A)) + \alpha R(T) \tag{1}$$

where $E_t(T)$ is the total energy to be minimized, D refers to the level of alignment between both images and R is related to the regularization term that help to create a well-defined displacement field at all locations. This term is bearing in order to fulfil the Hadamard's postulate [12], and $\alpha \in \Re^+$ gives a balance between both terms. In this case, the transformation, T, is represented by the changes occurred on the heart given by the displacement field u. Then, defining $\mathbf{x} \in \Re^d$ as a vector Eq. 1 can be reformulated as:

$$\arg\min_u E_t(u) = \arg\min_u D(F(\mathbf{x}), A(u(\mathbf{x}) + \mathbf{x})) + \alpha R(u(\mathbf{x})) \tag{2}$$

Thus, before applying the optimization technique, it is necessary to define how the information will be managed and analyze the best regularizer, R, to be use. In this sense, the traditional Sum of Squared Differences (SSD) is selected as dissimilarity measure, D, due to its characteristics such as simplicity and effectivity that made it a good option for this application. On the other hand, even if Tikhonov [14] is the most common regularization technique, it seems to be too strong in this problem. In the same way, other regularizers like the Mumford and Shah model [15] are not suitable for this application due to characteristics such as complex mathematical formulation or high computational cost that make them not suitable for real-time operation. A good alternative is the proposal pioneered in computer vision by Rudin et al. in [16] called Total Variation model

(TV). This approach has been selected due to seems to offer a good balance between the computational consumption and complexity.

According to previous, using SSD as dissimilarity measure, TV as regularization technique and considering images as continuous functions Eq. 2 can be reformulated by

$$\arg\min_u E_t(u) = \arg\min_u \int_\Omega (F(\mathbf{x}) - A(u(\mathbf{x}) + \mathbf{x}))^2 d\mathbf{x} + \alpha \int_\Omega \mid \nabla u(\mathbf{x}) \mid d\mathbf{x} \quad (3)$$

As practical effects, an approximation of all integrals by a discrete sum is done. Thus, the total energy, E_t, from Eq. 3 can be rewritten as

$$\arg\min_u \sum_{x\in(1\leq x\leq\Omega)} (F(\mathbf{x}) - A(u(\mathbf{x}) + \mathbf{x}))^2 + \alpha \sum_{x\in(1\leq x\leq\Omega)} |u_1(\mathbf{x}) - u(\mathbf{x})| \quad (4)$$

In this work, he displacement field $u(\mathbf{x})$ is modelled using an interpolation technique. In particular, it is described by b-splines. This work is oriented to 2-dimensional space, however, it is easily extended to n-dimensions. The main idea is to define a grid of lattice points in R^2 that allows representing the deformation under the influence of these points. In this regard, $u(\mathbf{x})$ is represented by the summation of the tensor products of univariante spline. Then, $u(\mathbf{x})$ is presented as

$$u(x,y) = \sum_{l=1}^{L} \sum_{m=1}^{M} B_l(\mu) B_m(v) C_{i+l,j+m} \quad (5)$$

where the size of the grid is given by $L \times M$ with uniform spacing δ, $i = [x/\delta(x)] - 1$, $m = [y/\delta(y)] - 1$, $\mu = x/\delta(x) - [x/\delta(x)]$, $v = y/\delta(y) - [y/\delta(y)]$.

3 Optimization Based on Genetic Algorithms

The selection of the optimization technique has a significant impact over the global solution of the problem because efficiency is a common factor to bear in mind in a system. Since heart motion compensation is a complex problem and its solution is oriented to work in real-time, traditional optimization techniques like gradient-based algorithms are not able to offer robust results [17]. A good alternative is the use of Evolutionary Optimization Algorithms (EOA) since they can be adapted to any real-world problem.

EOAs have been used for long time but it was not until recently that their potential has been explored in depth in cases like multi-objective optimization. One of the most well-known and representative class within this paradigm is related to GAs. They were developed by Holland who presented a well-described explanation about how they work in [18]. The main idea behind these kind of algorithms is to imitate the biological natural process based on the natural selection theory.

One of the main reason to use GAs is because their central topic of research is related to *robustness*. That is, the balance between efficiency and efficacy necessary

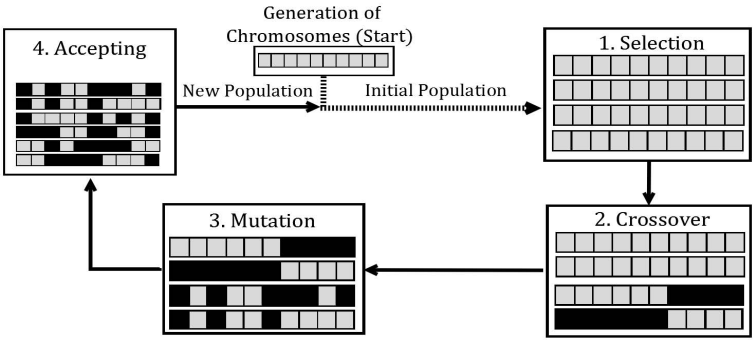


Fig. 2. The iterative process of GAs is carried out mainly in four stages

to survive in different environments [19]. In addition, they offer advantages that overcome traditional optimization techniques, being the most remarkable [20,21]: good results with both complex and large number of variables, parallelism, flexibility, use of probabilistic rules instead of deterministic, resistance to becoming trapped in local optima and hybridization with other methods. These characteristics are desired since the behaviour of this kind of applications depends on various factors such as reaction time and the amount of data.

In Fig. 2 the process of GAs starts with the generation of the initial population. Then, the selection of those with the best fitness. Next, the new offspring is formed by the crossover, and after applying mutation diversity is achieved. Finally, the new population is created. Particulary, in this work rank is used instead of a fitness proportionate as selection method. This is because with the use of the second approach the best individuals can be discarded. Therefore, in order to create the best population the rank method is selected. Also, a single point crossover was chosen and diversity is achieved by altering some genes from the chromosome (i.e. mutation). After applying the previous stages the best chromosomes from the last population can be lost. Hence, using elitism is the best way to avoid that situation. During this process, the best chromosomes are copied directly to the new population.

4 Experimental Part

In this section, the performance of the methodology is evaluated on the data set of Hamlyn Centre Laparoscopic at Imperial College of London [22]. In addition, all experiments were assessed under the same conditions, using a PC with Intel Core i7 CPU 2.00GHz, 8.00GB RAM under windows 7 operating system. This sequence consists of 900 frames of 355*285 pixels size, corresponding to the cardiac surface affected by both respiration and cardiac motion. As first step, the elimination of specular reflections is applied during the sequence. This process is required because the reflection of the light over the glossy surface generate errors

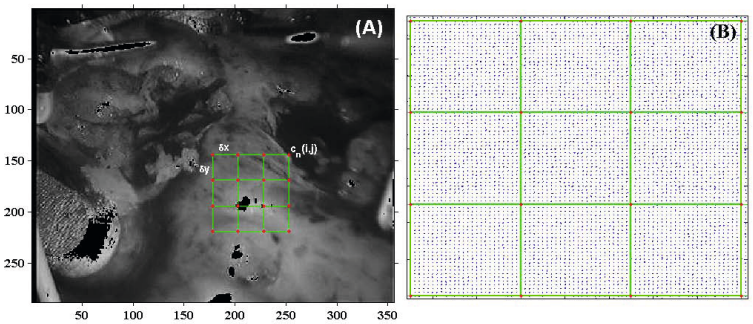


Fig. 3. A) A lattice points of $L \times M$ is defined over the heart surface, and B) after applying the deformation model based on b-splines the displacement field is obtained

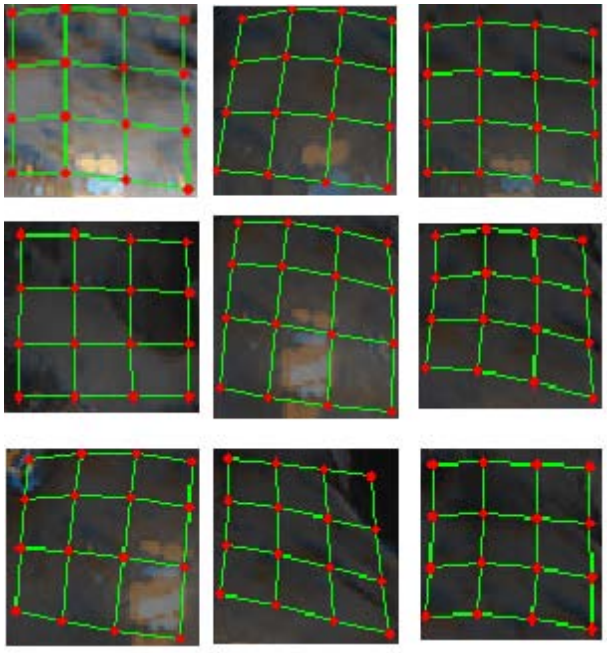


Fig. 4. A sequence of the tracking of the region of interest over the heart

during the execution of the algorithm. Despite different advances techniques can be used for solving this problem, in here, a simple threshold was applied in order to avoid this situation.

Thresholding was selected due to its simplicity and fast computational response. In Fig. 3 the grid used over the heart surface for tracking the region of interest can seen, in this area an interpolation based cubic b-spline is done. It allows obtaining the displacement field that is used during the minimization procedure. Here 16 control points were enough since the main objective is tracking

the movement in a certain point of interest. These actions will be imitated by the robot's instrument. Also, in Fig. 4 the effect caused by various heart cycles can be seen in the alteration of the grid.

5 Conclusion and Future Work

Heart motion compensation is a challenging and unsolved problem in medical robotics due to the lack of robustness. Therefore, the central topic of research in this context is related to robustness, this factor is crucial especially when the solution is oriented to work on human beings. In this sense, in this paper a new methodology for heart motion compensation is presented based on the combination of both b-splines as deformation model and GAs as optimization technique. According with the results, GAs have shown a good response time even with large amount of data. In addition, they offer a balance between efficiency and efficacy. That is, they are able to cover the demands of real-time problems such as reliability and speed, and have demonstrated to be a good line of research for solving this problem. However, even if GAs offer different advantages, it is necessary to bear in mind a set of factors in order to achieve optimal results like the strategy for preserving the best chromosomes. Finally, as future work hybridization will be tested. In other words, using a combination of both GAs with other traditional techniques. The method will be extended to $\Re^3$.

Acknowledgment. This work has been done in the frame of project DPI2011-29660-Co4-01 and DPI2011-29660-C04-03, under the Spanish Research Program, MINECO and with FEDER funds, EC. Also, this research is supported by a FPU national scholarship from the Spanish Ministry of Education with reference AP2012-1943.

References

1. Livesay, J.J.: The benefits of off-pump coronary bypass: A reality or an illusion? Texas Heart Institute Journal (2003)
2. Karamanoukian, H.L.: Decreased incidence of postoperative stroke following on-pump coronary artery bypass. Journal of the American College of Cardiology (2002)
3. World Health Organization (2011), `http://www.who.int/en/`
4. Lemma, A., Mangini, A., Redaelli, A., Acocella, F.: Do cardiac stabilizers really stabilize? experimental quantitative analysis of mechanical stabilization. Interactive CardioVascular and Thoracic Surgery (2005)
5. Jacobs, S., Holzhey, D., Mohr, F., Falk, V.: Limitations for manual and telemanipulator-assisted motion tracking and dexterity for endoscopic surgery. In: Proc. Int. Congr. Comput. Assisted Radiol. Surg., pp. 673–677 (2003)
6. Salvendy, G.: Manual control and tracking. In: Handbook of Human Factors, pp. 182–218. Wiley, New York (1987)

7. Nakamura, Y., Kishi, K., Kawakami, H.: Heartbeat Synchronization for Robotic Cardiac Surgery. In: International Conference on Robotics and Automation, Seoul, Korea, pp. 2014–2019 (2001)
8. Ortmaier, T., Groger, M., Boehm, D.H., Falk, V., Hirzingerw, G.: Motion estimation in beating heart surgery. IEEE Trans. Biomed. Eng., 1729–1740 (2005)
9. Noce, A., Triboulet, J., Poignet, P.: Efficient tracking of the heart using textures. In: Proceedings of the 29th Annual International Conference of the IEEE EMBS (2007)
10. Lo, B., Chung, A.J., Stoyanov, D., Mylonas, G., Yang, G.Z.: Real-time intra-operative 3D tissue deformation recovery. In: Proceedings of IEEE International Symposium on Biomedical Imaging, pp. 1387–1390 (2008)
11. Richa, R., Poignet, P., Liu, C.: Three-dimensional motion tracking for beating heart surgery using a thin-plate spline deformable model. The International Journal of Robotics Research (2009)
12. Hadamard, J.: Lectures on the Cauchy Problems in Linear Partial Differential Equations. Yale University Press, New Haven (1923)
13. Aviles, A.I., Casals, A.: Interpolation Based Deformation Model for Minimally Invasive Beating Heart Surgery. In: Roa Romero, L.M. (ed.) XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013. IFMBE Proceedings, vol. 41, pp. 372–375. Springer, Heidelberg (2014)
14. Lewis, B., Reichel, L.: Arnoldi-Tikhonov regularization methods, pp. 1–21. Elsevier Science (2008)
15. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. 42, 577–685 (1989)
16. Rudin, L., Osher, S.J., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
17. Rouet, J.M., Jacq, J.J., Roux, C.: Genetic algorithms for a robust 3-D MR-CT registration. IEEE T. Inf. Technol. B. 4(2), 126–136 (2000)
18. Holland, J.H.: Adaptation in Natural and Artificial Systems. MIT Press (1975)
19. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addision-Westley (1989)
20. Mitchell, M.: An introduction to genetic algorithms. The MIT Press (1998)
21. Sivanandam, S.N., Deepa, S.N.: An introduction to genetic algorithms. Springer (2008)
22. Stoyanov, D., Mylonas, G., Deligianni, F., Darzi, A., Yang, G.Z.: Soft-tissue Motion Tracking and Structure Estimation for Robotic Assisted MIS Procedure. Medical Image Computing and Computer Assisted Interventions 2, 139–146 (2012)

Haptic Feedback in Surgical Robotics: Still a Challenge

Arturo Marbán[1], Alicia Casals[1,2], Josep Fernández[1], and Josep Amat[1]

[1] Universitat Politècnica de Catalunya, BarcelonaTech, 08034, Barcelona, Spain
{arturo.marban,alicia.casals}@upc.edu
[2] Institute for Bioengineering of Catalonia, 08028, Barcelona, Spain

Abstract. Endowing current surgical robotic systems with haptic feedback to perform minimally invasive surgery (MIS), such as laparoscopy, is still a challenge. Haptic is a feature lost in surgical teleoperated systems limiting surgeons capabilities and ability. The availability of haptics would provide important advantages to the surgeon: Improved tissue manipulation, reducing the breaking of sutures and increase the feeling of telepresence, among others. To design and develop a haptic system, the measurement of forces can be implemented based on two approaches: Direct and indirect force sensing. MIS performed with surgical robots, imposes many technical constraints to measure forces, such as: Miniaturization, need of sterilization or materials compatibility, making it necessary to rely on indirect force sensing. Based on mathematical models of the components involved in an intervention and indirect force sensing techniques, a global perspective on how to address the problem of measurement of tool-tissue interaction forces is presented.

Keywords: surgical robotics, haptic feedback, indirect force sensing, machine learning, data fusion, mathematical models.

1 Introduction

Robotic surgery improves traditional minimally invasive surgery (MIS), such as laparoscopy, through teleoperation. Integrating haptic feedback (HF) as a feature in such surgical robotic systems is still a challenge because there are many technical constraints imposed by MIS procedures. To design and develop a HF system, two problems must be addressed: a) Measurement of forces and b) displaying the obtained information to the user, [1]. Measurement of forces can be implemented following either a *direct* or an *indirect force sensing* approach. In the former, the sensors are located in the point of interaction between the tool and tissue (i.e. the grasping mechanism at the instrument tip); in the latter all electronics (i.e. sensors and other hardware) are moved apart from the patient, and also it is suitable for a large variety of standard instruments, but unfortunately, it may lead to inaccurate results [2].

In robotic surgical systems that perform MIS procedures, such as laparoscopy, all natural HF is lost because the surgeon no longer manipulates the instrument

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_18,

directly. Haptic tehcnology can solve this problem through a force feedback system that implements *indirect force sensing* as depicted in Fig. 1. The need for applying these techniques is due to restrictions imposed by the surgical site, [1]: Several constraints on force sensor design such as size, geometry, cost, biocompatibility and sterilizability, make difficult the placement of such sensors in the tool tip. Also, the forces applied on the patient would ideally be estimated without using a force sensor.

The process of extracting haptic information can be implemented considering a set of *mathematical models* and *indirect force sensing* techniques applied to different scenarions. Each component of a surgical robotic system that forms a chain, starting from the *slave robot manipulator*, followed by the *surgical instrument*, and ending with human *organs/tissues*, as shown on Fig. 2 for a laparoscopic procedure, should be modeled as follows. The first component, the *slave robot*, should be modeled through the dynamic equation, [3]. The second, the *surgical instrument*, should be modeled as: a) A single rod (without the grasping or cutting mechanism) [4] and b) just modeling the grasping [2] or cutting force [6] at the instrument tip. Finally, human *organs/tissues* can be represented by deformable object models.

The set of techniques that deal with the problem of measuring haptic information in an *indirect* way, most of them implemented in scenarios different than the surgical site, and considered in this paper are: (i) Estimation of forces on deformable objects using vision (*Vision based force measurement*, VBFM). (ii) *Soft tissue modeling* using visual and force information. (iii) *Learning deformable objects models* through robot interaction, using visual and force information. (iv) Implementation of a *contact/collision detection* method using a single 6D force/torque sensor. (v) *Slip detection* and *prediction* based on a multi-dimensional time series of data. (vi) *Surface* and *texture recognition* based on information that come from 3-axis accelerometers. (vii) *Detection of contact* and *tactile events* in the surgical area through high frequency acceleration signals.

The aim of this paper is to present a set of methods and techniques that can be used to design a force feedback system, through *indirect force sensing* in the context of robot assisted MIS with the objective of performing surgical procedures such as laparoscopy.

2 Haptic Feedback in Robotic Surgery

The goal of haptic feedback (HF), as depicted in Fig. 1, is to provide force (FF) and/or tactile feedback (TF) to the human operator to achieve a sense of transparency, in which the surgeon does not feel as if he/she is operating a remote mechanism, but rather that his/her own hands are contacting the patient [1]. From the Fig. 1 it can be seen that FF and TF systems, require the use of: a) Sensors on the patient side to acquire haptic information and b) haptic displays to convey the information to the surgeon, [1]. In the next paragraphs these concepts are explained; TF systems are not considered in this work, however a detailed explanation can be found in [1].

Force feedback (FF) systems as shown in Fig. 1, typically measure and display the forces applied to the patient by the surgical instrument. The first task can be implemented under a *direct* or *indirect* approach, as explained in the last section. The second task consists in the transmision of this information to the user through *direct force feedback* (DFF) or a *sensory substution* (SS) technique, [1]; DFF refers to the use of conventional force display technology, in which the motors of the master manipulator are programmed to recreate the forces sensed by the patient side, [1]; in SS the haptic information is presented through graphics on screen (visual force feedback) [18] [19], or a combination of vibration and sounds (high frequency acceleration feedback) [16] [17], to give the user some feeling about the tool-tissue interaction forces. Virtual fixtures (VF) can provide complementary information through force feedback. They are software-generated force and position signals applied to human operators in order to improve the safety, accuracy, and speed of robot-assisted manipulation tasks (i.e. a virtual wall may be placed around a delicate anatomical structure to keep the surgical instruments from contacting it) [1].

The measurement of tool-tissue interaction forces under an *indirect force sensing* approach, considering sensors located outside the body, should deal with the kinematic constraints of the surgical tool, perturbing forces (i.e. friction force at the trocar), distortion effects (i.e. scaling and mirroring of tip forces due to lever-arm effect) and restrictions on force sensor design (ie. size, geometry, cost, biocompatibility, and sterilizability) imposed by the MIS procedure and surgical environment.

In the following sections, a set of *mathematical models* for each component of the surgical system followed by *techniques* that deal with the acquisition of haptic information are presented, suggesting how to address the problem of *indirect force sensing*.

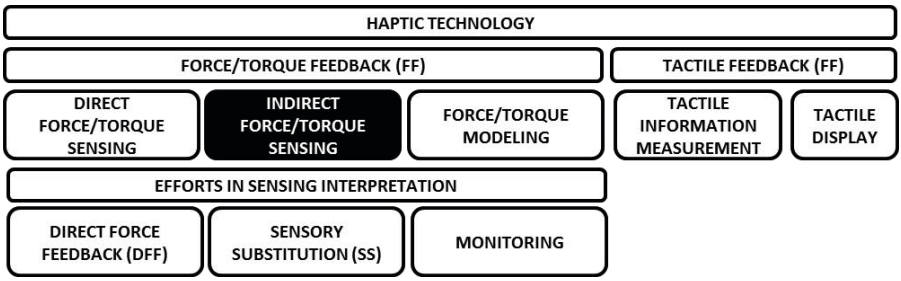


Fig. 1. Haptic technology: Force feedback (FF) & Tactile feedback (TF)

3 Mathematical Modeling of a Surgical Robotic System

A surgical robotic system has some key components that form a chain that starts with the *slave robot manipulator*, followed by the *surgical instrument*, and

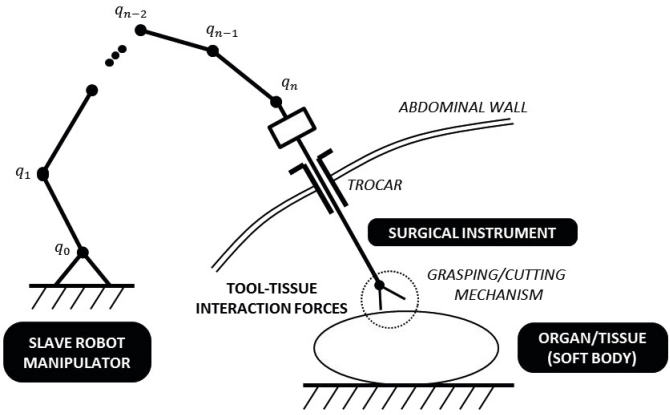


Fig. 2. Modeling the components of the surgical robotic system: Slave robot manipulator, surgical instrument and human organs/tissue

ending with the *human organs* or *tissues*. These components in the context of laparoscopic surgery are ilustrated on Fig. 2 and modeled as follows.

The *slave robot manipulator* can be modeled using the manipulator dynamic equation, derived from the Lagrange or Newton-Euler formulation, or expressed in any convenient form (such as operational space). See [3] for more details about this topic.

The *surgical instrument* can be decomposed into two parts: a) Modeling the surgical instrument as a rod, and b) modeling the force at the grasping or cutting mechanism. In the first case, the grasping or cutting mechanism at the tip is neglected; nonetheless, the following elements should be considered:

1. *Tool-tissue interaction forces.* According to [4] the forces and torques generated at the tool-tip vary with the surgical task and specific procedure; nonetheless, experimental data suggests that they are in the range of 0.5 to 10N and 0 to 0.1 Nm, respectively.
2. *Perturbation forces.* They come from the restrictions imposed by the surgical MIS procedure itself and other effects, and add noise to the real tool-tissue interaction forces: (i) *Friction force at the trocar.* It can be modeled by a non-linear function that depends on the linear velocity of the instrument [4]. The forces vary with the trocar used; for example, for a specific model of trocar, the forces can be as large as 3N. However, by taking care of the selected trocar and/or adding a lubricant, the kinetic friction can be reduced from 15% to 45%, [5]. This force is in the range from 0 to 3N, [4]. (ii) *Torque developed at the abdominal wall.* It is generated while performing movements with the surgical tool and it mainly depends on 2 DOF: The direction (α) and tilting angles (β). A linear model can be obtained when considering only the tilting angle, ranging from 0 to 0.7 Nm, [4]. (iii) *Inertial and gravity*

forces. They are generated only due position, velocity and acceleration of the surgical tool.
3. *Distortion of forces.* Scaling and mirroring of tip forces are originated by the lever-arm effect imposed by the MIS procedure: As the instrument tilts about a theoretical axis located in the abdominal wall, the force is transmitted via lever, the arm of which depends on the depth of the instrument insertion into the cavity, [4].

In the second case, some linear/non-linear models found in literature can be used to approximate the *grasping* or *cutting force* at the tip of the instrument:

1. *Pulling/pinching force.* In [2] a mathematical model to accurately determine the pulling and pinching forces of a laparoscopic grasper from force measurements at the shaft of the instrument (even when the exact geometry of the grasping mechanism is not known) is described; based on an experimental study, it was found that the force transmission from rod to forceps can be approximated by a two dimensional function of the forces applied on the rod (internal mechanism of the instrument) and the opening angle of the grasper.
2. *Cutting force.* An analytical model (with application in haptic rendering in laparoscopic surgical simulation) for cutting human iliac artery with laparoscopic scissors is presented in [6]. The model is based on the concepts of shearing deformation and fracture mechanics.

Human tissue and *organs* can be represented by deformable models. In the context of surgery simulation, [7] presents some methods for tissue modeling based on heuristic and continuum mechanical approaches (also suggest a third cathegory of hybrid models not commented here). In the former the mass-spring model (MSM) is used to represent a body as a set of discreted masses connected with spring (or spring-damper) elements; it is computational efficient, and commonly used in computer animation to render the behavior of soft bodies. However, it doesn't describe the real mechanical behavior of an object. The latter is based on connected elements (a 2D/3D mesh or boundary) and continuum mechanics equations; the finite element method (FEM) and boundary element method (BEM) fall in this cathegory. They are computationally expensive, but they describe the mechanical behavior of a soft body given its material properties (ie. Young Modulus and Poisson Ratio).

4 Indirect Force Sensing Techniques

A set of methods and techniques (most of them defined in environments different than the surgical site) designed to measure forces under an indirect approach are described in this section.

Interaction forces between a instrument driven by a haptic device and a deformable object, are inferred in real time in [9] based on the visual information from the slave environment, without force sensor. The method was evaluated in

two scenarios: Micro and macro-scale environments. In the micro-scale environment, cell manipulation (zebrafish embryos) is improved by allowing the operator to feel the forces through a haptic system, while performing a specific task such as punturing the cell; in this case the concept of *vision based force measurement* (VBFM) is implemented by capturing the deformation of the membrane (through a camera and computer vision techniques) and predicting the force generated, by modeling the cell membrane using boundary element method (BEM). In the macro-scale environment a similar procedure is performed, but using a piece of silicone as a deformable object. The same concept of VBFM, is implemented in [10] to model an elastic object (such as biological tissue) using a feedforward neural network (FNN) in conjunction with a deformable template matching algorithm; here the FNN is trained with a set of images that provide information about deformations of elastic objects, and the applied loads measured by a force sensor.

Deformable objects are found in every day life environments, so it's important to endow robots with the capability to interact with them. An interesting approach to this problem is described in [11], where mechanical properties of deformable objects are modeled through *robotic interaction* in a *learning process*, using *visual* (object deformation is captured through a depth camera) and *force* information (captured through a force/torque sensor). Here deformable objects are modeled through a 3D mesh using FEM. The objective is to estimate the Young modulus (E) and Poisson ratio (ν) that characterizes the material object. A similar approach is presented in [12] to *model* and *simulate soft tissues*; in this case the lighting conditions are controlled (three sources of light are used), three cameras capture the deformation of the object, and the magnitude and direction of interaction forces are measured by a 6D force/torque sensor.

A *contact/collision detection* method is developed and implemented for a humanoid robot in [13], so that it can move safely in an unstructured environment, using an *indirect force sensing* technique; in this case a single 6D force/torque sensor is mounted on the humanoid robot's arm. Here two approaches are considered and evaluated: The *model* and *machine learning* approach. The former uses the manipulator dynamic equation to detect contact states. The latter takes advantage of supervised machine learning algorithms to perform the same task; in this case the use of Least-Square Supported Vector Machines (LS-SVM) and Feedforward Neural Networks (FNN) algorithms are evaluted. Based on experimental data, the *machine learning* approach seems to be more accurate than the *model* approach.

A method for *slip prediction* is developed in [14] in which three *states* (slip, pre-slip and stationary contact) are detected an predicted, based on a multidimensional time series of data that come from a 6D force/torque sensor, and an artificial finger which has randomly distributed strain gauges and polyvinylidene fluoride films (PVDF) embedded in silicone. An incipient slip is detected by studying temporal patterns in the data (at least 100 ms before a slip takes place). Here data is analyzed using probabilistic clustering that transforms it into a sequence of symbols, which is used to train a Hidden Markov Model

(HMM) classifier; furthermore, two unsupervised learning algorithms are used: Principal Component Analysis (PCA) and Gaussian Mixture of Models (GMM).

In [15] surface and texture recognition is performed by a humanoid robot that interacts with different materials by scratching them; here a frequency domain analysis of the vibrations detected by the 3-axis accelerometers and machine learning algorithms, such as k-Nearest Neighbor (kNN) and Supported Vector Machines (SVM), are implemented to perform this task.

A system (denominated *VerroTouch*) is developed in [16] to partially restore the sense of touch in a commercial surgical robotic system: It measures the *vibrations* (captured by 3-axis accelerometers) produced by *tool contacts* and immediately recreates them on the master handles (through vibration actuators). This augmentation enables the surgeon to feel the *texture* of rough surfaces, the start and end of *contact* with manipulated objects, and other important *tactile events*; however it does not provide low frequency forces.

5 Conclusions

The *mathematical models* and *indirect force sensing* techniques presented in the last two sections, address the problem of measuring tool-tissue interaction forces in robotic surgery considering an indirect approach. They suggest the use of three powerful tools: *Machine learning* algorithms, *data fusion* and *mathematical modeling*. This is justified in the next paragraphs.

Machine learning algorithms implement a variety of tasks such as prediction of values and states, classification, clustering, and dimensionality reduction, that mimic human learning processes.

Data fusion leads to the possibility of estimating tool-tissue interaction forces using redundant information. Using different kind of sensors, the system can predict with a better accuracy haptic information. For example, the combination of *visual* and *force information* can be used to *model* or *learn deformable objects models*, as described in [11] and [12], respectively. If only visual information is presented, forces can be approximated with some degree of fidelity, as defined by the VBFM approach, [9] [10]. Finally, if only *high frequency signals* that represent *contact* and *tactile events* are captured by the haptic system through accelerometers, they can be processed and sent to the master console, and the surgeon will perceive them as vibrations and/or sound, [16] [17].

Mathematical models of the components of the surgical system in Fig. 2, allow to interpret interaction forces. Some of them can be suitable to implement in real time, for example, the *pulling/pinching force model* for a laparoscopic instrument presented in [2]. However, other can be computationally expensive, such as modeling deformable objects with the FEM method [7]. To approximate complex mathematical models, techniques such as machine learning algorithms can be used to model elastic objects (similar in nature to biological tissue) as explained in [10] or to detect contact states, as described in [13].

Acknowledgments. This work has been done in the frame of project DPI2011-29660-Co4-01 and DPI2011-29660-C04-03, under the Spanish Research Program, MINECO and with FEDER funds, EC.

References

1. Okamura, A.M.: Haptic feedback in robot-assisted minimally invasive surgery. Current Opinion in Urology 19(1), 102 (2009)
2. van den Dobbelsteen, J.J., Lee, R.A., van Noorden, M.: Indirect measurement of pinch and pull forces at the shaft of laparoscopic graspers. Medical & Biological Engineering & Computing 50(3), 215–221 (2012)
3. Sciavicco, L.: Robotics: modelling, planning and control. Springer (2009)
4. Picod, G., Jambon, A.C., Vinatier, D., Dubois, P.: What can the operator actually feel when performing a laparoscopy? Surgical Endoscopy and Other Interventional Techniques 19(1), 95–100 (2005)
5. Van den Dobbelsteen, J.J., Schooleman, A., Dankelman, J.: Friction dynamics of trocars. Surgical Endoscopy 21(8), 1338–1343 (2007)
6. Yang, T., Xiong, L., Zhang, J., Yang, L., Huang, W., Zhou, J., Liu, J., et al.: Modeling cutting force of laparoscopic scissors. In: 2010 3rd International Conference on Biomedical Engineering and Informatics (BMEI), pp. 1764–1768. IEEE (2010)
7. Meier, U., Lpez, O., Monserrat, C., Juan, M.C., Alcaniz, M.: Real-time deformable models for surgery simulation: a survey. Computer Methods and Programs in Biomedicine 77(3), 183–197 (2005)
8. Rosen, J., Brown, J.D., Chang, L., Sinanan, M.N., Hannaford, B.: Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete Markov model. IEEE Transactions on Biomedical Engineering 53(3), 399–413 (2006)
9. Kim, J., Janabi-Sharifi, F., Kim, J.: A haptic interaction method using visual information and physically based modeling. IEEE/ASME Transactions on Mechatronics 15(4), 636–645 (2010)
10. Greminger, M.A., Nelson, B.J.: Modeling elastic objects with neural networks for vision-based force measurement. In: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003, vol. 2, pp. 1278–1283. IEEE (2003)
11. Frank, B., Schmedding, R., Stachniss, C., Teschner, M., Burgard, W.: Learning Deformable Object Models for Mobile Robot Navigation using Depth Cameras and a Manipulation Robot. In: Proc. of the IEEE Intl. Conf. on Robotics & Automation, ICRA (2010)
12. Bickel, B., Bcher, M., Otaduy, M.A., Matusik, W., Pfister, H., Gross, M.: Capture and modeling of non-linear heterogeneous soft tissue. ACM Transactions on Graphics (TOG) 28(3), 89 (2009)
13. Fumagalli, M., Gijsberts, A., Ivaldi, S., Jamone, L., Metta, G., Natale, L., Nori, F., Sandini, G.: Learning to exploit proximal force sensing: a comparison approach. In: Sigaud, O., Peters, J. (eds.) From Motor Learning to Interaction Learning in Robots. SCI, vol. 264, pp. 149–167. Springer, Heidelberg (2010)
14. Jamali, N., Sammut, C.: Slip prediction using Hidden Markov models: Multidimensional sensor data to symbolic temporal pattern learning. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 215–222. IEEE (2012)

15. Sinapov, J., Sukhoy, V., Sahai, R., Stoytchev, A.: Vibrotactile recognition and categorization of surfaces by a humanoid robot. IEEE Transactions on Robotics 27(3), 488–497 (2011)
16. Kuchenbecker, K.J., Gewirtz, J., McMahan, W., Standish, D., Martin, P., Bohren, J., Mendoza, P.J., Lee, D.I.: VerroTouch: high-frequency acceleration feedback for telerobotic surgery. In: Kappers, A.M.L., van Erp, J.B.F., Bergmann Tiest, W.M., van der Helm, F.C.T. (eds.) EuroHaptics 2010, Part I. LNCS, vol. 6191, pp. 189–196. Springer, Heidelberg (2010)
17. McMahan, W., Gewirtz, J., Standish, D., Martin, P., Kunkel, J.A., Lilavois, M., Wedmid, A., Lee, D.I., Kuchenbecker, K.J.: Tool contact acceleration feedback for telerobotic surgery. IEEE Transactions on Haptics 4(3), 210–220 (2011)
18. Reiley, C.E., Akinbiyi, T., Burschka, D., Chang, D.C., Okamura, A.M., Yuh, D.D.: Effects of visual force feedback on robot-assisted surgical task performance. The Journal of Thoracic and Cardiovascular Surgery 135(1), 196–202 (2008)
19. Talasaz, A., Trejos, A.L., Patel, R.V.: Effect of force feedback on performance of robotics-assisted suturing. In: 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp. 823–828. IEEE (2012)

Robot Collaborative Assistance for Suture Procedures via Minimally Invasive Surgery

E. Bauzano, B. Estebanez, I. Garcia-Morales, and V.F. Muñoz-Martinez

University of Malaga, Spain
vfmm@uma.es

Abstract. Surgical robotics has usually provided the handle of surgical tools by using teleoperated systems or the automation of certain surgical tasks. However, the final goal of this field has always consisted of allowing the surgeon to perform an intervention without additional human assistance (i.e. solosurgery). In this way, this paper is focused on the design and implementation of a semi-autonomous surgical robot capable of assisting the surgeon during a suture procedure on a Hand Assisted Laparoscopic Surgery (HALS) scenario. The methodology proposed is based on a dispatcher system which manages the actuations of the robot depending on the recognized gestures of the surgeon's tool. The performance of the whole architecture has been tested by means of in vitro trials.

Keywords: Surgical Robotics, Task Planning, Human-Robot Interaction, Autonomous Robots, Hidden Markov Models.

1 Introduction

Minimally Invasive Surgery (MIS) has become one of the most important surgical techniques in the last twenty years due to its capability of reducing the size of incisions, shortening postoperative convalescence and diminishing other complications. These improvements involve several social and economic consequences, for example minimizing the time that patient stays at hospital.

In this way, one of the current trends in MIS consists of the Hand-Assisted Laparoscopic Surgery (HALS). This technique allows the surgeon to insert a hand through a small incision via a pressurized sleeve. Thus, the surgeon can use his non-dominant hand intra-abdominally while he operates a standard laparoscopic instrument with the other hand. With this technique the surgeon gains control over the operation, sense of depth and tactile feedback in comparison with a conventional laparoscopic surgical technique while still being a MIS intervention. This strategy has been proved as a valid option in certain kind of interventions, for example on a colorectal cancer surgery [1] where the patients presented a similar blood loss, operative time and stance in the intensive care unit than MIS interventions. Similar results have been obtained for colon cancer surgical interventions [2].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_19,

However, the HALS methodology only allows the surgeon to manipulate one laparoscopic tool, so it is necessary additional assistance for handling the camera or any other surgical instruments. This technique also requires that the surgeon remains in direct contact with the patient. Conventional systems like ZEUS by Computer Motion and DaVinci by Intuitive Surgical are not designed to satisfy such requirements. Therefore, the replacement of a human assistant needs a sophisticated robot system capable of taking autonomous decisions as well as communicating through an advanced Human-Machine Interface (HMI) system for a natural collaboration.

The automation of surgical maneuvers by using the robot co-worker concept allows surgeons to concentrate on the surgical procedures while the robot assistant automatically helps in a collaborative way. Visual Servoing techniques are defined as the most common methodology for designing automated tasks and it is used in surgical applications like safe movements of the endoscope on cardiac surgery [3]. Other works based on the automation of surgical maneuvers are devoted to actively assist the surgeon during the intervention, for example: by performing autonomous stitching and knot tying procedures [4]; by providing automatic conversion from laparoscopic navigation to open-surgery motion [5]; or by automatically navigating on cholecystechtomy interventions without pre-operative information [6].

A natural and collaborative interaction between the surgeon and the robot assistant requires an advanced HMI capable of imitating the communication with a human assistant during an intervention [7]-[8]. In this way, some studies focus on the location of the surgeon's arms [9], whereas others devote to the tracking of the surgeon's tools [10] or even the detection of face gestures [11]. However, these HMIs add new and unnatural tasks to the surgeon. For this reason, a new generation of interfaces have to provide the robot with the knowledge of the surgical protocol [12]-[13], in such a way that the current stage of the intervention can be known and the robot assistant can react autonomously in order to help the surgeon without direct commands.

This paper proposes a robotic assistance designed for a HALS scenario with capabilities of both, automation of surgical maneuvers and assistance in a collaborative way regards an improved HMI. For this purpose, firstly section 2 will introduce the HALS scenario for a suture procedure, as well as the control architecture proposed in this work in order to solve the automatic assistance to the surgeon. The main element of this architecture consists of a dispatcher system, which is devoted to follow each step of the suture procedure and will be explained on section 3. The evolution of such dispatcher occurs if the robot finishes a task or a recognizer of the surgeon's tool movements detects the fulfillment of a gesture, all of what will be covered on section 4. The implantation and experiments which validate the methodology are exposed on section 5. Finally, section 6 presents some conclusions and future lines of researching.

2 Surgical Scenario and Control Architecture

The development of the robotic assistance previously stated requires an initial description of its environment and the constraints that will limit its movements. Therefore, this section firstly introduces the HALS scenario as well as the suture method selected for the automation of a surgical task.

On the other hand, the analysis of a surgical protocol must take into account the coordination between all surgical tools involved on the intervention. The knowledge of such coordination allows distributing the tasks considered between the surgeon and the robot assistant. For this purpose, this section secondly shows the control architecture proposed in this work to aid the surgeon without additional human assistance.

2.1 HALS Scenario and the Surgical Protocol

Fig. 1 shows a surgical HALS scenario which includes the robot assistance to the surgeon. In this way, the robot interacts with the patient's body and the surgeon in a constraint space inside the Abdominal Cavity. One of the Surgeon's Hands is inserted through a special incision in order to directly manipulate the anatomical structure of interest. Thus, only one hand remains free to use another instrument, which has been denoted as the Surgeon's Tool. Therefore, a robot assistant should be able to handle at least the laparoscopic Camera and an extra tool labeled as the Robot Tool. Such limitation for the surgeon's freedom of movements can be particularly relevant in certain surgical procedures which require the use of additional tools.

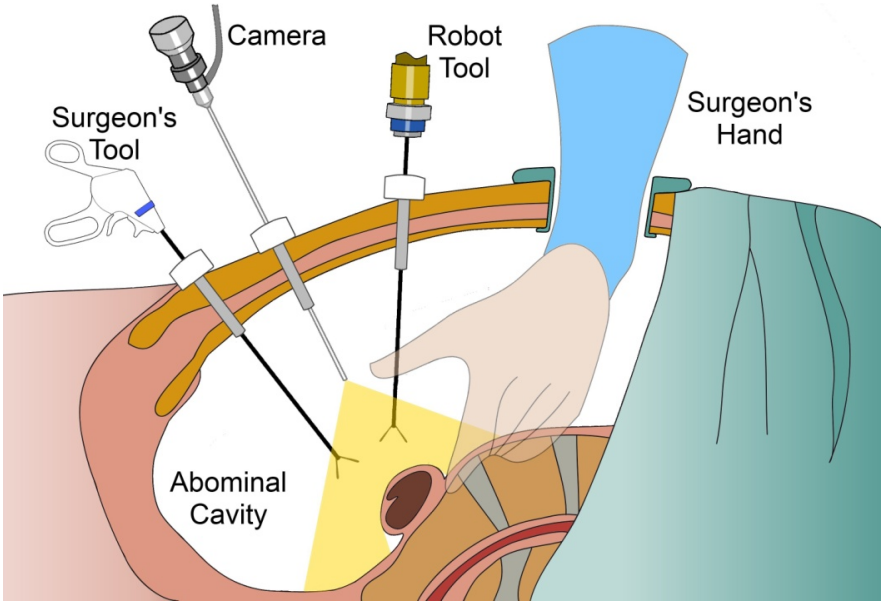


Fig. 1. Navigation problem for a robot assistant on a HALS scenario

More specifically, the traditional suture and knot tying consist of a procedure that involves two laparoscopic tools, and can be performed through several methodologies. This work focuses on the Rosser technique because it separates the tasks of both tools in two roles: the main and the supporting tools [14]. First one performs the suture and knots over the tissue; on the other hand, the supporting tool is devoted to assist the procedure by holding the needle or stretching the thread. Thus, a

surgical robot may provide automated assistance in a collaborative way by handling such supporting tool.

2.2 The Control Architecture

The previous description of the suture & knot tying procedure on a HALS scenario states the three main features for the use of a robotic assistant:

- The navigation of the supporting tool with several restrictions to the movement.
- The detection of the suture gestures performed by the surgeon.
- The monitoring of the suture workflow in order to know the current task.

In this way, Fig. 2 introduces the control architecture proposed to automatically assist a suture procedure. The overall system, so-called *State Assistant*, consists of three elements which solves each of the problems described above. The *Suture Assistant* models the workflow of the suture procedure through a state diagram that will be detailed on section 3. Such workflow can be modified by means of a surgeon's Voice Command processed by the *Voice Decoder* device. Depending on the current task of the suture, the state diagram is devoted to dispatch the detection of the current surgeon's surgical gesture or the actuation of the robot assistant with an automatic

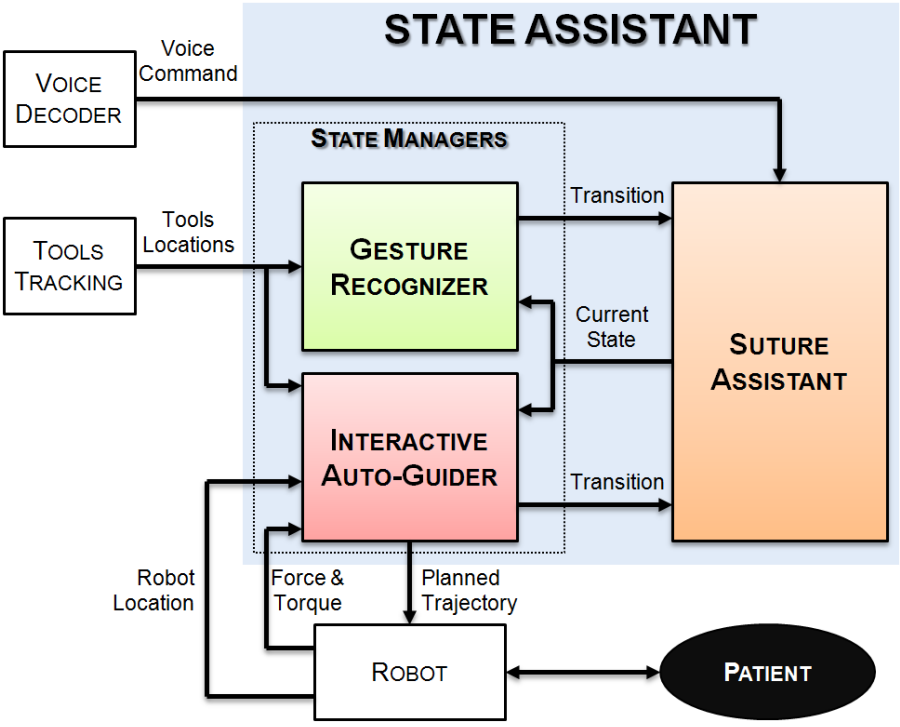


Fig. 2. Control architecture for an automatic assistance to the surgeon

movement. These tasks are applied by two *State Manager* systems, respectively: the *Gesture Recognizer* or the *Interactive Auto-Guider*. They communicate with the suture assistant by sending the activation of the Transition and receiving the Current State of the suture procedure. The gesture recognizer requires the Tools Locations measured by a *Tools Tracking* sensor for both, the surgeon and the robot. On the other hand, the interactive auto-guider receives the Force & Torque exerted and the current Robot Location in order to compute a suitable Planned Trajectory for the robot tool. The overall explanation about these inputs as well as the methodologies proposed to solve these state managers will be exposed on section 4.

3 The Suture Assistant

The main task of the suture assistant consists of following the suture procedure online. More specifically, a suture using the proposed Rosser technique can be divided into four steps denoted as maneuvers. Each maneuver is also subdivided into several elemental movements, so-called gestures. This work has considered two simplifications in order to adapt the original Rosser technique to an automated assistance:

1. All consecutive gestures performed by the same hand are grouped into a single gesture.
2. Only one hand can perform a gesture at the same time.

The description of the suture procedure starts with the Needle Setup. This maneuver is devoted to insert the needle into the abdominal cavity, so the surgeon is able to locate it in the center of the screen. Secondly, the Needle Puncture maneuver stitches the tissue with the surgical needle. Afterwards, the Knot Tying consists of three knots that must ensure and close the tissue, so this maneuver has to be repeated three times. Finally, the Cut Thread maneuver removes the extra thread and finishes the suture procedure.

Table 1 summarizes the sequence of gestures already adapted for an automated assistance by applying the simplifications previously commented. Each gesture has been labeled as S_k depending on its order on the sequence, and is included into its corresponding maneuver. Besides, this table also shows which subject (surgeon or robot) performs its corresponding gesture.

The sequence of gestures described on Table 1 can be implemented into the Suture Assistant as the state diagram proposed on Fig. 3. As it has been stated above, the assumption of allowing only one subject (surgeon or robot) to make a task at a time avoids possible concurrency issues on this finite-state diagram.

Each state of the diagram matches with a gesture of the table, and each frame covers a maneuver with their corresponding series of states. All states are connected with transitions T_k that allows the evolution to the following state. The robot states S_k^R in red color denotes a gesture performed by the robot supporting tool. On the other hand, surgeon's states labeled as S_k^S in green color identify the gestures of the surgeon's main tool. Robot states compute a trajectory planning based on a specific

Table 1. Suture Procedure with Robotic Assistance

Maneuver	Tool*	Gesture	Description
Needle Setup	R	S_1^R	Insert needle and center on screen
	S	S_2^S	Take needle and locate on tissue
Needle Puncture	R	S_3^R	Hold tissue next to the suture
	S	S_4^S	Puncture tissue, extract needle and stretch thread
	R	S_5^R	Release tissue and open grasp
	S	S_6^S	Locate needle into robot's grasp
Knot Tying	R	S_7^R	Locate needle for knot
	S	S_8^S	Rotate tool around the thread
	R	S_9^R	Stretch thread and close knot
Cut Thread	R	S_{10}^R	Get the needle closer to the suture
	S	S_{11}^S	Take thread's both ends & stretch
	R	S_{12}^R	Grasp thread under surgeon's tool tip
	S	S_{13}^S	Change to cutting tool and cut thread
	R	S_{14}^R	Extract thread and needle

*Tools: (R) Robot and (S) Surgeon.

target force or location, whereas surgeon's states recognize the current gesture performed by the surgeon's tool. In particular, the state diagram always starts from an initial state S_0 in order to represent the beginning of the suture procedure.

The normal workflow of the diagram which follows the suture procedure is represented by solid arrows tagged with the transitions T_k. As a specific situation, the knot tying maneuver needs to be completed three times. Therefore, the transitions T_9 and T_{10} of state S_9^R are conditioned by an iterator i that indicates the number of loops already performed.

The state diagram on Fig. 3 also shows additional transitions between states represented by dotted arrows and denoted as E_k or ${}^{k+1}E_k$. The notation depends on the source and target of the transition in such a way that E_k returns to its same state S_k, ${}^{k+1}E_k$ starts from state S_k and goes to state S_{k+1}. These special transitions are included in order to provide the surgeon a methodology to modify the normal workflow of the suture procedure. Such situations consider bad gestures recognition, unexpected events during the surgical protocol or even an anomalous behavior of the robot assistant. For this purpose, the transitions related to a specific state S_k are activated through a respective command voice of the surgeon for each transition E_k, ${}^{k-1}E_k$ or ${}^{k+1}E_k$.

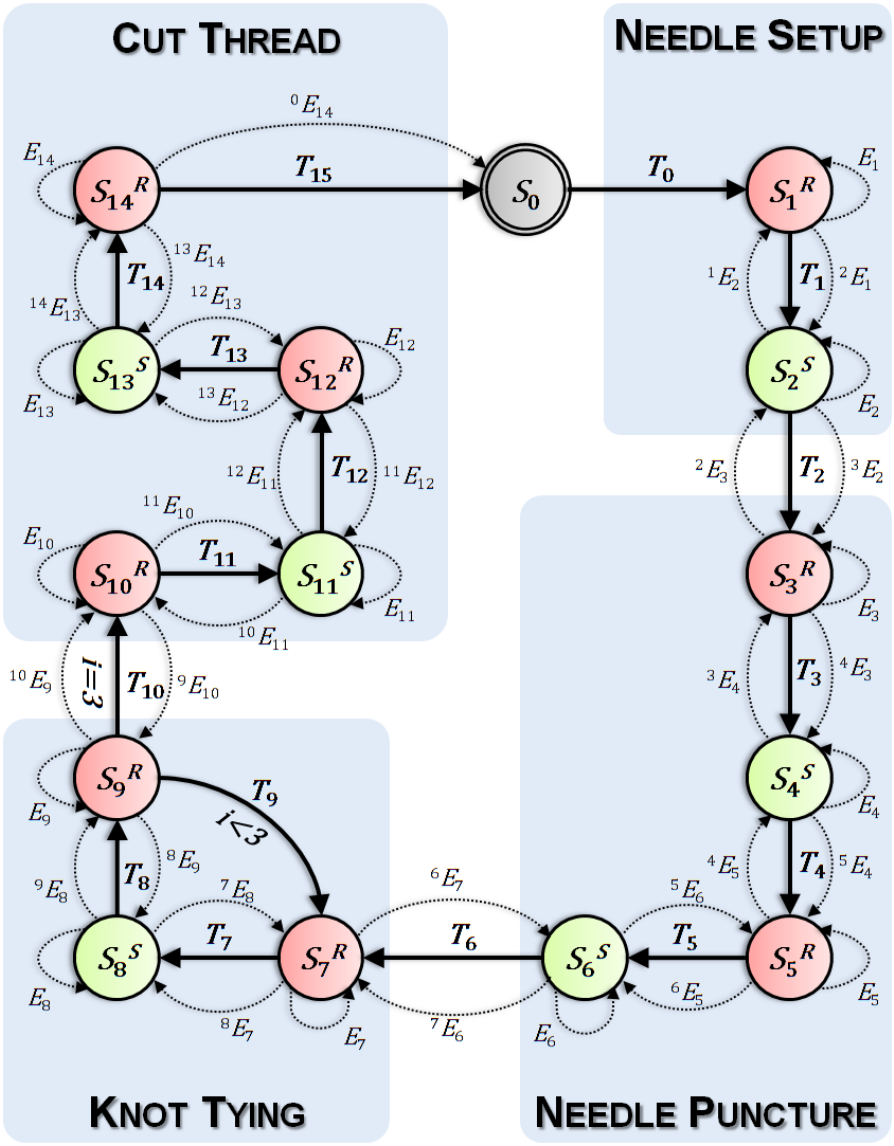


Fig. 3. State diagram based on the workflow of the suture procedure

4 The State Managers

States of the Suture Assistant require the actuation of two systems that handles the evolution of the state diagram to its next step. Firstly, section 4.1 will describe the interactive auto-guider proposed for the automation of the robot tool tasks. Secondly,

section 4.2 will be focused on the gesture recognizer developed for analyzing the surgeon's movement in order to detect the current gesture.

4.1 The Interactive Auto-Guider

The suture procedure exposed in Table 1 determines the main tasks that the robot must be able to handle:

- Move the robot tool tip towards a target location.
- Apply a force over a specific tissue of the patient.
- Stretch the suture thread.

In this way, the interactive auto-guider has been designed as shown in Fig. 4 to manage these tasks during the evolution of any robot state S_k^R of the Suture Assistant. The trajectory planning can be computed in terms of a target location vector $\mathbf{r_k}$ or the exertion of a vector force $\mathbf{F_k}$. This decision is taken by the *Target Selector* element depending on the current robot state S_k^R, and its outputs are obtained with the aid of the current tools locations vector $\mathbf{r_t}$. The final input of the robot is the controlled location vector $\mathbf{r_c}$, which consists of the contribution between the planned location $\mathbf{r_p}$ and the incremental displacement $\mathbf{\Delta x_f}$. Vector $\mathbf{r_p}$ is computed by means of the *Auto-Guide Planner* and represents the next location of the trajectory, whereas vector $\mathbf{\Delta x_f}$ requires the actuation of the force control loop shown below Fig. 4.

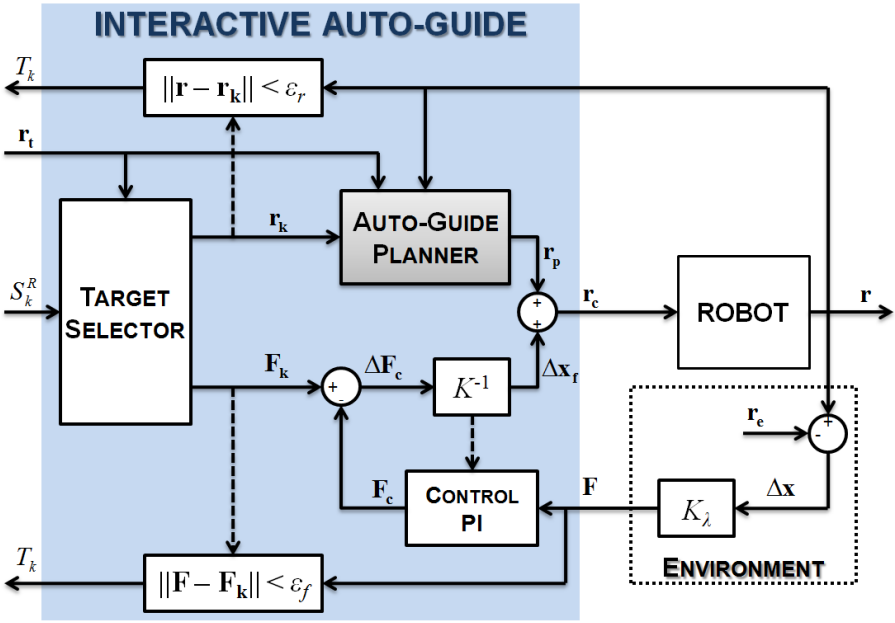


Fig. 4. Interactive Auto-Guide System for autonomous actuations

The transition T_k of S_k^R will be triggered once the following conditions are accomplished:

$$\begin{aligned} \left\| \mathbf{r} - \mathbf{r_k} \right\| &< \varepsilon_r \\ \left\| \mathbf{F} - \mathbf{F_k} \right\| &< \varepsilon_f \end{aligned} \tag{1}$$

Where ε_r and ε_f are the thresholds considered to reach both, the target location and the exertion of a desired force respectively.

Firstly, the force loop is based on a force-accommodative controller with a PI feedback that receives the vector force $\mathbf{F}$ measured by the robot and outputs the controlled vector force $\mathbf{F_c}$. This signal is substracted to $\mathbf{F_k}$ afterwards and transformed to $\mathbf{\Delta x_f}$ by means of a gain K which models the robot-environment relative stiffness with (2):

$$\mathbf{\Delta x_f} = K^{-1}\left(\mathbf{F_k} - \mathbf{F_c}\right) \tag{2}$$

Thus, in case of a non-zero $\mathbf{F_k}$ the robot will move with the direction of $\mathbf{\Delta x_f}$ until that force is applied. The stiffness K may be different depending on the current state S_k^R (i.e. holding tissue of the patient or stretching the suture thread). On the other hand, the *Environment* has been modeled with a K_λ that represents its real stiffness, and $\mathbf{r_e}$ is the position where the robot contacts the environment surface with the exertion of a null force.

Secondly, the Auto-Guide Planner element of Fig. 4 focuses on finding a path towards $\mathbf{r_k}$ by avoiding obstacles within the environment through the Artificial Potential Fields (APF) algorithm. Though this method computes a trajectory for the robot tool by itself, it has been added a velocity corrector system that modifies the robot velocity regards the prediction of a future collision in case that an obstacle is expected to cross the APF planned trajectory [15].

The APF associates a virtual repulsion field U_m^{rep} to each of the m obstacles and a virtual attraction field U^{att} to the target location. Such fields generate virtual forces of attraction $\mathbf{F^{att}}$ and repulsion $\mathbf{F_m^{rep}}$ over the robot tool. Thus, the force vector required to move the robot to the next planned location $\mathbf{r_p}$ is:

$$\mathbf{F_p} = -\nabla U^{att} - \sum_m \nabla U_m^{rep} = \mathbf{F^{att}} + \sum_m \mathbf{F_m^{rep}} \tag{3}$$

The obstacles considered in this work consist of the surgeon's tool and the inner tissue of the abdominal cavity, whose surface is represented by a grid of m–1 vertices generated through a monocular SLAM technique with reallocation for laparoscopic sequences [16]. On the other hand, expression of $\mathbf{F^{att}}$ has been chosen to be quadratic with the distance towards the target location whereas each $\mathbf{F_m^{rep}}$ is generated by a cylindrical field around each obstacle, thus:

$$\mathbf{F}^{\mathbf{att}} = 2K^{att}\Delta\mathbf{r_k}$$
$$\mathbf{F}_{\mathbf{m}}^{\mathbf{rep}} = \begin{cases} K^{rep}\left(\frac{1}{\rho_m} - \frac{1}{\rho_m^0}\right)\left(\frac{\Delta r_k \boldsymbol{\rho}_{\mathbf{m}}}{\rho_m^2} + \frac{\mathbf{n_m}}{2}\right) & \rho_m \le \rho_m^0 \\ 0 & \rho_m > \rho_m^0 \end{cases} \quad (4)$$

The gains K^{att}, K^{rep} on (4) represent the relevance of that virtual force relative to the others. Parameter ρ_m defines the minimal distance between the robot tool and that m-th obstacle, $\boldsymbol{\rho}_{\mathbf{m}}$ is the unitary vector with that direction and $\rho_m{}^0$ is the maximum distance where $U_m{}^{rep}$ may affect the robot trajectory. The distance between the robot's tool tip and the target location is denoted by Δr_k, and modifies the contribution of the repulse force nearby the target location. Moreover, $F_m{}^{rep}$ includes an extra unitary vector $\mathbf{n_m}$ which prevents the trajectory to find a local minima position [17]. In this way, the trajectory can be followed by means of the velocity vector $\mathbf{v_p}$, which results from integrating the vector $\mathbf{F_p}$ (5):

$$\mathbf{v_p} = \int \mathbf{F_p} dt \quad (5)$$

4.2 The Gesture Recognizer

This state manager interprets the gestures performed by the surgeon. Each of these gestures corresponds to a surgeon's state S_k^S as described on section 3, and consists of a specific sequence of interactions that occur between the surgical tools. Such interactions are considered basic actions and are defined by observable characteristics, which include information about the tools pose and their kinematics [18]. Each gesture is modeled by a Hidden Markov Model (HMM) and is described by a stochastic diagram with a finite number of states, where each state represents a basic action. The topology of the network used in this work for the HMMs is based on a modified *Bakis Model*, which allows several backward steps in each state [19].

The recognition of the current surgeon's gesture requires its comparison with all the HMMs $\Gamma=[\lambda_1\ \lambda_2\ \ldots\ \lambda_k]$ of each gesture S_k^S that are included into the *Gestures Library* shown in Fig. 5. This scheme considers both, the Training Mode of the HMMs and the Recognizer Mode of the current gesture. These processes share the acquisition of the sequence of observable characteristics $E=[e_1\ e_2\ \ldots]$. In this way, the *Data Pre-Processing* element receives as input the tools location vector $\mathbf{r_t}$ and transforms the velocities of the tools tips, the angle between tools and the distance between tips into a features vector C. Afterwards, the *Data Coding* module assigns an observable characteristic e depending to the combination of the components of C. The *Sequence Selector* module is devoted to stack the sequence of observable characteristics E of each surgical task. This sequence represents the whole gesture once the velocities of both tools V_t are null or the mean time required to complete this gesture t_m has been exceeded.

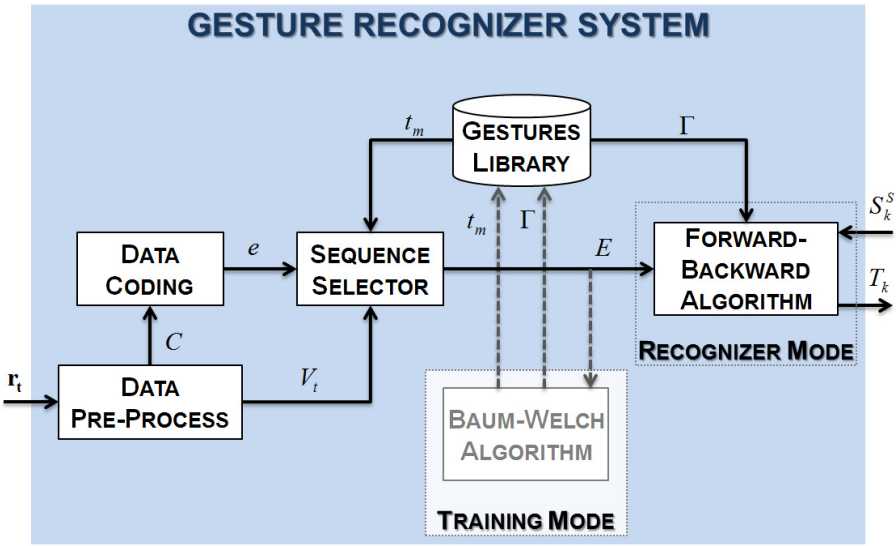


Fig. 5. Gesture Recognizer System of surgeon's tool movements

On one hand, Fig. 5 shows in dashed line the Training Mode, which processes the training of the HMMs Γ stored in the Gestures Library. For this purpose, the *Baum-Welch Algorithm* adjusts the parameters of each HMM λ_k with the aid of several trials of its corresponding gesture [20].

On the other hand, the Recognizer Mode uses the *Forward-Backward Algorithm* in order to compare the sequence E with the set of HMMs Γ and find the most probable gesture [19]. If this gesture is equal to the current state S_k^S established by the Suture Assistant module, the transition T_k is activated and the state diagram already described on Fig. 3 evolves to its next state.

5 Implantation and Experiments

The surgical scenario of Fig. 6 exposes the required elements for the robotic assistance on a HALS intervention with a simulated patient. The surgeon follows the intervention through a Screen which shows the image from the laparoscopic Camera. This tool is hold by the Endoscopic Robotic Machine (ERM), a robot assistant developed at the University of Malaga which moves the endoscope through voice commands [21]. The other surgical tools are handled by both, the surgeon and a Mitsubishi PA10 manipulator which is navigated with a low-level force controller [22]. The location of all the surgical tools is registered by means of a 3D Tracker device. Meanwhile, the other Surgeon's Hand holds an organ or tissue inner the Abdominal Cavity.

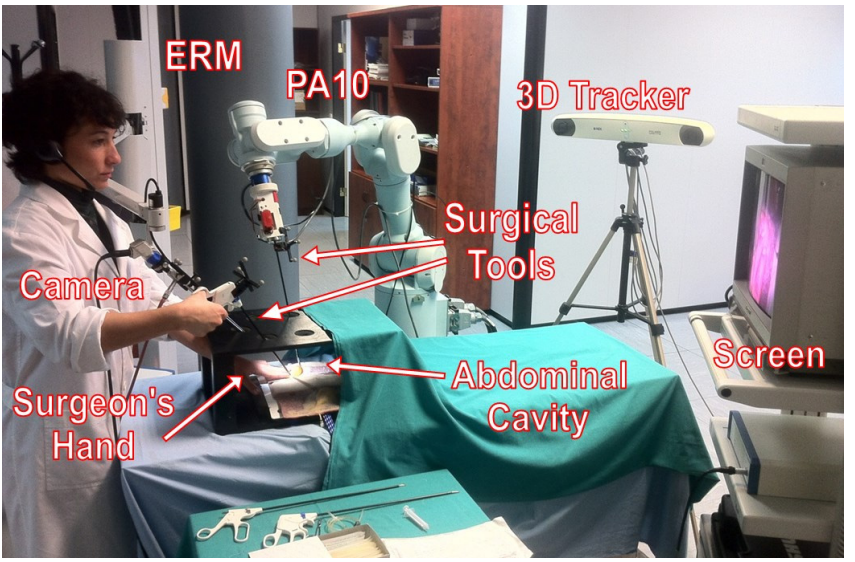


Fig. 6. Experimental setup for the validation of the State Assistant

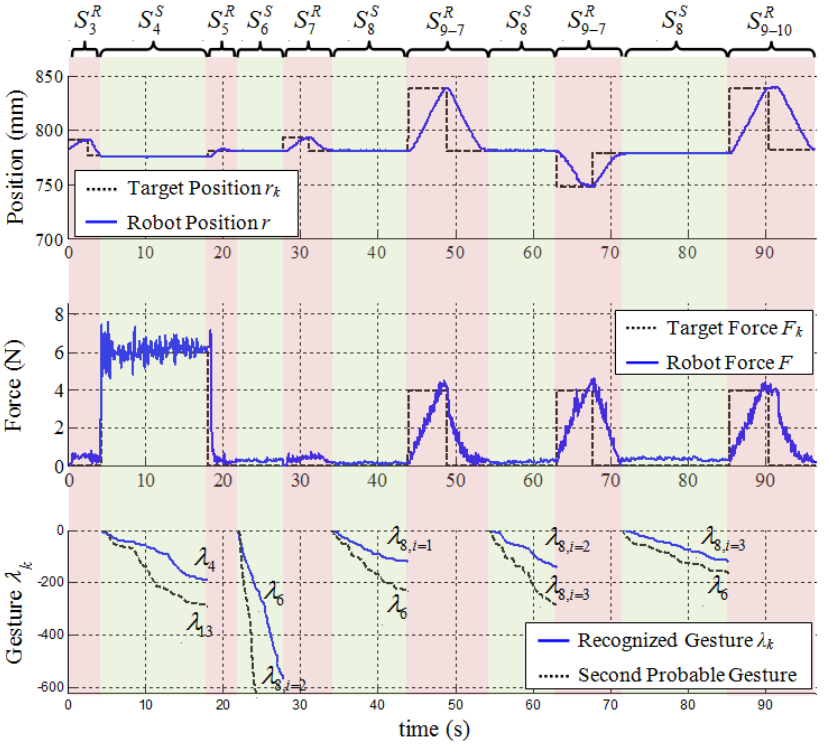


Fig. 7. Sequence of Needle Puncture and Knot Tying maneuvers

The experiment proposed consists of analyzing the evolution of the main maneuvers for the suture procedure: the needle puncture and knot tying (see section 3). Thus, Fig. 7 presents the results obtained during the sequence of gestures for these maneuvers from S_3 to S_9 (see Table 1). The two upper graphs plot the evolution on the robot tool tip position r and force exerted F. These values change whenever the suture assistant evolves to a new robot state that sends its corresponding target position r_k and/or target force F_k. More specifically, in the end of the gesture ${S_3}^R$ the robot applies a constant force over the tissue which aids the surgeon to insert the needle in ${S_4}^S$, whereas states ${S_7}^R$ and ${S_9}^R$ are devoted to stretch the thread in order to close the three knots. Below, a third graph exposes the recognition of the most two probable gestures made by the surgeon's tool (higher values are more probable). In this way, it can be deduced that the corresponding gestures λ_k are recognized with confidence due to the high difference with the second choice.

6 Conclusions

This paper has presented the control architecture for a robotic assistance on a HALS scenario. More specifically, two issues have been identified in order to use such systems with this technique: a flexible auto-guide planning for the robot tool trajectories and the exertion of forces; and a natural surgeon-robot communication that implements both, direct orders from the surgeon and real-time gestures recognition. The commercial robotic assistants are usually focused on a teleoperation interface; therefore, they are unable to address a HALS technique due to the requisite of direct contact between the surgeon and the patient.

Another extra advantage of this work is that the methodology proposed for a suture can be extrapolated to other types of surgery. The only requirements would be the adaptation of the state diagram and additional training for the gesture recognizer.

From the automatic guidance point of view, the authors believe that a deeper interaction between the camera movements and the robot tool would improve such automatic tasks. Moreover, other surgical procedures could be considered for their automation in addition to the suture like cutting, transporting or holding a tissue. Besides, all these techniques have emergency protocols that may also be considered for automation, for example in case of an unexpected broken vessel or tissue.

References

1. Orcutt, S.T., Marshall, C.L., Balentine, C.J., Robinson, C.N., Anaya, D.A., Artinyan, A., Berger, D.H., Albo, D.: Hand-assisted laparoscopy leads to efficient colorectal cancer surgery. J. Surgical Research 177(2), e53–e58 (2012)
2. Pendlimari, R., Holubar, S.D., Pattan-Arun, J., Larson, D.W., Dozois, E.J., Pemberton, J.H., Cima, R.R.: Hand-assisted laparoscopic colon and rectal cancer surgery: Feasibility, short-term, and oncological outcomes. Surgery 148(2), 378–385 (2010)

3. Elhawary, H., Popovic, A.: Robust feature tracking in the beating heart for a robotic-guided endoscope. Int. J. Medical Robotics and Computer Assisted Surgery 7, 459–468 (2011)
4. Knoll, A., Mayer, H., Staub, C., Bauernschmitt, R.: Selective automation and skill transfer in medical robotics: a demonstration on surgical knot-tying. Int. J. Medical Robotics and Computer Assisted Surgery (2012), doi:10.1002/res.149
5. Kranzfelder, M., Schneider, A., Gillen, S., Feussner, H.: New technologies for information retrieval to achieve situational awareness and higher patient safety in the surgical operating room: the MRI institutional approach and review of the literature. Surgical Endoscopy 25, 696–705 (2011)
6. Staub, C., Can, S., Knoll, A., Nitsch, V., Karl, I., Farber, B.: Implementation and evaluation of a gesture-based input method in robotic surgery. In: IEEE Int. Workshop on Haptic Audio Visual Environments and Games, HAVE (2011)
7. Wasen, K.: Person-friendly Robot Interaction: Social, Phychological and Technological Impacts in Health Care Work. In: IEEE Int. Workshop on Robots and Human Interactive Communication, pp. 643–648 (2005)
8. Herman, B., Hassan Zahraee, A., Szewczyk, J., Morel, G.: Ergonomic and Gesture Performance of Robotized Instruments for Laparoscopic Surgery. In: IEEE Int. Conf. Intelligent Robots and Systems (IROS), pp. 1333–1338 (2011)
9. Casals, A., Frigola, M., Amat, J., Laporte, E.: Quasi Hands Free Interaction with a Robot for Online Task Correction. In: Ang Jr., M.H., Khatib, O. (eds.) Experimental Robotics IX. STAR, vol. 21, pp. 175–184. Springer, Heidelberg (2006)
10. Hsu, J., Payandeh, S.: Toward Tool Gesture and Motion Recognition on a Novel Minimally Invasive Surgery Robotic System. In: IEEE Int. Conf. on Robotics and Automation (ICRA), Orlando, Florida, pp. 631–636 (2006)
11. Nishikawa, A., et al.: FAce MOUSe: A novel human-machine interface for controlling the position of a laparoscope. IEEE Trans. Robotics and Automation 19(5), 825–841 (2003)
12. Ko, S.Y., Lee, W.J., Kwon, D.S.: Intelligent Interaction based on a Surgery Tak Model for a Surgical Assistant Robot: Awareness of Current Surgical Stages based on a Surgical Procedure Model. Int. J. Control, Automation and Systems 8(4), 782–792 (2010)
13. Staub, C., Can, S., Jensen, B., Knoll, A., Kohlbecher, S.: Human-Computer Interfaces for Interaction with Surgical Tools in Robotic Surgery. In: 4th IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics, Roma, Italy, pp. 81–86 (2012)
14. Fischer, J.E., Bland, K.I.: Mastery on Surgery, vol. 1. Lippincott Williams & Wilkins (2007)
15. Bauzano, E., Muñoz, V.F., Garcia-Morales, I.: A Multi-Behaviour Algorithm for Auto-Guided Movements in Surgeon Assistance. Int. J. Mechanics and Control 12(1), 35–41 (2011)
16. Grasa, O.G., Civera, J., Montiel, J.M.M.: EKF monocular SLAM with relocalization for laparoscopic sequences. In: IEEE Int. Conf. Robotics and Automation (ICRA), pp. 4816–4821 (2011)
17. Enxiu, S., Tao, C., Changlin, H., Enxiu, S., Junjie, G.: Study of the New Method for Improving Artificial Potential Field in Mobile Robot Obstacle Avoidance. In: IEEE Int. Conf. Automation and Logistics, pp. 282–286 (2007)
18. Estebanez, B., Saz-Orozco, P., García-Morales, I., Muñoz, V.F.: Interfaz Multimodal para un Asistente Robótico Quirúrgico: Uso de Reconocimiento de Maniobras Quirúrgicas. Revista Iberoamericana de Automática e Informática Industrial (RIAI) 8(2), 24–34 (2011)
19. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications on Speech Recognition. Proceedings of the IEEE 77(2), 257–286 (1989)

20. Baum, L.E., et al.: A maximization technique ocurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics 41(1), 164–171 (1970)
21. Muñoz, V.F., Garcia-Morales, I., Perez-DelPulgar, C., Gomez-DeGabriel, J.M., Fernandez-Lozano, J., Garcia-Cerezo, A., Vara-Thorbeck, C., Toscano, R.: Control movement scheme based on manipulability concept for a surgical robotic assistant. In: IEEE Int. Conf. Robotics and Automation (ICRA), pp. 245–250 (2006)
22. Bauzano, E., Muñoz, V.F., Garcia-Morales, I., Estebanez, B.: Three-Layer Control for Active Wrists in Robotized Laparoscopic Surgery. In: IEEE Int. Conf. Intelligent Robots and Systems, IROS, pp. 2653–2658 (2009)

Part V

Intelligent Systems and Robotics

Multi Hypotheses Tracking with Nonholonomic Motion Models Using LIDAR Measurements

Jorge Almeida[1] and Vítor Santos[1,2]

[1] Department of Mechanical Engineering, University of Aveiro
[2] IEETA, University of Aveiro
{almeida.j,vitor}@ua.pt

Abstract. This paper presents an implementation of the Multiple Hypothesis Tracking (MHT) algorithm in the Advanced Driver Assistance Systems (ADAS) context. The proposed algorithm uses laser data received from two front mounted sensors on a mobile vehicle. The algorithm was tested with simulated and real world data and shown to obtain a very good performance. Nonholonomic motion models were used to model the movement of road agents instead of the more traditional constant velocity/acceleration models. The use of the nonholonomic motion models allows to obtain not only the linear velocity, but also the steering angle of vehicles, improving this way the future prediction and handling of occlusions. The MHT algorithm possesses some well-known critical disadvantages due to its complexity and computational growth, in this work we circumvent these limitations in order to achieve real time performance in real work conditions.

Keywords: LIDAR, MHT, Nonholonomic.

1 Introduction

Robotic systems are ever more present in our society. As their interaction with humans increases so does the need for robust safety measures. In order to have autonomous systems performing even more meaningful roles, their capability to perceive and understand our environment must improve.

In this context, tracking algorithms are a basic tool in any situation awareness system. The need to assess the behavior of other participants is paramount to any decision making algorithm. Tracking algorithms appear in a wide variety of contexts: vision or laser-based people tracking for mobile robotics and surveillance systems, sonar-based submarine tracking, tracking of animals to study their behavior, etc. [1].

The purpose of Multi Target Tracking (MTT) is to estimate the state of multiple different targets recursively. One of the most significant task in a tracking algorithm is the data association. In this step new unlabeled measurements need to be associated with previous detected targets.

This paper presents a solution to the data association problem based on the well known Multiple Hypothesis Tracking (MHT) algorithm. This method was

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_20,

coupled with nonholonomic motion models to estimate targets behavior. Two planar laser sensors provide the range data used to describe targets. Additional details on the test vehicle can be found in [2].

As opposed to other more simplistic methods, the MHT method permits the delay of ambiguous association decisions until additional data relives the ambiguity. MHT is widely regarded as an important data association method in the tracking community due to its probabilistically handling of parallel hypotheses, whereas most competing techniques are suboptimal in nature [3].

The algorithm, introduced in [4], creates a set of possible association hypotheses for each target at each time frame. It exhaustively enumerates all possible combinations creating an exponentially growing hypotheses tree. The creation of every possible combination, instead of just the best combination, allows the algorithm to delay the association in ambiguous cases.

The MHT algorithm can be implemented in several different fashions as presented in [3]: hypotheses oriented, track oriented, multidimensional (or multiframe) assignment method, and Bayesian MHT. This work implements the hypotheses oriented MHT as the method of choice.

In this method each hypothesis presents a different interpretation for all past measurements, consisting of a set of non-conflicting disjoint tracks [1]. These hypotheses are created recursively from previous hypotheses as new measurements are received. These hypotheses can be visualized in a hypotheses tree, where each node represents a hypothesis.

The MHT algorithm, initially proposed for radar applications [5–7], has been previously used in applications in the Advanced Driver Assistance Systems (ADAS) context with various purposes. In [8, 9] the authors implement the MHT in a tracking application for a mobile vehicle. The authors use the multi hypotheses approach to compute several possible classification results. The different hypotheses allowed to better interpreter the clusters created with the laser data. In their application the MHT algorithm is not used to perform data association.

The algorithm is also commonly applied to pedestrian detection and tracking. In [10] a redefinition of the probabilities hypotheses in proposed. The authors explicitly model track occlusion to reflect the fact that legs frequently occlude each other. In [11, 12] the authors reformulate the MHT algorithm to handle information from multiple robots.

This paper introduces in section 2 the proposed implementation of the algorithm. In section 3 the results of the trials with synthetic and real data are presented. Section 4 presents a discussion on the main issues and section 5 reports the final conclusions.

2 Multiple Hypothesis Tracking

2.1 Overview

The proposed algorithm starts by checking the valid possible associations between the current measurements and the existing targets. The measurements are checked against the predicted positions of existing targets using a Mahalanobis

distance gate. Nonholonomic motion models are used to predict the targets positions. Valid possible associations are used to assign measurements to clusters (section 2.3).

Each cluster is treated as an independent data association problem. The likelihoods of valid associations between measurements and targets create ambiguity matrices. The Murty's linear assignment algorithm [13] is then used to create the k-best possible assignment hypotheses.

Out of all created hypotheses only a subset is propagated for each cluster limiting in this fashion the grow of the hypotheses tree.

The output targets correspond to the targets in the most likely hypothesis in each cluster.

2.2 Hypotheses Probabilities

As proposed in [14] and presented in [10] the probability of each individual j hypothesis Ω_j^k given the new measurements z_k, at iteration k, can be calculated using the probability of the assignment set $\Psi_j(k)$ and the parent hypothesis $\Omega_{p(j)}^{k-1}$, denoted by the index $p(j)$, as:

$$p\left(\Omega_j^k|z_k\right) = p\left(\Psi_j(k), \Omega_{p(j)}^{k-1}|z_k\right) \tag{1}$$

Applying the Bayes' rule yields:

$$\begin{aligned} p\left(\Omega_j^k|z_k\right) = \eta\, p\left(z_k|\Psi_j(k), \Omega_{p(j)}^{k-1}\right)\; p\left(\Psi_j(k)|\Omega_{p(j)}^{k-1}\right) \cdot \\ p\left(\Omega_{p(j)}^{k-1}\right) \end{aligned} \tag{2}$$

The first term in the right-hand side, η, is a normalizer that ensures that all the probabilities of hypotheses in a cluster add up to 1. The next term is the measurement likelihood. Assuming a Gaussian pdf for a measurement associated with a target ($\mathcal{N}(z_k^i)^{\delta_i}$, $\delta_i = 1$ if the association is true, $\delta_i = 0$ otherwise), and a uniform association probability for a new track over the observation volume V, the measurement likelihood is calculated for all M_k measurements as:

$$p\left(z_k|\Psi_j(k), \Omega_{p(j)}^{k-1}\right) = \prod_{i=1}^{M_k} \mathcal{N}(z_k^i)^{\delta_i} V^{-(1-\delta_i)} \tag{3}$$

The center right term of equation (2) is the probability of an assignment set, $p\left(\Psi_j(k)|\Omega_{p(j)}^{k-1}\right)$. This probability is dependent on the probability of the number of targets with a certain label, the probability of a specific distribution of measurement assignments, and the probability of a specific distribution of target assignments.

The final term of equation (2) is the recursive term, the probability of the parent hypothesis.

The combination of all these probabilities wields a simplified version of the probability of a single hypothesis, equation (4), given that many terms cancel out. The final probability is independent of the observation volume V. The detailed deduction can be consulted in [10].

$$p\left(\Omega_j^k|z_k\right) = \eta'' \prod_{i=1}^{M_k} \mathcal{N}(z_k^i)^{\delta_i} \cdot \\ p_{det}^{N_{det}}\ p_{occ}^{N_{occ}}\ p_{del}^{N_{del}}\ \lambda_{new}^{N_{new}}\ \lambda_{fal}^{N_{fal}} \cdot \\ p\left(\Omega_j^{k-1}|z_{k-1}\right) \tag{4}$$

2.3 Clusters

The MHT implementation here presented revolves around the notion of cluster. Although introduced by Reid [4], the idea of clustering hypotheses is not usually discussed nor implemented. A cluster arises from the need to deal with conflicting targets, targets that share the possibility of association with one measurement. This possibility indicates that a valid association between each target and the measurement can occur. When two targets share a measurement the data association must contemplate the fact that only one of the targets can actually be associated with that measurement, in practice these two targets must be evaluated together. In this simplistic case two different valid associations can be made, each resulting association is now treated as a hypothesis. In a more complex case, with several conflicting targets and measurements, a hypothesis is a valid association of all targets with all measurements, different valid associations create different hypotheses within the same cluster (figure 1).

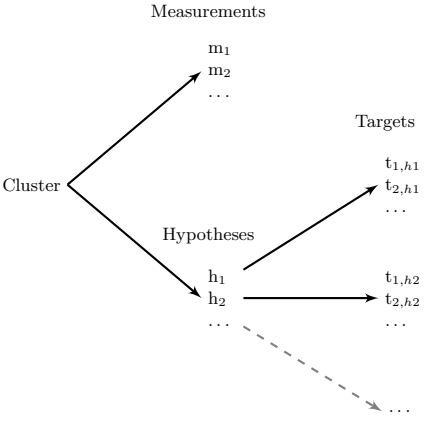


Fig. 1. Cluster composition. A cluster is composed of a set of measurements and hypotheses, each hypothesis has a variable number of targets.

2.4 Hypotheses Creation

The first step of the algorithm is to assign current measurements to existing clusters. Once all clusters are updated new hypotheses can be created. The clusters are solved separately and in parallel given that different clusters do not share conflicting targets. Each cluster is composed of a set of measurements and a set of hypotheses.

The purpose of this step is to associate the measurements to the targets of the cluster. Each hypothesis presents a different set of targets that can be associated with the measurements. The *measurement-to-track* association may have several possible solutions, creating this way the ambiguity problem. Instead of enumerating all possible combinations, the proposed implementation uses the Murty's algorithm to generate the k-best possible associations, making this way feasible the implementation of this algorithm. Of the complete set of k-best associations, in each cluster, only a small group is propagated according to a minimum representativity rule.

In each cluster, for each hypothesis, an ambiguity matrix is created. This matrix expresses the association likelihood between each target and each measurement (table 1). These probabilities are calculated from the bivariate normal distribution of the target position at the measurement position. Take notice that the probabilities in this matrix do not necessarily add up to 1 when adding columns or rows.

Table 1. Example ambiguity matrix. The ambiguity matrix expresses the probability of association between each measurement and each target.

	$\mathbf{t}_1$	$\mathbf{t}_2$	$\mathbf{t}_3$
$\mathbf{m}_1$	0.8	0.2	0.1
$\mathbf{m}_2$	0.3	0.1	0.9

From the ambiguity matrix all the possible associations can be extracted. In order to obtain these associations in an efficient manner Murty's algorithm is implemented.

2.5 Murty Algorithm

The Murty's algorithm introduced in [13], determines the k-best assignments in a linear assignment problem in polynomial time.

The algorithm, as described in [15], starts by finding the single most probable association by interpreting the problem as a weighted bipartite matching problem. A bipartite graph is created with the targets on one side, measurements on the other, and the negative log likelihood of each association as the arcs connecting them. To solve this classic assignment problem the Hungarian method was employed [16]. From this solution the Murty's algorithm partitions the main

problem into a list of new problems. These new problems follow two rules: first there are no duplicated problems, and second the union of the sets of solutions to these new problems contains all solutions for the main problem except the solution already calculated.

The k-best algorithm obtains the solutions iteratively. The best solutions is found and removed from the problem by replacing the problem with its partition. The best solution of the partition is found and the same methodology is applied, then algorithm continues until k solutions are obtained or there are no more valid solutions.

The Murty's algorithm termination criteria may be modified to better suit the MHT problem [10]. Given the formulation the Murty's algorithm, it provides an ordered list of solutions, from the best to the worst. We can set a lower limit that would stop the search for additional solutions by analyzing the probability of each solution as they are created. Another hypothesis would be to examine the representativity of all the solutions found so far, setting a minimum required value as the threshold.

The output of the assignment step is a list of possible children hypotheses for each cluster. After assignment, the targets in each hypothesis are labeled as one of the following: detected, occluded, deleted, or new.

2.6 Hypotheses Propagation

Each of the child hypotheses has a probability that is calculated using equation (4).

The propagation of all these hypotheses would cause the exponential growth of the tree even with the application of the Murty's algorithm. Each hypothesis in a cluster would create k new hypotheses at each iteration.

To avoid this growth another pruning strategy is applied. For each cluster only a subset j of the best hypotheses is propagated.

The first step to obtain this subset is to normalize the probabilities of all the new hypotheses, the sum of the probabilities of all new hypotheses inside a cluster must be 1. After normalization the best j hypotheses are appended to their respective parent hypothesis in the cluster. Hypotheses from the previous iteration that have no new children are considered dead.

In this step an additional stopping criterion was applied. If the sum of the probabilities of the hypotheses added so far account for a minimum representativity of 95%, no more hypotheses are added.

2.7 Motion Models

To predict the future positions of targets and reduce the search space, motion models are used [17]. In this implementation a nonholonomic model is proposed. This model, obtained from [18] and previously used in [19], incorporates the constraints that dominate the motion of a car-like vehicle contrary to the most typically used models [20].

The model consists of five state variables (5): the global x_p and y_p, linear velocity v_l, global vehicle orientation θ and steering angle φ. The global reference system corresponds to the mission start position.

$$x = [x_p, \ \ y_p, \ \ v_l, \ \ \theta, \ \ \varphi]^T \tag{5}$$

The model is obtained as follows:

$$\begin{aligned} \dot{x_p} &= \cos(\theta) \times \cos(\varphi) \times v_l \\ \dot{y_p} &= \sin(\theta) \times \cos(\varphi) \times v_l \\ \dot{v_l} &= 0 \\ \dot{\theta} &= \sin(\varphi) \times v_l / l \\ \dot{\varphi} &= 0 \end{aligned} \tag{6}$$

This model undergoes discretization to be used in a Extended Kalman Filter (EKF) framework. The variables x_p, y_p and θ are inputted as the filter measurements from the data. The filter parameter matrices were experimentally tuned to achieve a good performance.

The use of such a model has some important benefits over more traditional models. This model allows to obtain not only the linear velocity, but also the steering angle of the vehicles, this is of extreme importance given that it allows for large occlusions in maneuvering situations.

3 Results

Two main sets of experiments were conduced. A set with simulated data and a second experiment with real world urban data.

The first set allowed to test the data association step with large amounts of data. The simulated data provided a straightforward ground truth which easily allowed the quantification of the total number of errors. This quantification enabled the comparison of different parameterizations of the algorithm in both association performance and computational cost.

The second experiment allowed to test the algorithm with real world data. The data consisted of a key situation for ground vehicles safety and autonomy, namely roundabouts. This trial is especially important due to the very high complexity which limits the application of simpler algorithms.

3.1 k-j Parameterization

The MHT algorithm is heavily dependent on the hypotheses creation parameterization for both tracking and computational performance. The increase in the allowed number of hypotheses should, in theory, increase tracking performance but substantially increase the computational load.

This experiment evaluates the tracking performance dependency on the k and j parameters. Since the two parameters are heavily interconnected a simultaneous analyses of both was performed.

The experiment consisted of a single set of 10 trials tested with various k-j configurations, ranging from *1-1* to *10-10*. Each trial consisted of a set of 30 targets moving in linear trajectories with similar speeds (figure 2). These targets started in different normally spaced apart positions ($x = 0$ for all targets and y with incremental spacing with distribution $\mathcal{N}(0.66, 0.2^2)$) but crossed each other while moving. The trajectories orientations were uniformly distributed in the interval $]-\pi/9, \pi/9[$, while the velocities were normally distributed $\mathcal{N}(15.0, 1.0^2)$ with a short acceleration ramp at the beginning. As can be seen in the figure 2, in these trials an error rate of zero is impossible due to their very high complexity.

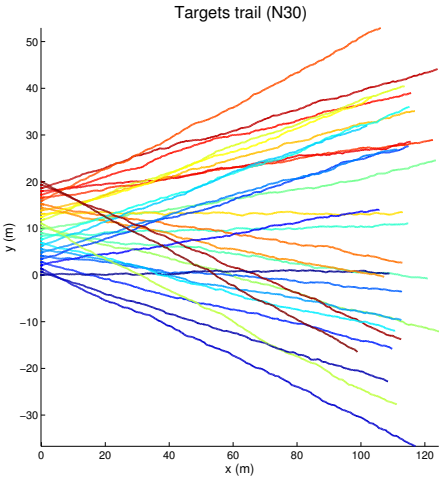


Fig. 2. Simulated targets raw data trails. The targets are color coded for easy distinction. At the start of the trial, on the left side of the graph, the targets are ordered by id.

Figure 3 presents the results of the experiment. Each bar plots the mean percentage of error per target per trial using a total of 10 trials. Take notice that $k > j$ and $j > 1$ when $k = 1$, do not appear because not all combinations of k-j are valid or interesting. For instance, it is useless to extract from a single hypotheses more new hypotheses (k) than the total we will evaluate (j).

As can be observed the worst performance is obtained with *1-1*, 12.1%. A sharp increase in performance is obtained with a small increase in the number of hypotheses, a 28% increase in performance from *1-1* (12.1%) to *2-2* (8.7%).

After the initial dip, the performance stabilizes around 8%, improving to low 7% only with a large number of hypotheses. The best performance, 7.08%, is obtained with *4-10*.

While the performance increase with j is clear, the parameter k does not significantly influence the results.

Additionally a computational cost comparison was performed. Figure 4 presents the mean iteration time for the 10 trials (each with 500 iterations)

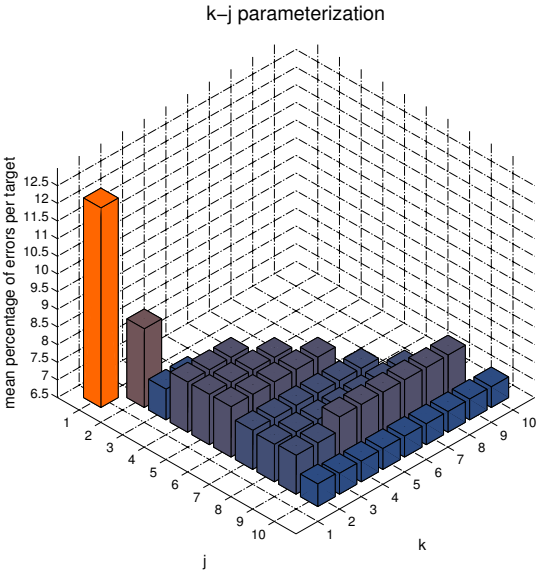


Fig. 3. *k-j* parameterization influence. Each bar presents the results for 10 trials with specific *k-j* values.

with different *k-j* parameterizations. Two different implementations of the algorithm were tested: a single thread implementation and a multi thread implementation. In the multi thread implementation each cluster was processed in a parallel thread. Inside each cluster, the evaluation of each hypothesis was also implemented in parallel.

As expected, with parameterization *1-1*, both implementations have the same and best performance. The performance of each implementation degrades exponentially with the increase of *k-j*. The multi thread implementation presents the best performance, with mean iteration time lower by around 30% (29.8% at *2-2* and 35.8% at *10-10*), compared to the single thread implementation.

The tests were conducted in computer equipped with a dual core processor. In a computer with additional cores, additional performance gains are expected.

3.2 Roundabout Trial

In this trial, real data was provided to the algorithm. The data was obtained in the city of Aveiro-Portugal in a three lane large roundabout (figure 5). The vehicle entered twice in the roundabout, in both occasions the vehicle had to stop due to traffic. A total of 34 different vehicles were encountered and visible in the laser data.

Egomotion was provided by internal sensors in the vehicle as presented in [19].

In order to obtain a set of metrics of the algorithm performance, ground-truth information was needed. The acquired raw laser data was hand labeled, with

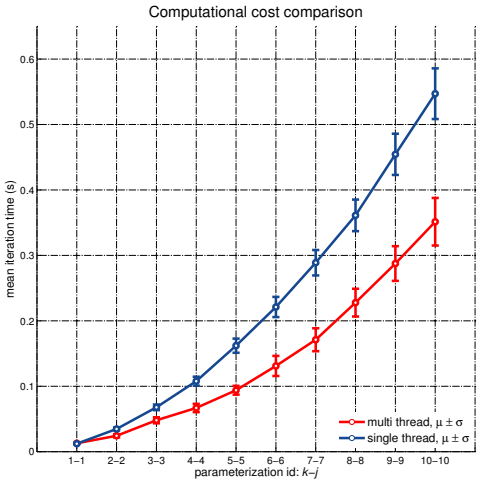


Fig. 4. Mean iteration time of different k-j parameterizations. Each parameterization was tested with 10 trials.

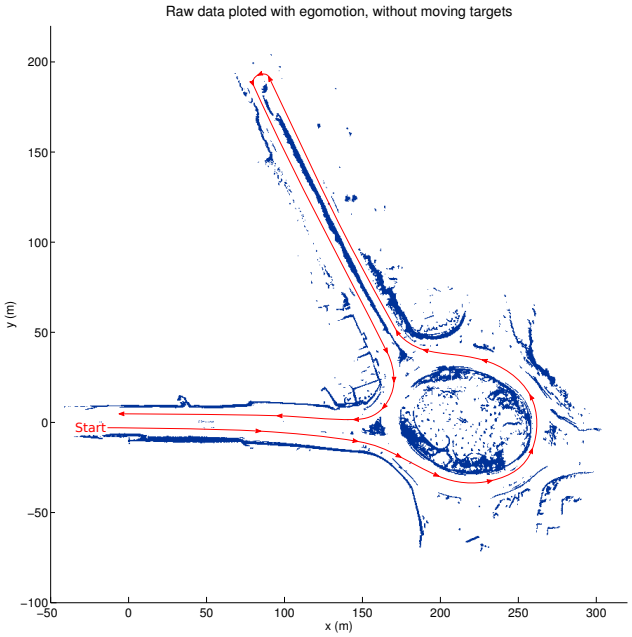


Fig. 5. Raw data acquired in the trial, corrected with egomotion. The moving targets points are not displayed. The red line plots the vehicle path during the trial.

all 34 vehicles segmented and tagged. To take advantage of the nonholonomic motion model proposed, nonholonomic measurements are needed, measurements must correspond to the position of center rear axis and orientation of the vehicle.

This paper focuses on the data association step of the tracking algorithm, the segmentation is out of scope of this paper.

In this trial the total percentage of association errors over all performed associations was 4.22%.

In order to detect the loss of a target while it was still in scene the total created targets were compared to the ground-truth total count, figure 6. The loss of a target most often occurs due to occlusion from other targets and fixed objects. This is particularly critical at longer ranges where a target may only be represented by a single point.

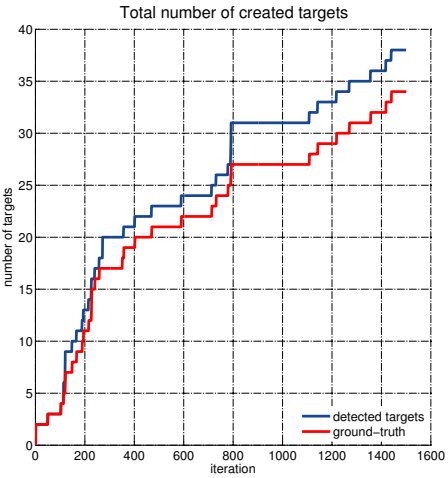


Fig. 6. Accumulated number of targets. The difference between the ground-truth and the detected targets identifies situations were a target was lost and subsequently reinitialized. At the end of the trial the accumulated detected targets were 11.7% higher that the ground-truth.

The distribution of the distance from the Kalman filter estimated position to the measurement data is presented in figure 7. This figure presents only the distance for correct associations. The data indicates that estimation is performed correctly and accurately.

4 Discussion

The trials using simulated data do not allow for an evaluation of the true performance of the algorithm. Nonetheless they allow the comparison of different parameterizations and the influence of difference factors. These comparisons would be very hard to achieve using real data due to the lack of ground truth.

The higher number of available hypotheses typically leads to a better performance, but also increases dramatically the computational load of the algorithm.

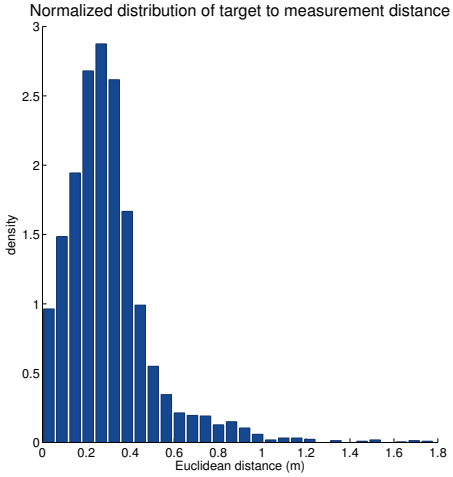


Fig. 7. Distribution of the distance between the estimated position of targets and the hand labeled measurements, excluding bad associations

For instance, at the beginning of the trials all targets belong to the same cluster (due to their proximity), in the trial *1-1* with 30 targets, a single 30×30 cost matrix must be evaluated, but in the *10-10* a total of 10 matrices (30×30) must be evaluated 10 times using the k-best algorithm.

The performance increases due to the higher number of hypotheses is obvious when the number of hypotheses is low, but at higher numbers this performance increase fluctuates and the computational cost oversteps this gain. This performance curve is very hard to extrapolate to the real scenario. The size in the number of hypotheses can be correlated to the targets dynamics, lower dynamics indicate that some hypotheses must be kept during a long time, given that 2 or 3 iterations are not enough to ascertain which one is correct. The use of the *k-j* approach completely limits the exponential growth of hypotheses but has some potential drawbacks. On the short-term, high probability hypotheses may be oversampled completely obfuscating other lower probability valid hypotheses, that on the long run would be better alternatives.

The evaluation of each individual hypothesis is independent of all the others, due to this fact their processing may be performed in parallel. The heavy parallelization of the algorithm is of crucial importance to improve performance in the most complex trials. This parallelization feature is also of extreme importance to allow the implementation in dedicated systems.

The trial with real data allowed to ascertain the true behavior of the algorithm. The algorithm very accurately estimated the total count of targets. This demonstrates the algorithm's ability to deal with occlusions. In this trial several other vehicles moved in close proximity to our vehicle creating large occlusions zones. These zones often caused vehicles to become occluded during large periods of time. The nonholonomic motion model allowed to correctly estimate the

true position of the vehicle until it exited the occlusion zone, preventing this way the reinitialization of its tracking.

5 Conclusions

This work presents a hypotheses oriented implementation of the multiple target MHT algorithm.

The MHT algorithm applies the notion of multiple valid hypotheses to an association problem thus delaying critical decisions that could be proved wrong, to a time when more information relieves the ambiguity. At each iteration, a set of hypotheses expresses the different possible, within gating distance, combination of measurement to track associations as well as the different assumptions on the number of actual tracks and false alarms. The hypotheses clustering allowed the partition of the main problem into independent subsets, both simplifying and improving the computational speed by allowing parallel processing.

The algorithm demonstrated high performance and robustness with both simulated and real data.

Synthetic data was used to evaluate effect of the hypotheses limitation via the *k-j* method. The increase in the total number of hypotheses leads to a initial large increase in performance that quickly stabilized.

The polynomial Murty ranked assignment algorithm was used to replace Reid's original NP-hard exhaustive hypotheses creation, evaluation and branching. The hypotheses limitation and pruning, though the j limit algorithm, completely avoid the exponential growth of the hypotheses tree. This limitation scheme although necessary imposes some important drawbacks that should be addressed.

The algorithm was tested using real world data. The data was obtained in a key situation for road autonomous system safety, namely a large roundabout. The association algorithm performed very well and the use of an advanced motion model allowed to overcome most occlusions, preventing the creation of surplus targets.

References

1. Tinne, D.: Rigorously Bayesian Multitarget Tracking and Localization. PhD thesis, K. U. Leuven, Leuven, Belgium (May 2010)
2. Santos, V., Almeida, J., vila, E., Gameiro, D., Oliveira, M., Pascoal, R., Sabino, R., Stein, P.: ATLASCAR - technologies for a computer assisted driving system on board a common automobile. In: IEEE International Conference on Intelligent Transportation Systems, Funchal, Portugal, pp. 1421–1427 (September 2010)
3. Blackman, S.: Multiple hypothesis tracking for multiple target tracking 19, 5–18 (January 2004)
4. Reid, D.: An algorithm for tracking multiple targets 24, 843–854 (December 1979)
5. Koch, W.: On bayesian MHT for well-separated targets in densely cluttered environment. In: IEEE International Radar Conference, pp. 323–328 (May 1995)

6. Blackman, S., Dempster, R., Busch, M., Popoli, R.: IMM/MHT solution to radar benchmark tracking problem 35, 730–738 (April 1999)
7. Danchick, R., Newnam, G.: Reformulating reid's MHT method with generalised murty k-best ranked linear assignment algorithm. In: IEE Proc. on Radar, Sonar and Navigation, pp. 13–22 (February 2006)
8. Streller, D., Dietmayer, K., Sparbert, J.: Object tracking in traffic scenes with multi-hypothesis approach using laser range images. In: 8th World Congress on Intelligent Transport Systems (2001)
9. Streller, D., Dietmayer, K.: Object tracking and classification using a multiple hypothesis approach. In: IEEE Intelligent Vehicles Symposium, pp. 808–812 (June 2004)
10. Arras, K., Grzonka, S., Luber, M., Burgard, W.: Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In: IEEE International Conference on Robotics and Automation, pp. 1710–1715 (2008)
11. Tsokas, N., Kyriakopoulos, K.: Multi-robot multiple hypothesis tracking for pedestrian tracking. Autonomous Robot 32, 63–79 (2012)
12. Tsokas, N.A., Kyriakopoulos, K.J.: A multiple hypothesis people tracker for teams of mobile robots. In: IEEE International Conference on Robotics and Automation, pp. 446–451 (2010)
13. Murty, K.G.: An algorithm for ranking all the assignments in order of increasing cost. Operations Research 16, 682–687 (1968)
14. Cox, I., Hingorani, S.: An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking 18, 138–150 (February 1996)
15. Cox, I., Hingorani, S.: On finding ranked assignments with application to multitarget tracking and motion correspondence 31, 486–489 (January 1995)
16. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2, 83–97 (1955)
17. Streller, D., Furstenberg, K., Dietmayer, K.: Vehicle and object models for robust tracking in traffic scenes using laser range images. In: IEEE International Conference on Intelligent Transportation Systems, pp. 118–123 (2002)
18. Laumond, J.P.: Robot motion planning and control. Springer (1998)
19. Almeida, J., Santos, V.M.: Real time egomotion of a nonholonomic vehicle using LIDAR measurements. Journal of Field Robotics 30, 129–141 (2013)
20. Schubert, R., Richter, E., Wanielik, G.: Comparison and evaluation of advanced motion models for vehicle tracking. In: International Conference on Information Fusion, pp. 1–6 (2008)

Programming by Demonstration: A Taxonomy of Current Relevant Methods to Teach and Describe New Skills to Robots*

Jordi Bautista-Ballester[1,2], Jaume Vergés-Llahí[1], and Domènec Puig[2]

[1] ATEKNEA Solutions,
Víctor Pradera, 45, 08940 Cornellà de Llobregat, Spain
{jordi.bautista,jaume.verges}@ateknea.com
[2] Department of Computer Engineering and Mathematics,
Universitat Rovira i Virgili, Tarragona 43007, Spain
jordi.bautista@estudiants.urv.cat,
domenec.puig@urv.cat

Abstract. Programming by Demonstration (PbD) covers methods by which a robot learns new skills through human guidance and imitation. PbD has been a key topic in robotics during the last decade that includes the development of robust algorithms for motor control, motor learning, gesture recognition and the visual-motor integration. Nowadays, PbD deals more with learning methods than traditional approaches, and frequently it is referred to as Imitation Learning or Behavioral Cloning. This work will review and analyse existing works in order to create a taxonomy of the elements that constitute the most relevant approaches in this field to date. We intend to establish the categories and types of algorithms involved so far in PbD and describing their advantages and disadvantages and potential developments.

Keywords: Mobile Robotics, Programming by Demonstration (PbD), Imitation Learning, Learning from Demonstration (LfD), Taxonomy.

1 Introduction

Programming by Demonstration (PbD) covers methods by which a robot learns new skills through human guidance and imitation. Also referred to as *imitation learning, lead through teaching, tutelage, or apprenticeship learning*, PbD takes inspiration from the way humans learn new skills by imitation in order to develop methods by which new tasks can be transmitted to robots. In this paradigm, the programmer become an *instructor* where both the decomposition and the programming of a skill is performed through the observation of a demonstration done by the instructor, which can be either a human being or another robot.

* This research has been partially supported by the *Industrial Doctorate* program of the Government of Catalonia.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_21,

1.1 Motivation

The motivation of this paper is connected with our current research which is focused on envisaging the most relevant techniques that allow us to teach and share skills to a group of autonomous outdoor mobile robots called **VinBot**[1].

The purpose of the project is to provide the necessary navigation and behavioral skills to a number of agriculture robots so they can perform tasks such as monitoring the growth of crops and estimating very valuable information such as the yield. These robots will be networked, that is, robots connected to a cloud-based service which will provide the off-board computational resources and all the necessary tools for communication, storage, process and share of the data obtained by the on-board sensors.

Robots in VinBot must be capable of learning certain skills involved in autonomous outdoor navigation, such as the creation of maps of the fields and how to cope with changes in this mutable environment while moving through them in different periods of the time. Also the avoidance of unknown obstacles and potential risks to the integrity of the robot will require learning new skills or, at least, sharing sets of already acquired strategies from other sources. Finally, we need natural ways to specify and control the missions, as well as learning certain tasks to tend a specific crop will also be necessary.

Here the importance of autonomously learning skills from demonstrations is even more evident since the interaction will mostly be performed with users unaware of robot programming who might not even be located in the same place as the robot. Additionally, we intend this newly acquired knowledge can benefit any other robot in the system. This implies that the representation of skills must be adequate both for transferring not only to similar but also to not strictly identical robots, as well as shared and replicated onto a number of units.

Consequently, we intend to employ the PbD paradigm for the tasks of skill learning and transference in the context of networked autonomous mobile robots. As it will be shown in this paper, PbD is a natural approach to deal with both the problems of learning skills from demonstrators and the representation of skills among different robotic *embodiments*. Despite most of the approaches analysed here were usually applied to more human-like platforms, such as humanoids or robotic arms, we also want to investigate what type of approaches best fit our specific mobile robot platform.

1.2 Outline of the Paper

This paper will first review the state of the art in the area of imitation learning techniques, analysing both the defining elements that compose the most relevant approaches and review the **taxonomy** of techniques describing their advantages and disadvantages. The objective of this paper is dual: *summarizing* and *standardizing* the current state of this problem and its constituent *elements*, and

[1] VinBot is an FP7 European project starting in 2014 whose main object is to develop a cloud-based mobile robotic system for agricultural applications.

also analysing the scope of each of them. Such categorization will be useful for a better understanding of the techniques and deciding the most effective methods and posterior research lines in the development of a system using PbD.

2 Programming by Demonstration (PbD)

Since the 80s research in this topic has grown steadily and has become a central topic in robotics. Learning robot skills for complex platforms that interact in complex and variable environments is faced with two key challenges.

First, the complexity of the task to learn is such that learning by only *trial-and-error* would be impractical. In this way, PbD appears as a strategy to speed-up and to facilitate the process of learning by reducing the search space and yet allowing the robot to refine its model of demonstration by trial-and-error. Also PbD permits the robot to incorporate usual tasks by means of a non-specialized instructor.

Second, PbD favours a closer relation between the learning process and control stage, so the latter can be adapted in real time to perturbations and changes that will likely happen in the environment.

In this section we describe the building elements encounter in PbD and organize them in Tables 1, 2, and 3.

2.1 Elements in Programming by Demonstration

The challenges faced by PbD were enumerated in [4] as a set of key questions: *What to imitate? How to imitate? When to imitate? Whom to imitate to?* To date only the first two questions have actually been addressed in PbD. Fig. 1 shows an example of the first two questions.

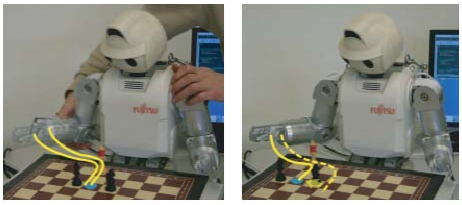


Fig. 1. Observation of multiple demonstrations and reproduction of generalized motion. In this case, extracted from [57], moving the chess queen forward. The robot records its joint's trajectories and learn to extract (what to imitate?) and after that it reproduces the skill (how to imitate).

In a similar manner, Breazeal & Scassellati in [5] went in further detail into the ideas of how, in fact, a robot is able to know what elements to imitate when attempting to learn the movements of a human being, what perceptual aspects

are relevant to this task and how a robot is able distinguish the parts that will be emulated once a relevant action is perceived, as well as the moment the robot must convert this perception into a sequence of motor responses in order to achieve the same result.

According to the work by Argall et al. [6], PbD can be divided in two stages, namely, *collection of examples*, where all the information forming a demonstration is collected, and *derivation of a policy*, also known as *mapping*, where a set of examples are used to define a group of actions able to reproduce the behavior outlined in the demonstration.

What to Imitate: Collection of Examples. In this first stage, a set of informations from the demonstrator, be it a robotic or human, and possibly also from the environment, is collected from the readings of a capturing system. This can be a device mounted either on the demonstrator or on the learner, the commands of a remote control operated by the demonstrator, or a sensor located externally in the environment, like a camera.

Due to the *correspondence problem*, i.e., how to correspond actions in different embodiments and robotic platforms [4] (Fig. 2), in the collection stage we must be aware of the particular structure for both the demonstrator and the robot learner. Consequently, two successive mapping steps are required in order to cope with this problem, namely, *record mapping* and *embodiment mapping*, where the first is a mapping between sensor readings and motor commands and the latter, a mapping between motor commands from the demonstrator's body onto that of the learner.

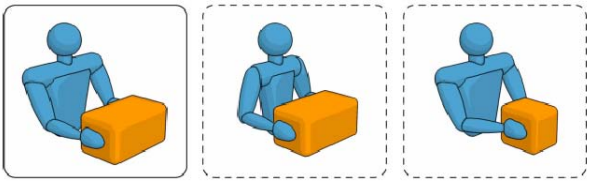


Fig. 2. Illustration of the correspondence problem, extracted from [57]

Hereafter we describe the sort of techniques summarized in Table 1 showing their requirements with respect of record and embodiment mappings. Two main groups of techniques can be separated depending on whether the demonstrator conveys directly the learner. Also the main related works are referred in Table 1.

In teleoperation (Fig.3), the demonstrator operates the robot learner and its sensors capture the motion. No record mapping is necessary since the sensory system is that from the robot learner. In shadowing, the demonstrator shows the task and the robot learner captures the motion with its sensors while attempting to match that of the instructor. In this case, a record mapping is necessary. Both

Table 1. What to imitate? Techniques in the first stage –Collection of Examples according to [6]–. First column tells the method the demonstrator employs to convey the learner. The use of record or embodiment mappings is shown respectively in 3^{rd} and 4^{th} columns. The most extreme cases are *teleoperation*, where no mapping is needed, and the case of *external observation*, where both mappings are required.

Demonstrator Methods	Technique Names	Record Mapping	Embodiment Mapping	Related Works
Robot Learner	Teleoperation	No	No	[7, 9–14]
	Shadowing	Yes	No	[15, 16]
Externally	Sensors on Instructor	No	Yes	[2, 17, 18]
	External Observation	Yes	Yes	[19–22]

in teleoperation and in shadowing no embodiment mapping is needed since the robot captures information directly from its sensors.

In case sensors are on the instructor, record comes directly from these sensors and need no record mapping. With this imitation technique, the demonstrator executes the task and the sensory system information is recorded. If sensor readings come from an external observation, the demonstrator executes the task and the external sensory system records the execution which will be translated by means of a record mapping onto the learner motor commands. In case no sensory device is located on the robot learner, an embodiment mapping is also required both in sensor-on-instructor and external-observation techniques.

Fig. 3. Teleoperation is one of the most common way for wheeled mobile robots. In this case, extracted from [58], a Pioneer 3-DX robot is teleoperated by a Phantom Premium haptic device.

How to Imitate: Derivation of a Policy. The second stage consists in executing a group of actions that allows the robot to reproduce the behavior that was demonstrated from a set of examples. The following are the three most common approaches: 1) learning a function mapping states to actions, 2) learning a model of the world dynamics, or 3) using a planner that produces the sequence of actions after learning the model of an action.

We relate the most interesting works corresponding to this section in Table 2, classifying them with respect to the learning approach utilized.

Table 2. How to imitate? Methods in the second stage –Deriving a Policy according to [6]–. Most of the works in PbD are within the *Mapping Function* category, either in Classification or in Regression.

Approaches	Learning Techniques		Related Works
Mapping Function	Classification	Low Level Robot Actions	[23, 24, 29]
		Robot Movement Primitives	[7, 19, 25, 28]
		High Level Behaviors	[23, 26, 27]
	Regression (Mapping Function Approx)	At Run Time	[13, 32]
		Prior Run Time	[28, 33, 34]
		Prior Execution Time	[14, 17, 20, 21]
System Models	Reward Based Learning	Engineering Reward Function	[35, 36]
		Learned Reward Function	[9, 22, 37, 38]
Plans	Using a Planner		[12, 39]

In the first approach, learning a mapping function, the algorithms can be grouped in two different families depending on whether the output is discrete –*classification*– or continuous –*regression*–.

In the second approach, learning a model of the world dynamics, a reward function is maximized. It can be a user-defined or learned in optimization process. It is typically formulated within the framework of Reinforcement Learning (RL).

In the third approach, goal executions can be represented as plans. Therefore, the planning framework represents a policy as a sequence of actions leading from an initial state to the target state.

2.2 Performance of Programming by Demonstration

Recent studies demonstrate that the performance achieved by the different exposed techniques may vary substantially. Different metrics to compute the imitation performance for each proposed approach can be used as in [41], where several metrics were utilized to evaluate a reproduction attempt with respect to the set of demonstrations (e.g. Root Mean Square Error (RMSE), Norm of Jerk, Learning Time and Retrieval Duration).

Argall et al. [6] also exposes the limitations of the dataset provided in the demonstration. The first limitation is caused by the dataset sparsity, which can occur when the demonstrator cannot demonstrate all possible states –*underdemonstrated states*–. To deal with this problem, [8, 39] proposed a generalization

from existing demonstrations and [10, 14, 23], an acquisition of new demonstrations. The second, by the poor quality of a set of examples, which can happen whenever the instructor's demonstrations is ambiguous, unsuccessful or suboptimal in certain areas of the state space.

Different solutions were proposed to improve demonstration data when the demonstration is suboptimal or ambiguous [7, 11, 18, 23, 40] and attempts to learn from experience by means of feedback from the demonstrator or a reward function were tried, as in RL [8, 13, 17, 39, 40]. Table 3 shows main works dealing with the causes of low performance.

Table 3. Evaluation of the Apprenticeship. Two main reasons were identified in [6] as the cause of low performance: underdemonstrated states and poor quality data. We show some of the works dealing with this problems and the approaches followed.

Reasons	Approaches	Related Works
Underdemonstrated State	Generalization from Existing Demonstration	[8, 39]
	Acquisition of New Demonstrations	[10, 14, 23]
Poor quality data	Suboptimal or Ambiguous Demonstrations	[7, 11, 18, 23, 40]
	Learning from Experience	[8, 13, 17, 39, 40]

3 Taxonomy of Programming by Demonstration

Many approaches were proposed in relation with PbD. In this section we consider the most relevant ones, as enumerated in Table 4, and we intend to describe the advantages and drawbacks of each of them.

A great majority of the approaches in PbD mainly focuses on the spacial position and velocity of the end effector or the joint angles. First attempts depended on an explicit temporal indexing and virtually operated in an open loop. Main drawback of these techniques is that time dependence makes them very sensitive to both temporal and spatial perturbations.

To compensate these shortcomings, a heuristic to re-index the new trajectory in time is required, while simultaneously optimizing a measure of how well the new trajectory follows the objective one. This heuristic search is highly task-dependent and non-trivial and becomes less intuitive in high-dimensional spaces. One of these time-dependent approaches [46] employs Expectation Maximization (EM) and an extended Kalman's filter to follow a given trajectory. This algorithm also learns a local model of the robot's dynamics along the chosen trajectory.

EM was also used in [44] to optimize a Gaussian Mixture Model (GMM) for the estimation of the parameters of existing models. In order to find a statistical noise-free estimation of the dynamical model several approaches using either

Table 4. Approaches on Programming by Demonstration. There are two main sets of applications where PbD has been applied. For each set, several metrics were used to evaluate the performance of the techniques. Main approaches are related to their corresponding reference.

Applications	Metrics	Approaches	Related Works
Robot 3D position and velocity coordinates of end-effector and/or joint angles	RMS, RMSE, Learning Time, Norm of Jerk, Retrieval Duration, Max/Rel. Likelyhood, Stability, External Reward	HMM+GMR	[41]
		DMP	[42]
		MoMP	[3]
		BM	[30]
		SEDS	[29]
		RL	[37]
Robot position, orientation and velocity. Obstacles distances and discrete actions	Task Performance, Instructor Evaluation, Max. Reward Function	IRL	[9]
		RL	[49]
			[52]

Gaussian Process Regression (GPR) [31], Locally Weighted Projection Regression (LWPR) [33] or Gaussian Mixture Regression (GMR) [41] were proposed. GMR and GPR find a locally optimal model of the function by maximizing the likelihood for a complete model to fit the data, while LWPR minimizes the RMSE between the estimates and the data. One of the main drawbacks of these approaches is that they cannot guarantee a stable estimate of the motion since no stability constraint is forced near the optimization attractor point.

Dynamic Movement Primitives (DMP) [45], originally proposed by Ijspeert et al. [2], offer a method by which a non-linear dynamical model can be estimated while ensuring global stability at the optimization attractor point, that is, robustness and precision encoding of complex dynamics. DMP is also robust against perturbations and allows changing parameters of the trajectory without altering the overall shape of the movement. These models are straightforward learned by imitation and well suited for reward-driven self-improvement. MoMP is an extension of these models recently proposed by Mülling et al. [3] to cope with complex motor tasks requiring several movement primitives. MoMP creates a framework based on the idea that complex motor tasks can frequently be solved using a relatively small number of movement primitives and do not require a complex monolithic approach to solely cope with an entire task.

A different robust approach complementary to DMP is that of SEDS by Khansari et al. [29], which intends to ensure time-independent learning generalized dynamics from multiple demonstrations. SEDS also outperforms BM, previously proposed by Khansari et al.[30], in that it ensures globally asymptotic stability instead of local stability and can generalize better the motion for trajectories far from those in the demonstrations. BM is more accurate, offers more flexibility and ensures the motion is locally stable. SEDS is more constraint

because it fits a motion with a single globally stable dynamics. SEDS and DMP are complementary in the following way. DMP must be used whenever a motion is intrinsically time-dependent and only one single demonstration is available. In contrast, when the motion is time-independent and multiple demonstrations are available, SEDS would be the choice. A third time-independent approach based on HMM and GMR is proposed in [41]. This method evaluates the eigenvalues of each linear dynamical system and ensures that all of them have negative real parts (stable). Nevertheless, asymptotic stability is not guaranteed.

Considering objectives like position, orientation, and velocity of the robot, distances to the obstacles and a discrete set of actions, most of the techniques are *reward-based*, that is, a known reward function to guide the exploration is assumed [49–53]. A robot's policy will then consist in choosing actions that maximize an expected reward function. In order to avoid the former selection, Inverse Reinforcement Learning (IRL) [9] offers a framework to automatically determine the reward and discover the optimal control policy using a discrete state action space. Alternative approaches derive a cost function in a continuous space [22, 55].

Recent IRL works consider multiple experts and identify multiple reward functions [56]. The goal here is this multiplicity of policies will make the controller more robust by offering alternative ways to complete the task whenever the context no longer allows the robot to perform the task in the optimal way.

More recently, Grollman et al. [54] proposed an approach based on learning from a set of failures where the evaluation of metrics is a binary value representing the success of the learned task. This work offers an interesting alternative to approaches that combine IL and RL since no reward function needs to be explicitly determined.

4 Discussion

As seen in the previous sections of the paper, selection of a certain type of PbD approach is determined by a number of steps we must take into account. First of all, we must consider what information from the instructor during a demonstration will be recorded and employed to teach our robot learner. Choosing a certain approach will be determined, let's say, by the possibility or desire of using the capturing system directly on the instructor or instead obtaining the information from the images of a set of cameras looking at the scene. Several possibilities are available and the actual scenario of the task to learn will determine to a great point the most proper way to capture the movements.

The creation of a set of policies that will control the robot is a question of both the type of information we obtain from the sensors and the quality of such data, as well as the algorithm to perform the learning process. Data can be continuous or discrete, but also have different levels of quality, from motor commands to higher semantic behavioral levels, like in *'bring an apple from the kitchen'*. On the other hand, considerations about the construction of the learning algorithms –p.e., the mapping functions– and at what precise moment the resulting actions will be required also determine the approach to chose.

A series of drawbacks must be taken into account in designing the procedure for PbD. Since the learning process usually consists in iterative algorithms which will adjust the results according to some objective function until a goal is met, it is important to keep in mind several measures of performance in order to obtain an adequate solution, such as the total time and convergence speed of a particular algorithm. Also the robustness to perturbations and precision of the solutions obtained in the demonstrations and the possibility to include new examples to the learning process. In this aspect, stability is a key element for most current approaches. However, sometimes stability can only be maintained locally, while other times it can be globally. Two of the most important aspects that generate low performance are under-demonstrated states, which turn into poor generalizations and problems when incorporating new data, and data of poor quality from ambiguous and not relevant demonstrations.

Looking at the taxonomy of the most relevant current approaches to PbD, we can appreciate that the two important steps to resolve are the measure of the performance of the result and the procedure to model the dynamic of the system with respect to the learning algorithm. For the first, most approaches use norms that measures the discrepancy with respect to the ideal demonstration provided by the instructor. The advantage of this approach is that we can rely on the instructor experience and goodwill, but the dependence on the instructor might also limit the overall performance of the system as the instructor become less competent or tasks grow in complexity and a correct demonstrations are more difficult to obtain. For the second problem, the main limitation is the scope of the task, i.e., a combination of primitives that can be learned independently and generalized or a longer scope task engulfing more extensive behaviors. This might compromise not only the stability of the solution but also the codification of the task into primitive ones, if we think of large differences between instructor and learner embodiments.

With respect to the future of PbD, Cangelosi et al. [48] proposed a *roadmap* of action learning research starting from 2010 till following 20 years. Until 2012, PbD was focused on how to solve *action learning*, where simplest actions or movements are considered, intended as complete motor primitives. Presently, we are on-going the second milestone in the roadmap, which refers to the flexible acquisition of action patterns and their combination to achieve more complex goals. For further details please refer to the work by Karaoguz et al. [47] or also to already mentioned Mülling et al. [3], who proposed to cope with complex motor tasks employing several movement primitives.

The acquisition of hierarchical and compositional actions is expected to be solved during the forthcoming years, and by 2016, the association between syntactic constructions and composite actions via social learning is likely to become the main focus of the investigation. On our behalf, future developments of PbD might also consider the semantic content of human commands, which can be found in natural language, but specially in visual content provided by cameras or instructional videos.

5 Conclusions

In this work we showed the categories and types of existing algorithms in PbD and discussed their advantages and drawbacks. We reviewed the state of the art in the area of imitation learning, analyzing both the building elements that compose the most relevant approaches and proposed a taxonomy of techniques describing. A summary and a standardization of the PbD techniques was accomplished in this paper, as well as the usage of each technique. Such categorization is expected to be useful for a better understanding of the PbD approaches and to decide the most effective methods and posterior research lines in the development of systems using PbD.

References

1. Calinon, S., Guenter, F., Billard, A.: On learning, representing, and generalizing a task in a humanoid robot. IEEE Trans. on Systems, Man and Cybernetics 32, 286–298 (2007)
2. Ijspeert, A., Nakanishi, J., Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots. In: IEEE Int. Conf. on Robotics and Automation, ICRA (2002)
3. Mulling, K., Kober, J., Kroemer, O., Peters, J.: Learning to Select and Generalize Striking Movements in Robot Table Tennis. The International Journal of Robotics Research 32(3), 263–279 (2013)
4. Nehaniv, C.L., Dautenhahn, K.: Like me? Measures of correspondence and imitation. Cybernetics and Systems 32(1-2), 11–51 (2001)
5. Breazeal, C., Scassellati, B.: Robots that imitate humans. Trends in Cognitive Sciences 6(11), 481–487 (2002)
6. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A Survey of Robot Learning from Demonstration. Robotics and Autonomous Systems 57(5), 469–483 (2009)
7. Pook, P.K., Ballard, D.H.: Recognizing teleoperated manipulations. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 1993 (1993)
8. Smart, W.D.: Making Reinforcement Learning Work on Real Robots Ph.D. Thesis, Department of Computer Science, Brown University, Providence, RI (2002)
9. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proc. of the 21st Int. Conf. on Machine Learning, ICML 2004 (2004)
10. Chernova, S., Veloso, M.: Multi-thresholded approach to demonstration selection for interactive robot learning. In: Proc. of the 3rd ACM/IEEE Int. Conf. on Human-Robot Interaction, HRI 2008 (2008)
11. Breazeal, C., Berlin, M., Brooks, A., Gray, J., Thomaz, A.L.: Using perspective taking to learn from ambiguous demonstrations The Social Mechanisms of Robot Programming by Demonstration. Robotics and Autonomous Systems 54(5), 385–393 (2006)
12. Rybski, P.E., Yoon, K., Stolarz, J., Veloso, M.: Interactive robot task training through dialog and demonstration. In: Proc. of the 2nd ACM/IEEE Int. Conf. on Human-Robot Interactions, HRI 2007 (2007)
13. Argall, B., Browning, B., Veloso, M.: Learning from demonstration with the critique of a human teacher. In: Proc. of the 2nd ACM/IEEE Int. Conf. on Human-Robot Interactions, HRI 2007 (2007)

14. Grollman, D.H., Jenkins, O.C.: Dogged learning for robots. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 2007 (2007)
15. Nehmzow, U., Akanyeti, O., Weinrich, C., Kyriacou, T., Billings, S.: Robot programming by demonstration through system identification. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2007 (2007)
16. Ogino, M., Toichi, H., Yoshikawa, Y., Asada, M.: Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping. The Social Mechanisms of Robot Programming by Demonstration, Robotics and Autonomous Systems 54(5), 414–418 (2006)
17. Calinon, S., Billard, A.: Incremental learning of gestures by imitation in a humanoid robot. In: Proc. of the 2nd ACM/IEEE Int. Conf. on Human Robot Interactions, HRI 2007 (2007)
18. Aleotti, J., Caselli, S.: Robust trajectory learning and approximation for robot programming by demonstration. The Social Mechanisms of Robot Programming by Demonstration, Robotics and Autonomous Systems 54(5), 409–413 (2006)
19. Billard, A., Mataric, M.: Learning human arm movements by imitation: Evaluation of biologically inspired connectionist architecture. Robotics and Autonomous Systems 37(2-3), 145–160 (2001)
20. Ude, A., Atkeson, C.G., Riley, M.: Programming full-body movements for humanoid robots by observation. Robotics and Autonomous Systems 47, 93–108 (2004)
21. Steil, J., Rthling, F., Haschke, R., Ritter, H.: Situated robot learning for multimodal instruction and imitation of grasping. Robot Learning by Demonstration, Robotics and Autonomous Systems 2-3(47), 129–141 (2004)
22. Ratliff, N., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Proc. of the 23rd Int. Conf. on Machine Learning, ICML 2006 (2006)
23. Chernova, S., Veloso, M.: Confidence-based learning from demonstration using Gaussian Mixture Models. In: Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2007 (2007)
24. Saunders, J., Nehaniv, C.L., Dautenhahn, K.: Teaching robots by moulding behavior and scaffolding the environment. In: Proc. of the 1st ACM/IEEE Int. Conf. on Human Robot Interactions, HRI 2006 (2006)
25. Kober, J., Peters, J.: Imitation and Reinforcement Learning. Practical Learning Algorithms for Motor Primitives in Robotics 17(2), 1–8 (2010)
26. Rybski, P.E., Voyles, R.M.: Interactive task training of a mobile robot through human gesture recognition. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 1999 (1999)
27. Lockerd, A., Breazeal, C.: Tutelage and socially guided robot learning. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2004 (2004)
28. Calinon, S., Sauser, E.L., Billard, A.G., Caldwell, D.G.: Evaluation of a Probabilistic Approach to Learn and Reproduce Gestures by Imitation. In: IEEE Int. Conf. on Robotics and Automation (x), pp. 2671–2676 (May 2010)
29. Khansari-Zadeh, S.M., Billard, A.: Learning Stable Nonlinear Dynamical Systems with Gaussian Mixture Models. IEEE Trans. on Robotics 27(5), 943–957 (2011)
30. Khansari-Zadeh, S.M., Billard, A.: BM: Aniterative algorithm to learn stable nonlinear dynamical systems with Gaussian Mixture Models. In: Proc. Int. Conf. Robotics and Automation, pp. 2381–2388 (2010)
31. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. Springer, New York (2006)

32. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning rhythmic movements by demonstration using nonlinear oscillators. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2002 (2002)
33. Vijayakumar, S., Schaal, S.: Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space. In: Proc. of the 17th Int. Conf. on Machine Learning, ICML 2000 (2000)
34. Grollman, D.H., Jenkins, O.C.: Sparse incremental learning for interactive robot control policy estimation. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA 2008 (2008)
35. Chersi, F.: Learning Through Imitation: a Biological Approach to Robotics. IEEE Trans. on Autonomous Mental Development 4(3), 204–214 (2012)
36. Merrick, K.: Intrinsic Motivation and Introspection in Reinforcement Learning 4(4), 315–329 (2012)
37. Guenter, F., Billard, A.: Using reinforcement learning to adapt an imitation task. In: Proc. of the IEEE/RSJ Int. Conf.onIntelligent Robots and Systems, IROS 2007 (2007)
38. Abbeel, P., Coates, A., Quigley, M., Ng, A.Y.: An application of reinforcement learning to aerobatic helicopter flight. In: Proc. of the Advances in Neural Information Processing, NIPS 2007 (2007)
39. Nicolescu, M.N., Mataric, M.J.: Methods for robot task learning: Demonstrations, generalization and practice. In: Proc. of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2003 (2003)
40. Chernova, S., Veloso, M.: Learning equivalent action choices from demonstration. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2008 (2008)
41. Calinon, S., Dhalluin, F., Sauser, E.L., Caldwell, D.G., Billard, A.G.: Learning and Reproduction of Gestures by Imitation. An Approach Based on Hidden Markov Model and Gaussian Mixture Regression 17(2), 44–54 (2010)
42. Ijspeert, A., Nakanishi, J., Schaal, S.: Trajectory formation for imitation with nonlinear dynamical systems. In: Proc. IEEE Intl Conf. on Intelligent Robots and Systems (IROS), pp. 752–757 (2001)
43. Vijayakumar, S., Dsouza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation 17(12), 2602–2634 (2005)
44. Dempster, A., Rubin, N.L.D.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Statist. Soc. B 39(1), 1–38 (1977)
45. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: Proc. Int. Conf. Robotics and Automation, pp. 1293–1298 (2009)
46. Coates, A., Abbeel, P., Ng, A.Y.: Learning for Control from Multiple Demonstrations. In: Proc. of the 25th Int. Conf. on Machine Learning (ICML 2008), pp. 144–151 (2008)
47. Karaoguz, C., Rodemann, T., Wrede, B., Goerick, C.: Learning Information Acquisition for Multitasking Scenarios in Dynamic Environments. IEEE Trans. on Autonomous Mental Development 5(1), 46–61 (2013)
48. Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., Tani, J., et al.: Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics. IEEE Trans. on Auton. Mental Develop. 2(3), 167–195 (2010)
49. Nicolescu, M.N., Jenkins, O.C., Olenderski, F.: Learning behavior fusion from demonstration. Interaction Studies 9(2), 319–352 (2008)
50. Grollman, D.H., Jenkins, O.C.: Incremental Learning of Subtasks from Unsegmented Demonstration. In: Int. Conf. on Intelligent Robots and Systems (2008)

51. Chernova, S., Veloso, M.: Interactive Policy Learning through Confidence-Based Autonomy. Journal of Artificial Intelligence Research 34, 1–25 (2009)
52. Argall, B., Browning, B., Veloso, M.: Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. Robotics and Autonomous Systems 59(3-4), 243–255 (2011)
53. Argall, B., Sauser, E.L., Billard, A.G.: Tactile Guidance for Policy Adaptation. Foundations and Trends in Robotics 1(2), 79–133 (2010)
54. Grollman, D.H., Billard, A.G.: Donut as I do: Learning from failed demonstrations. In: Int. Conf. on Robotics and Automation (2010)
55. Ratliff, N., Ziebart, B., Peterson, K., Bagnell, J.B., Hebert, H.: Inverse Optimal Heuristic Control for Imitation Learning. In: Proc. of the 12th Int. Conf. on Artificial Intelligence and Statistics (2009)
56. Choi, J., Kim, K.: Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In: Advances in Neural Info. Proc. Sys. 25, NIPS 2012 (2012)
57. Billard, A., Calinon, S.: Robot Programming by Demonstration. In: Handbook of Robotics, ch. 59 (2007)
58. Farkhadinov, I., Hwan, J.: A User Study of a Mobile Robot Teleoperation. In: Proc. of the 4th International Conference on Ubiquitous Robotics and Ambient Intelligence (2007)

Collaborative Architecture Design for Automated Deployment and Positioning of Beaconing Robots

G. Martín[1], J. López Martínez[2], S. Aparicio[1], and I. González[2]

[1] ALVAC S.A, A Coruña, Galicia, Spain
proyecto.guillermomartin@gmail.com, sapariciohill@alvac.es
[2] Universidad de Oviedo, Mieres, Asturias, Spain
proyecto.joseluismartinez@gmail.com,
gonzalezaloignacio@uniovi.es

Abstract. The rising number of applications involving service robots demands more specific and custom control strategies. Under those special working conditions on-road signaling and beaconing robots are framed. Control strategies that transcend each individual robot and focus on the whole set are required, instead of treating each one as a single entity. Thus, control methodology - which usually involves both planned control as behavioral based control - will depend not only on each single robot's parameters, but also on the parameter analysis of other group robots, taking them into account within the decision process. This paper describes an approach to a collaborative control architecture that enhances the mapping and positioning tasks of the robot, as well as the functional organization of the whole robot group. That enhancement is achieved by means of mutual recognition between system's robots, the use of specific sensors and the sharing of robot's position data together with its movement parameters and the information about its surroundings.

1 Introduction and State of the Art

Nowadays road beaconing tasks employ signals and cones that must be placed manually on the road by a team of workers, and that must be relocated manually too every time that road works need it. That implies a hazard to the workers health due to the risks derived from the proximity of traffic.

In order to minimize that kind of risk a robotic beaconing system is proposed. It consists on mobile robots that carry the beaconing elements and can be deployed on the road in an automated way. Such a robotic system needs a complex control strategy, and in order to do so a collaborative architecture is proposed as the best choice to manage it.

Collaborative architectures are far from uncommon in today's robotics scene [1]. We can find them in a wide number of devices, ranging from household cleaning equipment (Roomba) or toys (AirDrone) to tracker robots, control appliances and surveillance solutions. The variety of robotic appliances also leads to an increasing need of communication and interoperability among them [2].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_22,

Robot control architectures can be classified in three main types [3]: hierarchical (or deliberative), reactive and hybrid.

Hierarchical architectures use a control structure based on a perception-planning-acting loop scheme. They use high level environment models that allow complex artificial intelligence algorithm designs, but they use to have latency problems when reacting to fast environment changes.

Reactive architectures are based on behaviors, receiving data from the sensor systems and generating control commands based on them. Each one of the different programmed behaviors runs on an isolated layer of the robot's control system. This type of architecture has a high response rate, but limited functionality if there are no environment models available.

Hybrid architectures combine elements from both previous architectures, incorporating a high level planning layer supported by a low level reactive layer. The architecture described in this paper is comprised within this last type, assigning mapping and action planning to the high level layers while reactive behaviors (as the obstacle avoidance) are assigned to the low level layers.

An illustrative example of collaborative architectures is robotic soccer: a competition game in which two robot teams play. Just as in the real life game, robots of the same team communicate and cooperate with each other in order to achieve the game objective: manage a ball through the playing arena and score a goal. This task can be implemented via 3-D simulation or using real robots, taking into account that there are several approach possibilities depending on the architecture employed. There are mainly three possibilities: teams controlled by global vision equipment (a camera is placed over the playing arena and its images are processed by a computer), local vision teams (each robot has its own camera and processes the images by itself) and humanoid teams (biped robots with local vision). The collaborative architecture proposed in this paper uses an approach similar to the one of the local vision teams, in which each robot has its own vision system and share the relevant information retrieved with the rest of the robots of the working group. New technologies and programming approaches involving cooperative architectures are shown at the Robocup event, a robotic contest that takes place every year [4] (Fig. 1).

Another collaborative architecture example is so-called swarm behavior [5]. This is a behavior that can be found in wild animals and has its origin in a set of simple rules which they follow without any apparent centralized coordination. The term "swarm" comes from swarms of insects, although the concept can also be used to refer to the flocks of bird, herds of vertebrates and also to schools of fish. This behavior is applied to robotics in order to solve problems as different as object transportation, task assignation or nano-medical issues. One of the most extended uses of the robotic swarm is the collaborative mapping [6]. Collaborative mapping uses several robots that navigate through the same area while mapping it by means of different types of algorithms (probabilistic approach, triangulation, relative positioning). Using this method, the result is substantially better than the any single map provided by one of the robots. The proposed architecture uses swarm behaviors to perform some collaborative tasks as mapping or obstacle avoidance.

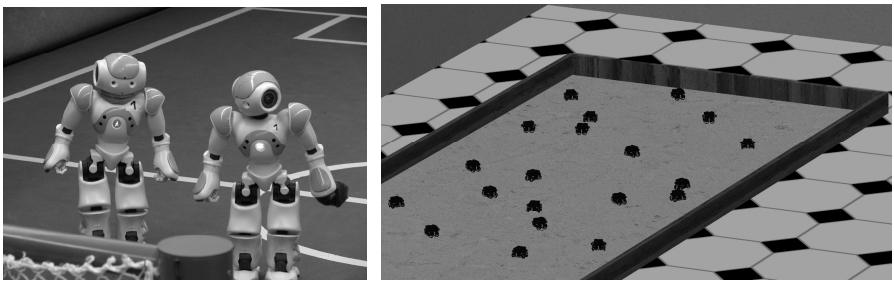


Fig. 1. Robotic soccer and robotic swarm mapping

2 Beaconing System: A Description of the Proposed Collaborative Architecture

Road signaling and beaconing is used for a variety of tasks, ranging from alerting road users of traffic accidents and road works (from roadside weed clearing to asphalt crack sealing), to undertaking road or lane cuts (due to weather or police work, for instance).

The system described in this paper (called RoboCone) consists of a group of autonomous robots, each of them carrying a work cone and a light beacon on top of their structure. The aim of this automated system is to minimize the risks that appear for workers during the tasks of manual deployment and recovering of the signaling and beaconing equipment (most accidents involving road workers occur when sign are being deployed or removed). Each robot can be remotely positioned at a desired position on the road. This automated deployment and subsequent positioning can be performed both manually (a worker controls the process remotely) or automatically (a control system manages the positioning tasks following a previously introduced pattern). The number of robots will depend on the particular requirements of each case, and will stay between ten units (minimum quantity of robots to perform a proper lane cut) and one hundred units (maximum number of robots that the processing equipment can manage).

RoboCone system is designed to work with ROS [7]. ROS (Robotic Operative System) is an open source framework containing libraries and tools specifically designed for robots development and control. This system is based on "nodes" (each node represents a running process) that communicate to each other over TCP/IP using a multiple subscriber/multiple publisher scheme. ROS offers a functional abstraction of the robotic control, regardless of the hardware used, acting as a middleware. The ROS repository includes implementations of algorithms libraries for purposes as different as SLAM systems (Simultaneous Localization and Mapping), auxiliary device drivers (cameras, GPS, gamepad) or 3D models of real robots ready to be used within the Gazebo simulation environment.

RoboCone architecture comprises a node for each cone robot or guide robot belonging to the system. Each node acts as a driver, whose objective is to communicate with the robot both sending as receiving messages. Sent messages are received as operative commands (movement commands, data requests) by the robot,

while the received ones (usually raw sensor data sent by the robot) are processed and re-formatted into a new type of message that is published through the proper channel, called “topic”. This data is used by the action planner, whose mission is to coordinate the robots by sending and receiving information to each node driver.

Along with these nodes, Gazebo 3D simulator also adds an extra node for each robot represented in the simulation. Each one of these new nodes emulates the behavior of the real robot by the use of real physic laws (gravity, collisions, friction…).

Two types of robots participate in the beaconing tasks. The first (called “guide-robot”, Fig. 2) is a robot whose mission is to map the road as it ”drives” along it, retrieving information and details about road condition as well as possible obstacles on it. In order to perform this task, it carries a laser scanner that provides SLAM data [8], which is used together with video cameras that take images of the track. It also carries an Inertial Measurement Unit (IMU) that provides movement information [9] and a GPS system used to get an estimated position of the robot [10][11].

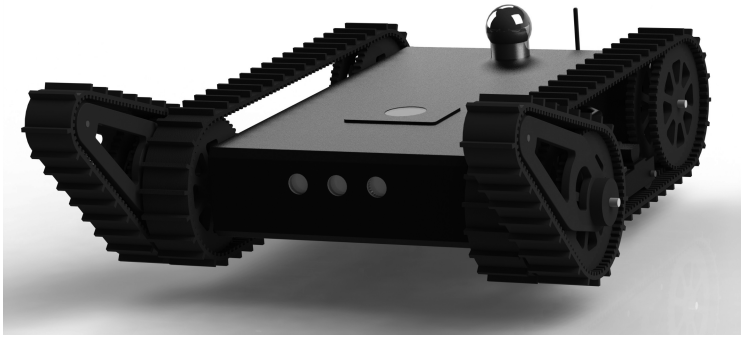


Fig. 2. Guide robot

The second type of robot (called “cone robot”, Fig. 3) is the one that carries the beaconing system and is positioned on the road to alert road users about tasks and events on their route. To do so it carries a signaling element (roadwork cone) with a light beacon that can be remotely operated. The cone robot also has a video camera, an ultrasonic distance measurement system and an infrared line detection system. They provide detection capability both for obstacles and road lines. It implements an IMU to retrieve odometry data to manage its movement, as well as a GPS system to provide precise positioning of the robot [12].

Every robot system consists on one guide robot and several cone robots, whose number depends on the area that is going to be signaled. The deployment process is performed in an automated way, following the previously programmed parameters and directives (length of the road cut, width, distance between cones, number of robots…). The operation scheme followed by the robots during the deployment

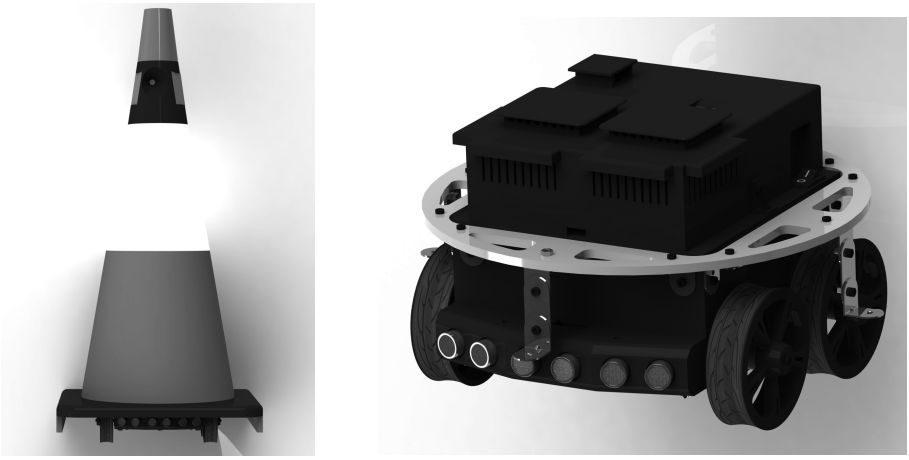


Fig. 3. Cone robot with and without mounted work cone

process is as follows. The guide robot leads the way, moving forward on the road shoulder while cone robots follow it [13]. The guide robot maps the road and sends the information about which path they have to follow to the cone robots. Once they reach the desired deployment spot, the system starts placing each cone robot into their appropriate position. After being deployed, the robots can stand still or move along the road, following the work in progress (ie. linear maintenance tasks, such as roadside weeding or road mark painting). After the completion of those tasks, cone robots move back to the road shoulder and then follow the guide robot to the place at which the collection or storage takes place. A diagram of the whole deployment process is shown in Fig. 4. Fig. 5 shows a simulation capture of the deployed system in a typical lane cut scenario. Both cone robot and guide robot simulation models have been designed specifically for this project, including all the real robots capabilities and functionalities. Simulation was performed with Gazebo simulation environment, using ten cone robots and one guide robot and specifying the lane cut parameters (cut width: 2.5m, cut length: 27m, type of formation: lane cut). The starting position of the robots was lined on the road shoulder.

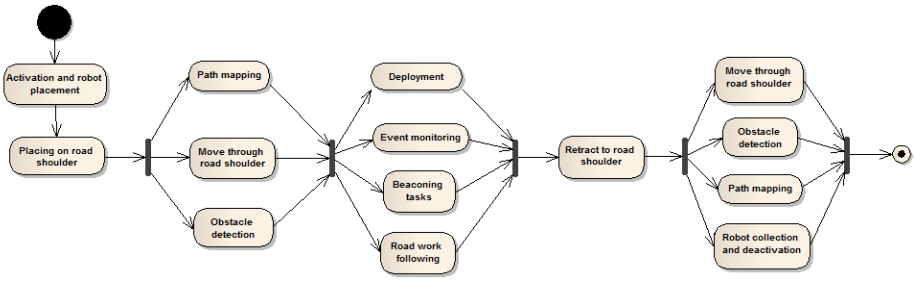


Fig. 4. Automated deployment process diagram

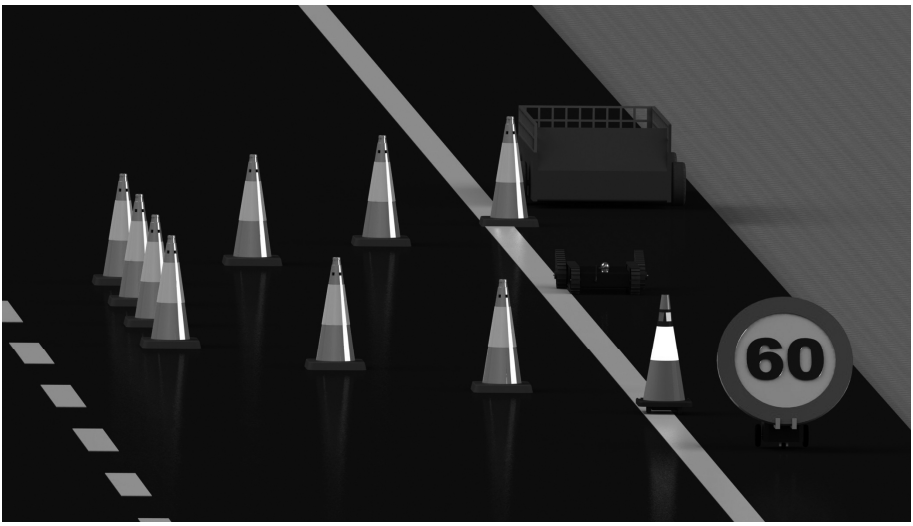


Fig. 5. Deployed beaconing system

Within this working scheme, collaborative tasks performed by the system can be grouped into four areas:

- Positioning tasks: each robot uses its own position measurements (GPS, odometry, IMU and camera) together with the information processed by other robots to correct and enhance their position estimation.
- SLAM tasks and obstacle mapping: in cooperation with the guide robot, each cone robot sends information to the mapping system, allowing a better system performance and enhanced robot path calculations.
- Monitoring, positioning and tracking tasks: the system uses camera and sensor information to scan the surrounding area and detect positioning errors, obstacles or events in order to apply the correct actions.
- Management and auxiliary tasks: the system reacts to changes in the placement of components, as well as to new needs in their placement or structuring.

Each one of these tasks is described in greater detail below.

2.1 Positioning Tasks

Each one of the robots gets its own position information from the GPS module, the odometry system, the IMU and the line detection sensors. This information is used to estimate robot's position, and is then sent to the control system in order to be processed. This positioning data includes an error that depends on the precision grade of each sensor. Differences in the error percentage from one robot to another lead to a global lack of precision of the whole system.

The information concerning a single robot given by the rest of robots of the system is used to enhance the positioning of the first one as a possible way to minimize that error. The way in which this is implemented is as follow:

The camera mounted on each robot allows it to take images of its field of view. Those pictures are processed by image recognition software in order to find other robots which are part of the system in the pictures [14]. This can be done with shape recognition algorithms, color recognition filters or by means of identification of QR tags placed on the robots (in case that QR codes are used)[15] [16].

Once the system detects other robots, the images are processed to calculate their positions (distance and orientation) in relation to the robot that took the picture. These calculations are possible thanks to previous knowledge of the robot's size and geometry. After applying these operations to the whole group of robots, a large number of relative measurements are obtained: this allows for the performance of a correction of the individual measurements and thus a fine adjustment of each robot's position. The information provided by the line detection sensors (when available) also helps the system to adjust robot position. The operation diagram of the position correction system is as shown in Fig. 6. Fig. 7 shows the use of a detection algorithm to find the relative distance and orientation of a cone using a real image.

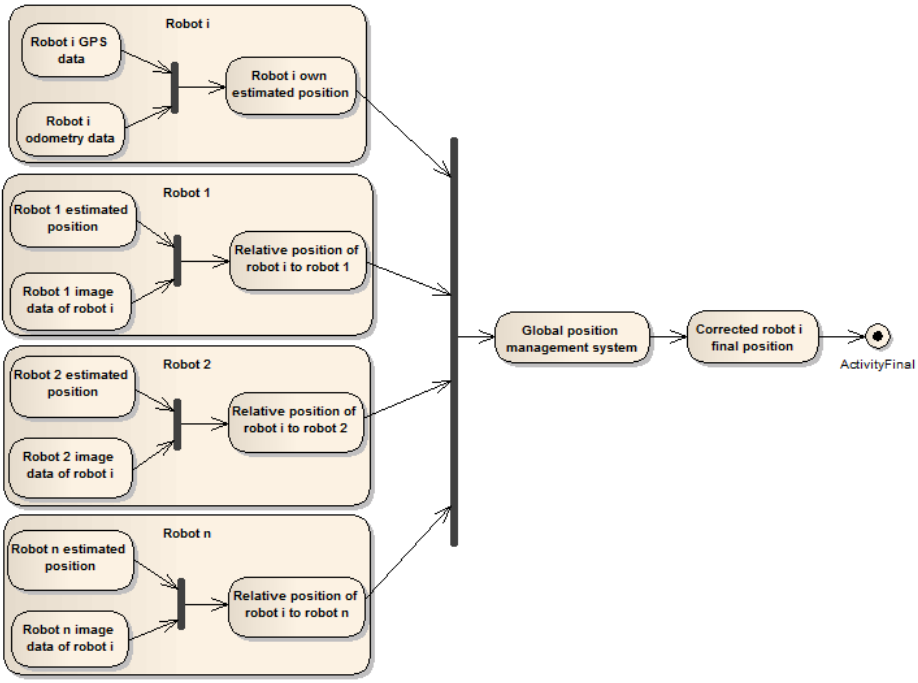


Fig. 6. Position correction process diagram

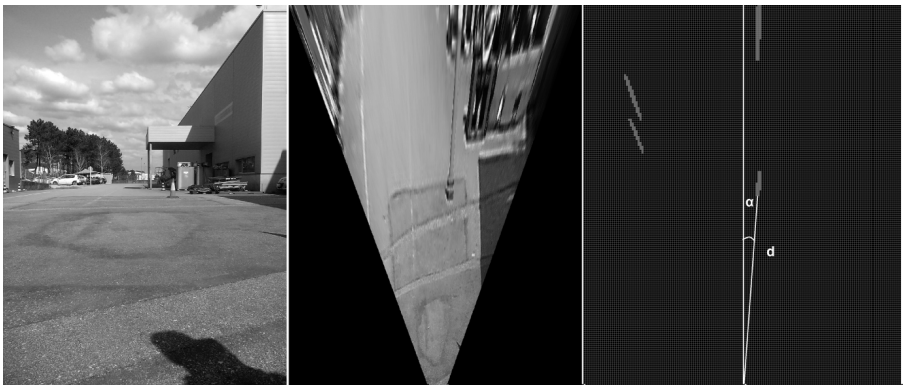


Fig. 7. Cone detection algorithm applied on a real image. On the left, original image. Center: bird's eye transformation. On the right: cone detection and measurement (distance and angle).

2.2 SLAM Tasks and Obstacle Mapping

While moving along its route, the guide robot uses its laser scanner to map its surroundings and then combines that information with the images taken by the camera to obtain a more accurate result. However, the mapping is not complete because the field of view of the robot sensors does not cover a full 360º angle. This causes the appearance of "dead zones" or "blind spots" in which little or no information is available. As the mapping information is employed to plot the path that the system will follow it is absolutely crucial to improve, in as much as possible, the accuracy of this map.

The map used by the system is initially created as a blank flat square grid whose dimensions are the ones of the desired working area. In that grid each square represents the minimum area unit needed for the proper operation of the system, which will be the cone robot's area (40x40cm square plant). Information received about those "cells" by the laser scanner or the camera is used to fill each one of them with the appropriate attributes (presence/absence of obstacles, presence/absence of lines, presence/absence of another robot, relative inclination, luminosity data, presence/absence of irregularities on asphalt…).

The tool used within ROS to create the base of the map is the ROS SLAM package "gmapping", which creates a 2D complementary map from the laser sensor and the guide robot orientation data. The information included in this complementary map is exported to the system map, fitting the obtained SLAM data to the predefined square grid of fixed dimensions. At this point, every cell containing part of an obstacle will be set as obstacle. Then system uses ROS topics to merge the robot sensors information with each robot's position information in order to assign all this data to its corresponding cell.

In order to achieve a more accurate and complete map, each cone robot collaborates in its generation process, filling the gaps of information with its own sensor data. To accomplish that task, each cone robot takes a picture

when it finds itself in a specified position determined by the system. This way a full group of images can be assigned to the scanner data associated with one position, instead of a single picture. This methodology not only allows an improvement of the information stored in the map, but also the addition of new data (objects, line positions) or even the detection of changes in map information over time.

Apart from the information supplied by the cameras, the cone robot's sensor systems allow the inclusion of more detailed information to the map. They can determine or confirm the position of road lines (using an infrared based color detection sensor system), detect irregularities on the asphalt (bumps and holes can be found via robot-to-road infrared distance sensors) or find little obstacles on the surface. An approach to this strategy is the one shown in Fig. 8.

With all this data the system can accurately calculate trajectory and points of passage (waypoints) for each robot [17]. This trajectory is calculated by means of the use of path planning algorithms, and the result is a collection of waypoints that the robots must follow in a specific order. The spacing between those points is small enough (between 0.5m and 3m) to allow the robot travelling between them using straight movements, which increase both the movement precision as the system positioning accuracy. Proximity between waypoints also allows optimized obstacle avoidance strategies and path recalculation tasks (for example when a new obstacle is reported to the system).

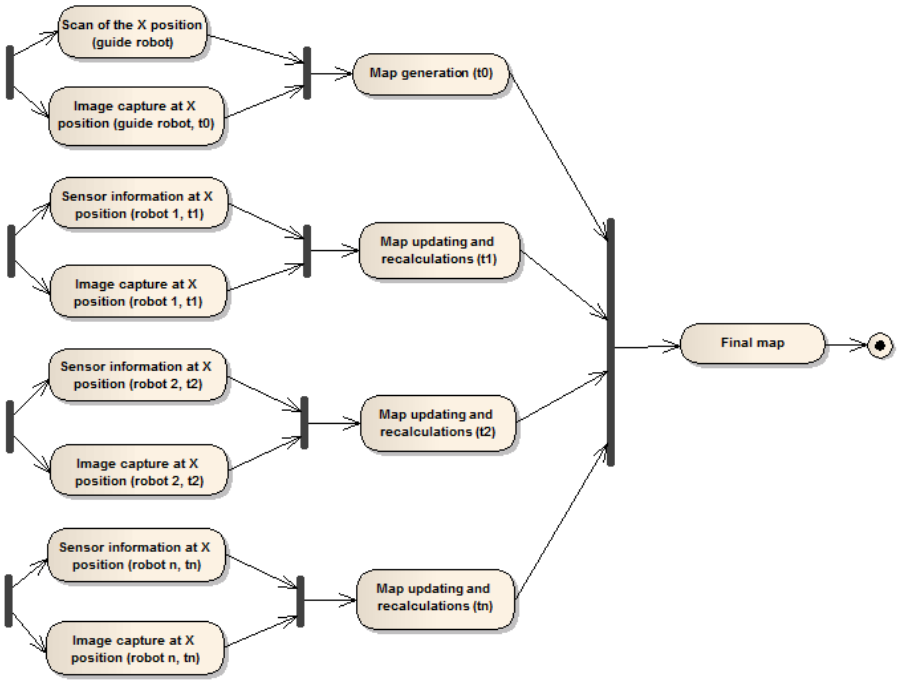


Fig. 8. SLAM mapping process diagram

2.3 Monitoring, Positioning and Tracking Tasks

Both during the deployment period and the beaconing tasks, the information obtained from sensors and cameras is used to test the system is operating correctly. During the movement and deployment phases the visual information and the sensor data allows for the correct movement and path monitoring. This would be the case of the combined robot movement, in which the cone robots follow the guide robot using the same path, simply leaving a predetermined distance between them. Any unforeseen deviation taken by a robot is registered by the rest, both in the images taken and in the distance measured by the ultrasonic sensors. The analysis of that information allows the correction of that deviation almost instantly.

If the deployment has already been completed, data received from each robot's camera is used to check both the correct positioning and the proper operation of the light beacons. It is also possible to perform change detection in the deployment area, in order to find undesired operations - such as access violations (ie. invasion of the working area by a vehicle) - that would be recorded by the camera system and the obstacle detection system. A diagram of this operation scheme can be found in Fig. 9.

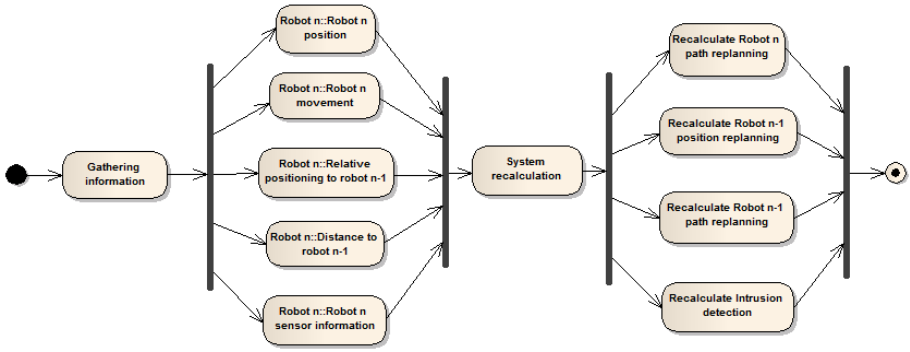


Fig. 9. Monitoring, positioning and tracking process diagram

2.4 Management and Auxiliary Tasks

If, through any of the previously described methods, the system detects any kind of anomaly in the positioning or functionality of the robots, it takes the appropriate measures to solve it. There are some different possible scenarios. The two main ones are:

- Detection of an unexpected obstacle while in movement: in this situation the robot that finds the object applies a dodging maneuver (when possible) and reports the position of the obstacle to the system. This way the map is corrected and the paths and points of passage are recalculated to avoid it.
- Detection of failures in the deployment scheme: if a robot is detected outside of its assigned position (eg. due to a malfunction, a battery that ran out or a collision) the

system recalculates the whole formation's positions to accomplish with the most important programmed parameters (length, width and placement of the beaconing system). The rest of robots therefore move to cover the gap left by the "missing" robot and continue to fulfill the main beaconing requirements of the road maintenance operation (eg. length and width of the "protected" area). That means, for example, that if one robot fails and the whole formation must cover a defined area (given parameter) the distance between robots (another system parameter, but less important) must change. Detection of the failure can be achieved through image processing, through sensor data from the other robots, or through the very information provided by the malfunctioning robot (low battery level, wrong GPS position data…).

3 Collaborative Robot Architectures: Technological Comparison

Three type of collaborative robot architectures have been compared: a robotic soccer architecture (Robot Soccer Nao [18]), a multi-robot mapping architecture ([19][20][21]) and the proposed design for automated deployment and positioning of beaconing robots (Robocone system). Its main characteristics are described in the following chart (Table 1).

Table 1. Technological comparison of collaborative robot architectures

	Robot Soccer Nao	**Multi-Robot Mapping**	**Robocone**
Global objective	Scoring a point in the opponent goal and preventing the rivals to do so in self goal	Creating a map	Road beaconing and mapping
Collaborative tasks	Sharing of cartographic data, opponent robot blocking, strategy planning and tactics implementation	Sharing of cartographic data	Sharing of cartographic data, movement and position correction, event detection
Positioning	Through computer vision and odometry	Through GPS, odometry, range finder system, computer vision	Through differential GPS, odometry, range finder system, line detection, computer vision
Vision system	Yes: local vision and global vision	Yes: local vision	Yes: local vision

Table 1. (*continued*)

Equipped sensors	2 HD cameras 1280x960, 4 microphones, 9 tactile sensors, 8 pressure sensors, 1 IMU, 1 sonar rangefinder (0–70 cm)	Infrared range finder, sonar, IMU	Laser range finder (30m), sonar sensors (0-3m), IMU, camera, GPS, line sensors, infrared edge sensor
Communi-cations	Wi-Fi	Bluetooth, Wi-Fi	Wi-Fi
Data processing	Internal: local vision External: global vision, map fusion, role assignment, path planning	Internal: computer vision External: map merging	Internal: computer vision, obstacle avoidance External: map merging, path & action planning, system coordination

The main pros of the proposed architecture against the other ones within its field of application are the following:

- Differential GPS correction. This allows improved and more accurate position estimation, essential for on-road tasks.
- Robot recognition. This allows the correction of the robots position based on computer vision.
- Sensors' information. Line detection sensors, obstacle detectors and edge sensors supply important road information that enhances the mapping and path planning tasks.
- Surveillance activities. The use of the images taken by the robots during road works allows the detection of ongoing zone intrusions or other risky situations.

4 Further Work and Conclusions

At its current stage, the collaborative system is still being developed. Despite this, there are some interesting preliminary results that can already be presented in this paper already. Both the robotic platforms (guide robot and cone robot) have already been designed and built (Fig. 10), and that their control algorithms have been implemented. Moreover, the sensor systems are fully operative (GPS, IMU, distance and line detection, cameras) and part of the sensor software has already been programmed (odometry tasks, single robot positioning, single robot movement planning and execution). The communications system (through WiFi technology) is under development, although the first tests have been already completed successfully.

Furthermore, simple mapping software with obstacle detection and recognition has been already built.

All this said, in order to achieve further developments, the image recognition and relative positioning software, currently under development, should be finished before this project can be completed and finally tested.

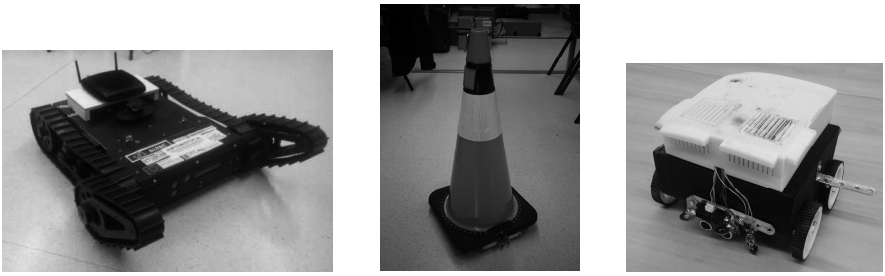


Fig. 10. Guide robot and cone robot (physically built)

Finally, the collaborative architecture basics have been designed but, however, they have not been fully implemented. The pending challenges are, therefore, collaborative position correction, SLAM tasks, cone (object) tracking and group behavior management .They must be achieved to provide the system with the tools needed to make it fully functional in a real use scenario.

References

1. Balchm, T., Parker, Y.L.E.: Robot teams: from diversity to polymorphism. AK Peters Wellesley (2002)
2. del P.A.G. Fuente, M., de la Pinta, J.R., García, A.L.: Interoperability Systems. In: Service Robotics within the Digital Home, pp. 1–47. Springer, Netherlands (2011)
3. Murphy, R.: An introduction to AI robotics. The MIT Press (2000)
4. Veloso, M., Stone, P.: Video: RoboCup robot soccer history 1997 – 2011. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5452–5453 (2012)
5. Şahin, E.: Swarm Robotics: From Sources of Inspiration to Domains of Application. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics WS 2004. LNCS, vol. 3342, pp. 10–20. Springer, Heidelberg (2005)
6. Burgard, W., Moors, M., Fox, D., Simmons, R., Thrun, Y.S.: Collaborative multi-robot exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2000, vol. 1, pp. 476–481 (2000)
7. Quigley, M., Conley, K., Gerkey, B., 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, vol. 3 (2009)
8. 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, pp. 1556–1563 (2006)

9. Sukkarieh, S., Nebot, E.M., Durrant-Whyte, Y.H.F.: A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. IEEE Transactions on Robotics and Automation 15(3), 572–578 (1999)
10. Caron, F., Duflos, E., Pomorski, D., Vanheeghe, Y.P.: GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. Information Fusion 7(2), 221–230 (2006)
11. Cox, I.J.: Blanche: Position estimation for an autonomous robot vehicle. In: Autonomous Robot Vehicles, pp. 221–228. Springer (1990)
12. Chenavier, F., Crowley, J.L.: Position estimation for a mobile robot using vision and odometry. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2588–2593 (1992)
13. Litzenberger, G.: World Robotics (Service Robots 2012), IFR Statistical Department (2012)
14. Besl, P.J., Jain, Y.R.C.: Three-dimensional object recognition. ACM Computing Surveys (CSUR) 17(1), 75–145 (1985)
15. Liu, Y., Yang, J., Liu, Y.M.: Recognition of QR code with mobile phones. In: Control and Decision Conference, CCDC 2008, pp. 203–206 (2008) (Chinese)
16. Chang, Y.-H., Chu, C.-H., Chen, M.-S.: A General Scheme for Extracting QR Code from a non-uniform background in Camera Phones and Applications. In: Ninth IEEE International Symposium on Multimedia, ISM 2007, pp. 123–130 (2007)
17. Bekey, G.A.: Autonomous Robots: From Biological Inspiration To Implementation And Control. MIT Press (2005)
18. Coltin, B., Liemhetcharat, S., Mericli, C., Tay, J., Veloso, M.: Multi-humanoid world modeling in Standard Platform robot soccer. In: 2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 424–429 (2010)
19. Rockel, S., Klimentjew, D., Zhang, J.: A multi-robot platform for mobile robots—A novel evaluation and development approach with multi-agent technology. In: 2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 470–477 (2012)
20. 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, vol. 1, pp. 321–328 (2000)
21. Rothermich, J.A., Ecemiş, M.İ., Gaudiano, P.: Distributed localization and mapping with a robotic swarm. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics 2004. LNCS, vol. 3342, pp. 58–69. Springer, Heidelberg (2005)

Bimanual Robot Manipulation and Packaging of Shoes in Footwear Industry

Ricardo Morales[1], Francisco J. Badesa[1], Nicolás García-Aracil[1], Richard Bormann[2], Jan Fischer[2], and Birgit Graf[2]

[1] Department of Systems Engineering and Automatic Control, Miguel Hernandez University, Spain
[2] Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Germany

Abstract. The present paper presents the ongoing research undertaken as a research experiment, called HERMES, inside Echord european integrated project framework. The goal of the HERMES experiment is to study, analyze and finally implement the packaging of shoes on a robot system that mimics the required degree of flexibility and dexterity provided by the human workers. The use of a bimanual system with anthropomorphic hands has been chosen as it could also be applied to solve other processes of similar or even higher complexity in the future and thus provide a holistic approach towards automization in footwear industry. In this paper, the first experimental results of the HERMES experiment are presented.

Keywords: Robot vision, bimanual robotic systems, automation, control.

1 Introduction

The present paper presents the ongoing research undertaken as a research experiment, called HERMES, inside Echord european integrated project framework with an overall aim of studying, analyzing and finally implementing the packaging of shoes on a robot system that mimics the required degree of flexibility and dexterity provided by the human workers. The use of a bimanual system with anthropomorphic hands has been chosen as it could also be applied to solve other processes of similar or even higher complexity in the future and thus provide a holistic approach towards automization in footwear industry.

The footwear sector is an important part of the European manufacturing industry and it plays a crucial role on the economy and social well-being in numerous localised regions within the EU27. The footwear market has annual sales of around 2.1 billion pairs and a value of around €49 billion; the EU alone accounts for about 5% of global production [1]. The European footwear industry is highly competitive, both on the European Union (EU)'s internal market, and on global markets mostly due to its quality, design and fashion attributes. The industry is dominated by small and medium sized enterprises (SMEs) and is particularly exposed to international competition.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_23,

One of the main challenges faced by the footwear industry is the increasing costs of production, which have led EU firms to relocate and outsource their products and processes to developing countries as a cost-reducing strategy. In addition to cost savings, more efficient supply chains help to reduce cycle times and meet customers' demands. The most innovative companies use technological investments to partially automate formerly manual processes, streamline operations, improve customers' service and knowledge, enable new imaginative ways of innovating products, speed up distribution [1].

The footwear industry accounts for some of the shortest production runs to be found (eight pairs of shoes is the average order size). Automation is more and more required in order to ensure competitiveness in this growing market. The introduction of intelligent robotic technologies can contribute to overcome the complexity in the automation of the associated production processes.

This paper presents the final implementation of the packaging of shoes on a bimanual robotic system with anthropomorphic hands that mimics the required degree of flexibility and dexterity provided by the human workers.

2 Robotics and Automation in the Footwear Industry

From 1996, there are some research projects dealing with the automation of manufacturing processes in footwear industry, such as, SPI6 project financially supported by Italian Ministry of University and Research. The main goal of SPI6 was the integration between product design and manufacturing and the automation of some of the most important phases of footwear manufacturing. Another project funded by EU that have dealt with the idea of enhancing the productivity and flexibility of footwear industry from different perspectives is the INTELISHOE project [2]. The main objective of INTELISHOE was developing innovative capacities of SMEs of a traditional sector such as the footwear and auxiliary industry by the taking-up of IS technologies aiming at the reduction of time-to-market through the implementation of a distributed design concept and subsequent prototype production which will allow to establish new work procedures. The SSHOES project addresses the development and demonstration of customized product concepts (notably, footwear and insoles), the creation of an engineering framework for footwear products and processes, the assessment of materials and production processes to achieve full eco-sustainable product concepts, and on the creation of a technology framework for the digital engineering of customized diabetic footwear and insole products. The EURO Shoe project provided a first approach towards an engineering framework of customized shoes, where the issue of designing and manufacturing a single pair of shoes was achieved successfully from a manufacturing perspective [3]. Low cost 3-D digitizing technologies and quick production process for footwear and components yielded for the first time to customised shoes in an industrial way. The CEC-MADE-SHOE project has achieved progress in the field of eco-sustainable materials (biodegradable leather and components) for shoes and components, advanced tools for customization process (magic mirror), first step towards simulation tools (virtual shoe test bed) and new product concepts (bio-shoes and

active shoes) in the domain of fashion footwear [4]. The FIT4U project funded by EU inside Seventh Framework Programme has developed an engineering framework for customized safety footwear and gloves [5]. The ROBOFOOT project has recently achieved progress in the introduction of robotic technologies [6]. ROBOFOOT has demonstrated that robotic solutions for manual tasks that are considered by the industry as being among the most complex and challenging are possible with the re-design of manufacturing processes, new robot programming and controlling tools, the development of smart end-effectors and the support of machine vision [7–9]. However, there are still a lot of work to do to develop a commercial robotic application for the most complex and challenging processes, such as, packaging of shoes.

3 Packaging Process Inside Footwear Production Process

There are four main activities involved in the shoe manufacturing: i) the manufacturing of uppers; ii) the making (assembling); iii) the finishing; and iv) the delivery. The finishing phase of production is shown in Figure 1. It can be split on the following operations: polishing of the shoes, addition of various accessories, laces and insocks, final check for possible inaccuracies and preparation of the box for delivery or stock in warehouse. The "To put shoes in the box" block is the part of the finishing process which is called Packaging. Operators at the end of the production line manually do all the tasks involved in the Packaging process. Workers take the shoes, visually inspect them and if everything is fine and no defect is identified, proceeds with the packaging process, i.e. introducing a soft piece into the shoe (usually a simple piece of crumpled paper plus a plastic rib), and finally introducing the pair of shoes into the box.

Packaging process is one of the operations with higher workforce impact. Workers verify the correspondence in the pair (numbers), they write the pair number on the box, the pair of shoes is put into the box (sometimes having to introduce some piece of paper (or similar) to separate them and the box is closed with the lid (see Figure 2).

4 Design of the Bimanual Robotic System

4.1 Analysis of Packaging Process

The process of putting one shoe into the box, introducing a piece of paper, putting the other shoe and closing the box with the lid takes from 20 to 25 seconds to a human worker. Therefore, the robotic system should carry out this task in the less or equal time.

From the analysis, it can be extracted that the process will require a re-design of the process (such as an automatic box feeding system) to adapt it to the proposed robotic system. In the scope of the HERMES project, the packaging process will not include any further operation, such as, classification of the box

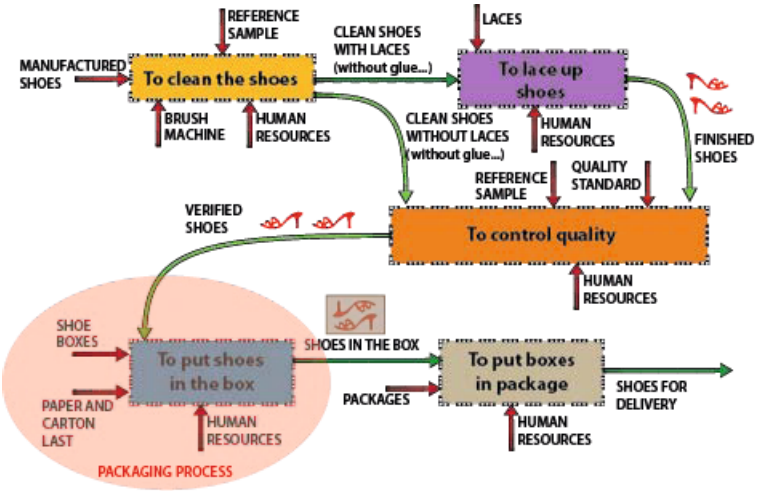


Fig. 1. Description of the finishing phase [10]

Fig. 2. Packaging in LAGAL SHOES manufacturing process

or building of the pallet. To understand how humans interact with shoes and manipulate them in the packaging process, a motion capture system has been used. In short, a motion capture setup composed by 12 V100 cameras and ARENA mocap software manufactured by Natural Point has been used to monitor the human behaviour and the object trajectories in a general packaging task. ARENA mocap software incorporates camera calibration, easy skeleton creation, multiple actor and objects tracking, real time solving and streaming, editing tools, and flexible data export options for full body motion capture. Marker tracking is the foundation of ARENA's motion capture technology. Using reflective markers placed on actors and objects, calibrated cameras create a cloud of 3D points. These 3D points can be labeled and exported to other applications or mapped to ARENA's skeleton solver for tracking full body motion. Captured actor and object data can be exported to 3D animation and analysis packages using standard 6DoF digital file formats. In Figure 4, some snapshots taken during the motion capture process are shown. In this process, the information provided by the camera was used to compute joint angles, velocities, trajectories and so on.

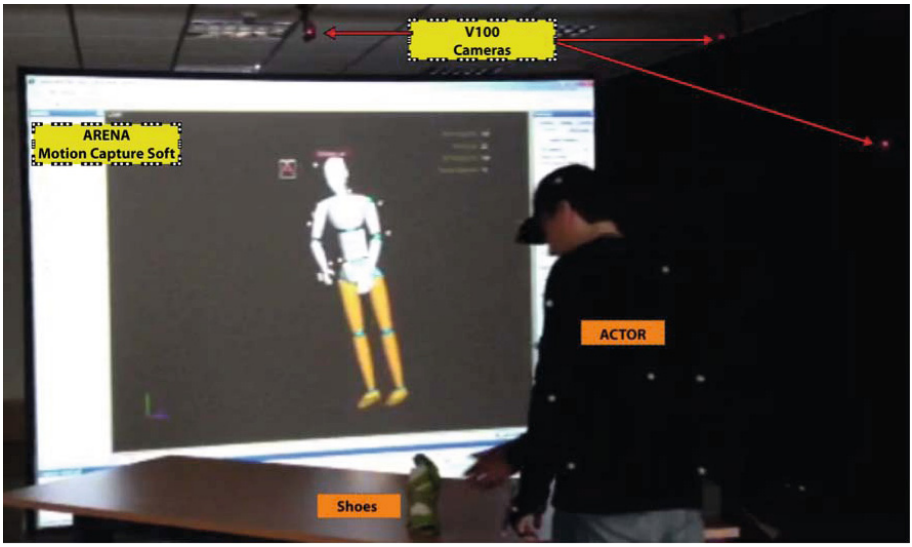


Fig. 3. Detail of motion capture system to monitor the human behaviour

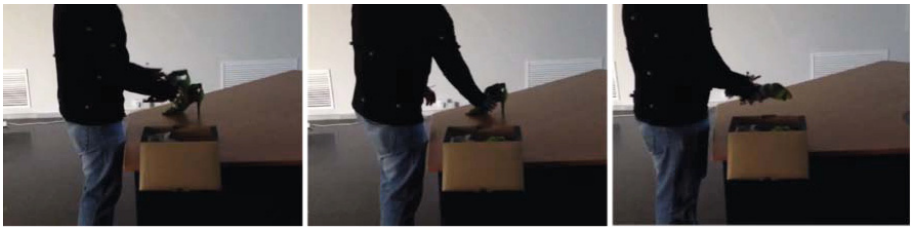


Fig. 4. Snapshots of motion capture process

4.2 Required Workspace

To compute the required workspace, 10 subjects were carried out the packaging process (putting the pair of shoes into the box and closing the box with the lid) inside our motion capture facilities. Using the information provided by the motion capture system, such as, joint angles, velocities, trajectories and so on, a required workspace was identified. In Figure 5 , the upper-limbs and head trajectories acquired by the motion capture system for one subject in a packaging process and volumetric workspaces for each limb computed with the information extracted from the experiment with 10 subjects are shown.

Different solutions, specifically kinematic configurations of the robot hardware (e.g. length of connectors) and sensor positions were designed and simulated for the robotic system for automatic packaging of shoes and validated with respect to the specifications. The design and preliminary evaluation of the robotic solution with special attention to mimic the human behaviour during the packaging

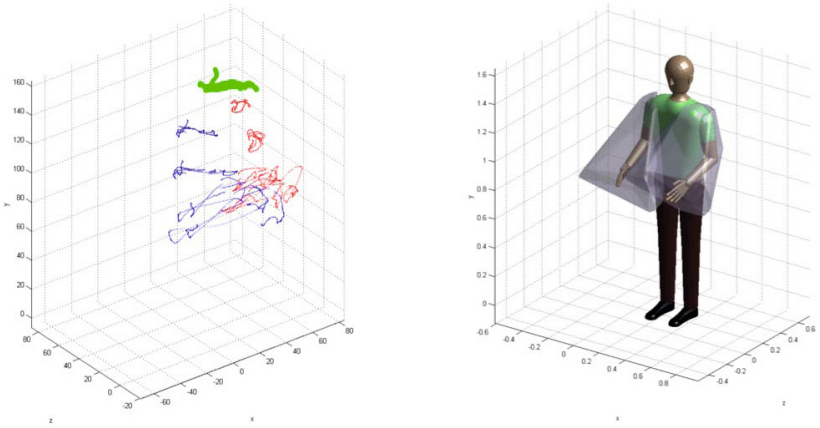


Fig. 5. Left: Upper-limbs (blue, red) and head (green) trajectories for one subject, Right:Upper-limbs workspace

process will be carried out using CAD software and simulation tools in dynamic environments. Taking into account the information provided by these simulations and required upper-limbs workspace for packaging process, a bimanual robotic system based on a 7 DOF robot configuration using PRL modules manufactured by Schunk and custom-made links has been selected as the optimal solution. The designed links along with the Schunk modules creates a kinematic chain. A world coordinate frame is established in the base of the robot. In order to determine the transformation matrix from the world frame to the end-effector frame, a local coordinate frame needs to be established for every link of the robot. Each local coordinate frame is defined using Denavit-Hartenberg (D-H) convention. In Figure 6, bimanual robot workspace (green colour) and upper-limbs workspace (yellow colour) are shown to demonstrate that the robot workspace is enough to reach the human workspace in packaging process.

5 Hardware and Software Components

In this section, the main hardware and software components developed up to now inside HERMES project framework are described in detail.

5.1 Hardware

The main components of a customized bimanual manipulator (see Figure 7):

- 7 PRL ROTARY UNITS (2xPRL120, 2xPRL100, 2XPRL80, PRL60) from Schunk for each robotic arm
- 1 FT Sensor FTN-Gamma from Shunck for each robotic arm
- 2 Antropomorphic hands from Prensilia

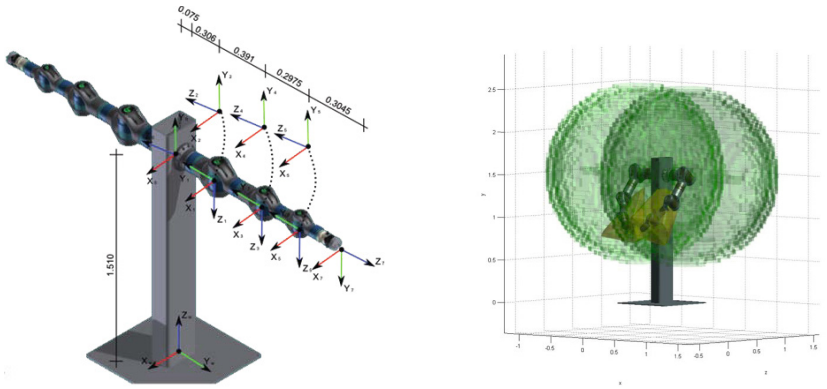


Fig. 6. Left: Bimanual robotic system with the coordinate frames using the Denavit-Hatenber convention. Right:Robot workspace vs upper-limbs workspace for packaging process.

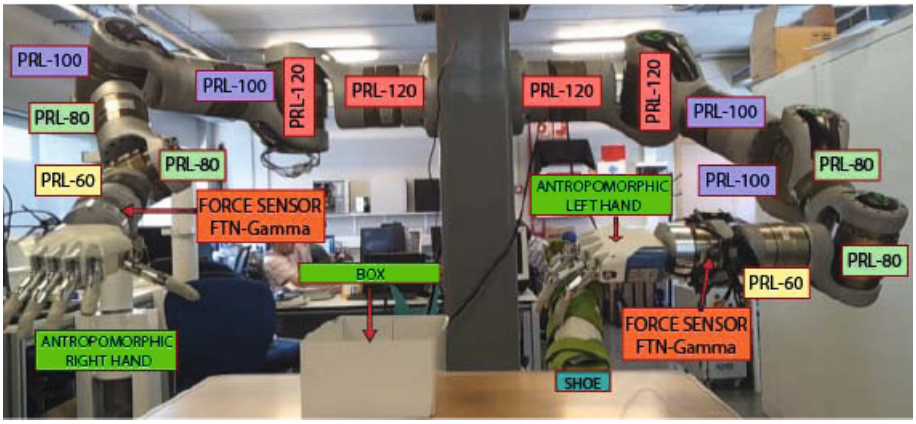


Fig. 7. Low level control functions of the robot have been encapsulated in a MATLAB class called HERMES

5.2 Low Level Control Functions for Bimanual Manipulation

Software modules to move and control the two robotic arms and hands has been implemented. First, the software modules to move and control the robotic devices in *open-loop* were developed and tested. After that, the direct and inverse kinematics of the bimanual robotic system was designed and implemented. Finally, basic trajectory planning algorithms were integrated and applied to the project demonstrator. All the low level control functions of the robot have been encapsulated in a MATLAB class called HERMES (Figure 8). HERMES class can be decomposed in the following modules:

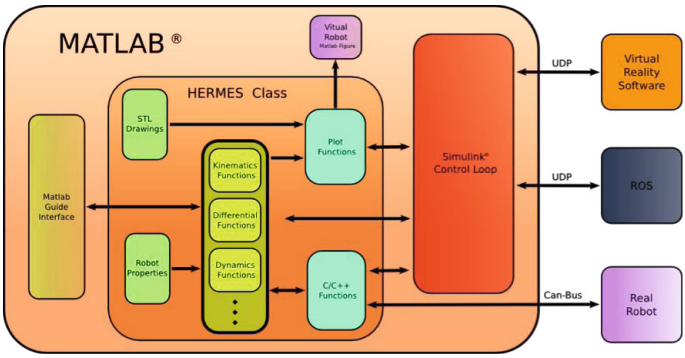


Fig. 8. Low level control functions of the robot have been encapsulated in a MATLAB class called HERMES

- Matlab Guide Interface: MATLAB Guide Interface developed to com- mand the virtual and real robot. The interface has been generated with the MATLAB toolbox guide. Using this interface, robot motions in the joint space and cartesian space can be performed.
- HERMES Class: MATLAB Class allows to generate a HERMES object that has access to all the control functions, parameters and tools of the robot.
- STL (STereoLithography) Drawings: MATLAB data files with graphical information about vertices and faces of the mechanical components of the robot extracted from STL files. With the help of this information, a 3D image of the robot is represented in a MATLAB figure.
- Robot Properties: Matlab data files with the information about kinematic and dynamic parameters of the robot.
- Kinematics, Differential, Dynamics and other functions: Set of functions to compute forward and inverse kinematic algorithms, to solve redundancy and motion trajectories, to compute direct and inverse dynamics, etc.
- Plot Functions and Virtual Robot: A toolset to represent a high quality 3D model of the robot in Matlab graphical environment. These plot function can be used to test control algorithms in the virtual robot and environment before to run them in the real robot.
- C/C++ Functions: Set of functions written in C / C++ that enable communication between the robot controller and the MATLAB environment. These functions are compiled into the MATLAB environment using the mex build tool. The result is a transparent communication between MATLAB and the robot controller in real time.
- Simulink Control Loop: The MATLAB class and its functions are accessible from the Simulink environment, allowing the development control schemes in a simple way. The real robot can also be controlled from Simulink environment using block C / C + + Functions.
- Virtual Reality Software: Simulink schemes controlling the real robot can be linked virtual reality software using UDP communication blocks.

– ROS(Robot Operating System): It can be linked with ROS
– Real Robot: Block representing the real robot. The connection between a PC running Matlab and the HERMES class to the real robot is carried out using a CAN network

5.3 High Level Control System

Based on the analysis of the manual packaging process and the technical specification for the robotic system, the high-level control loop suitable to solve this task were designed and implemented. The high level control system has been implemented by defining generic ROS interfaces to single capabilities of the robot. Specifically, functions like object detection and localization, arm movements and grasping were considered as single modules whose functionality can be easily accessed from the control script in a standardized way using the service and action concepts of ROS. Both concepts define a client-server mechanism in which the client triggers the server to fulfil some function, e.g. object localization or an arm movement. The difference between both approaches is that services are blocking calls to the server whereas actions are non-blocking, i.e. the script could execute further instructions while the request is handled. Object detection and grasping can be executed quite fast by the system, so it is sufficient to implement them as services. The arm movements, however, can take some more time and are therefore provided as an action to the control script. The figure illustrates the implemented interfaces. This work allows for quickly assembling the real scenario from these basic capabilities by simply sequencing calls to different functionalities in the high level control script. The communication protocol for the interfaces is detailed in the boxes of figure 9.

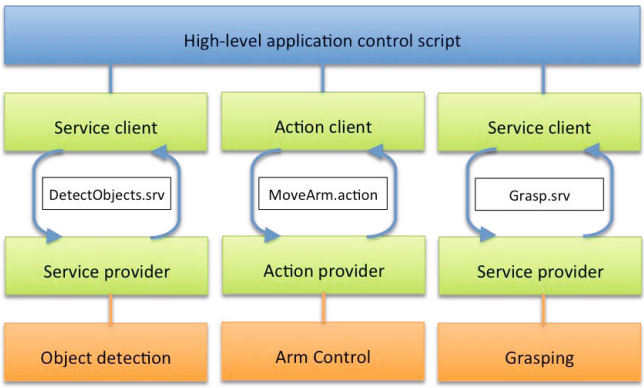


Fig. 9. Communication Protocol in ROS

5.4 Computer Vision System

An essential prerequisite for applying robotics technologies in footwear packaging and the automatic manipulation of relevant objects such as shoes, shoe boxes and packaging paper is the ability of the robotic system to detect these objects in the environment automatically. By acquiring 3-D sensor data and colour information of an object from different viewpoints, object models can be created as a basis for their future detection in a scene.

Local methods [11] extract interest points, so called feature points, from the object views and perform classification on the regions around these points. Global object representations are based on statistical classification techniques where complete object views are used to train an object classifier [12]. Global approaches are powerful in deciding whether or not an object appearance that is not included in the training set e.g. from a different perspective, belongs to a known object. On the other hand, local interest point matchers have advantages in detecting an object even if only parts of it are visible e.g. in cluttered scenes. Both methods give only limited information about the objects precise position and orientation required for manipulation in 3-D space. Here so-called (geometric) model-driven approaches are needed that fit a 3-D shape model into the scene and track this shape in a sequence [13]. These methods reliably compute the pose information under the constraint that there is a priori knowledge about the scene location where model fitting needs to be started.

Object detection and learning has been the topic of numerous projects in the past. Initial work on 3-D data acquisition, object body fitting and feature point classification was done at Fraunhofer IPA in the context of their Care-O-bot developments [14, 15], e.g. within the EC-funded Cogniron project. In prior work existing methods could be improved by early fusion of range and colour images and the extraction of 6-D (3-D position and orientation) feature points, meaning that full coordinate systems are defined for each feature point. With the current approach, objects can be detected very efficiently (less than a second) at very large spatial ranges [18]. The detection result is available in real 3-D point coordinates and orientations. A first trial to apply the learning and detection system to shoes showed promising results (Figure 10) if enough texture is available.

While the shoe box and the wrapping paper usually have enough texture to use this kind of detection systems, many shoes have too few texture to detect a sufficient amount of feature points. Hence, a texture-less detection approach has to be used for shoe detection. The initial approach based on a point cloud segmentation and description of the resulting clusters with the SAP descriptor [20, 21], but it depended a lot on the success of the segmentation step. However, especially on spatially sparsely distributed objects like shoes or shoe boxes, uninformed segmentation is likely to fail and so does the following classification step consequently. To avoid this error mode, a new descriptor for the detection of texture-less objects has been developed [19]. It is based on pixel-wise computations that are aggregated during model training. Objects are detected in real scenes by applying a sliding window method that turns a previous segmentation step unnecessary and

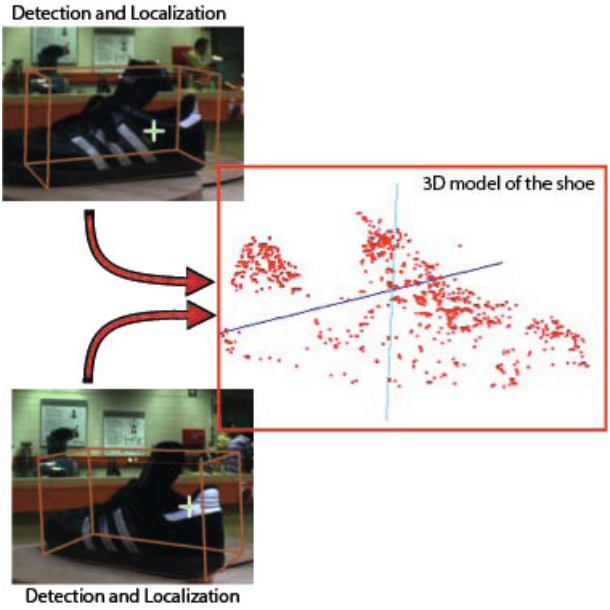


Fig. 10. First results: Detection and localization of a sports shoe and 3-D model of the shoe

therefore is better suited for detecting complex objects in cluttered scenes than a solution based upon previous segmentation. The proposed descriptor operates on single shot RGB-D images originating from a RGB-D camera device like the Microsoft Kinect. Initially, the computation of the local point feature descriptor captures 2D image and 3D depth cues with a binary descriptor. This per-pixel information is aggregated into a histogram over the object related image area. The histogram is reordered and normalized in order to achieve rotation and scale invariance.

6 First Experimental Results

6.1 Basic Trajectory Planning to Put a Shoe into the Box

The first experiments of putting a shoe into the box has been carried out with success using the information about human behaviour obtained using the motion capture system. Basic trajectory planning algorithms have been implemented to follow a learned trajectory made by a real worker in our motion capture facilities. In Figure 11, some frames of the packaging process are shown.

6.2 Computer Vision System

The recognition and localisation of the object can be divided into two steps: 1) we have to train the appearance of an object. Therefore, we developed a

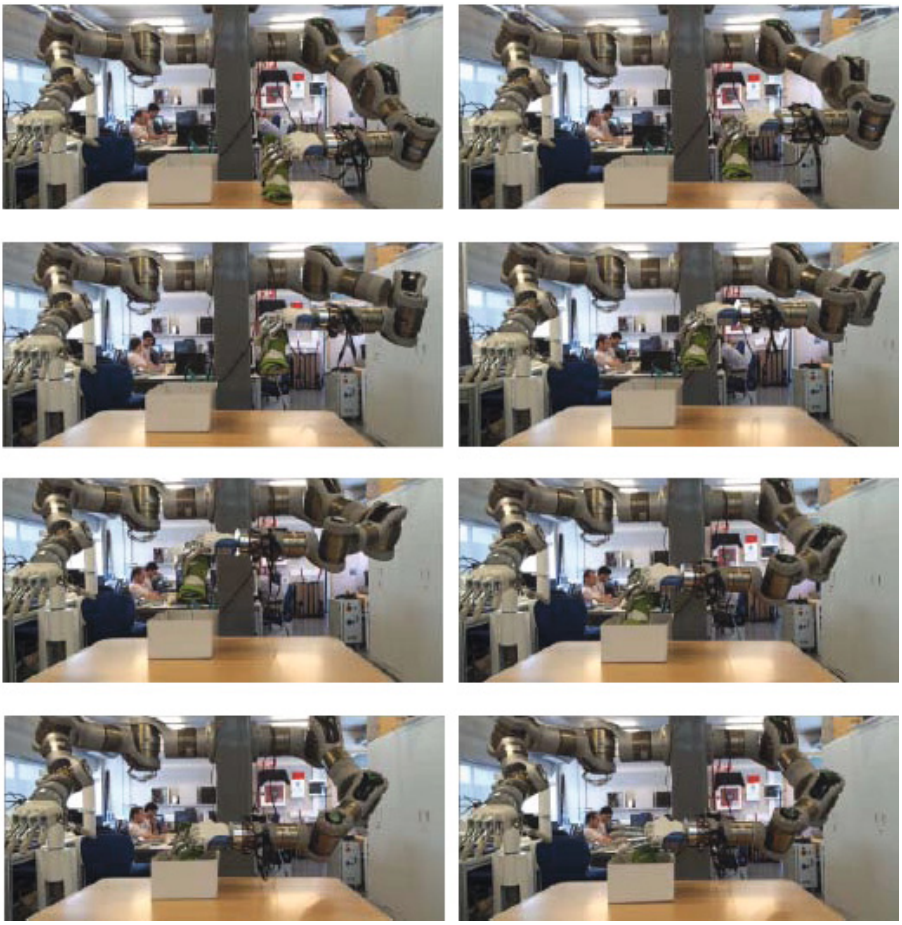


Fig. 11. First experimental results: Basic trajectory planning to put a shoe into the box. Each frame of the sequence is shown from frame 1 to 8(from Left to Right and from Top to Bottom).

simple training procedure that only requires capturing different perspectives of the object of interest placed next to a checkerboard. The training procedure can segment the object automatically as shown in Figure 12; and 2) Once a new object is trained it can be detected in images as shown in Figure 12. The highlights of this detection method are that the object may be placed in a cluttered scene and that we get a pretty accurate initial guess of the object pose. The pose estimate can be refined afterwards by aligning the captured data with the stored model using the ICP algorithm as demonstrated in Figure 13.

The perception module requires to obtain a recognition model for each object that should be recognized, e.g. shoes, the shoe box or a shoe divider. To generate an object model it is necessary to capture the object from multiple perspectives,

Fig. 12. First results: Training procedure (left) and recognition procedure (right)

i.e. all those perspectives that should be recognized later. As recognition also includes the pose estimate of the object, a fixed coordinate system must be attached to the object during modeling so that the viewing angles of each recorded perspective can be stored. This data allows to determine the orientation of the objects during the recognition step. So far, the coordinate system has been fixed during recording by placing the objects next to a checker board. However, when the object shall be recorded from all sides the checker board will be occluded from a range of perspectives. Hence, no coordinate system will be available for those recording. To overcome this limitation for object modeling the checker board has been replaced with board that contains 4 Pi-Tag markers. The pattern of this board is shown in Figure 13. The relations of all marker locations to the center of the board are exactly known so that the object coordinate system can be established whenever at least one marker is visible to the camera. Consequently, modeling of objects from all sides is now possible with the new solution.

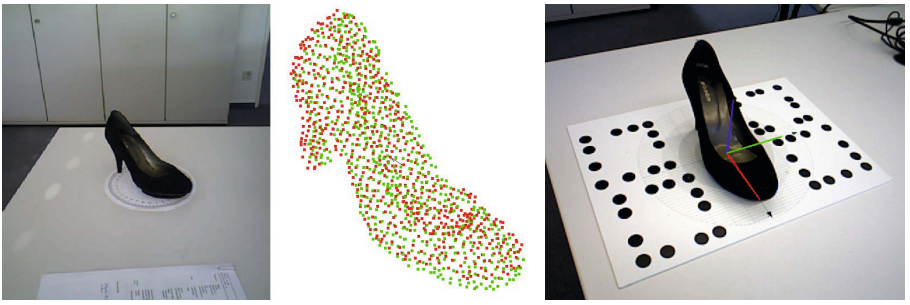


Fig. 13. Left: pose refinement with the ICP algorithm (red=model points, green=captured points), right: object recording using the Pi-Tags

7 Conclusions

The present paper presents the ongoing research undertaken as a research experiment, called HERMES, inside Echord european integrated project framework. The HERMES experiment will demonstrate the potential to apply robotic technologies in the packaging process of the high added value shoe production. Important key technologies for bimanual manipulation, specifically in terms of detection and manipulation of non-rigid parts, are being developed and will be transferred to potential system integrators for future commercialisation. The consortium is finishing the integration of all the modules to make the final demonstration of the project at the end of November. The use of anthropomorphic robotic hands is justified by two main reasons: the first one is to mimic the human behaviour in the packaging process and the second one is that simple parallel and/or suction grippers would only be useful for some kind of shoes and materials and anthropomorphic hand would be adaptable to most of the shoes and materials.

Acknowledgment. This work was supported by the EU Project EC FP7-ICT-231143 ECHORD.

References

1. Floyd, P., Patel, S., Kantor, E.: Survey on the Situation of the European Footwear Sector and Prospects for its Future Development: Final Report. In: Depth Assessment of The Situation of The European Footwear Sector and Prospects for Its Future Development, DG Enterprise and Industry, European Commission (November 2011)
2. INTELISHOE - Integration and linking of shoe and auxiliary industries, 5Th FP, IST-1999-20949 (2000-2002)
3. EURO-ShoE: Development of the processes and implementation of the management tools for the extended user oriented shoe enterprise, G1RD-2000-00343 (2001-2004)
4. CEC made shoe-Custom, Environment and Comfort made shoe, FP6, contract 507378 (2004-2008)
5. FIT4U-Framework of Integrated Technologies for User Centred Products, FP7, 229336 (2009-2012)
6. Robofoot-Smart robotics for high added value footwear industry, FP7, Project Reference:260159 (2010-2013)
7. Maurtua, I., Ibarguren, A., Tellaeche, A.: Robotics for the Benefit of Footwear Industry. In: Su, C.-Y., Rakheja, S., Liu, H. (eds.) ICIRA 2012, Part II. LNCS, vol. 7507, pp. 235–244. Springer, Heidelberg (2012)
8. Ibarguren, A., Martínez-Otzeta, J.M., Maurtua, I.: Particle Filtering for Industrial 6DOF Visual Servoing? Journal of Intelligent and Robotic Systems (in press)
9. Maurtua, I., Ibarguren, A., Tellaeche, A.: Robotic solutions for Footwear Industry. In: 2012 IEEE 17th Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–4, 17–21 (2012)
10. Robots in Footwear Industry: requirements. Deliverable 1.1. ROBOFOOT-Smart robotics for high added value footwear industry, `www.robofoot.eu`

11. Moreels, P., Perona, P.: Evaluation of feature detectors and descriptors based on 3-D objects. International Journal of Computer Vision (2006)
12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)
13. Teulire, C., Marchand, E., Eck, L.: Using multiple hypothesis in model-based tracking. In: IEEE Int. Conf. on Robotics and Automation, ICRA 2010, Anchorage, Alaska (May 2010)
14. Kubacki, J., Giesler, B., Parlitz, C.: Active Autonomous Object Modeling for Recognition and Manipulation. In: Autonome Mobile Systeme, pp. 227–233 (2005)
15. Kubacki, J., Baum, W.: Towards Open-Ended 3-D Rotation and Shift Invariant Object Detection for Robot Companions. In: Intelligent Robots and Systems, IROS, pp. 3352–3357 (2006)
16. Viola, P.A., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)
17. Hough, P.: Method and means for recognizing complex patterns. U.S. Patent 3069654 (1962)
18. Fischer, J., Arbeiter, G., Bormann, R., Verl, A.: A framework for object training and 6 DoF pose estimation. In: Proceedings of the 7th German Conference on Robotics (ROBOTIK), München (2012)
19. Fischer, J., Bormann, R., Arbeiter, G., Verl, A.: A Feature Descriptor for Texture-Less Object Representation Using 2D and 3D Cues from RGB-D Data. In: IEEE International Conference on Robotics and Automation, Kongresszentrum Karlsruhe, Karlsruhe, Germany, May 6-10 (2013)
20. Bormann, R., Fischer, J., Arbeiter, G., Verl, A.: Efficient Object Categorization with the Surface-Approximation Polynomials Descriptor. In: Stachniss, C., Schill, K., Uttal, D. (eds.) Spatial Cognition 2012. LNCS, vol. 7463, pp. 34–53. Springer, Heidelberg (2012)
21. Bormann, R., Fischer, J., Arbeiter, G., Verl, A.: Adding Rotational Robustness to the Surface-Approximation Polynomials Descriptor. In: Proceedings of the IEEE Conference on Humanoid Robots (2012)
22. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
23. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
24. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
25. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
26. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)

Human-Robot Collaborative Scene Mapping from Relational Descriptions

Eloy Retamino Carrión and Alberto Sanfeliu

Institut de Robótica i Informática Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain

Abstract. In this article we propose a method for cooperatively building a scene map between a human and a robot by using a spatial relational model employed by the robot to interpret human descriptions of the scene. The description will consist in a set of spatial relations between the objects in the scene. The scene map will contain the position of these objects. For this end we propose a model based on the generation of scalar fields of applicability for each of the available relations.

The method can be summarized as follows. In first place a person will come into the room and describe the scene to the robot, including in the description semantic information about the objects which the robot can't get from its sensors. From the description the robot will form the "scene mental map". In second place the robot will sense the scene with a 2D range laser building the "scene sensed map". The objects positions in the mental map will be used to guide the sensing process. In a third step the robot will fuse the two maps, linking the semantic information about the described objects to the corresponding sensed ones. The resulting map is called the "scene enriched map".

1 Introduction

In many everyday life tasks and situations we make use of spatial relations for resolving references (eg. "The scissors are in the left drawer"), explaining a wished objects layout (eg. "put the plant close to the left corner") or focusing someone's attention into a certain region (eg. "look behind the table"). In our work a person will use this kind of relational language for describing a scene to a robot in order to help it in the mapping of the scene and to improve the map itself. The scene will consist in a set of objects distributed inside a room which dimensions are known by the robot.

We find three possible situations which can occur in this kind of collaborative mapping process: (1) the person and the robot are together while mapping the scene, (2) the robot first senses the scene and afterward the person describes it, (3) the person describes the scene and afterward the robot senses it.

In our work we will face the third situation. The person will first describe the whole scene using the aforementioned spatial relations. From the description the robot will build a map of the scene, composed of a representation of the objects located at their respective positions. This "mental map" has two purposes in the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_24,

scene mapping process. In first place, it will be used for guiding the robot sensing process. That is, after interpreting the description the robot will go into the room and seek the described objects in the positions they have in the mental map. In second place, during the description the person may add semantic information about the objects which the robot could need for fulfilling its duties. For example, a post-delivering robot who has to leave a letter at Mike's desk should know who is the owner of each desk in its map. As the described objects are matched with their corresponding sensed objects, this semantic information can be trespassed to the latter.

In the rest of the article we will distinguish between four different scene maps. (1) The "human mental map", the map the person forms in his mind from viewing the scene. (2) The "robot mental map", the map the robot builds from interpreting the human description of the scene. (3) The "robot sensed map", the map the robot builds from sensing the scene. (4) The "enriched map", built from the fusion of the robot sensed and mental maps. We call this last map enriched because it mixes the accuracy of the robot sensors with the semantic information provided by the person in his description.

We would like to remark that the generation of the "robot mental map" can facilitate by itself the human robot interaction in many different cooperative tasks apart from mapping an scene. Reorganizing the objects in a room, defining an strategy for searching an object or exploring an area. For all of these tasks the robot has to interpret the "mental map" the person has in mind and to verify it using its own sensors.

The contributions of the present work are: (1) A model for interpreting spatial relations. (2) A method built upon the former to let the robot to map a scene in collaboration with a human. In the rest of the article we will, present the related work, the relational model, the method and the results of the experiments conducted for testing it.

2 Related Work

There are numerous attempts of interpreting qualitative spatial relations. In [1] Moratz et Al. define a computational model for the projective spatial relations similar in spirit to the one developed here. In their work each relation defines a canonical direction and depending on the position of the referenced object with respect to this direction the relation became true or false. In their experiments they tried to understand how people express spatial knowledge by asking the subjects to tell the robot to go towards one of the objects in the scene (ie. to uniquely determine that object from the others in the scene). The drawback of their work is that the conditions for a spatial relation to be fulfilled are boolean which is against the intrinsic vagueness of these relations.

In [2] Stopp et Al. use a computational model of the topological (near) and projective (front, behind, left, right, above, bellow) relations for accessing to a robotic arm through natural language. Their model [3], as the one presented in here, is based on the concept of the continuous decay of the applicability of

the spatial relations as we separate from an ideal condition which defines the relation.

In the same direction, Kelleher et Al. [4] introduce the fact, also contemplated here, that the applicability of the projective relations decay with the distance to the referent object. They also state that the "size" of the generated fields is proportional to the size of the referent object.

As far as we know, all the previous works related with the interpretation of qualitative spatial relations face the case of single relations between objects which positions are deterministically known, therefore not considering possible uncertainties. Just in [3] Gapp defines the way to perform single compositions between relations. In the work presented here however, is required to interpret the description of a whole scene without the support of any perceptual information, ie. to "imagine" the whole scene. This requirement led to the development of a general framework for composing spatial relations, estimating probability distributions for objects which positions are known just by spatial relations with other objects and the interpretation of spatial relations between these "imagined objects". Regarding this last point, in [5] Mavridis et Al. let a robot to imagine objects on top of a table at positions expressed by spatial relations, but they are more focused in the maintenance of a 3D representation of the environment from the robot sensors and in the interpretation of spatial relations between sensed objects. In their work the variety of spatial relations is very limited. Also they can't be composed neither be expressed between imagined objects.

Regarding the existing approximations for building maps from relational descriptions, they result quite simplistic, or lets say, the accuracy of the map neither the interpretation of the spatial relations itself weren't the main objective of the research. In [6] Coyne et Al., develop a model for interpreting narratives (ie. for generating a 3D representation from them), but they are more focused in the aesthetics of the representation and in the natural language processing. Though their work contemplates an extensive vocabulary including many spatial relations, the model for the latter seems to be too deterministic. That is, the spatial relations seems to define fixed distances between the objects no matter the context (though they don't provide any details at all about this model).

The process of building a map from a description presents several similarities with the well known in robotics SLAM problem [7,8], though it presents important differences which forbid to undertake it using the same approaches. In the former a person uses a set of qualitative spatial relations for expressing to the robot the positions of the objects in a scene and his own. Since the point of view from which the spatial relations are referred influences their interpretation, the robot must infer the position of the objects and the person's at the same time. Up to here the similarities. The person's position is expressed also through relations with other objects introduced in the description, that is, there is no explicit odometry information. This makes a distinction with most of the situations faced in robotics, for example [9]. More than that, as it will be seen, the objects probability distributions extracted from the spatial relations are far from being gaussians, which makes inapplicable any EKF Slam algorithm. As

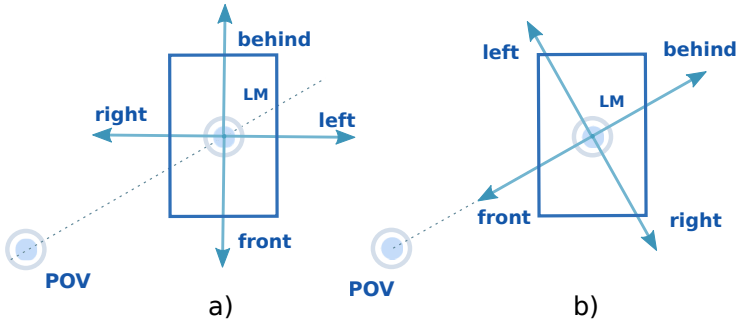


Fig. 1. Canonical directions for the projective relations in the intrinsic (a) and extrinsic (b) cases

said above, the approach taken here for building the "mental maps" will be to extract probability distributions for the positions of the referenced objects in the given spatial relations and to reduce the uncertainty by directly composing the relations given for each object.

Finally, as it's been said, the mental map built from the description is used by the robot to guide the sensing process, ie. to direct its sensors to the positions where according to this map the actual objects more probably are. In the same direction, [10] Aydemir et Al. use background knowledge and the interpretation of spatial relations to perform indirect object searching. For example, if the robot has to look for a cup in its environment and it knows that cups are usually on top of tables; it looks first for a table, computes the region determined by the expression "on the table" and looks for the cup inside that region. Their model though, just contemplates two spatial relations: in, on. Logically they don't consider compositions and the spatial relations can't be directly used by a person for communicating spatial knowledge.

3 Relational Description Model

In this section we specify the model used by the robot to interpret the scene descriptions. The approximation taken for this interpretation is based of the generation of scalar "Fields of Applicability" (FOAs from now) for each of them. These scalar fields represent the distribution of the applicability of a certain spatial relation in every point of space. The characteristics of each FOA will depend on the spatial relation which it represents and on the pose and geometry of the objects involved in the relation (eg. the table in "on the left of the table").

3.1 Preliminary Concepts

In any grounded spatial relation there are several objects implied, each of them fulfilling a different function. Before going with the generation of the FOAs is important to define them.

In our syntax, the *Point of View* (POV) is the object which states the position from which the relation is expressed (usually corresponding to the person referring the relation). The *Landmark* (LM) will be the object used as referent (eg. the table in "the chair on the right of the table"). Finally, we will designate the *Trajector* (TR) as the referenced object (eg. the chair in the previous sentence). The FOAs will be generated using the POV and the LM. The applicability of a potential TR for a relation will be measured by evaluating the corresponding field on its center of gravity.

We will consider two types of relations, being the FOAs generated by each type closely related: topological and projective. The *topological relations* ("near", "far" and "close to" in our model) are proportional to the distance between the LM and the TR. The *projective* ones ("left", "right", "front", "behind") define a canonical direction. In this case the applicability will decay with the angular deviation of the vector $\overrightarrow{LM, TR}$ from that direction.

The next idea we must take in consideration is that there are more than one possible frame of reference in which a relation can be interpreted [11]. In the *intrinsic* case the frame is defined exclusively by the LM orientation (Fig 1a). The front direction is determined by its physical or semantic characteristics (eg. the side in the direction of motion in a mobile object).

In the *extrinsic* case, the frame is defined by the positions of the POV and the LM (Fig 1b), being the front direction the one going from the LM to the POV.

In the performed experiments all the employed objects are cylindrical, thus they don't have an intrinsic front. In other case the influence of each frame of reference should be decided. Generally is accepted that in case of competition, the intrinsic frame dominates [12], though the concrete "weight" of each frame is not a decided matter.

3.2 Fields of Applicability

Their values goes from 0 to 1, being the applicability for a relation null if 0 and maximum if 1.

As the purpose of the FOAs is to specify objects positions in a 2D map, they will be defined in the euclidean plane. In the expressions where the POV, LM or TR appears, it must be understood that they refer to their projections on the XY plane.

Proximity Field. This field doesn't semantically correspond to any of the mentioned relations, but it will form part of the rest of the FOAs expressions. It expresses the concept of proximity between two objects and obviously decreases with the distance between them.

The reason for using the "proximity" instead of directly the distance between objects is that the former encompasses contextual factors which must be taken into account for the correct interpretation of the spatial relations. These are: (1) the size of the involved objects and (2) the size of the scene itself. For example,

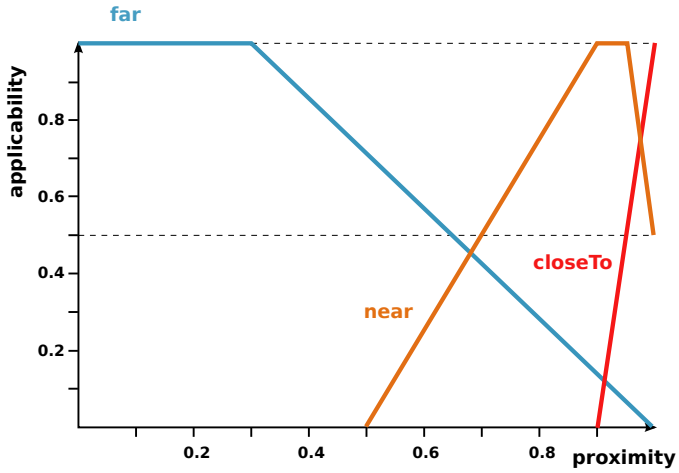


Fig. 2. Parametrization of the topological relations as a function of the proximity

the relation "near the house" clearly implies a larger area than "near the pen". By defining the "near" FOA in terms of the "proximity" instead of the distance this circumstance can be implicitly considered. In the same way, a correct interpretation of "near the house" must lead to a larger region if the considered scene is the whole city than if it's reduced just to the house neighborhood.

The former intuitive concepts are considered in the proximity expression as two constant factors which affect the proximity rate of decay with the distance (d_{max}, a_n). In turn, this decay is modeled as a linear function:

$$proximity(LM, P) = \begin{cases} 1 - \left(\frac{d}{d_{max}}\right)/a_n \ , \frac{d}{d_{max}} \leq a_n \\ 0 \ , \frac{d}{d_{max}} > a_n \end{cases}$$

$$a_n = \frac{a}{a_{max}}$$

where 'P' is a point of the plane, 'd' the euclidean distance between 'P' and the closest point of the LM and 'a' the area of the LM. 'a_n', the normalized area, expresses the fact that the proximity must decay slower for larger LMs [4]. It's defined as the area of the LM divided by the area of the larger object in the scene ('a_{max}').

'd_{max}' corresponds to the maximum distance in the scene. This factor expresses that the same distance must correspond to larger proximity values in larger scenes.

Projective Relations. The values for the FOAs generated from these relations will decay with two factors. (1) the angle α between the vector $\overrightarrow{LM, P}$ and the canonical direction. (2) the distance with the LM. The latter assertion is supported by Kelleher et Al. [4], but intuitively projective relations "loose" definition with the distance. As a extreme case, if a chair is 20 Kms far from a

table it would never be said to be on the left of it, independently of the angle they form.

The canonical direction for each relation in the extrinsic and intrinsic cases are the ones in the Fig 1 [11]. Once set the canonical direction, the expression of the field for all the projective relations is:

$$projective(POV, LM, P) = \begin{cases} \left[1 - \frac{\alpha}{\alpha_{max}}\right] \cdot proximity(LM, P) \ , \alpha < \alpha_{max} \\ 0 \ , \alpha > \alpha_{max} \end{cases}$$

where $\alpha_{max} = 90^{o}$. As it was argued in the former subsection, the proximity was used instead of the distance between P and the LM. The decreasement with the angular deviation has been modeled as a linear function of the angle α.

Topological Relations. The topological relations have been modeled as lineal parametrizations of the proximity, in accordance with the interval of distances for which each of them is conceptually acceptable ("close to" for very short distances, "near" for mid-length distances and "far" for points very separated from the LM).

The parametrizations used in the model are the ones represented in Fig 2. They were fitted in order to improve the interpretation of the descriptions in the performed experiments.

3.3 Virtual Objects

When the person describes the scene, the robot just knows about the position of the objects from the relations made in the description. For taking this circumstance into account we introduce the concept of *virtual objects* in contrast with the sensed ones, ie., the ones acquired from the sensors.

The position of these objects, created from the description, is defined by the spatial relations in which they are the TR. For example, in "there is a chair in front of me" the position of the chair is defined by "in front of me".

As the FOAs represent the distribution of a spatial relation applicability, it's natural to express the uncertainty in a virtual object position in function of the FOAs corresponding to the relations in which it was TR.

Concretely, we define the probability density function for the center of gravity of a virtual object 'TR' of being located at the point 'P' if its position was specified by the spatial relation 'rel' as:

$$f_{TR}(P|POV = Q, LM = R) = \frac{1}{n} rel(POV, LM, P)$$
$$n = \int_P rel(POV, LM, P) dP$$

being 'rel(POV,LM,P)' the FOA corresponding to the spatial relation 'rel' and 'P,Q,R' points of space.

Three things must be noted in the former expression. (1) there is a conditional dependence with the positions of the the POV and LM. This is natural, as those positions appear in the expression of the FOAs. (2) The integral in the normalization of the FOAs should be evaluated over the region corresponding

to the interior of the room, representing that the probability for a virtual object of being inside the room in which the description is performed must be 1. (3) It's normalized and it just can take non-negative values, hence fulfilling the conditions for being a density function.

If the POV or the LM are also virtual objects, we will need to extract the marginal distribution for the TR position by making use of the law of total probability:

$$f_{TR}(P) = \frac{1}{n} \iint_{Q,R} rel(POV, LM, P) f_{POV}(Q) f_{LM}(R) dR dQ$$

The former expression should be autonomously evaluated by the robot for any relation, POV and LM when interpreting a description. As a practical workaround, we opted for discretizing the probability distribution for the virtual objects position. In this way, $f_X(P)$ turns into $p(X = P)$ and the integral into a double summation.

The discretization was made by evaluating the FOAs over a grid covering the scene region (ie. the interior of the room).

The resulting expression for the marginal distribution is:

$$p(TR = P) = \frac{1}{n} \sum_Q \sum_R rel(POV, LM, P) p(POV = Q) p(LM = R)$$
$$n = \sum_P rel(POV, LM, P)$$

where 'P,Q,R' are points of the grid.

In the worst case (POV and LM virtual), when processing a spatial relation the FOA must be evaluated for each grid point being the POV and the LM also at any grid point. This makes the algorithm to be $O(n^3)$ with the number of grid points.

As an approximated solution to the former marginal distribution expression, we can take the assumption that the POV and the LM are located at their mean positions according to their own distributions. That is, to assume that all the terms in the summation in the marginal distribution expression are zero but the one in which $p(POV = P_{POV})$ and $p(LM = P_{LM})$, where P_{POV} and P_{LM} are the mean positions for the POV and the LM.

With this approximation the algorithm for processing a spatial relation turns to be $O(n)$ with the number of grid points. In the experiments the exact and the approximate solutions were tested in order to decide if the first one deserves its higher time complexity.

Regarding how theoretically appropriate is taking the former assumption, it's partially supported by the "gricean principle" which states that when an utterance is given from a speaker to a listener both of them expect the contextually most typical interpretation of the utterance [13]. In our case this principle can be translated as that the speaker will give a reference expecting the listener to "imagine" the objects in their more typical positions, ie. in the ones corresponding to the maximums of applicability for the relations which defined their positions.

Finally, we have to consider that the spatial relations included in the model just give information about the relative positions between the implied objects,

ie. they say nothing about their size or orientation. In our model this implies that as all the information about the virtual objects comes from the description there will be a complete uncertainty in these magnitudes.

In the expression of the "proximity" field it appears the distance to the LM and its area. These parameters obviously depend on the concrete geometry of the LM which in turn depends on the orientation and size (jointly with the position and the type). Thereby, some assumptions must be taken. These will be that the virtual objects are of a "standard size" (see the architecture subsection) for the computation of the area and that they are point objects for the computation of the distance.

3.4 Composition Model

In a description, more than one relation can be provided for the same virtual object in order to better specify its position. That is, they can be referred more than one relation with the same TR. Therefore it's needed to determine how this spatial information will be composed.

Two relations with the same TR can be given with two purposes: (1) to delimit the region in which it can be located. This case corresponds to an intersection between the regions in which each of the relations is applicable. In logical terms it will be a conjunction (eg. "the ball is on the right on the table and close to the wall"). (2) to enlarge the region in which it can be located. This case corresponds to a union or disjunction (eg. "The ball is on the right of the table or on its left").

We will suppose that when two relations are referred for the same TR the intention is to concrete its position. Hence, in this case it will be performed a conjunction. In contrast, when the LM in a relation is ambiguous (eg. "near the wall" if there are more than one wall), a constructive composition (disjunction) will be performed for that relation being the LM each of the possible candidates (eg. each of the walls in the former example).

Regarding how these two compositions are performed, it can be remembered from the last paragraph that from a spatial relation in which the TR is a virtual object, it can be deduced a probability density function for its position ($f_{TR}(P)$). When two relations are given for the same TR we will have two different densities:

$$f^1_{TR}(P) = \frac{1}{n} \iint_{Q,R} rel_1(POV_1, LM_1, P) f_{POV_1}(R) f_{LM_1}(Q) dR dQ$$
$$f^2_{TR}(P) = \frac{1}{n} \iint_{Q,R} rel_2(POV_2, LM_2, P) f_{POV_2}(R) f_{LM_2}(Q) dR dQ$$

An intersection (or conjunction) will correspond to the joint probability density function, and a union (or disjunction) to the union of the two density functions.

Before concreting the expressions for the composition a reflection must be done about the independence of the composed densities. Each of them represents the probability for the same virtual object of being located at a certain point of space given a different spatial relation. Or more precisely, they represent the probability for the relations which define the respective densities of being fulfilled when the object is located at a certain point. With the latter interpretation,

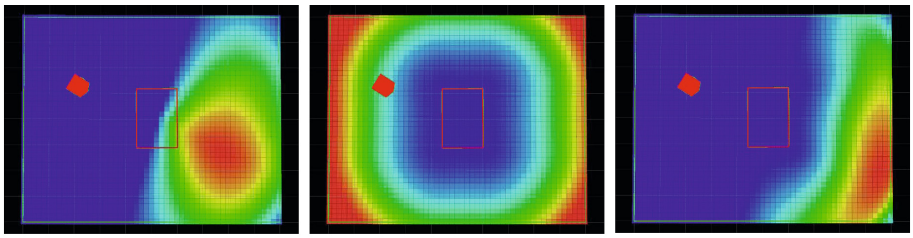


Fig. 3. Probabilities distributions for and object a) "behind the table", b) "far from the table" and c) and the conjunction of the two previous ones. Red color corresponds to the maximum value in the distribution and blue to the minimum. The table is the hollow rectangle and the POV the filled polygon.

and remembering the expressions of the FOAs, it can be stated that the fact of fulfilling one of the relations doesn't directly conditions the fulfilling of the other. Although there can be an indirect dependence. For example, the statement "the chair is on the left of the table" indirectly conditions the veracity of the statement "the chair is on the right of the table", but in terms of the definitions of the FOAs which represent both relations, there is no dependence.

Basing in the former argument, we will suppose the density functions corresponding to different spatial relations to be independent. This statement leads to the following expressions for the composition of two relations:

$$f^1_{TR}(P) \bigcap f^2_{TR}(P) = f^1_{TR}(P) \cdot f^2_{TR}(P);$$
$$f^1_{TR}(P) \bigcup f^2_{TR}(P) = f^1_{TR}(P) + f^2_{TR}(P) - f^1_{TR}(P) \bigcap f^2_{TR}(P);$$

In the discrete case (ie. discrete probability distributions) the expressions for the the composition are analogous to the latter. In Fig 3 is shown the distribution for a TR which is "behind the table" and "far from the table" (ie. the final distribution is the intersection of the former ones).

4 Collaborative Scene Mapping

In this section we overview the method used for building the scene maps described in the introduction and the architecture of the implemented system. The presented collaboration procedure tries to exploit the inherent capacities of each of the parts: the robot and the human. The robot can contribute with the accuracy of its sensors and its computational capacities for processing and composing information. On the other hand, the human have a unique facility in the segmentation and classification of objects. He also may possesses background knowledge and information about particularities or functionality of the objects in the environment to map.

In first place the person describes the scene using the contemplated set of spatial relations and the robot forms a mental map of it by interpreting those

relations according to the presented model. This mental map provides information about the type of the objects in the scene and the region where they are most probably located. When sensing the scene, this information is used for sense guidance. That is, the robot looks for the actual objects in the regions corresponding to the covariance ellipses of the virtual objects in the mental map. Also, as it's been said, after sensing the scene the robot tries to match these virtual objects with the sensed ones, linking the semantic information associated to the former to the latter.

4.1 Architecture

The system has three main components: (1) a spatial relations library which, given a relation, a POV and a LM (sensed or virtual), evaluates the corresponding FOA over a 2D grid following the model described in the previous section. (2) A simple parser which translates the descriptions given by the human. (3) A Geometric Scene Description component (GSD), which keeps a list of the scene objects and actualize the information about them when new relations are processed. The objects can be added to the GSD in two ways, being perceived by the robot sensors (sensed objects) or introducing them in the description (virtual objects).

Each object in the list is an instance of a type of object (eg. chairs) which has associated a 3D mesh used by the spatial relations library in the computation of the FOAs. Each instance has a pose, a covariance, an a scale. The scale is set as the proportion between the dimensions of the actual object and the 3D mesh corresponding to its type. The virtual objects will be considered to be of a "standard size", ie. for them the scale will be set to one.

The virtual objects in the GSD have also a grid corresponding to their position probability distribution. From this distribution is extracted a mean position and a covariance matrix used in the matching of the robot mental and sensed maps.

4.2 Robot Mental Map: Processing the Descriptions

In the first part of the mapping process the person describes the spatial layout of the scene without the robot being there. The idea is that as he provides relations for the same TR he will be imposing additional constraints over its position which will reduce its covariance.

In order to ensure a correct interpretation of the description, the person must follow some rules. These are:

1. Before making any movements he must specify them by describing his target position in the same way as he would do with any other object, ie. using the same spatial relations (eg. "I move in front of the second table"). This ensures a correct position for the POV in the interpretation of the relations.
2. He must specify his intrinsic orientation when providing relations in which he is the LM (eg. "The table is on my left"). He will do that by telling where is he looking when referring the relation (eg. "If I look to the back wall, the

table is on my left"). This ensures a correct definition of the intrinsic frame in these cases.
3. The objects involved in a spatial relation must be unequivocally specified. That is done by naming them with its type and the ordinal number corresponding to their order of appearance in the description (eg. "the second chair is on the right of the first table").

The robot mental map is formed by all the virtual objects in the GSD after processing the whole description.

4.3 Enriched Map: Scene Check and Map Refinement

In the next step the robot senses the scene by using its 2D range laser. For this task the robot is supposed to be able to detect and classify the objects present in the scene (at least the ones introduced in the description) and to localize itself inside the scene (ie. in a map of an empty room with the known dimensions).

As it's been said, in order to ease this process the robot uses the formerly computed mental map for guiding the sensing. This is performed by following the next method: after coming into the room, the robot places itself in the middle of it and direct the laser to the regions corresponding to those in the mental map where the virtual objects are more probably located. That is, to those regions corresponding to the covariance regions of the virtual objects.

The sensed map will be formed by all the sensed objects in the GSD after the sensing process.

After sensing the scene, the robot performs the matching of the virtual objects with the sensed ones. To achieve an assignment (called objects fusion) the system looks for three requirements:

1. The sensed and virtual object are of the same type.
2. The probability for the virtual object of being at the sensed object position is greater than a certain threshold (according to its final probability distribution). For the experiments this threshold was set to the half of the maximum in the probability distribution.
3. For each relation in the description in which the virtual object acts as LM, there is a sensed object of the TR type with an applicability greater than a threshold when the relation is evaluated using the candidate sensed object as LM. This threshold was set to 0.5. This second check helps to prevent wrong fusions in complex scenes (when there are several objects of the same type).

If the three requisites fulfill the sensed object is taken as a candidate for the fusion. From all the candidates is chosen the one closest to the mean position of the virtual object. Once a fusion is accomplished, the semantic information associated with the virtual object is added to the sensed one and the former is deleted from the GSD. The final position for the object is the one corresponding to the sensed object.

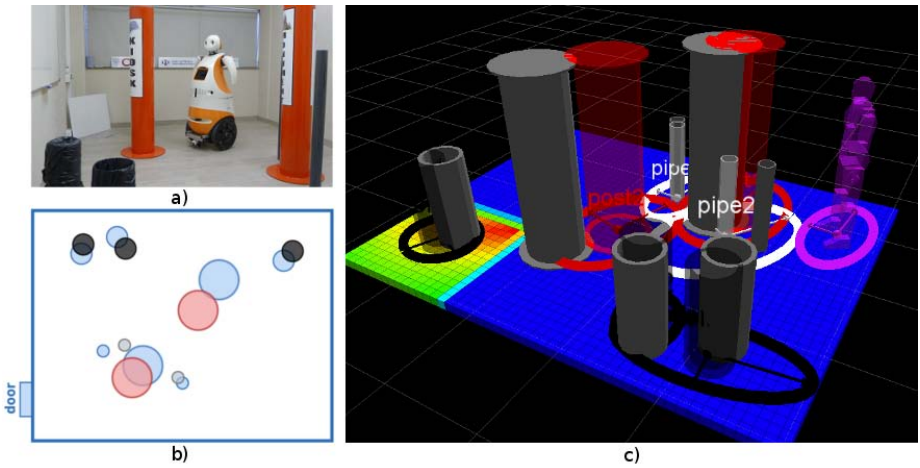


Fig. 4. a) Robot "Tibi" sensing the second scene. b) 2D view of the virtual map showed in the picture (c) contrasted with the ground truth (blue objects). c) Mental and sensed maps for the second scene used in the experiments. The sensed objects are the gray ones. The virtual ones has the color the person told in the description. They are visualized the covariance ellipses for the virtual objects.

If after trying to fuse all the virtual objects there are still any of them in the GSD the robot infers that the sensed scene doesn't correspond to the described one. As at the time being there is no mechanism to detect wrong fusions, if one virtual object is fused with a sensed object not corresponding to it, that will probably prevent the correct fusion of the rest of the objects. Anyways this never happened in the performed experiments.

The objects in the GSD after the fusion forms the enriched map. In our experiments we symbolize the semantic information given by the person with the color of the objects (as the robot is using a range laser which provides no color information). That is, with the "name" of the color of the objects as the person perceives them (eg. "red"). In the maps an RGB value is associated with each color just for visualization purposes.

5 Experiments

The conducted experiments were focused on testing the accuracy of the robot mental maps and the efficiency when matching them with the sensed ones. For that mean, we placed a set of objects inside a room and asked ten people among the researchers of the institute to describe the scene using the relations specified in section 3 and following the rules in subsection 4.2. The only information the robot had about the scene were the dimensions of the room, that there was a door and the location of the latter. The initial position of the person describing the scene was supposed to be "close to the door".

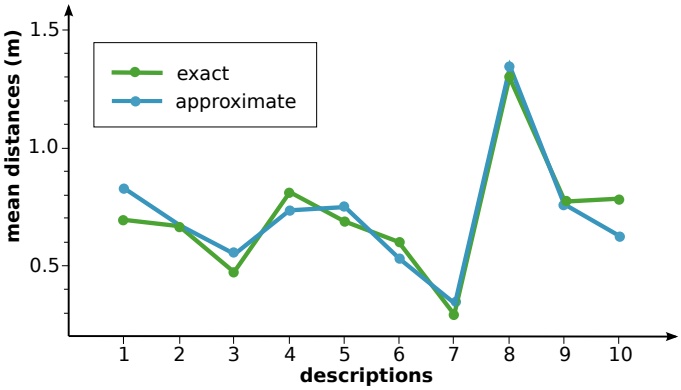


Fig. 5. Mean distances from the virtual objects to the ground truth in the ten descriptions processed for the scene2

From each description the robot built a map by generating and composing each relation and afterward came into the scene, sensed it and performed the matching. The objects could be of three types: post, bin or pipe. Two scenes were tested. A simple one with one object of each type (scene1). A more complex one with several objects of each type (scene2).

As an example of the provided descriptions, next is shown the one from which was computed the most accurate mental map of the second scene (description 7 in Fig 5).The built mental map is shown in Fig 4.

"There is a red post in front of me. The post is near me. There is a white pipe behind the post. The pipe is close to the post. There is a black bin on my left. The bin is close to the left wall. There is a black bin on my left. The second bin is close to the left wall. The second bin is close to the first bin. The second bin is on the right of the first bin. There is a black bin close to the back wall. The third bin is close to the left wall. There is a white pipe near me. If I look to the third bin the second pipe is in front of me. There is a red post near the back wall. The second post is near the left wall. If I look to the third bin the second post is in front of me. The third bin is behind the second post."

The descriptions were processed using exact and approximate solutions when generating FOAs from virtual POV or LM (subsection 3.3).

The results were evaluated according to three parameters:

1. Mean distance from the objects in the robot mental map to the ground truth for all the objects in all descriptions ("distance" in table 1).
2. Mean covariance in the virtual objects position, expressed as the length of the radius of a circumference of the same area as the covariance ellipse, $r = \sqrt{area/\pi}$ ("covariance").
3. The percentage of times that the robot succeed to match the robot mental and sensed maps ("matches").

Table 1. Mean distances, mean covariances and percentage of success in the matching of the virtual and sensed maps for the two scenes using the two approaches for generating the FOAs

		distance (m)	covariance (m)	matches (%)
approximate approach	scene1	0.9 ± 0.2	0.6 ± 0.1	90
	scene2	0.6 ± 0.2	0.6 ± 0.1	80
exact approach	scene1	0.9 ± 0.2	0.73 ± 0.09	100
	scene2	0.6 ± 0.1	0.73 ± 0.08	90

The results achieved are summarized in Table 1. There is no data about the performance in the sensing process as the robot achieved to detect all the objects in every case. This success is no doubt due to the simplicity of the geometry of the chosen object. This choice was made in order to isolate the assessment of mental map and fusion processes from possible errors in the object detection.

The first thing we realized after conducting the experiments is that not all the people have the same "descriptive skills" (or the same understanding of space). As it can be seen in Fig 5 the results are very description dependent, going the mean distances from 0.29 m to 1.43 m (though almost all of them stay in the interval 0.4 - 1.0).

For using the mental map as a tool by itself the interesting numbers are those corresponding to the mean distances and covariances. But for using it as an intermediate step in the mapping process (as done in this article), the most relevant data is the percentage of matches. The accuracy of the mental map is important for guiding the sense and fusing the objects. But if after all the robot achieve to match the two maps, the final object positions will be the sensed ones. In this sense, the success percentage is around 90 %.

Regarding the options in the FOAs generation, the mean distances to the ground truth were the same when using the exact and approximate solutions in the generation of FOAs from virtual objects. Though the former option tend to generate larger covariances, which helps to improve the matches percentage. On the other hand, the larger covariances could lead to wrong fusions. But this didn't happen in the conducted experiments.

6 Conclusions

We have presented a model which let a robot to build the map of a scene in collaboration with a person. To that end, the person must describe to the robot the layout of objects in the scene using qualitative spatial relations being able to include semantic information about the objects in the description. An extension of the model which let to express more precise information about distances and orientations would no doubt improve the accuracy of this map. Although we plan to research in that direction in the future, in the present work we preferred to limit ourselves to a more ordinary language which people usually use when expressing spatial knowledge.

The mental map built from the description provides information about the type of the objects in the scene and the region where they are most probably located. For the time being just the position is used for guiding the sensors. An interesting expansion for a future work would be to study the joint use of the type and position information to improve the segmentation process. Also taking into account possible occlusions in the sense guidance would make the method more robust.

Finally, the mental and sensed maps are fused, being trespassed the semantic information in the virtual objects to the sensed ones.

Acknowledgments. This work has been partially funded by Spanish Ministry of Economy and Competitiveness under project TaskCoop DPI2010-17112.

References

1. Moratz, R., Fischer, K., Tenbrink, T.: Cognitive modeling of spatial reference for human-robot interaction. International Journal on Artificial Intelligence Tools (IJAIT) 10(4), 589–611 (2001)
2. Stopp, E., Gapp, K.-P., Herzog, G., Laengle, T., Lueth, T.C.: Utilizing spatial relations for natural language access to an autonomous mobile robot (1994)
3. Gapp, K.-P.: Basic meanings of spatial relations: Computation and evaluation in 3d space (1994)
4. Kelleher, J., van Genabith, J.: A computational model of the referential semantics of projective prepositions. In: Saint-Dizier, P. (ed.) Syntax and Semantics of Prepositions. Text, Speech and Language Technology, vol. 29, pp. 211–228. Springer, Netherlands (2006), http://dx.doi.org/10.1007/1-4020-3873-9_14
5. Mavridis, N., Roy, D.: Grounded situation models for robots: Bridging language, perception, and action. In: Proceedings of the AAAI 2005 Workshop, pp. 32–39 (2005)
6. Coyne, R., Sproat, R.: Wordseye: an automatic text-to-scene conversion system. In: SIGGRAPH, pp. 487–496 (2001)
7. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. In: UAI, pp. 435–461 (1986)
8. Dissanayake, G., Newman, P.M., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (slam) problem. IEEE Transactions on Robotics 17(3), 229–241 (2001)
9. Thrun, S., Burgard, W., Fox, D., Hexmoor, H., Mataric, M.: A probabilistic approach to concurrent mapping and localization for mobile robots. In: Machine Learning, pp. 29–53 (1998)
10. Aydemir, A., Sjöö, K., Folkesson, J., Jensfelt, P.: Search in the real world: Active visual object search based on spatial relations. In: To appear in Proc. of the IEEE International Conference on Robotics and Automation, ICRA 2011 (2011)
11. Retz-Schmidt, G.: Various Views on Spatial Prepositions. AI Magazine 9(2), 95–105 (1988)
12. Miller, G.A., Johnson-Laird, P.N.: Language and perception. Harvard (1976)
13. Grice, H.: Logic and conversation. In: Cole, P., Morgan, J.L. (eds.) Syntax and Semantics: Speech Acts, pp. 41–58. Academic Press (1975)

Probabilistic Performance Evaluation for Multiclass Classification Using the Posterior Balanced Accuracy

Henry Carrillo[1], Kay H. Brodersen[2], and José A. Castellanos[1]

[1] Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, C/ María de Luna 1, 50018, Zaragoza, Spain
{hcarri,jacaste}@unizar.es

[2] Translational Neuromodeling Unit, Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology (ETH Zurich), 8032 Zurich, Switzerland
brodersen@biomed.ee.ethz.ch

Abstract. An important problem in robotics is the empirical evaluation of classification algorithms that allow a robotic system to make accurate categorical predictions about its environment. Current algorithms are often assessed using sample statistics that can be difficult to interpret correctly and do not always provide a principled way of comparing competing algorithms. In this paper, we present a probabilistic alternative based on a Bayesian framework for inferring on balanced accuracies. Using the proposed probabilistic evaluation, it is possible to assess the balanced accuracy's posterior distribution of binary and multiclass classifiers. In addition, competing classifiers can be compared based on their respective posterior distributions. We illustrate the practical utility of our scheme and its properties by reanalyzing the performance of a recently published algorithm in the domain of visual action detection and on synthetic data. To facilitate its use, we provide an open-source MATLAB implementation.

Keywords: multiclass classifiers, accuracy, balanced accuracy, probabilistic performance.

1 Introduction

A central theme in the development of intelligent, autonomous robots has been the challenge of decision problems. Typical examples include the critical tasks of object detection [3,14], scene recognition [8,27], active SLAM [10,11] or loop closing [15,17]. All of these domains have seen significant progress in the development of increasingly accurate classification algorithms.

By contrast, there has been less focus on the evaluation of the *performance* of such algorithms. Assessing the performance of a given classifier is crucial as it allows us to (*i*) obtain an interpretable estimate of the degree to which its results generalize to unseen examples from the same distribution from which the existing data were drawn, (*ii*) compare competing approaches, and (*iii*) tune the (hyper)parameters of a classifier in light of the estimated performance in a given domain.

A common basis for evaluating the performance of a classifier is the confusion matrix. It provides a summary of classification outcomes and permits the inspection of

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_25,

the number of correct and incorrect predictions in each class. However, in the absence of an appropriate summary statistic, reporting a confusion matrix by itself is generally insufficient and easily leads to highly subjective interpretations of performance.

Commonly used summary statistics that are based on confusion matrices include the overall sample accuracy; the per-class, or balanced, sample accuracy; the Kappa coefficient; and the F_μ-score. Other statistics include the area under the receiver-operating characteristic (ROC) curve and the area under the precision-recall (PR) curve, although these are typically limited to two-class (binary) classification problems (see [21] for a generalization).

While all of the above performance metrics can be helpful in understanding the behaviour of a classifier, the key quantity of interest in most practical domains of application is the generalization ability, i.e., the probability of the classifier to make a correct prediction on an unseen example. It is tempting to try and answer this question by considering the *accuracy* of a classifier alone. However, classification accuracy is a misleading performance indicator when the data are not perfectly balanced [1,6,12,18].

A straightforward way of resolving the above limitation is to replace the accuracy by the *balanced accuracy*, defined as the arithmetic mean of class-specific accuracies. Critically, however, it is not sufficient to report the mean of class-specific *sample* accuracies. Rather, we must infer on the latent class-specfic accuracies of which the observed sample accuracies are an instantiation. Inferring on the balanced accuracy then means, for example, to report a point estimate as well as a measure of uncertainty about this estimate.

This paper describes a simple Bayesian framework for assessing the performance of classifiers. The proposed model makes it possible to compute the full posterior distribution of the balanced accuracy given the available classification outcomes. This approach extends previous work [6, 7] by providing a generalization to multiclass classifiers. In addition, we suggest a concrete method for comparing two balanced accuracies based on the posterior distribution of their difference. This method allows one to rank competing classifiers in a probabilistically interpretable fashion.

Using a Bayesian model for multiclass balanced accuracies offers three strengths over previous schemes: (*i*) the useful properties of the balanced accuracy are generalized to a multiclass setting; (*ii*) a Bayesian perspective allows us to explicitly incorporate prior knowledge (e.g., domain-specific information or a cost function that assigns a measure of importance to each class), account correctly for posterior uncertainty, and easily derive other posterior inferences; (*iii*) the model enables cross-algorithm comparisons that correctly account for the posterior uncertainty about each algorithm's performance.

The paper is structured as follows: Section 2 briefly reviews the merits of a Bayesian approach to performance evaluation. Section 3 provides a concrete example of a decision problem from the domain of robotics, followed by a brief overview of previous approaches to performance evaluation. Section 4 develops the proposed Bayesian model for performance evaluation in multiclass classification and a method for comparing competing classifiers. Section 5 presents a set of experiments in order to characterize the properties of the approach and illustrate its application. We conclude the material in Section 6 with a brief discussion.

2 Bayesian Inference on Classification Performance

In most situations, evaluating the performance of a classifier aims at characterizing the classifier's ability to predict the correct class of data that has not yet been seen. Abstracting away from specific implementations, we can denote the performance of a classifier by the variable $\lambda \in [0, 1]$. The two limits 0 and 1 refer to the ability of making incorrect or correct predictions on all future instances, while 0.5 refers to classifications at random. Despite the advantages of a Bayesian framework, λ has mostly been evaluated by adopting a classical, or frequentist, approach to inference.

Classical inference considers distributions over data but does not permit distributions over parameters such as λ. As a result, it is restricted to point estimates, $\hat{\lambda}$, and, most commonly, 95% confidence intervals, representing the interval in which the true value would be in 95% of cases if the experiment was repeatedly carried out an infinite number of times. A 'test' is then carried out by asking whether the value of a summary statistic (i.e., a t-score), or a more extreme value, could be observed under a 'null' hypothesis. The main advantage of point estimates, confidence intervals, and hypothesis tests is their computational simplicity. However, their correct interpretation is prone to errors [4].

In a Bayesian framework, inference proceeds by passing from a prior distribution, $p(\lambda)$, to a posterior distribution, $p(\lambda|\mathcal{D})$, that is informed by the data y. Depending on the given cost function, the posterior mean, mode, or median then replaces classical point estimates. Posterior intervals replace confidence intervals. And Bayesian model comparison replaces hypothesis tests. The main advantage of Bayesian inference is its conceptual simplicity (providing a probabilistic statement about the quantity of interest rather than providing a sampling statistic about a summary statistic) and the flexibility with which posterior inferences can be summarized [16]. A downside is that Bayesian inference is often computationally more complex than classical inference. In the application considered in this paper, however, this is not an issue, since the classifier evaluation is usually small-scale and carried out offline.

3 A Motivating Example

To illustrate the importance of assessing overall classification performance, we consider the example of a service robot, designed to clean the dishes in a kitchen. A core component of such a robot is the capability of visually classifying objects as a 'mug', a 'sink', or a 'bottle of wash-up liquid'. Clearly, any classification algorithm designed for this task must be proficient at detecting all three types of object, since all are required to complete the task. Thus, we are interested in the overall performance of the classifier (i.e., the variable λ) rather than its performance on individual classes.

3.1 Confusion Matrices

A common way of reporting classification results, especially in a multiclass setting, is to compile a confusion matrix, also referred to as a contingency table or accumulation matrix. Let each element $x_{r,c}$ of a confusion matrix $C \in \mathbb{N}^{l \times l}$ represent the number of

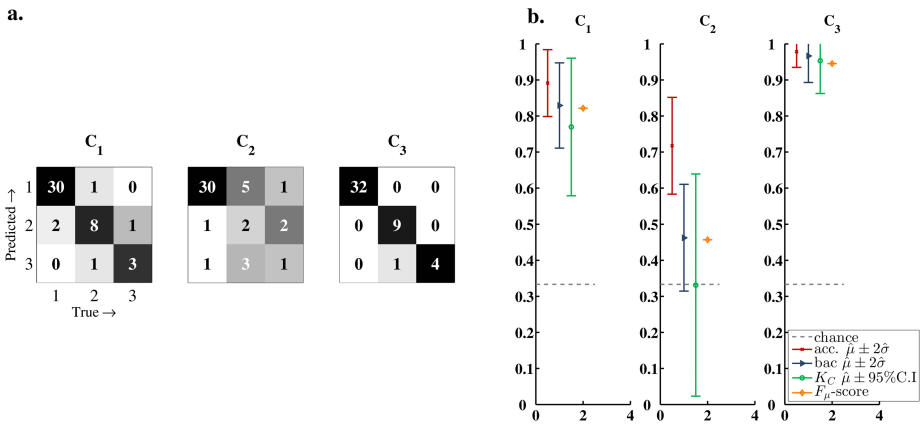


Fig. 1. (a) Three example confusion matrices, summarizing the classification outcomes described in Section 3. (b) Conventional performance metrics with classical error bars stemming from the standard error of the mean. To reproduce these results, see `MotivatingExample.m`, available online [9].

times a classifier predicted class r when the true class label was c. Thus, diagonal and off-diagonal elements indicate the number of correct and incorrect predictions, respectively.

Let us suppose that three competing classifiers were tested on a given dataset, resulting in three matrices C_1, C_2, and C_3 (see Fig. 1). Using a graphical representation of the matrices, one can easily obtain an intuitive sense about which classifier performed best (C_3). However, such an assessment remains vague and does not obviate the need for a quantitative evaluation. For instance, how confident are we that classifier 2 performed better than chance? How certain is it that classifier 3 outperformed classifier 1? Several metrics have been proposed to address such questions. In the next part, we will review some of these, focusing on their properties regarding the assessment of prediction accuracy of a classifier.

3.2 Performance Metrics

The literature on performance metrics on the basis of confusion matrices is large and diverse, comprising both frequent propositions of new statistics and the development of statistical models for their estimation. Here, we briefly consider some of the most common metrics and outline where balanced accuracies fit in.

The most common statistic for reporting the performance of a multiclass classifier is its *sample accuracy* (acc), defined as the number of correct predictions across all classes, k, divided by the number of examples, n. While conceptually simple, assessing performance using the sample accuracy alone has long been known to be prone to erroneous interpretation. This is because the accuracy does not account for the degree of class imbalance that may be present in a given dataset [1, 6, 12, 18], which means it can only be correctly interpreted in relation to a dataset-dependent baseline accuracy.

An example of the misleading nature of inferences based on accuracies can be seen in the results of the example in Fig. 1b. The plot shows that the sample accuracy of C_2 is close to that of C_1, despite C_2 misclassifying classes 2 and 3 most of the time.

In robotics, a common way of overcoming the above limitation is to resort to the F_μ-*score* [24, p. 183], defined as the across-classes average of the F_β-*score* [26]. The F_β-*score* itself, frequently used in binary classification, is given by the β-weighted average between precision and recall. Setting $\beta = 2$, as is commonly done, yields the harmonic mean of precision and recall. Thus, as depicted in Fig. 1b, the F_μ-score accounts for the degree of class imbalance in the test examples. At present, however, there has been no established convention of computing its corresponding confidence or credible intervals.

Another approach to overcoming the limitation of sample accuracies is based on the *Kappa coefficient* [13], which has been one of the dominating metrics in the domain of remote sensing. It quantifies the degree of overall agreement within a given confusion matrix $C \in \mathbb{N}^{l \times l}$,

$$K_c = \frac{p_0 - p_e}{1 - p_e} \tag{1}$$

with

$$p_0 = \frac{1}{l} \sum_{i=1}^{l} k_i, \quad p_e = \frac{1}{l^2} \sum_{i=1}^{l} C_{i+} \times C_{+i}, \tag{2}$$

where k_i is the number of correct predictions in class i and l is the number of classes. C_{i+} and C_{+i}, respectively, are the row-wise and column-wise sums of row and column i in the confusion matrix.

Like the F_μ-score, K_c accounts for the degree of class imbalance in the data. However, it can be invariant to the number of misclassifications and does not necessarily reflect what one would intuitively consider prediction strength [25].

An alternative is the *balanced accuracy* (bac), defined as the average accuracy obtained on all classes. In the case of a multiclass classification problem, its sample statistic is given by

$$\hat{\lambda} = \frac{1}{l} \sum_{i=1}^{l} \frac{k_i}{n_i}, \tag{3}$$

where k_i is the number of correct predictions in class i, l is the number of classes and n_i is the number of examples in class i. The balanced accuracy is frequently used in practice and has several conceptual strengths over the conventional accuracy while maintaining its simplicity. However, a probabilistic approach is not always being adopted when interpreting it, despite the fact that the limits of a frequentist confidence interval, as can be seen in classifier 3 in Fig. 1b, can easily lie outside of its $[0, 1]$ domain. (One possible remedy is to apply a z-transform prior to computing the interval.)

In summary, current approaches to multiclass performance evaluation face multiple challenges: (*i*) assessing performance on the basis of sample statistics does not replace principled probabilistic inference; (*ii*) error bars are often based on ill-justified distributional assumptions, such as in the case of classical confidence intervals without a z-transform; (*iii*) classification accuracy remains a popular metric even in those cases where an imbalanced dataset leads to misleading conclusions; (*iv*) alternatives to the

above, such as the use of a Bayesian framework [6, 7] have not yet been generalized to multiclass classification problems.

4 Theory

To help address the challenges outlined above, we describe a Bayesian approach to estimating the accuracy and balanced accuracy of a classifier in a multiclass setting. The adoption of a Bayesian perspective has long been considered helpful in this context (cf. [5, p. 68-74]; [24, p. 72-78]) and has in particular been described previously for binary classifiers [6, 7], where $C \in \mathbb{N}^{2\times 2}$. Here, we develop a generalization to the multiclass case where $C \in \mathbb{N}^{l\times l}$ with $l \geq 2$. In order to keep our treatment self-contained, we will begin with the binary case and then demonstrate its extension to multiclass classification.

It is worth pointing out that the approach in this paper differs from the implementation in [6, 7] in that we suggest a feasible strategy for computing the posterior distribution of λ in a multiclass setting. Our strategy is based on the characteristic function of the per-class posterior distribution and its Fourier transforms. This strategy is computationally more efficient than a direct extension of the previous implementation, which would otherwise require an l-fold convolution, where l is the number of classes in the classifier.

Another focus of the present paper is the comparison of competing algorithms based on the posterior distribution of the difference of their respective balanced accuracies.

4.1 Problem Statement

We consider the solution to a decision problem in which each one of n i.i.d. examples (or trials) is associated with a class label from a finite set of categories $\{1, \ldots, l\}$. We wish to assess the generalizability of the classifier. In others words, we wish to characterize the classifier's ability of predicting the correct class on future, unseen data, i.e., estimate the variable λ.

4.2 Solution Sketch

In a Bayesian framework, the performance of a classifier is considered a latent (unobservable) variable, and we use probabilities to express our uncertainty about classification performance before and after observing actual classification outcomes. Under this view, evaluating the performance λ of a classifier means passing from a prior distribution $p(\lambda)$ to a posterior conditioned on observed data $p(\lambda|\mathcal{D})$. The posterior encodes the plausibility of all possible true performance values in light of the observed data, and there are many ways in which it can be summarized, for example, in terms of the posterior mean, mode, or a posterior interval.

To model classification performance, we code correct predictions as $y = 1$ and incorrect predictions as $y = 0$. A classification result can then be viewed as a sequence of outcomes $y_1, \ldots, y_n$. Modelling each outcome as the i.i.d. result of a Bernoulli experiment, we obtain

$$p(y_i|\lambda) = \text{Bern}(y_i|\lambda) = \lambda^{y_i}(1-\lambda)^{1-y_i}, \tag{4}$$

where λ is the probability of any one trial being classified correctly. This implies that the number of correct predictions k in a sequence of trials $y_1, \ldots, y_n$ follows a Binomial distribution:

$$p(k|\lambda, n) = \text{Bin}(k|\lambda, n) = \binom{n}{k} \lambda^k (1-\lambda)^{n-k} \tag{5}$$

Finally, we express any available prior knowledge about classification performance by placing a prior on λ. A natural choice for this is to use the conjugate prior of the Binomial distribution, i.e., the Beta density. In the absence of any preceding classification results, we express maximal prior uncertainty (i.e., all values in the domain $[0 \ldots 1]$ of performance λ are considered equally plausible *a priori*) using a uniform distribution

$$p(\lambda) = \text{Beta}(\lambda|a, b) = \text{Beta}(\lambda|1, 1). \tag{6}$$

Thus, given the observed data k and n, we obtain the posterior performance as:

$$\begin{aligned} p(\lambda|k, n) &= \frac{\text{Bin}(k|\lambda, n) \times \text{Beta}(\lambda|1, 1)}{p(k)} \\ &= \text{Beta}(\lambda|k+1, n-k+1) \end{aligned} \tag{7}$$

This posterior encodes our knowledge about λ in light of the observed classification result. Critically, it goes beyond point estimates of performance (such as, e.g., the sample accuracy k/n), since it reflects how uncertain we are about our estimate. For instance, observing only very few classification outcomes will be correctly accounted for by a wide posterior distribution; whereas the observation of a large additional number of outcomes would cause the posterior to shrink to a more precise distribution.

Having described the model in general terms, we will now turn to the special cases of posterior accuracies and balanced accuracies in a multiclass setting, as described in the following two sections.

4.3 The Posterior Multiclass Accuracy

In what follows, we describe how to obtain the posterior distribution of the accuracy for a multiclass setting. It should be noted that this section merely serves as a preparation of the next section; using the accuracy to describe the performance of a classifier is often misleading and is discouraged [1, 6, 12, 18].

In order to infer on the posterior classification accuracy of a multiclass classifier, we can use the model described above as is. In the multiclass setting, the variable[1] θ then simply represents the probability with which an individual trial is classified correctly, i.e., classification accuracy. In other words, the posterior distribution of the overall accuracy is given by

$$p(\theta|k, n, a, b) = \text{Beta}(\theta|a+k, b+n-k), \tag{8}$$

where $a = 1$ and $b = 1$ encode our prior ignorance about the classifier's performance (see [19] for a comparison of alternative priors). The key point to note here is that the

[1] The variable θ represents the same as the variable λ, i.e., the performance of a classifier. We changed its notation to prevent abuse of notation in the next section.

availability of a full posterior distribution yields a plethora of useful ways of forming summary statistics. For example, we could report a central 95% credible interval,

$$\left[F^{-1}_{\mathrm{B}(a+k,b+n-k)}(\tfrac{0.05}{2});\ F^{-1}_{\mathrm{B}(a+k,b+n-k)}(1-\tfrac{0.05}{2})\right], \tag{9}$$

where $F^{-1}_{\mathrm{B}(\cdot)}(\cdot)$ is the inverse cumulative density function of the Beta distribution, evaluated at the desired quantile.

Alternatively, we could derive a point estimate of classification accuracy that minimizes the expectation of a given loss function. For example, the optimal point estimate under an ℓ_2-loss function is the posterior mean:

$$\langle\theta\rangle_{p(\theta|k,n,a,b)} = \frac{a+k}{a+b+n} \tag{10}$$

In contrast, the expected loss of a $(0,1)$-loss function is minimized by the posterior mode:

$$\arg\max_{\theta} p(\theta|k,n,a,b) = \frac{k+a-1}{a+b+n-2} \tag{11}$$

This shows that we can reinterpret the conventional sample accuracy k/n as the optimal estimate under a flat prior and a loss function that is $L(\theta,\hat{\theta}) = 0$ if $\theta = \hat{\theta}$ and 1 otherwise.

4.4 The Posterior Multiclass Balanced Accuracy

Classification accuracy, as defined above, is a misleading measure of performance when the data are not perfectly balanced. This is because a classifier may take advantage of an imbalanced dataset and trivially achieve a classification accuracy equal to the fraction of the majority class, and thus potentially much higher than the $1/l$ baseline. Put differently, the baseline accuracy that can always be achieved by a classifier, even in the case of zero mutual information between data features and class labels, depends on the degree of class imbalance; it is not always $1/l$ (i.e., 0.5 in the case of binary classification).

This issue can be resolved by replacing the accuracy by the balanced accuracy, i.e., by the arithmetic mean of class-specific accuracies,

$$\lambda := \frac{1}{l}\sum_{i=1}^{l}\theta_i, \tag{12}$$

where θ_i is the (latent) accuracy of the classifier on class i. When the data are perfectly balanced (i.e., the data contain the same number of examples from each class), the balanced accuracy reduces to the conventional accuracy. Critically, however, its baseline performance is always $1/l$, regardless of the degree of class imbalance. Thus, if a classifier has achieved class-specific accuracies above $1/l$ only by exploiting the class imbalance, its balanced accuracy will drop to $1/l$, as desired [7].

Under a Bayesian perspective, we wish to pass from a prior distribution over the balanced accuracy to a posterior distribution in light of the observed classification outcomes $\mathcal{D} = \{(k_1,n_1),\ldots,(k_l,n_l)\}$. We have seen in (8) how we can obtain the posterior distribution of the overall accuracy. Thus, to obtain the posterior balanced accuracy, we first apply (8) to each class in turn; we then find the conditional distribution over λ.

Computing $\boldsymbol{\lambda}$. In [6, 7], a convolution is used to compute λ for the binary case. The direct extension of its convolution approach would require an l-fold convolution, where l is the number of classes in the classifier. This would in turn require the numerical computation of an l-dimensional integral, which can be both computationally complex and numerically unstable.

An alternative is to consider the inverse Fourier transform of the products of the characteristic functions of the individual distributions, as described next.

The probability distribution of a given random variable θ is fully specified by its characteristic function Φ_θ which is given by the Fourier transform $\mathcal{F}\{\theta\}$ [22, p. 145]. Thus, owing to the product property of the Fourier transform, the convolution of two functions is identical to the product of the functions' Fourier transforms. In the context of classification, we can exploit this fact to obtain the posterior distribution of the balanced accuracy as

$$p(\lambda|\mathcal{D}) = \mathcal{F}^{-1}\{\Phi_{\check{\theta}_1} \times \cdots \times \Phi_{\check{\theta}_l}\}, \tag{13}$$

where $\Phi_{\check{\theta}_i}$ is the characteristic function of $\check{\theta}_i := \frac{1}{l}\theta_i, \forall i = 1 \ldots l$, and where $\mathcal{F}^{-1}\{\cdot\}$ is the inverse Fourier transform [22, p. 272]. This step is facilitated by the fact that the posterior distributions of all individual class-specific accuracies, $p(\theta_i|k_i, n_i)$, are Beta densities and can be obtained using (8).

Just as in the case of the posterior accuracy, it is useful and important to obtain summary statistics, such as the mean, mode, or a credible interval. In contrast to the closed-form expressions we saw in the previous section, **analytical solutions for balanced-accuracy statistics are not available**. To address this limitation and facilitate their use, we provide a numerical implementation in MATLAB.[2]

4.5 Comparing Competing Multiclass Classifiers Using Their Posterior Performance Distribution

Given the posterior distribution of the performance of one classifier $p(\lambda_1|\mathcal{D})$, a critical question is how this given classifier compares to a competing classifier with posterior performance $p(\lambda_2|\mathcal{D})$ or others, e.g., $p(\lambda_3|\mathcal{D})$. This question can be answered by considering the pairwise posterior differences between the respective competing classifiers,

$$p\left(\delta \mid \mathcal{D}\right), \tag{14}$$

where $\delta := \lambda_{(j)} - \lambda_{(i)}$ denotes the difference between the posterior balanced accuracies $\lambda_{(i)}$ and $\lambda_{(j)}$ of two competing classifiers (cf. the *difference between proportions* [20, p. 175-176]) and $\mathcal{D} = \{k_1^{(1)}, \ldots, k_l^{(1)}, k_1^{(2)}, \ldots, k_l^{(2)}\}$ is the classifiers outcomes.[3]

Properties of $\boldsymbol{\delta}$. The domain of the random variable δ is continuous in $[-1 \ldots 1]$, Its distribution can summarized, for example, by reporting the posterior expectation. It represents the expected algebraic distance in performance between the two classifiers.

[2] The software can be downloaded from: `http://mloss.org/software/view/447/`

[3] The dataset sizes $\{n_1^{(1)}, \ldots, n_l^{(1)}, n_1^{(2)}, \ldots, n_l^{(2)}\}$ have been omitted for brevity.

Computing δ. Algorithmically, we can compute the posterior density of the difference of two balanced accuracies using a stochastic approximate inference approach. Specifically, under a Monte Carlo scheme [24, p. 154-155], we repeatedly draw samples from $p(\lambda_{(i)}|\mathcal{D})$ and $p(\lambda_{(j)}|\mathcal{D})$ and collect the differences between each pair. This approach will result in an approximation of $p\left(\delta \mid \mathcal{D}\right)$. In practice, a high number of samples (e.g., 5 000) is required for a suitable approximation.

A simple heuristic method for comparing competing classifiers can be done by ranking them by their performance's algebraic distance against each other. We suggest the following simple scheme:

1. Given T competing classifiers, compute the pairwise posterior difference between them (using eq. 14) and its posterior mean. Decide on a winning classifier for each comparison based on the sign of the posterior mean.
2. Count the number of times each classifier wins.
3. Rank the competing classifiers according to its number of victories.

This simple scheme assumes that the classifiers are tested with the same dataset. It also assumed that the posterior mean of δ approximates the $p(\lambda_{(i)} > \lambda_{(j)}|\mathcal{D})$ for every pairwise comparison, which could be an optimistic approximation. Non-heuristic schemes of comparison are of interest and objective of future research.

5 Experiments

In this section, we present a set of experiments to compare the proposed probabilistic evaluation to previous frequentist approaches such as sample accuracy, sample balanced accuracy, Kappa coefficient and F_μ-score. We begin by considering the synthetic data from the motivating example in Section 3 and conclude the section by reanalyzing an empirical dataset from the domain of action detection.

5.1 A Bayesian Look to the Motivating Example

Using the confusion matrix of Fig. 1a as input, the posterior distribution of the balanced accuracy (henceforth PDBAC) of each classifier can be computed as described in Section 4.4. The PDBAC of each classifier is depicted in Fig. 2.

As expected, the posterior mean of the classifiers show a clear difference in performance between (C_2) and the others. This behavior is also exhibited by the sample balance accuracy, the F_μ-score and the Kappa coefficient, but not for the sample accuracy, as discussed in Section 3. Moreover, the uncertainty associated with the posterior mean is in the correct domain of the random variable λ for all classifiers, unlike the (non-z-transformed) accuracies, which for C_3 exhibit non-sensical uncertainty bounds.

To illustrate the effect of sample size on posterior inferences, Table 1 details the performance metrics after evaluating C_1, C_2, and C_3, as well as two scaled versions of them. As expected, as the number of trials grows, the limits of the classical confidence intervals assume more sensical values. It is further worth noting that the classical point estimate itself is ignorant to the sample size (since the confusion matrices retain the same proportions). By contrast, the posterior mean reflects a shrinking influence from

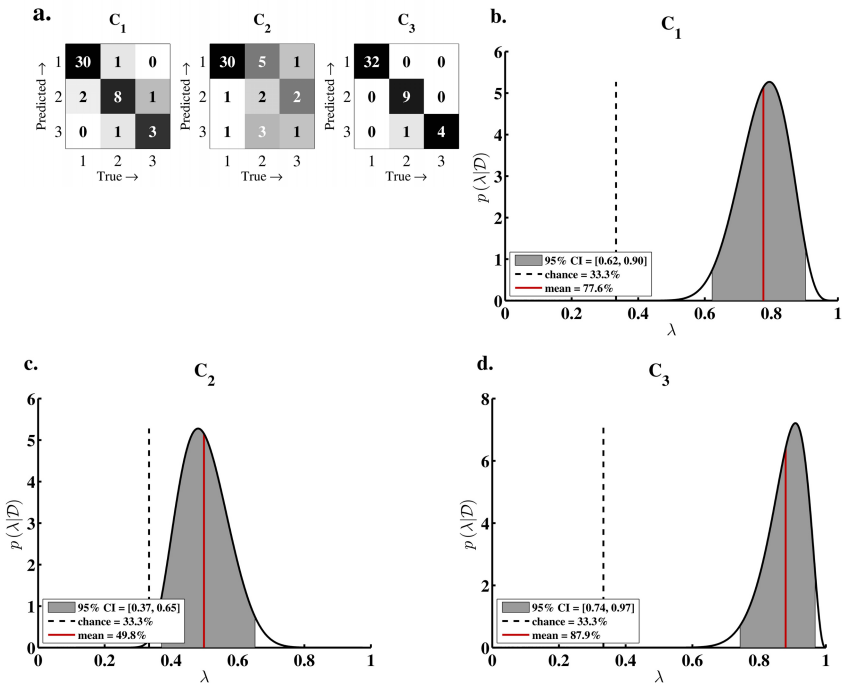


Fig. 2. (a) Confusion matrix of the classifiers studied in Section 3. (b)(c)(d) Posterior distribution of the classifier's multiclass balanced accuracy. For each distribution the posterior mean and 95% confidence interval is indicated. Figure generated by `IllustrativeExample0.m`, available online [9].

the prior and a growing influence from the data as its posterior interval tightens. The posterior mean and the sample balanced accuracy would agree in the limit of infinite data.

Ranking Competing Classifiers. The procedure described in Section 4.5 makes it possible to rank the competing classifiers C_1, C_2 and C_3 according to their posterior distributions. We begin by computing the pairwise distribution $p(\delta \mid \mathcal{D})$ for each competing classifier. The three resulting distributions are depicted in Fig. 3.

It is worth recalling that the posterior mean of δ is not simply the difference between the involved posterior means. The posterior mean of δ takes into account the uncertainty of the λs. Continuing with the next steps of the proposed method, the resulting rank is: 1. C_3, 2. C_1 and 3. C_2.

5.2 Action Detection

There has been increasing momentum in studying models for visual recognition of human actions from images. One recent study [23] proposed a system for action modelling

Table 1. This table summarizes performance statistics of the classifiers studied in Section 3. C_1, C_2 and C_3 are the confusion matrices of the classifiers considered in section 3. C_4, C_5 and C_6 provide the same information scaled by a factor of 10. Likewise, C_7, C_8 and C_9 are scaled by a factor of 100. The statistics computed for all matrices is the mean and the 95% confidence interval. The table compares the PDBAC with the sample balanced accuracy, the sample accuracy, the F_μ-score, and the Kappa coefficient. The grey figures represent 95% C.I. whose limits escape the domain $[0, 1]$.

	PDBAC		bac		acc		F_μ-score		K_C	
	μ	95%C.I	μ	95%C.I	μ	95%C.I	μ	95%C.I	μ	95%C.I
C_1	0.776	[0.62 - 0.90]	0.829	[0.73 - 0.92]	0.891	[0.79 - 0.98]	0.821	-	0.769	[0.57 - 0.96]
C_2	0.498	[0.37 - 0.65]	0.462	[0.32 - 0.59]	0.717	[0.58 - 0.85]	0.457	-	0.331	[0.02 - 0.63]
C_3	0.879	[0.74 - 0.97]	0.966	[0.92 - 1.01]	0.978	[0.93 - 1.02]	0.945	-	0.953	[0.86 - 1.04]
C_4	0.822	[0.77 - 0.87]	0.829	[0.80 - 0.85]	0.891	[0.86 - 0.92]	0.821	-	0.769	[0.70 - 0.82]
C_5	0.468	[0.42 0.52]	0.462	[0.42 - 0.50]	0.717	[0.67 - 0.75]	0.457	-	0.331	[0.23 - 0.42]
C_6	0.955	[0.93 - 0.98]	0.966	[0.95 - 0.98]	0.978	[0.96 - 0.99]	0.945	-	0.953	[0.92 - 0.98]
C_7	0.828	[0.81 - 0.85]	0.829	[0.82 - 0.83]	0.891	[0.88 - 0.90]	0.821	-	0.769	[0.75 - 0.78]
C_8	0.463	[0.45 - 0.48]	0.462	[0.44 - 0.47]	0.717	[0.70 - 0.73]	0.457	-	0.331	[0.30 - 0.36]
C_9	0.966	[0.96 - 0.97]	0.966	[0.96 - 0.97]	0.978	[0.97 - 0.98]	0.945	-	0.953	[0.94 - 0.96]

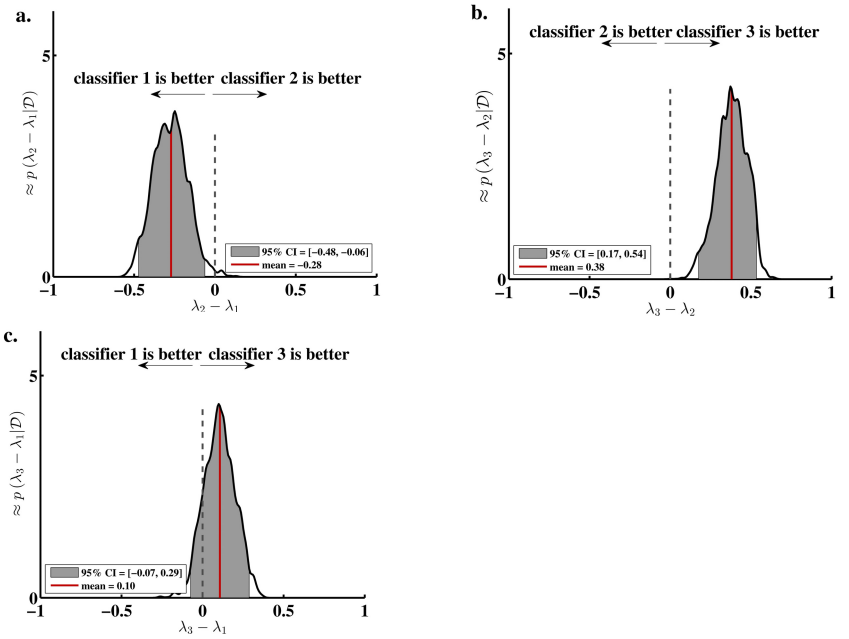


Fig. 3. (a)(b)(c) Posterior distribution of the difference of two posterior balanced accuracies computed as explained in Section 4.5. The posterior balanced accuracies λ_1, λ_2, and λ_3 stem from C_1, C_2, and C_3, respectively. For each distribution the posterior mean and the 95% confidence interval are indicated. Figure generated by `IllustrativeExample0.m`, available online [9].

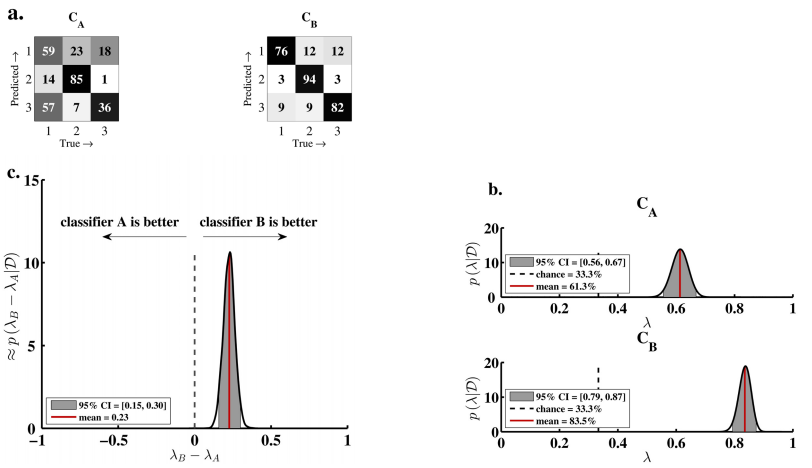


Fig. 4. (a) Confusion matrices of two classifiers [2, 23] for visual action detection (see Section 5.2). (b) Posterior distributions of the two classifiers' multiclass balanced accuracies. (c) Posterior distribution of the difference in balanced accuracy between the two classifiers. Figure generated by `IllustrativeExample1.m`, available online [9].

based on a classification method for human actions from image sequences. The authors compared their classifier to an alternative approach [2] using a dataset composed of three classes representing the acts of 'moving an object', 'making a sandwich', and 'opening a book', respectively.

We revisited the reported results and, based on the published confusion matrices (Fig. 4a), computed the posterior multiclass balanced accuracies along with several summary statistics (Fig. 4b). Our results show that, in contrast to conventional sample accuracies, posterior balanced accuracies provide a rich representation of our knowledge about each classifier's performance. Critically, using the approach outlined in Section (Sec. 4.5), we can infer on the difference between the two performances. Specifically, given the two competing classifiers , C_B is ranked number one with a posterior expectation of the difference of 0.23. We illustrate the full posterior distribution of δ using a kernel density estimator (bandwidth 0.0125), as shown in Fig. 4c.

6 Discussion

Classification algorithms frequently form a critical part of complex systems for pattern recognition and machine learning, such as those found in the domain of robotics. However, evaluating the performance of a given system, and comparing it to others, is often subject to methodological limitations. Reporting overall classification accuracy, for example, is statistically unwarranted because it can only be interpreted in relation to a baseline level that depends on the degree of class imbalance at hand. Sample statistics, such as the sample balanced accuracy for instance, rectify this problem but do not readily embrace our uncertainty about the performance metric of interest.

In this paper, we described how the above limitations can be overcome in a natural way using a Bayesian framework for inferring on the balanced accuracy. The approach generalizes our previous work on balanced accuracies for binary classification problem. More importantly, a method of ranking competing classifiers using their posterior balanced accuracy is proposed. The main advantage of this methods is that it permits to account for the uncertainty in the classifiers' performance during evaluation.

One critical feature of this approach is the flexibility with which posterior inferences can be summarized. In particular, we can obtain derivative inferences, such as our confidence with which one classifier is better than another. In this context, it is worth noting that, in order to ensure a fair comparison, it is important that the algorithms whose performances are compared were applied to the same dataset.

To facilitate its widespread use, we provide an open-source MATLAB toolbox which we have made available for download [9]. With this toolbox we hope that balanced accuracies may help improve the correct evaluation and comparison of multiclass classifiers in future classification systems.

Acknowledgments. This work was supported by the MICINN-FEDER projects DPI2009-13710 and DPI2012-36070 as well as by DGA-FSE (grupo T04).

References

1. Akbani, R., Kwek, S.S., Japkowicz, N.: Applying Support Vector Machines to Imbalanced Datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
2. Aksoy, E., Abramov, A., Worgotter, F., Dellen, B.: Categorizing Object-action Relations from Semantic Scene Graphs. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 398–405 (May 2010)
3. Andreopoulos, A., Hasler, S., Wersing, H., Janssen, H., Tsotsos, J., Korner, E.: Active 3D Object Localization Using a Humanoid Robot. IEEE Transactions on Robotics 27(1), 47–64 (2011)
4. Berger, J.O.: Could fisher, jeffreys and neyman have agreed on testing? Statistical Science 18(1), 1–32 (2003)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer-Verlag New York, Inc., Secaucus (2006)
6. Brodersen, K.H., Mathys, C., Chumbley, J.R., Daunizeau, J., Ong, C.S., Buhmann, J.M., Stephan, K.E.: Bayesian Mixed-Effects Inference on Classification Performance in Hierarchical Data Sets. Journal of Machine Learning Research 13, 3133–3176 (2012)
7. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The Balanced Accuracy and Its Posterior Distribution. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 3121–3124 (August 2010)
8. Cadena, C., Galvez-Lopez, D., Tardos, J.D., Neira, J.: Robust Place Recognition With Stereo Sequences. IEEE Transactions on Robotics 28(4), 871–885 (2012)
9. Carrillo, H.: GBAC (2013), `http://www.mloss.org/software/view/447/`
10. Carrillo, H., Latif, Y., Neira, J., Castellanos, J.A.: Fast Minimum Uncertainty Search on a Graph Map Representation. In: IEEE / RSJ International Conference on Intelligent Robots and Systems (IROS 2012), Vilamoura, Algarve, Portugal (October 2012)

11. Carrillo, H., Reid, I., Castellanos, J.A.: On the Comparison of Uncertainty Criteria for Active SLAM. In: IEEE International Conference on Robotics and Automation, pp. 2080–2087 (2012)
12. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research (JAIR) 16, 321–357 (2002)
13. Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement 20(1), 37–46 (1960)
14. Ess, A., Schindler, K., Leibe, B., Van Gool, L.: Object Detection and Tracking for Autonomous Navigation in Dynamic Environments. The International Journal of Robotics Research 29(14), 1707–1725 (2010)
15. Galvez-Lopez, D., Tardos, J.D.: Bags of Binary Words for Fast Place Recognition in Image Sequences. IEEE Transactions on Robotics 28(5), 1188–1197 (2012)
16. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian data analysis. CRC press (2003)
17. Granstrm, K., Schn, T.B., Nieto, J.I., Ramos, F.T.: Learning to close loops from range data. The International Journal of Robotics Research 30(14), 1728–1754 (2011)
18. Japkowicz, N., Stephen, S.: The Class Imbalance Problem: A systematic Study. Intelligent Data Analysis 6(5), 429–449 (2002)
19. Kerman, J.: Neutral noninformative and informative conjugate beta and gamma prior distributions. Electronic Journal of Statistics 5, 1450–1470 (2011)
20. Kruschke, J.K.: Doing Bayesian Data Analysis: A Tutorial with R and BUGS, 1st edn. Academic Press / Elsevier, Amsterdam (2011)
21. Landgrebe, T., Duin, R.: Efficient Multiclass ROC Approximation by Decomposition via Confusion Matrix Perturbation Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 810–822 (2008)
22. Leon-Garcia, A.: Probability and Random Processes for Electrical Engineers, 2nd edn. Addison-Wesley, Reading (1994)
23. Luo, G., Bergstrom, N., Ek, C., Kragic, D.: Representing Actions with Kernels. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2028–2035 (September 2011)
24. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning series. The MIT Press, Cambridge (2012)
25. Nishii, R., Tanaka, S.: Accuracy and inaccuracy assessments in land-cover classification. IEEE Transactions on Geoscience and Remote Sensing 37(1), 491–498 (1999)
26. Rijsbergen, C.J.V.: Information Retrieval, 2nd edn. Butterworth-Heinemann, Newton (1979)
27. Siagian, C., Itti, L.: Biologically Inspired Mobile Robot Vision Localization. IEEE Transactions on Robotics 25(4), 861–873 (2009)

A Hierarchical Hybrid Architecture for Mission-Oriented Robot Control

Manuel Muñoz, Eduardo Munera, J. Francisco Blanes, and Jose E. Simó

Institute of Control Systems and Industrial Computing,
Polytechnic City of Innovation,
Polytechnic University of Valencia, Spain
mmunoz@ai2.upv.es
www.ai2.upv.es

Abstract. In this work is presented a general architecture for a multi physical agent network system based on the coordination and the behaviour management. The system is organised in a hierarchical structure where are distinguished the individual agent actions and the collective ones linked to the whole agent network. Individual actions are also organised in a hybrid layered system that take advantages from reactive and deliberative control. Sensing system is involved as well in the behaviour architecture improving the information acquisition performance.

Keywords: Intelligent Robotics, Limited Resources Management, Embedded Systems.

1 Introduction

Some years ago, no one beyond the filmmakers imagined the current capabilities of the robots, spread from those who are cloistered in the factories, to the accompaniment and protocol Asian robots. Technology evolution in terms of computing power, sensory capability and mechanical improvements, allows to perform more autonomous robots, with complex tasks or behaviours, even in a heterogeneous collaborative mess. From this fact arises the need for control systems capable of handle robots while perform their tasks, and coordinate them to achieve cooperative tasks.

There is a lot of solutions to this problem, some focus on part of the problem, while others are dependent on a certain platform or architecture. The alternative proposal, mixes some of these solutions or paradigms and seeks to avoid some problems encountered by addressing the problem from a general point of view, and offering a platform independent solution.

From another point of view, these tasks require a lot of information from the environment in order to interact properly with it. Therefore in addition to the advanced sensing systems mounted on robots, special behaviour are required to encourage and/or optimise the acquisition of information. The proposed solution also addresses this idea.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_26,

1.1 Previous Work

Robot control architectures have been a highly contested issue for years, and remains, in response to the large number of publications that keep coming on this subject. By hindsight, several trends can be observed: Deliberative Architectures, Behaviours-Based Architectures and hybrid solutions that mix both.

Early control architectures performed a deliberative execution because of its straight application despite of its requirement of a exhaustive previous knowledge of the situation to control. Nowadays this is still being used on some architectures like the one presented by Conor in [19], which is focused on the control of autonomous submarine vehicles.

The first steps in the model definition and formalisation of behaviour was exhibited by R. Brooks in [6]. The work describes an architecture for control mobile robots. Modelling some layers with different levels of competence. Each of these layers can access all the sensor data and generate control actions for the actuators. Separate tasks can suppress or inhibit inputs outputs. Thus, the lower layers can function as highly reactive mechanisms or reflections, while the upper layers are working to achieve the overall goal. Other classic behaviour based architectures like the proposed by and Arkin [3] remain strong influence over many current works like the presented by Cañas and Matellán in [7] where autonomous behaviours are generated by using dynamic hierarchies of small schemes. Thus, is obtained a scalable and extensible architecture which provide more flexibility than subsumption architecture and improves its performance in dynamic environments.

Some years later arises the trend of combine both previous paradigms. These kind of systems are called hybrid architectures or reactive-deliberative architectures. Stoytchev in [25] presents a hybrid robot architecture that combines three components: deliberative planning, reactive control, and motivational drives. Each of the three components addresses one of the challenges: To adapt quickly to changes in the environment; To understand high level human commands; and To be engaging and fun to use.

The amount of applications that requires a certain level of cooperation, and the increment of the capabilities of the agents involved, have promoted the number of researches focused in coordination architectures. Typical coordination system make use of a fusion of behaviours and an arbitration process in order to execute the most accurate action for each situation, a clear example was explained by Proetzsch in [21] where is proposed a solution called integrated behaviour-based control (IB2C). In this work is proposed the use of "behaviour modules" as the basic units of the architecture which contains information about the action, the rating, and the transfer functions of the behaviour. Those behaviours can be stimulated and inhibited in order to calculate the level of its relevance which in turns is reflected on the activity level of that behaviour to the current state. This activity level is used to perform a weighted fusion of the system behaviours for providing the output. One of the most relevant contributions of this work is the capability of implement different coordination methods separated in two main groups: arbitration or command fusion.

Actually most of the proposed solutions make use of fuzzy logic (FL) for the behaviour coordination. In [11] R. Huq offers a solution for behaviour-based control by combining the rigid state-based paradigm of the discrete events system (DES) with the flexibility of FL proposing the use of a fuzzy discrete event system (FDES) [15,22] as solution. The application of a DES ease the analysis of complex system that can be clearly described as a sequence of events. Transitions between states are usually based on sensory information, which implies that sensory uncertain can lead to an erroneous performance. Nevertheless, the application of FL can help to represent this uncertain, offering a mechanisms for dealing with that situations. In that architecture behaviours are defined by its associated action, the sensory information and a state-based modulation of its activity. That last element make use of fuzzy event matrices to generate predictions to estimate the activity of the behaviour. For the modelling of those matrix are used "fuzzy state vectors" which defines the probability of perform a transition between two states given a certain event. Based on the information contained on those vectors an arbitrator must determine the actions to perform. This paradigm is later extended on [12,13].

Although this, there are some coordination methods which offers some out the box solutions ,like the ones formulated on [16]. In this case is presented a multi-robot cooperation inspired in a biological system, a pack of wolves. That way is differentiated both alpha and betta roles which can be swapped between the pack members in function of its capabilities and collaborate to achieve a common goal, for the case, the simulation of an elk hunt.

The relevance of perceptual information for the execution of basic behaviours is clearly remarked in classical studies like Mataric in [17] where are remarked the advantages of centralised or distributed behaviours and its disadvantages, between some of them are related to the limitations in this time. Asama in [4] describes the importance of distributed information on robot systems making an analysis of functions distribution between the agents and path planning. On more actual works can be found some works where is implemented a communication between agents in order to share sensory information. A clear example of this can be found in [18] by Matellán where is compared three different fuzzy communication methods in order to share perceptual information about the environment and defines how this is reflected on its behaviours and the performance of its tasks.

1.2 Outline

As outline of this paper: After the introduction section 1.1 describes a review of related works explaining advantages and detected problems. A description of the used platforms and simulators is depicted in section 2. The proposed Mission Based System is shown in section 3. The group coordination approach is depicted in section 4. A clarify example is shown as a use case in section 5. A comment about the implementation issues is reported in section 6. The paper ends with some conclusions about the work in section 7.

2 Framework Overview

2.1 Development Platforms

The architecture introduced on this work is referred as a general solution for agents in networked environments. Nevertheless, all the proposed examples and validation tests will be focused on robotic platforms, including mobile and humanoid robots. Using mobile robots as agents of the system provides a reliable platform for validation, due to the wide range of sensors that can be equipped with, and its wheel-based locomotion. Humanoid robots on its side provides a more restricted platform, due to its complex locomotion an the limited number of sensors that can assemble. Validation tests have been performed on available robots and for which the research group has great experience in previous developments.

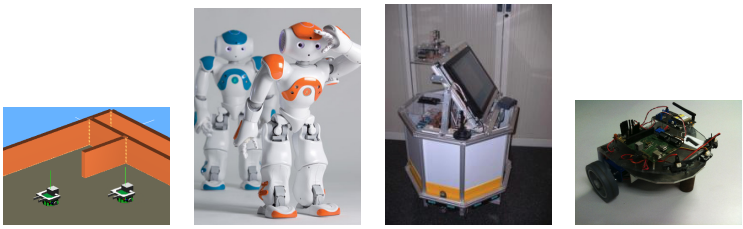
Mobile robots, just as has been described, are characterised by its displacement capabilities and its flexible configuration. Two different types of mobile robots are implied at this work, each one developed following its own philosophy. First is introduced the KertrolBot [26], as is showed on 1(d), a small size micro-controller-based robot which is propelled by a differential locomotion, is equipped with infrared sensors, and disposes of Wi-Fi communications. The goal of the KertrolBot is to provide solutions for small applications due to simplicity of its design and the low computational power.In the second case the Yet Another Intelligent Robot (YAIR), as can be observed in 1(c)[5], offers a similar locomotion but disposes of a bigger structure in which are integrated a screen, some more sensors as a camera or a laser range sensor, and the use of a computer as main operational unit. The YAIR offers a more wide range of applications because of its advanced capabilities and its computing power.

Giving way to the Humanoid Robot, the main challenge faced is dealing with biped walking. This locomotion is far more complex that the exposed on previous robots, and implies some restrictions on the mobility, the weight, and the stability, which limit the sensors that can be equipped with. In this particular case, we use the commercial robot Nao, displayed on 1(b) [23]. In spite of all the restrictions, this robot offers a good demonstrator for the development of some convoluted tasks like playing soccer.

For testing more complex environment situations, and disposing of more variety of robot models, we use a simulator, like Gazebo [14]. This one is a robot specific simulator which offers the capability of reproduce most of the real applications which are proposed on this paper, and enhance fast development and validation of the described architecture. An example of simulated robots can be observed in 1(a).

2.2 System Architecture

Beyond the execution of behaviours, agents must assemble mechanisms for performing some other type of jobs. Then, a scalable and flexible framework is established in order to group required tasks according to its functionality forming



(a) Simulator (b) Nao (c) YAIR (d) KertrolBot

Fig. 1. Presented Platforms

modules. These modules offer abstraction from their implementation providing an interface which can be easily accessed by other modules.

Perception Module. In almost every proposed situation, agents must obtain information about the physical world in order to perceive its environment. Depending on the environment and the possibilities, the appropriate sensor must be used. Thus the module must provide elaborated information as independent as possible from the source. Each one must be properly studied for adapting its raw sensory measures to a predefined data type. That adaptation process is the main objective of this module, which provides understandable information about the physical world to other modules that required it in order perform a proper operation. That is specially critical for the case of the Global Modelling and the Actuation modules.

Global Modelling Module. A reliability knowledge of the environment together with the agent's location in space brings the capability of use this information to take decisions about the best way to face its mission. That takes even more relevance when the agent have to perform some kind of path planing, or location tasks, in order to achieve its mission. To meet this need, current module offers a localisation method which allow to estimate the location of an agent in the space from the information provided by the perception module. But a model of the area is also needed in order to be able to discern its position in it. For that reason this module must also manage the environment map, generating it in case that no previous information is provided. Many techniques for both, localisation and environment mapping, are implemented as will be introduced on the Environment Interaction section.

Actuation Module. On the way to achieve its mission an agent may have to interact with the environment, which can involve from apply a control action to perform a displacement on the space. As will be introduced next, usually the actuation is raised as the result of a behaviour fusion, which reflects the desire of the agent to perform a certain action. Just as the opposite case of the adaptation phase described on the perception module, the main purpose of this

one is to translate the information provided by other modules, expressed in a defined data type, to a raw value required by the actuator that will depend of their characteristics.

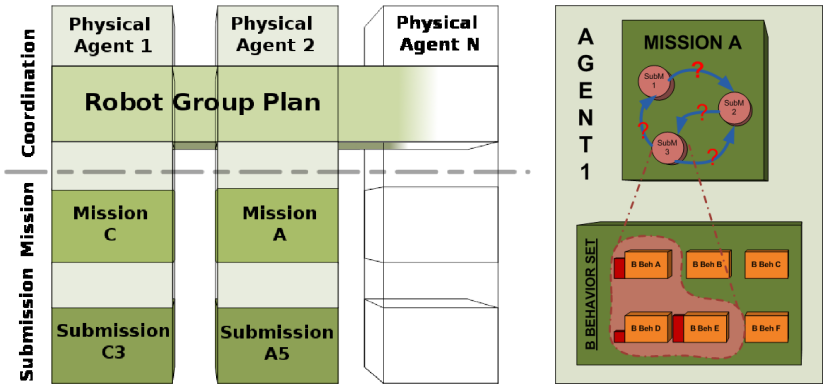
Communication Module. Some of the information managed in the previously mentioned modules, and specially the Perception Module, is sensitive to be shared with other agents of the system. In that way, sharing information between agents provides knowledge, including the state of the other agents, the progression of the mission and information about the environment. The establishment of this module enhances the coordination and brings the capability of setting up some levels of cooperation among robots, as well as the implementation of distributed consensus and data association mechanisms [1]. Those methods can be remarked as a hot topic according the amount of publications that refer to it. On next sections will be introduced how the proposed system offers a great base for integrating a distributed consensus layer.

Behaviours Module. The Behaviours module is responsible for generating commands. The work described in this paper mainly turns around this module. We propose an approach named "Hierarchical Hybrid Architecture for Mission-oriented Robot Control". System has been split hierarchically in two components:

- On one hand, the coordination side offers a high level of abstraction and manages all the information related to the collective strategy, consequently it can be represented as a module of distributed agents.
- On the other hand, the control side works with information from different levels of abstraction, but this information belongs to the individual behaviour of each agent. On Figure 2(a) can be noticed the general representation of a behaviour system.

3 Individual Behaviour – Mission/Submission System

The individual behaviour layer is responsible for executing the mission that has been assigned to that agent. An individual behaviour is defined as any behaviour carried out by one unique agent that leads its actions in a certain lapse of time. This behaviour is determined by a mission, which has been assigned to each agent from the upper layer as is show later on the section "Group Coordination". In most cases, individual behaviours can be composed by actions of heterogeneous complexity, for that reason this layer has been split in three different levels: Main behaviour, Mission, and Submission (or Mission State). On following sections will be described concepts ranging from the more basic levels to the mentioned Mission in ordered to ease its understanding. In Figure 2(b) is shown an overview of the interaction between this three levels.



(a) General Overview (b) Individual Overview

Fig. 2. Architecture Overviews

3.1 Basic Behaviour

A Basic Behaviour (or Basic Skill) is an action executed by an agent in response of stimulus, or precise commands transferred to a lower module to perform an action. Basic behaviours are strongly related to the agent performance in both input (perception capacity) and output (actuation capacity). It also is characterised by aiming to a simple and precise goal. A basic behaviour can not be divided and executed in a concurrent way.

Output Features: The output of a basic behaviour must actuate according the capabilities offered by the agent API, which is widely ranged between different kind of actions. In general case, it is defined as the establishment of a certain value as output information. On software agents this can be considered as a trigger between its own states. However, physical agents offers different options like, for example, establishing a voltage on an actuator, positioning a servomotor to a certain angle, making the final effector of a robot reach a position, or the establishment of a reference velocity for a mobile robot. It must be remarked that this performance is imposed by the kind of the current agent which is being working with.

Characterisation of Input Stimulus: Input signals that are used by the basic behaviours provides a low abstraction level. In most cases, this signals are used to close control loops which rule basic behaviours.

3.2 Behaviour Fusion

As has been explained on the basic behaviour section, basic skills must be simple and indivisible. One of the main reason for the establishment of that restrictions,

is to provide the system a mechanism to fuse those abilities, obtaining as a result the emergent behaviours that allows the resolution of more complex problems that can not be solved separately.

Motivational State: The concept of "motivation" is introduced for being used as a weighting element when a fusion of basic behaviours is performed. One behaviour, with a high level of motivation, will influence on the composition in a more representative way than other less motivated. In every behaviour, its motivation value, must be computed previously to the fusion. On this calculus may take part many factors as temporal and sensory data, the own logic state of the system, ifnextchar.etcetc..

Composition Function: The "composition function" is used in order to obtain the final value of the actuation for a certain variable based on the contributions brought by the different behaviours. As a basic composition function is proposed the weighted mean of the basic behaviours outputs, using the "motivation" as weighting element. Composition is calculated using equation 1. For the basic behaviour i, its motivation BM_i, is multiplied by its contribution BC_i. The result of the sum of all these products fro all the basic behaviours is normalised with the summation of the motivations. For that purpose can be used any other function, therefore the comparison of different functions, and the analysis of its influence on the output is relegated to a future work.

$$u() = \frac{\sum_{i=1}^{n} BM_i \cdot BC_i}{\sum_{i=1}^{n} BM_i}; \tag{1}$$

Disjoint Behaviours: Are called disjoint those behaviours that must/can not occur simultaneously because there is not physically possible, or because a design restriction. As more visual examples, when using biped robots as physical agents is not possible to perform a kick while the robot is walking, or in the case of wheeled differential robots is not possible to follow a straight line with a constant speed while is performing a turn on the vertical axis. For the establishment of the composition avoiding disjoint behaviours is proposed a modification of the previous Equation 1. The concept of composition table (equation 2) is introduced, in every row is indicated the composition coefficient of each behaviour with the rest. In this way, the element $c(i, j)$ shows the index of composition for the behaviour j with the i. The composition of a behaviour with itself must be 1, while the index with disjoints behaviours, which cant be executed at same time, must be 0. Therefore, this coefficient is dimensionless and could take real values between 0.0 and 1.0.

$$C_{i,j} = \begin{bmatrix} 1 & \cdots & [0.0 - 1.0] \\ \vdots & 1 & \vdots \\ [0.0 - 1.0] & \cdots & 1 \end{bmatrix} \tag{2}$$

The composition equation 1 has been modified for taking into account this composition matrix obtaining a new equation 3. The procedure to follow implies that composition coefficients must be extracted from the behaviour column with the highest motivation value. Thus new equation must be:

$$u() = \frac{\sum_{i=1}^{n} BM_i \cdot BC_i \cdot c_{i,j}}{\sum_{i=1}^{n} BM_i \cdot c_{i,j}};\ j = \text{more motivated behaviour index.} \quad (3)$$

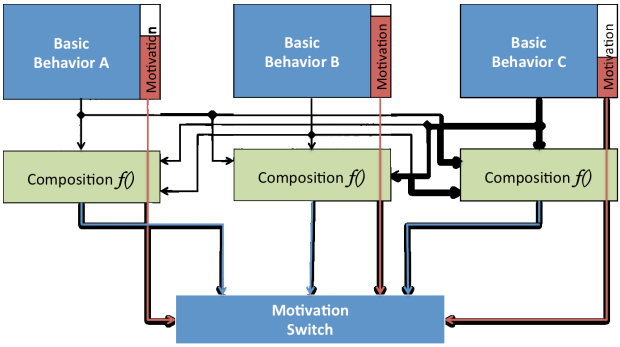


Fig. 3. The fusion of Behaviours A, B and C are conditioned by their levels of motivation

Fatigue Factor: The fatigue factor weights the contribution of a behaviour depending on the lapsed time from its activation. The contribution values are ranged from 0.0 to 1.0, and are obtained as the product of other 2 subfactors:

- Rise subfactor: Favours the progressive introduction of the behaviour. Its value is initially 0.0 and is linearly increasing until reaches the rising time when must be 1.0.
- Fall subfactor: This factor limits the action of a behaviour if is reached the fatigue time and remains active because has not accomplished its goal. Its initial value is 1.0 and is decreasing until 0.0. After the falling time the values is maintained till expires the blocking time.

These factors are calculated from 3 parameters, rising time, fatigue time and blocking time. This parameters will be established according to the dynamic of each behaviour.

3.3 Submission or Mission State

A submission is composed by a set of the available basic behaviours. When a state is active every behaviour of the set is also activated. The activation of several behaviours leads to the emergence of new behaviours as a result of the fusion of its control actions. A submission is also characterised by its table of composition and all the basic behaviours contained.

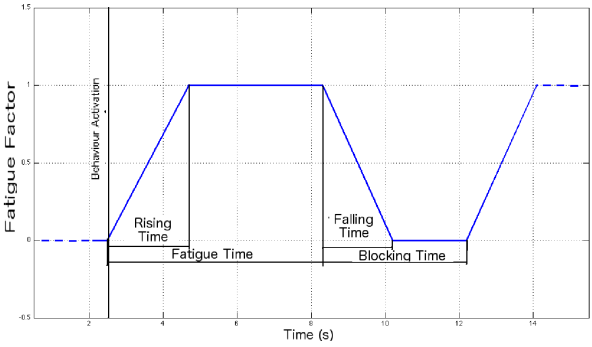


Fig. 4. The application of the Fatigue factor avoids to get stuck on the execution of non-progressing behaviours

3.4 Mission

Is defined as a "mission" the complex task realised by an agent, as a part of a group, in order to reach a certain objective assigned for the upper coordination level. Missions may have a concrete goal or may execute a repetitive task. If an agent is performing alone, the mission to achieve will not be modified along time. Missions are represented by a HFSM (Hierarchical Finite State Machine), in which graph are depicted two types of elements: states and transitions. States are explained in detail on the "Submission or Mission State" section. One of the states will be marked as the initial state and will be the first to activate once the mission starts. Transitions between states are performed after the evaluation of a set of sensory conditions, and can be traversed in one direction only. Those conditions ensures that the partial objectives of the state has been successfully accomplished. It can be considered that the mission does not provide an output for itself, but takes the output of the active state in the moment of the execution. Mission requirements will evaluate the state of accomplishment of the submission in execution. It must be to distinguish two cases:

- Global requirements: will be the sum of the requirements of every state in the mission, along with the requirements of every existing transition.
- Actual requirements: are the requirements of the actual states, with the sum of the requirements of its exit transitions.

3.5 No Goal Oriented Behaviours

Although the proposed architecture aims to the accomplishment of goals, not all the behaviours involved are directly related to the objective. The main functionality of these behaviours is to provide the needed support for assure that the goal-oriented behaviours dispose of all the requirements for its proper execution. Between these requirements most usually can be found the need of an environment interaction or the knowledge of the status of the other agents in

the network. Therefore, this section will fall on the characteristics of the behaviours employed for covering this demand and the way that is managed in the architecture.

Environment Modelling and Localisation: Most of the executed tasks that are relevant to the coordination system, are also related with the interaction with the environment. That is specially critical when the agents have to perform some kind of path planing or location tasks in order to achieve its objectives. For providing them the knowledge of the characteristics of the environment in which are located can be provisioned a previously generated map, or as alternative can make use of simultaneous localisation and mapping (SLAM) [8] techniques that allow the agents to generate a map of its initially unknown surrounding thanks to the sensory information.

Typically the use of mobile robots as agents implies the need of being able to locate its own position in the map for developing its tasks in an proper way. Many techniques have been used for that purpose, like the Montecarlo Particle Filter [9] Monte-Carlo Particle Filter (MCL) and the some variants of the Kalman Filter (KF) as the Unscented Kalman Filter (UKF) [24], although it also have been developed some derived techniques by taking profit of the advantages of both them. That way in [20] is presented a location solution for humanoid robots, in a soccer game environment, which propose a combination of these techniques.

A good knowledge of the environment and their location in space brings the possibility of using this information to take decisions about the progress of the mission. The exchange of data between agents allows to perform a better distribution of the tasks, enhance the environment model, or detect dynamic obstacles. That way, next section will describe the advantages provided by the data distribution between agents and the establishment of consensus procedures.

Cooperation and Consensus: In a network of agents, where a common objective is fixed, the fact that every agent is working on its own without caring about the state of other ones does not take profit of the advantages of these systems. Any kind of cooperation between agents will provide a remarkable enhancement for mission execution.

One of the tasks that will be improved by this cooperation is the environment modelling. If a map of the work area is provided the detection of dynamic elements is a simple decision, but it is not also on those situations where agents must generate its own map. Once an element of a new area are spotted there is no initial distinction about if is an static or a dynamic obstacle. This situation can be solved by a consensus phase where agents which also has been explored this area contribute with their information, as can be seen on [2].

Some of the most common application of cooperation to the environment modelling are intended to perform efficient area exploration. This kind of cooperation are widely used in mobile robot and Unmanned Aerial Vehicles (UAV) networks [10]. An exploration performed with several robots which provides a certain overlap between the sensed range allow a fast association between the distributed information an provides an agile method for environment modelling.

Focusing on the mission development, cooperation is also a clear benefit as it improves the performance and speeds up the execution. In a network of agents which does not perform any type of cooperation, tasks must be preassigned independently of its state. By enabling a consensus phase, this state, which characterises the agent (position, tasks in execution, ifnextchar.etcetc.) may be considered in order to decide which are the most qualified robot to execute a new task when its required.

Task Integration Approach: The need of localisation implies the execution of several tasks in order to guarantee a accurate estimation. That way several behaviours have to be exclusively or tightly related with that kind of tasks. As any other behaviour active they must also be fused in addition to other behaviours related purely related with the mission execution. In most of the cases required actions are intended to the sense of its surrounding, but in some other cases implies active motion in order to perform a pursuit of localisation marks.

Examples: Following the example explained in previous sections. Physical agents need to be localised and with a good sensory feedback. Thus additionally to the 2 Basic Behaviours described in section 5 a Non Goal Oriented Behaviour is joined. The behaviour is Landmark Following:

- Landmark Following.
  - The objective of the physical agent is to trace its trajectories as near as possible to the available landmarks. That way, positioning is not lost.
  - The action, output, or control action, taking into account the exposed on previous examples, can be one of the components of the velocity vector which guide the agent near to landmarks.
  - Perceptual requirements are the position of the surrounding landmarks.

4 Group Coordination

The lower layer need the indication of what mission must be executed, for that reason is implemented a group coordination layer. By the application of that layer can also be obtained some other additional benefits, thanks to a good coordination between agents any activity can be performed in a more efficient way than in a individual approximation. Next is introduced the different sublevels in which is decomposed this layer.

4.1 Tactics

Tactics will be defined as a set of missions to achieve by a group of agents. None tactic is preassigned to a mission nor an agent, but must establish certain restrictions referred to the minimum resource types that must have an agent to

perform a concrete task. The number of agents, which realises this tasks, does not have to be known a priori, therefore it must be established a priority task assignation criteria. In the same way, it is possible to exist duplicates (an agent perform more than one) according to the available resources.

Tactic Execution. Mission Assignation: The most adequate way for a proper execution, is the assignment of the different missions to each agent in order to reach the goal of the strategy. Each mission is assigned in a reactive way attending to its restrictions and the capacities of the agents. The assignation criteria can also depend on the sensitisation, or attend to an opportunistic pattern. Tactic objectives are the same as the strategy ones, but additionally must optimise the distribution of missions between agents.

4.2 Plan

The Plan describes the set of actions that the group of agents must perform for reaching the goal. This actions are grouped in Missions and this in turn are grouped in Tactics. This plan aims to the achievement of set of actions, Tactics and/or Missions, executed in a determined order, and carried out by a group of agents which can swap theirs missions in a opportunistic way. An HFSM describes the Plan in which each state is a Mission.

5 Experimental Use Case

This example tries to clarify the concepts explained above. The context for the example is the scenario shown in Figure 5. There a factory is simulated where the handling of materials is performed by 4 physical agents (gripper differential robots) that transport the freight from a charging dock and to the corresponding download box.

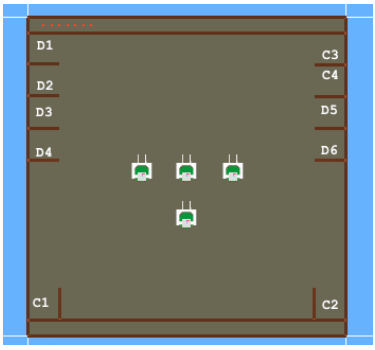


Fig. 5. Team Behaviour Example

5.1 Robots Coordination: Plan

In the coordination level is defined a three state plan and the corresponding tactic for each of these states, each one of this named as tactic.

1. **Merchandise reception:** A big amount of freight is receipt.
2. **Standard merchandise movement:** Some freight is receipt.
3. **Merchandise dispatch:** No freight is receipt.

5.2 Tactic

In the case that a big amount of freight is receipt a state machine evolves to the "Merchandise reception" state. The wheelbarrows must unload the freight and take the charge to one of the unload places. The tactic consists of 4 Missions, but these are not assigned in principle to any particular agent.

- **Mission I:** Freight Reception.
- **Mission II:** Freight Reception.
- **Mission III:** Freight Reception.
- **Mission IV:** Battery charging.

Tactic Execution. Mission Assignation: This scenario use 4 robots to achieve the goal. The tactic sets 3 equal missions to receipt the freight and leaving it in the proper place, while 1 robot reload its batteries. The assignment is done according to battery load criteria.

5.3 Mission

As example of Mission "Freight Reception" is explained: The goal of this mission is collect freight from a charging dock and unload it at the appropriate place. Another mission can be, relocating objects: load collection in a given box to move it to another.

5.4 SubMission

The first mission described above, could consist of a set of submissions as follows:

- Go to the loading dock.
- Adjust the freight to be picked up.
- Loading the freight.
- Go out of loading dock.
- Go to the unload site.
- Download freight.
- Go out of unloading dock.

As this is a simple example, all submissions are executed sequentially and in loop, but not always be this way.

5.5 Basic Behaviour

As example the first submission is explained. This submission is formed by 2 Basic Behaviour: *Displacement to site* and *Environment object collision avoiding.*

Displacement to site:

- The objective of the physical mobile agent is to perform a movement in the direction of a interest goal.
- The action, output, or control action, as the shown on the previous example, relies on the performance of the agent. In addition, the behaviour must obtain the direction and velocity of the displacement required for reaching the position of the interest place.
- The relative position between the robot and the goal site are the perceptual requirements.

Environment object collision avoiding:

- The objective of the physical agent is to avoid to collide with its surrounding objects while is in movement.
- The action, output, or control action, taking into account the exposed on previous examples, can be one of the components of the velocity vector which avoids the collision with a certain object.
- Perceptual requirements are the position of the surrounding objects.

6 Implementation

Although the model referred along this document is functional and has been tested, the evolution, tests and improvements are always progressing. Current implementation has been developed on C++, using a computer with a Linux distribution as main operative system, and has been tested using both a simulator which offers virtual representation of robots, and real mobile robots as elements to control. The robots used in this test are the humanoid platform Nao. In both cases tests have been done using the framework developed to attend the Standard Platform League (SPL) in the RoboCup[1] championship by the Hidalgos Team.

6.1 Development Model

The behaviour module has been implemented with the aim of be usable over several robots, so has been designed robot-API independent. Different behaviour components (Basic Behaviours, Missions, Submissions, Plans, Tactics, etc) use the c++ heritage mechanism to take the architecture benefits. Thus, one base class is generated by each component in a header file. The behaviour designer just must generate a class per component that extend the appropriate basis class, and add the behavioural functionality code. Next subsections explain each component in detail.

[1] `http://www.tzi.de/spl`

Basic Behaviour. In a Basic Behaviour derived class, the functionality has been split into 3 functions/blocks:

- Motivation Update: in this function user has to calculate and store the value of the current motivation for the behaviour in edition.
- Contribution Update: in this function user must set a proper values for contributions over which it has influence. Different behaviours could act over different actuators or system variables.
- Actuation function: takes the result of the fusion process and perform the actuation. Each behaviour decides which contributions need, and how to use them to actuate on the robot API.

Submissions. As was described, a submission is a set of Behaviours. In the constructor the designer must add all the required basic behaviours. Additionally user must afford a function that calculates and provide the accomplishment level of the current submission.

Missions. A Mission is a set of submissions that are executed following the criteria imposed by a finite state machine. That way, the programmer has to specify states and transitions to depict the flow of the machine.

- The states are conformed by submissions, the designer just need to instantiating the previously defined.
- Transitions are needed to indicate the relationships between states. The code that manage the transit condition, and the states of input and output, define the transition.

6.2 Behaviour Migration

After the design of behaviours, the code is compiled and linked as a dynamic library. The library could contain the complete behaviour or a subset. Availing of this mechanism, is possible to modify part of the behaviour (e.g. basic behaviour), link alone in a new library and send it to the robot, then the module is requested to load this behaviour in place of the current one.

7 Conclusions

The presented hybrid architecture presented in this paper take advantages from reactive and deliberative control. The aim of this architecture is to manage the behaviour of a heterogeneous group of physic agents. The layered structure allows a easy behaviour definition, obtaining a good reactive response in the low level and a simple planning in the deliberative level.

The group coordination level, deal with the missions assignment according the robots state and resource availability.

Finally architecture introduces sensory tasks in the behaviours system, thus resultant actions could improve the perceptual performance. System is being tested in a simulator and different mobile robots.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under the CICYT project Mission Based Control (COBAMI): DPI2011-28507-C02-02, under coordinated project High Integrity Partitioned Embedded Systems (HI-PartES): TIN2011-28567-C03-03, and under the collaborative research project supported by the European Union MultiPARTES Project: FP7-ICT 287702. 2011-14.

References

1. Aragues, R.: Consistent data association in multi-robot systems with limited communications. Robotics: Science and Systems, 97–104 (2010)
2. Aragues, R., Cortes, J., Sagues, C.: Distributed consensus on robot networks for dynamically merging feature-based maps. IEEE Transactions on Robotics (2012)
3. Arkin, R.C.: Motor schema based mobile robot navigation. The International Journal of Robotics Research 8(4), 92–112 (1989)
4. Asama, H., Habib, M.K., Endo, I., Ozaki, K., Matsumoto, A., Ishida, Y.: Functional distribution among multiple mobile robots in an autonomous and decentralized robot system. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation. IEEE (1991)
5. Benet, G., Blanes, F., Martínez, M., Simó, J.: A multisensor robot distributed architecture. In: IFAC Conference INCOM 1998 (1998)
6. Brooks, R.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2(1), 14–23 (1986)
7. Canas, J.M., Matellán, V.: Dynamic schema hierarchies for an autonomous robot. In: Garijo, F.J., Riquelme, J.-C., Toro, M. (eds.) IBERAMIA 2002. LNCS (LNAI), vol. 2527, pp. 903–912. Springer, Heidelberg (2002)
8. Choset, H., Nagatani, K.: Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. IEEE Transactions on Robotics and Automation 17(2), 125–137 (2001)
9. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. American Association for Artificial Intelligence, 343–349 (1999)
10. Hu, J., Xie, L., Xu, J.: Vision-based multi-agent cooperative target search. In: Control Automation Robotics & Vision (ICARCV), pp. 895–900 (2012)
11. Huq, R., Mann, G.K.I., Gosine, R.G.: Behavior-modulation technique in mobile robotics using fuzzy discrete event system. IEEE Transactions on Robotics 22(5), 903–916 (2006)
12. Jayasiri, A., Mann, G., Gosine, R.G.: Mobile robot behavior coordination using supervisory control of fuzzy discrete event systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, pp. 690–695 (2009)
13. Jayasiri, A., Mann, G.K.I., Gosine, R.G.: Behavior coordination of mobile robotics using supervisory control of fuzzy discrete event systems. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 41(5), 1224–1238 (2011)
14. Koenig, N., Howard, A.: Gazebo-3d multiple robot simulator with dynamics. Technical report (2006)
15. Lin, F., Ying, H.: Modeling and control of fuzzy discrete event systems. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 32(4), 408–415 (2002)

16. Madden, J.D.: Multi-robot system based on model of wolf hunting behavior to emulate wolf and elk interactions. In: 2010 IEEE International Conference on Robotics and Biomimetics, ROBIO, pp. 1043–1050 (2010)
17. Mataric, M.J.: Interaction and intelligent behavior. Technical report, DTIC Document (1994)
18. Olivera, V.M., Molina, J.M., Sommaruga, L., et al.: Fuzzy cooperation of autonomous robots. In: Fourth International System on Intelligent Robotics Systems, Lisboa, Portugal (1996)
19. McGann, C., Py, F., Rajan, K., Thomas, H., Henthorn, R., McEwen, R.: A deliberative architecture for auv control. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 1049–1054. IEEE (2008)
20. Munera, E., Muñoz, M., Simó, J., Blanes, F.: Humanoid Robot Self-Location In SPL League. In: Comité Español de Automática (CEA), XXXIII Jornadas de Automatica, 797–804 (2012)
21. Proetzsch, M., Luksch, T., Berns, K.: Development of complex robotic systems using the behavior-based control architecture iB2C. Robotics and Autonomous Systems 58(1), 46–67 (2010)
22. Qiu, D.: Supervisory control of fuzzy discrete event systems: a formal approach. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35(1), 72–88 (2005)
23. Aladebaran Robotics. NAO Software Documentation 1.12. Technical report (2012)
24. St-Pierre, M., Gingras, D.: Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system. In: 2004 IEEE Intelligent Vehicles Symposium, pp. 831–835 (2004)
25. Stoytchev, A., Arkin, R.C.: Combining deliberation, reactivity, and motivation in the context of a behavior-based robot architecture. In: Proceedings of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 290–295 (2001)
26. Nicolau, V., Muñoz, M., Simó, J.: KertrolBot Platform. SiDiReLi: Distributed System with Limited Resources. Technical report, Institute of Control Systems and Industrial Computing - Polytechnic University of Valencia, Valencia, Spain (2011)

Real-Time Vehicle Detection and Tracking Using Haar-Like Features and Compressive Tracking

Sara Maher Elkerdawi, Ramy Sayed, and Mohamed ElHelw

Nile University, Ubiquitous Computing Group,
Cairo, Egypt
{sara.maher,ramy.sayed}@nileu.edu.eg,
melhelw@nileuniversity.edu.eg
http://www.nileu.edu.eg

Abstract. This paper presents a real-time vision framework that detects and tracks vehicles from stationary camera. It can be used to calculate statistical information such as average traffic speed and flow as well as in surveillance tasks. The framework consists of three main stages. Vehicles are first detected using Haar-like features. In the second phase, an adaptive appearance-based model is built to dynamically keep track of the detected vehicles. This model is also used in the third phase of data association to fuse the detection and tracking results. The use of detection results to update the tracker enhances the overall framework accuracy. The practical value of the proposed framework is demonstrated in real-life experiments where it is used to robustly compute vehicle counts within certain region of interest under variety of challenges.

1 Introduction

Traffic is becoming an inevitable part of our daily lives especially in crowded cities which forces the demand for developing intelligent traffic surveillance systems. These systems help in monitoring, analyzing, and controlling traffic, while operating in real time and providing accurate information about traffic conditions. Traffic surveillance systems can be divided into intrusive and non-intrusive systems. The former include loops detectors that are inserted below road surface to detect metallic vehicle mass that crosses it. Such solutions incur high cost and safety risks associated with their installation and maintenance. The latter include traffic cameras that are easily installed and maintained to provide additional information such as vehicle trajectory and recognize activities of interest such as accidents and driving in the wrong direction. In this paper, we present a real-time vision framework that is suitable for traffic camera systems to detect and track vehicles within a pre-defined region of interest called the virtual loop. The paper is organized as follows: Section 2 states the related work, Section 3 introduces the proposed method, Section 4 discusses experimental results, and finally conclusion represented in Section 5.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_27,

2 Related Work

Most video-based traffic surveillance systems have a predefined method to select its region of interest (ROI) such as frame difference [1,2] and virtual loops [3,4]. Virtual loop methods employ prior knowledge to select the ROI directly. In this paper, the ROI is selected manually where road lanes are usually selected as ROI.

Video-based traffic monitoring is applied on three major stages: detection, tracking, and data association. In detection, most methods detect the vehicles using background subtraction [5] depending on motion information. Background reference frame is the core of background subtraction methods. Frame differencing between two successive frames may be applied [6] but it is sensitive to noise and illumination changes, parallax and motion in the background (*i.e.* trees and other non-vehicle objects). Other methods for background estimation are running average [7] or Gaussian Mixture model (GMM) [8] that estimates background reference to reduce parallax effect and periodic motions as well as illumination changes, but it will fail in crowded roads due to very slow motion. For the purpose of detecting vehicles independent on the motion and traffic flow, we used Haar-like features of vehicles instead of background subtraction techniques. For tracking, Kalman filter [9,10] or any dynamic model are usually used as a tracker for its simplicity and real-time performance. However, there are two limitations: linearity and the need to frequently update the object's state using detection/measurement in the correction. In order to bind detected targets and tracked targets, data association techniques such as one-to-one association and overlap-rate-based association can be used [16].

3 Framework Architecture

In this section we describe the proposed framework. Section (3.1) shows how we used Haar-like features and Adaboost cascade classifiers in the detection. Tracking is presented in Section (3.2) and finally linking the detection and tracking is described in Section (3.3).

3.1 Detection

The first stage in the framework is vehicle detection. First, preparing positive samples and negative samples manually so that vehicles are the positive samples and anything else that can form a background are the negative samples. Then Haar-like features as in Fig. 1. are used to extract different features of vehicles in different scale and shapes. More specifically, we used three kinds of features as used in face detection algorithm by Viola and Jones [11]. The value of a two-rectangle feature is the difference between the sum of the pixels within the rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a four-rectangle feature computes the difference between

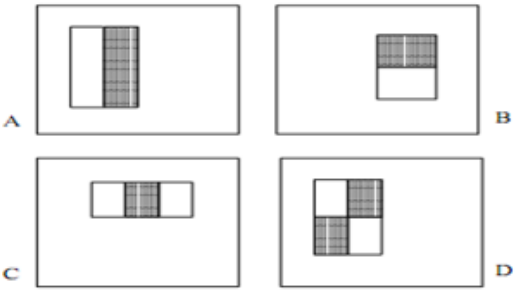


Fig. 1. Example rectangle features are shown relative to the enclosing detection window. The sums of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the gray rectangles. Two-rectangle features are shown in (A) and (B), (C) shows a three-rectangle feature, and (D) a four-rectangle feature.

diagonal pairs of rectangles. Given that the base resolution of the detector is 24x24, the exhaustive set of rectangle features is quite large. An integral image is computed in order to rapidly calculate these features.

Integral image can be computed using few operations per pixel and once it is computed, any Haar-like feature can be computed in constant time. A variant of Adaboost is used both to select a small set of features and to train the classifier. Adaboost is an algorithm for constructing a strong classifier as linear combination of weak classifiers. A weak classifier $h_j(x)$consists of a feature f_j, a threshold Θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1, & p_j f_j(x) \leq \mathrm{p}_j \Theta_j \\ 0, & otherwise \end{cases} \quad (1)$$

These weak classifiers are constructed in cascade manner which uses all the information collected from the output of a given classifier as additional information for the next classifier in the cascade. Constructing cascade of classifiers achieves increased detection performance while radically reducing computation time. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances (*i.e.* the threshold of a boosted classifier can be adjusted so that the false negative rate is close to zero). Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rate. The overall form of the detection process is that of a degenerate decision tree, what we call a "cascade" as shown in Fig. 2. A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window. Similar to decision tree, subsequent classifiers are trained using those examples which pass through all the previous stages. As a result, the second classifier faces a more difficult task than the first. The samples which make it through the first stage are harder than typical samples. At a given detection

rate, deeper classifiers have correspondingly higher false positive rates. After the learning phase, the detection stage concluded in extracting Haar-like features from an initial window size and scale it with a specified percentage that affect the accuracy of detection and performance time (i.e. the larger scaling percentage, the less computation time and the less false positives). As we'll see in the tracking system we choose this scaling to be large as we need real time performance and only care about detecting the target at least once during its life time within the virtual loop.

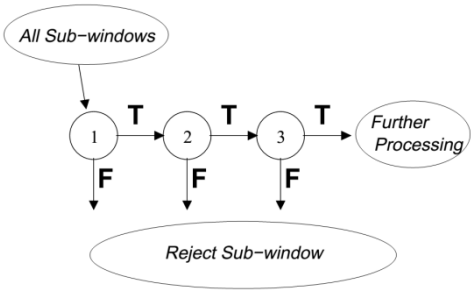


Fig. 2. Cascade Illustration. A series of classifiers are applied to every sub-window. The initial classifiers eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing the numbers of sub-windows have been reduced radically.

3.2 Tracking

Detection module only isn't enough; vehicles must be tracked over frames to eliminate counting the same target multiple times. Kalman filter or any dynamic model is a popular choice in tracking for its simplicity and real-time performance. But it needs to frequently use the detection/measurement to update the object state (position and velocity) in Kalman's correction. While our detection module focuses on accuracy (less false positive) than repeatability (detecting the vehicle multiple times); therefore it isn't suitable to integrate Kalman with our detection. Thus, a real time and less dependent on external measurements like Compressive Tracking (CT) is modified to integrate with the framework. CT is a generative model that represents the object efficiently as a feature vector extracted in a compressive domain and uses these features to discriminate the target from background including similar objects.

In this work we extend CT to deal with multiple objects tracking and use external information from detection if exists to enhance the object model with scaling.

Features

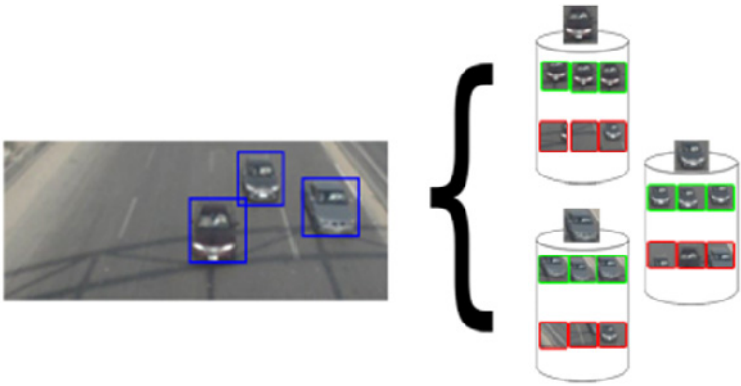
Using the same features as in [12], the features are extracted from Multi-Scale filter bank then a sparse measurement matrix is applied on the high-dimension features to generate low-dimension ones that preserve most of the salient information. Johnson-Lindenstrauss lemma [13] states that the distances between the points in a vector

space are preserved with high probability if they are projected onto a randomly selected subspace with suitably high dimensions. One of the ways to construct an approximation to such a projection matrix R is done by Achlioptas [14] defined as

$$r_{ij} = \begin{cases} 1, & with\ probability\ \frac{1}{2s} \\ 0, & with\ probability\ 1 - \frac{1}{2s} \\ -1, & with\ probability\ \frac{1}{2s} \end{cases} \tag{2}$$

Model Construction

For each target, set of positives and negatives samples are generated around the object centers to construct the model with Naïve Bayes classifier using the extracted features for each. Fig. 3. shows positive and negative samples subset with multiple objects.



(A) **(B)**

Fig. 3. Positive and Negative samples subset for multiple objects. (A) Detected vehicles, (B) For each vehicle, positive (top row) and negative (bottom row) samples are obtained to differentiate between the object and the background including similar objects.

As mentioned in [15], the random projections of high dimensional random vectors are almost always Gaussian. Thus, the conditional distributions $p(v_i|y = 1)$ and $p(v_i|y = 0)$ are assumed to be Gaussian distributed with four parameters $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ where v_i is each feature in the low dimensionality distribution and y represents sample label. Assuming the independency in the feature vector; the parameters are incrementally updated with

$$\mu_i^1 \leftarrow \eta\mu_i^1 + (1 - \eta)\mu^1 \tag{3}$$

$$\sigma_i^1 \leftarrow \sqrt{\eta(\sigma_i^1)^2 + (1 - \eta)(\sigma^1)^2 + \eta(1 - \eta)(\mu_i^1 - \mu^1)^2} \tag{4}$$

Where η is a learning rate declaring how adaptive the model is to new information, μ^1 is the feature average and σ^1 is the feature standard deviation. These parameters are updated using the extracted samples each frame. The samples are generated with fixed size (without scaling) to provide more representative model for the filters. Although the tracker successfully keeps track with scaling changes but it doesn't cover most of the object. This issue is fixed in data association phase as will be discussed in Section (3.3).

3.3 Data Association

While tracking seeks target localization continuity, data association stage seeks linking the independent detected targets of current frame with the tracked targets over frames. Data association must consider challenging factors like the absence of detected target, false detections and ambiguity. And to maintain a real time framework, we use a simple matching technique such as overlapping besides the constructed models built in the tracking phase as shown in Section (3.2). Algorithm (Table 1) describes data association algorithm. Before proceeding with the algorithm, several terms are defined:

Detected Target (DT): Every detected target returned from the detection phase explained in Section (3.1).

Tracked Target (TT): For each object's model, a prediction is generated which is called tracked target.

Get Confidence: apply the track's model on the detection target give the learnt parameters in model construction.

Update Classifier: updates the track's parameters by (3), (4) using a list of measurements.

Table 1. Shows how tracking and detection results are associated

Algorithm 1 DataAssociation(DT,TT)

```
FOREACH Track ∈ TT do
 MaxOverlap ← 0
 FOREACH Detection ∈ DT do
  MaxOverlap ← max(MaxOverlap,Detection ∩ Track)
  Confidence ← getConfidence(Detection,Track)
 end for
  if MaxOverlap > Thresh OR Confidence > ConfThresh
then
  updateClassifier([Track,Detection])
 else
  updateClassifier([Track])
 end if
end for
```

This algorithm ensures multiple criteria:

- **No ambiguity,** due to the backward-forward association where each tracked target either is associated with the most overlapped, a confident detection or none (absence of detections).
- **False detection,** is handled using the models' discriminative rules.
- **Scaling,** data association provides new scales detection to the tracker that helps updating the model with a more recent scale of the object.

4 Experiments and Results

As most of vehicle surveillance systems have been tested on different datasets; the problem is lacking representative datasets. Thus, self-collected dataset is used to test the performance. The dataset format is 640*480 pixels, 30 frames/sec and about 3 minutes long taken from desert road with variety of challenges scenarios:

- **Heavy flow,** as shown in Fig.4 there is slow motion and an overlap between targets, in which techniques with clustering or motion dependent may fail. Although detection doesn't detect all the vehicles present at each frame but each vehicle is detected at least once in the ROI over time. The tracking algorithm keeps track of the detected vehicles even with absence of detection in the next frames.
- **Illumination,** as shown in Fig. 5. there is a clear illumination change due to shadow. The detection module is not affected by it unlike frame differencing techniques besides it doesn't depend on the environment lighting condition. In addition adding blur noise to the samples in the cascade classifier training. The strong compressed features in the tracking are also not affected by the intensity changes.
- **Multiple Detection,** as shown in Fig. 6 resulted from many common detected vehicles features on the same vehicle that is successfully solved in the data association.

The following parameters were used in the experiments:

- **Detection,** Haar detection scale percentage parameter is set to 1.3 and thus balancing real-time and high true positive accuracy detection and search window is set to 15.
- **Tracking,** Given target location, the search window is relatively proportional to the target size. Positive samples generation are 4-pixels around the target. Negative samples' inner radii is set to 10 pixels around the object while the outer radii is 1.5 multiples the search window.

The framework achieves 93% accuracy and overall average running time of 0.2017 seconds. Table 2 gives insights about the error rate in each phase; notice how updating the tracker by the detections feedback from data association improves the results.

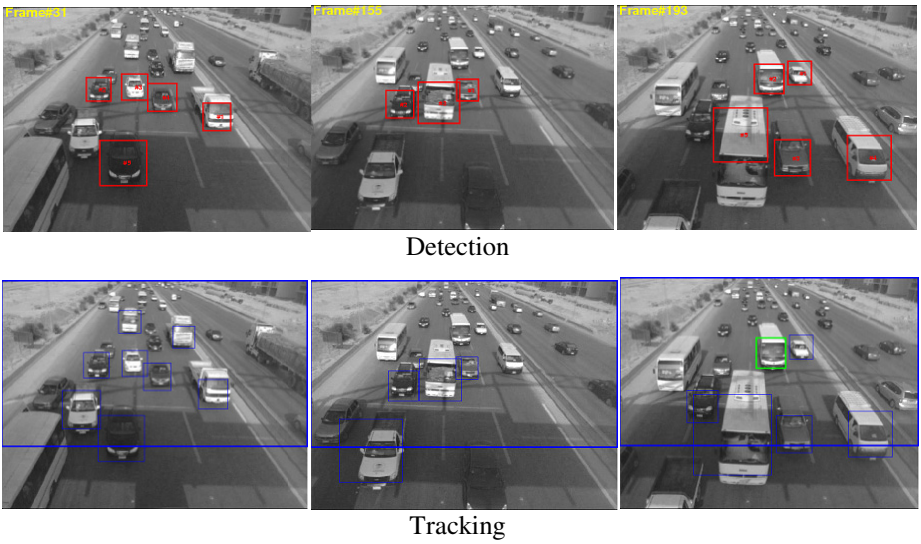


Fig. 4. Heavy flow: Top row shows detected targets (red), Bottom row shows tracked targets (blue) and first time detection (green). ROI controls tracked targets removal.

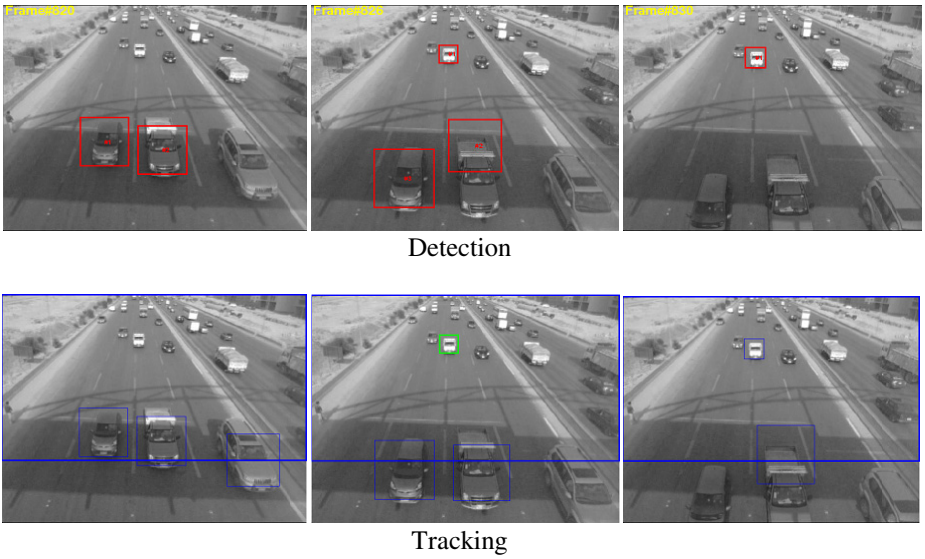


Fig. 5. Illumination: The technique successfully tracks vehicles with similar background intensities under different illumination

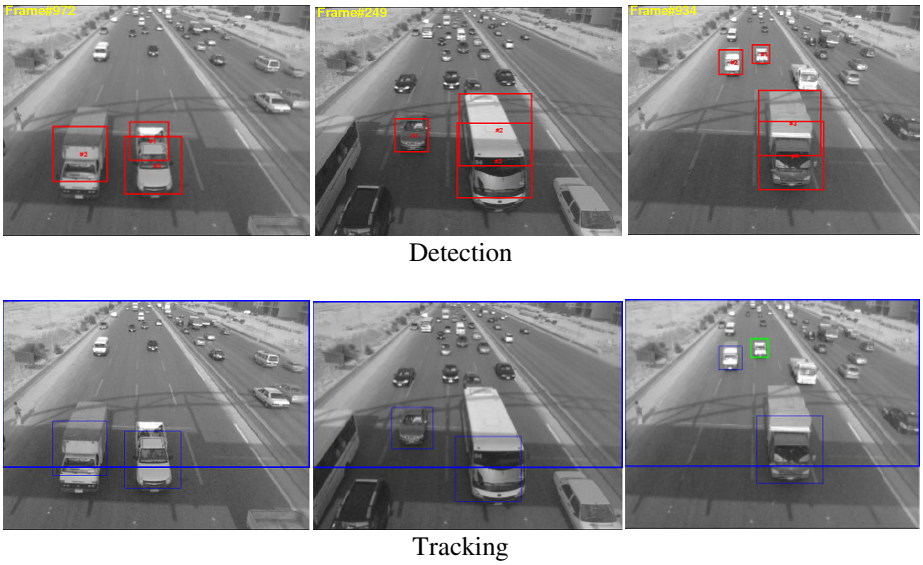


Fig. 6. Multiple Detection: Successful ambiguity elimination

Table 2. Error rate in each phase

Detection	Tracking only	Tracking updated by Data association
0.08%	0.11%	0.02%

5 Conclusions

Many applications in different fields are based on detection and tracking an object of interest, where high accuracy with real-time performance is usually required.

This paper proposed detection and tracking system applied in vehicles surveillance. In this paper, we proposed a simple yet robust framework that uses Haar-like features and Adaboost cascade classifier generating a general rule that learns vehicles appearance. Then the tracking keeps track of the detected targets using compressed features and naive-base classifier for each object to distinguish between objects and background. Experimental results show excellent performance.

This work can be extended as follows: First to enhance standalone system functionality, learning phase in detection can be done without human supervision by adding a robust algorithm that extract the positive samples with hundred percent confidence and then using them with a predefined negative samples to train the cascade classifier. Secondly, different scaled samples are to be generated in each vehicle's model to improve the tracking. Currently the scaling information are from the detection phase only so the tracking will predict fixed size bounding box unless a detection's sample corrects the model.

References

1. Paragios, N., Deriche, R.: Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 22(3), 266–280 (2000)
2. Xie, L., Zhu, G., Wang, Y., Xu, H., Zhang, Z.: Robust vehicles extraction in a video-based intelligent transportation systems. In: IEEE 2005 International Conference on Communications, Circuits and Systems (2005)
3. Pang, C.C.C., Lam, W.W.L., Yung, N.H.C.: A Novel Method for Resolving Vehicle Occlusion in a Monocular Traffic-image Sequence. IEEE Trans. on Intelligent Transportation Systems 5(3), 129–141 (2004)
4. Lai, A.H.S., Yung, N.H.C.: Vehicle-type Identification Through automated virtual loop assignment and blockbased direction-biased motion estimation. IEEE Transactions on Intelligent Transportation Systems (2000)
5. Mandellos, N.A., Keramitsoglou, I., Kiranoudis, C.T.: A background subtraction algorithm for detecting and tracking vehicles. Expert Systems with Applications 38(3), 1619–1631 (2011)
6. Cheung, S.-C., Kamath, C.: Robust Background Subtraction with Foreground Validation for Urban Traffic Video. EURASIP Journal on Applied Signal Processing 14, 1–11 (2005), UCRL-JRNL-201916
7. Tang, Z., Miao, Z., Wan, Y.: Background Subtraction Using Running Gaussian Average and Frame Difference. In: Ma, L., Rauterberg, M., Nakatsu, R. (eds.) ICEC 2007. LNCS, vol. 4740, pp. 411–414. Springer, Heidelberg (2007)
8. Zivkovic, Z.: Improved Adaptive Gaussian Mixture Model for Background Subtraction. In: Proc. ICPR (2004)
9. Robert, K.: Night-Time Traffic Surveillance: A Robust Framework for Multi-vehicle Detection, Classification and Tracking. In: Proceeding of the AVSS 2009, Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 1–6 (2009)
10. Yu, S.-H., Hsieh, J.-W., Chen, Y.-S., Hu, W.-F.: An automatic traffic surveillance system for tracking and classifying vehicles. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 379–386. Springer, Heidelberg (2003)
11. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of SimpleFeatures. In: Computer Vision and Pattern Recognition (2001)
12. Zhang, K., Zhang, L., Yang, M.-H.: Real-time Compressive Tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
13. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. Modern Analysis and Probability 26, 189–206 (1984)
14. Achlioptas, D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins. J. Comput. Syst. Sci. 66, 671–687 (2003)
15. Diaconis, P., Freedman, D.: Asymptotics of graphical projection pursuit. Ann. Stat. 12, 228–235 (1984)
16. Mao, H., Yang, C., Abousleman, G.P., Si, J.: Automated multiple target detection and tracking in UAV videos. In: Proc. SPIE 7668, Airborne Intelligence, Surveillance, Reconnaissance (ISR), vol. 2 (2010)

Evolutionary Learning of Basic Functionalities for Snake-Like Robots

Damaso Perez-Moneo Suarez and Claudio Rossi

Centre for Automation and Robotics, UPM-CSIC, Madrid, Spain
damaso.psuarez@alumnos.upm.es, claudio.rossi@upm.es

Abstract. The objective of the work presented in this paper is to investigate the optimal learning strategy for snake-like modular robots using a (1+1) Evolutionary Algorithm. We take into account three different but correlated tasks: efficient locomotion, reaching a given point and obstacle avoidance.

Starting from earlier results on locomotion, we have performed three different sets of experiments. In the first, the snake must learn to go to a goal point, and we investigate how employing different fitness functions affect the learning of this task. In the second experiment the snake must learn how to avoid obstacles. In this experiment we test two possible behaviors, called *mixed* and *switched* strategies. Finally, in the third set of experiments, we introduce the concept of *incremental learning* and compare it with the "all-at-once" learning schemes of the first two experiments.

The results of the simulations indicates that the modular robots are able to learn both tasks with the (1+1) Evolutionary Strategy adopted, and that the fitness function that explicitly rewards each of the tasks perform better that the fitness function that takes into account the locomotion task only implicitly, rewarding only the reaching of the target point. We also demonstrate that the obstacles avoidance configuration with only one behaviour (mixed) is better than the configuration with two behaviours (switched) and that incremental learning provide a faster evolution towards good controllers.

Keywords: Evolutionary robotics, embodied evolution.

1 Introduction

The aim of the research presented here is to develop new evolutionary strategies for generating robot controllers for real, physical robots.

The scenario we are working on is an ecosystem of robots that evolve both their physical structure (body) and control software (mind). In this ecosystem, like in the real world, individuals have to learn some basic skill to survive, or improve the abilities that they have inherited from their parents. They will have to learn to find energy sources, without which they would eventually starve to death. Occasionally, they will meet and mate. New offspring will be generated by crossing over their genetic material. These will be at all effects *new* individuals, whose morphology will be determined by the morphology of their parents. We assume that an instrument for their physical implementation exist. This shall be an automated production mechanism that, following

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_28,

their genetic blueprint, would build the newborn individuals, by assembling basic parts ("organs") or even by producing them on-the-fly by 3D printing technology [1].

At this stage, we are centering our research on *lifetime learning*, i.e. the process where newborn individuals have to learn (or improve) their skills. Note that new individuals *are not aware of their morphology*. The only information they have is the number of modules they are formed of. Thus, the first thing they have to learn is how to control themselves.

We implement lifetime learning with a (1+1) on-line and embedded evolutionary strategy. This means that the robots manage internally and in real time the evolution process. This simple strategy is preferred since our aim is to perform on-line evolution on real robots. In the case of classical EAs adopting a population on $n >> 1$ individuals, multiple candidate controllers need to be evaluated at each generation, meaning multiple test runs of the robot. This is clearly time consuming and is preferred when evolving off-line controllers, i.e. when controllers are first evolved, typically in simulations, and the best one is finally deployed to a real robot.

Lifetime learning implies that the robots have to learn some basic skill to survive, such as finding energy sources, finding mates and avoiding obstacles. All such tasks assume a basic skill, that is the ability to move efficiently.

In previous work [2], we have compared different encoding strategies for the robots controller, and showed that good controllers for modular snake-like robots to learn efficient locomotion can be generated. Thus, in this work robots controllers will adopt such encoding, which will be briefly described below. Here, we focus on the tasks of going to a goal point, that in future will be either a food source or a potential mate, and the task of avoiding obstacles that may be present in the environment.

When it comes to learn different tasks, the question of which is the best way to take them into account to reward good individuals arises. The first objective of the work presented in this paper is to investigate how employing different fitness functions affects the learning of the target reaching task. Clearly, in order to reach a target point, efficient locomotion must be employed. Thus, taking into account only the final distance to the target point takes intrinsically into account efficient locomotion. However, one can design a fitness function that is composed of different terms, each rewarding a particular skill, in this way taking into account explicitly the two (or more) tasks.

For the next task (obstacle avoidance) we compare two different strategies. It is easy to see how the task of avoiding an obstacle may generate conflict with the task of going to a goal point, in the case the obstacle is in the path to reach the goal point. In this case, two strategies can be adopted. In the first, the controller has to learn how to take the conflicting goals into account, properly weighting two terms of the fitness function. We call this the *mixed* strategy. In the second, a higher level behaviour controller switches between the "goto point" and "avoid obstacles" behaviours. This strategy is called *switched* strategy.

This first two sets of experiments was aimed at finding the best strategy for the robots to learn all tasks at once. In the third part of this paper we introduce a completely different learning scheme, based on *incremental* learning, where robots learn one task at a time, building from previous experience.

The robot ecosystem we work with exists in a virtual world, simulated adopting a realistic 3D simulator, that faithfully reproduces both the real world physics and the robots' kinematics and dynamics as well as the physical interactions with the world's objects.

Besides aiming at reducing the *reality gap*, we share the growing feeling in the research community that "matter matters"[3]. That is, the body, its shape, and its physical interaction with the environment have an impact on the way the mind is shaped. Therefore, we include realistic physics in our simulated ecosystem, in such a way that controllers will be evolved taking into account such physical interactions.

The physical realism of the simulations implies a high computational cost, making these simulations very slow[1]. For this reason, the modular robots we adopt at this stage are very simple. They are composed of a number of modules connected to each other by a motor that allows horizontal movement. However, by including an additional degree of freedom to the joints, a wide variety of movements in 3 dimensions could be generated. Also, allowing more connections, more complex morphologies (e.g. in form of H, that would resemble a four-legged robot) can be generated. Figure 1 depicts the basic modules and snake-like robot.

The parameters required by each motor are the three classical parameters of periodic undulatory motion: rotation speed, amplitude and phase, plus a bias value that allows curved paths (see below).

Related Work

There is a considerable body of work on controllers for snake-like robots controllers. Undulatory motion can be generated e.g. by central pattern generators [4], Neural Networks [5,6,7,8], or modulated architectures [9,10,11].

Central pattern generators and neural networks are both powerful techniques, but have the drawback of needing too much information (think, e.g., at the number of connection weights that the have to be evolved).

It must be pointed out that the most usual configuration for evolving robot controllers is off-line and off-board: an evolutionary algorithm is run in an external computer, and controllers are then deployed into the robots (either real or simulated) to be evaluated. This is the case e.g. of the one of the most known evolutionary robotics works, the evolving virtual creatures of C. Sims [7] and J. Bongard [8].

Recently, on-line evolution has also been proposed in the literature (see e.g. [12], where virtual agents are evolved in continuous time, but without lifetime learning). In particular, Eiben et al. [13] proposed an embodied on-line and on-board distributed evolution, where each robot evolves on board a population of candidate controllers, and at given times robots exchange controllers with each other. This strategy is called *island method*. The strategy we adopt here adopts just one controller for each robot, without exchanges between robots. This approach is similar to the one proposed in [14], where a (1+1)-online Evolutionary Algorithm is proposed. The main differences with this work lays in the genotypes (robot controllers) representation (Neural Networks vs.

[1] With 2 robots with 4 modules in the arena, simulation time was $t = 0.3\times$ with respect to real time, while with 6 modules, simulation time was $t = 0.1\times$

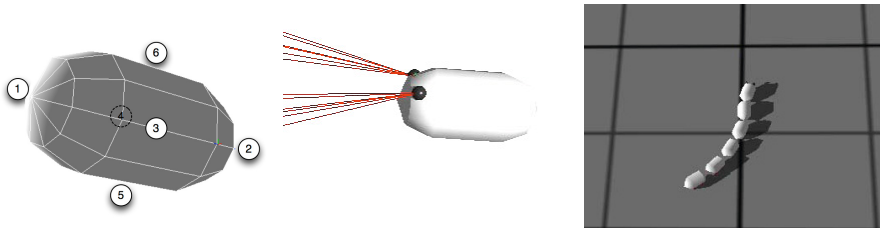


Fig. 1. Left: Body modules and possible connection points. Centre: "head" module equipped with two distance sensors. Right: Modular snake consisting of 6 body segments.

periodic functions) and in the fact that our approach adapts better to different robot morphologies.

Finally, the concept of incremental learning has been introduced by Bongard [8] with the purpose of evolving robot morphologies starting form simple configurations to more complex ones. Here, instead, we use incremental learning for learning increasingly difficult tasks.

2 Snake Locomotion

In this section, we briefly recap some basic concept on snake serpentine locomotion.

We model the snakes movement using the known "serpenoid" functions described by Hirose (see e.g. [15]):

$$\phi_i(t) = 2\alpha \sin(\omega t + (i-1)\beta) \tag{1}$$

$\phi_i(t)$ is a sinusoidal function that determines the position of joint i between two body segments and n is the number of body segments of the snake ($i = 1 \ldots n-1$). Parameters ω, α and β determine the shape of the serpentine curve realized by the snake-like robot, defining respectively angular speed, amplitude and phase of the oscillation of the joints.

The set of all such parameters, called *genes* is stored in an array of floating-point values, which is called the *chromosome* of the robot's controller. Chromosomes undergo *mutations* that result in different controllers. A mutation consist on modifying each gene adding a random quantity drawn from a Gaussian distribution $N(0, \mu)$, where μ is called the *mutation rate*.

As commented above, previous work [2] allowed us to reach the conclusion that the best configuration for the evolution of snakes was the so called *He1Sig* encoding, which uses the equation:

$$\phi_i(t) = A_i \sin(\omega_i \cdot t - \varphi_i) \qquad i = 1 \ldots n-1, \tag{2}$$

where n is the number of modules of the robot and each join has its own values for amplitude (A_i), phase (ω_i) and frequency (φ_i). The *He1Sig* encoding has a common mutation rate μ (range of the gaussian perturbation) for all the genes that is also

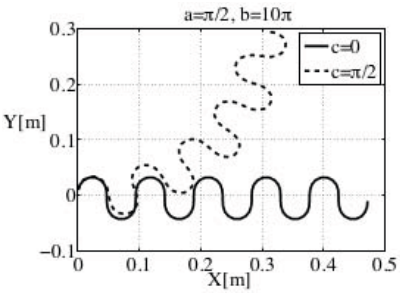


Fig. 2. Example of the effect of c in the trajectory of the snake locomotion

stored in the chromosome (and therefore undergoes evolution as well). Thus, the total chromosome length is of $3 \cdot (n-1) + 1$ genes. Gene values are normalized in the range $[0, 1]$.

2.1 Controlling Direction

In order to direct to a given location, the robots could rotate in place to point towards the location, and then move straight, a strategy commonly used in mobile robotics. This option, besides being somehow "unnatural", would require switching between to behaviours: rotate and move straight. A more elegant way to execute this task is to bias the locomotion of the snake so that in performs a curved trajectory, ending at the target point.

Equation (1) can be easily modified for such purpose (see [16] for more details) as:

$$\phi_i(t) = 2\alpha \sin(\omega_s t + (i-1)\beta) + \gamma \tag{3}$$

Adding the term γ makes the wave motion of the snake to be asymmetric, resulting in a curved path. The term γ can be expressed as a function of the desired curvature radius $c : \gamma = -\frac{c}{n}$ (see also Fig. 2).

3 Getting to a Goal Point

In our case, the desired curvature radius is not explicitly calculated. Instead, it is a parameter encoded into the chromosome that takes into account the angle between the current snake orientation and the line connecting the snake and the target point. In other words, the snakes have to learn how much to bias their locomotion in order to reach the target point. Therefore, Equation (2) is added a path curvature term:

$$\phi_i(t) = A_i \sin(\omega_i \cdot t - \varphi_i) + C_i \cdot angle(t) \tag{4}$$

where C_i are additional $n-1$ genes, each joint having its own bias term[2] and the term $angle(t)$ is the angle between the snake orientation $u(t)$ and the direction of the target

[2] Following the same logic that each joint has its own set of parameters, even though these would not be strictly necessary.

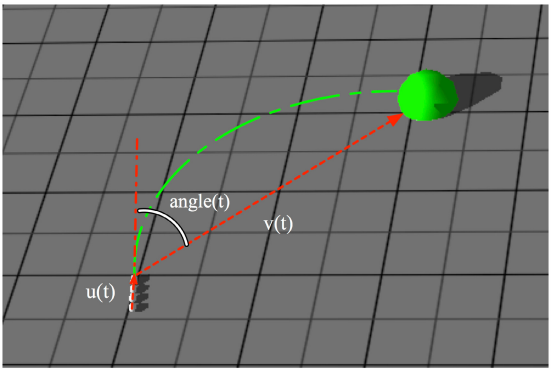


Fig. 3. Illustration of the angular component of the motion. The green line is the ideal trajectory.

point $v(t)$ (see Figure 3)[3]. Thus, chromosome length is finally $3(n-1)+1+(n-1)$ genes.

3.1 Fitness Function

The first question to be addressed is how to evaluate how good is each individual, that is, how to define the fitness function for the task "go to point".

For teaching the "goto point" skill, we compared two types of fitness function, one that only takes into account the distance from the target (F_d, Equation (5)), and one that takes into account separately the motion and the distance to the target ($F_m F_d$, Equation (6)):

$$Fitness_{Fd} = F_{distance} = (1 - \frac{Dist_{Final}}{Dist_{Initial}}) \tag{5}$$

where $Dist_{Initial}$ and $Dist_{Final}$ are respectively the initial and final distances from the target (note that reaching the objectives gives a fitness of 1);

$$\begin{aligned} Fitness_{FmFd} &= 0.5 \cdot F_{Movement} + 0.5 \cdot F_{distance} \\ &= 0.5 \cdot \frac{dist_{straight}}{dist_{accum}} + 0.5 \cdot (1 - \frac{Dist_{Final}}{Dist_{Initial}}) \end{aligned} \tag{6}$$

where $dist_{accum}$ is the distance travelled in relation to the initial position of the snake, and $dist_{straight}$ is the motion in relation to the orientation of the snake. The ratio $dist_{straight}/\ dist_{accum}$ expresses how much of the distance travelled was actually in the longitudinal direction (see Fig. 4).

Equation (5) rewards individuals that actually get close to the target point, regardless the way they reach it. Although it appears clear that in order to reach the target efficient

[3] Note that as the snakes moves, the angle varies and the theoretical ideal trajectory is an arc of circumference.

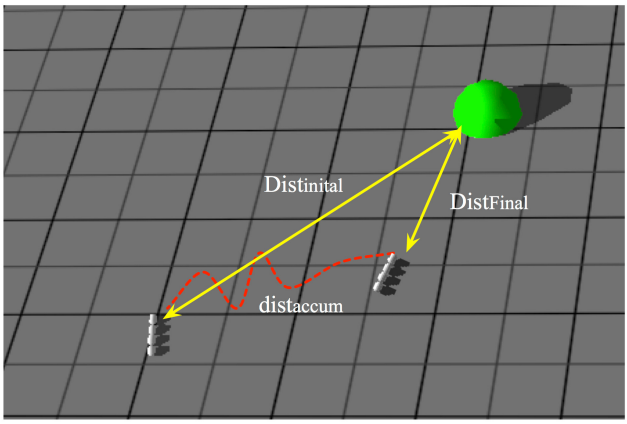


Fig. 4. Illustration of distance components of the fitness

locomotion must be developed, this function may reward erratic movement such as random jumps or rolling motion.

Equation (6) explicitly rewards motion pure serpentine motion, i.e. along the longitudinal direction of the snake.

3.2 Experiments

In order to assess the effectiveness of each of the two solutions described above, we have performed a set of experiments using the 3D simulator *Webots*, which adopts the ODE (Open Dynamics Engine[4]) for collision detection and simulation of rigid body dynamics. The ODE library allows accurate simulation of physical properties of objects such as velocity, inertia and friction.

Table 1. Summary of the parameters used in the experiments

Parameters of the experiments						
Configuration	FmFd4Mod	Fd4Mod	FmFd5Mod	Fd5Mod	FmFd6Mod	Fd6Mod
Fitness	$Fitness_{FmFd}$	$Fitness_{Fd}$	$Fitness_{FmFd}$	$Fitness_{Fd}$	$Fitness_{FmFd}$	$Fitness_{Fd}$
Size	4		5		6	
Representation	Heterogeneous (chromosome length: $3(n-1)+1+(n-1)$ real valued genes)					
Algorithm type	(1+1) evolutionary strategy					
Generations	500	500	500	500	500	500
Mutation	Gaussian perturbation $N(0,\mu)$, where μ is part of the chromosome					
Crossover	N/A					

We have performed experiments with three different configurations, consisting of 4, 5 and 6 modules. In order to avoid bias for a fixed position, for the snakes to learn to go

[4] `http://www.ode.org`

to a particular point, we calculate the fitness for 4 layouts. In this way, every controller has to learn to go to different target the points. In each layout the snake is placed in the center of the arena, and the target point is placed at one of the four corners of a square.

In each fitness evaluation, the snake was allowed 20 seconds (simulated world time) to get the target point. As they have four objectives in four layouts, each controller is tested for 80 seconds. Each snake was allowed to evolve for 500 generations.

Experiments have been performed for all combinations of snake size and fitness function. The parameters of the experiments are summarised in Table 1. For each configuration, 10 independent runs were performed.

Table 2. Summary of the experimental results for the "goto point" task

	Controller					
	FmFd4Mod	Fd4Mod	FmFd5Mod	Fd5Mod	FmFd6Mod	Fd6Mod
Best	0.89204	0.55795	0.84927	0.94464	0.93436	0.96827
Mean	0.64271	0.32916	0.75409	0.73339	0.89686	0.74027
Std. dev.	0.14616	0.15417	0.080097	0.22175	0.050982	0.29902

Figure 6 shows a comparison of the average fitness obtained for the different configurations (recall that fitness is normalised in such a way that the perfect behaviour has fitness 1).

The results of the experiments have shown that, for all different snake sizes, the best controllers are the ones that are generated using the fitness $Fitness_{FmFd}$ (Eq. 6), that is, the one that explicitly rewards efficient locomotion. Note also that standard deviations are smaller, indicating a more stable behavior. This can be because adding the locomotion term condition eliminates controllers that achieved getting close to the goal points "by chance" with a chaotic movement, and therefore there is a bigger variance between lucky runs and bad runs..

Note that longer snakes achieve better solutions. This is easily explained since in equality of conditions, longer snakes move faster.

4 Avoiding Obstacles

Also in the case of avoiding obstacles, the robot shall modify its trajectory performing a curved path. In this case, the curvature will depend on the readings of the two range sensors mounted in its head (see Figure 1, center).

Calling $d_s = Value\_left\_sensor - Value\_right\_sensor$ the difference between the values of the distance sensors (higher values mean closer obstacles), Equation (4) can be added a curvature term that let the snake avoid obstacles:

$$\phi_i(t) = A_i \sin(\omega_i \cdot t - \varphi_i) + D_i \cdot d_s \tag{7}$$

Again, the snakes have to learn how much to bias their locomotion in order to avoid the obstacles, i.e. discover the correct values of weights D_i. As before, D_i are additional $n-1$ genes, biasing the sinusoidal actuation of each of the joints.

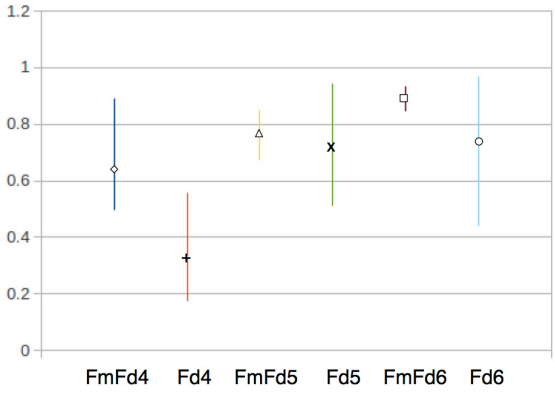


Fig. 5. Best fitness and standard deviation obtained for the different configurations (average over 10 runs)

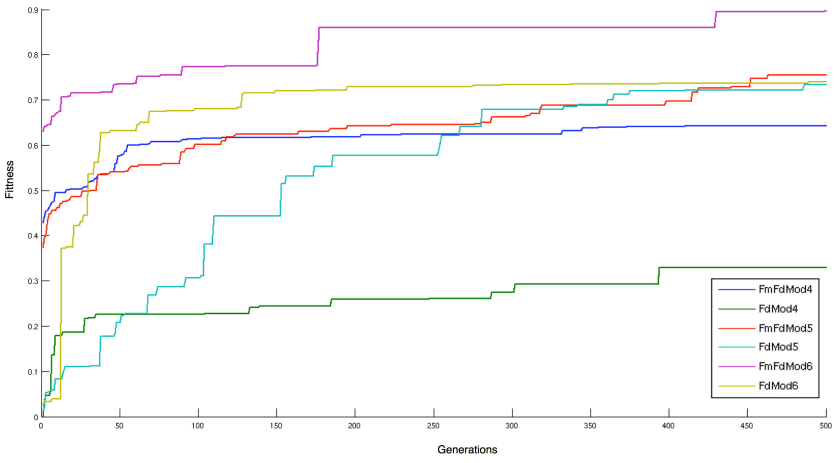


Fig. 6. Best fitness of the six configurations analysed (average over 10 runs)

However, this time the task is more complex, since obstacles are on the way to the goal point of the robot, and thus there are two conflicting objectives. There are two ways to deal with this situation. One is changing between behaviours go to point/avoid obstacle ("switching" strategy) and one consists on putting bias terms of Eq. (7) and Eq. (4) altogether ("mixed" strategy).

Switching Strategy. The idea in this first configuration is that the robot learns tow separate two behaviours. If one of the sensor values are greater than a given threshold T (an obstacle is detected) then the controller uses the *avoid* behaviour, if not it uses the *goto_point* behaviour.

```
// switched strategy
d_s=Value_right_sensor-Value_left_sensor;
if (Value_right_sensor>T || Value_left_sensor>T)
   targetPositionH[i] = amplitudeH[i]*sin(omegaH[i]*t+phaseH[i])
                                      + D[i]*d_s;
else
   target_positionH[i] = amplitudeH[i]*sin(omegaH[i]*t+phaseH[i])
                                       + C[i]*angle;

---------

// mixed strategy
d_s=Value_right_sensor-Value_left_sensor;
target_positionH[i] = amplitudeH[i]*sin(omegaH[i]*t+phaseH[i])
                                     + C[i]*angle + D[i]*d_s;
```

Fig. 7. Pseudo-code of the two strategies for avoiding obstacles and reaching a target point. Top: switching strategy, bottom: mixed strategy. The variable $targetPositionH$ is servo's reference position.

Figure 7 (top) shows the pseudo-code of this strategy. Note that in the case of the switching strategy, threshold T is not a fixed parameter, but it is encoded in the chromosome and therefore undergoes evolution as well.

Mixed Strategy. The second option considers only one behaviour. The controller must learn to give sufficient importance to each aspect of behaviour in order to go to a point while avoiding obstacles. In other words, we let evolution discover the correct mixing (i.e. values for the terms C_i and D_i).

Fitness Function

In either case, in order to design the fitness function for this task, we have to define a term that evaluates how good is the robot in avoiding obstacles. This can be measured by how much time it takes to actually avoid it and how close it got.

We take into account the worst case, the higher value of the two sensors, and compute its accumulated value F_{avoid} that sums the values of the sensors, and the time during which the sensors detected the obstacles, measured in number of steps N_{steps}. Thus:

$$Fitness_{Fa} = (1 - \frac{F_{avoid}}{N_{steps}}) \tag{8}$$

In the previous section we have seen that the fitness function that explicitly rewards the different tasks performed better. So, in for the experiments involving learning to avoid obstacles described below we used the following fitness equation (see also Eq. (5) and (6):

$$Fitness_{FmFdFa} = \big(F_{movement} + F_{distance} + (1 - \frac{F_{avoid}}{N_{steps}})\big)/3$$

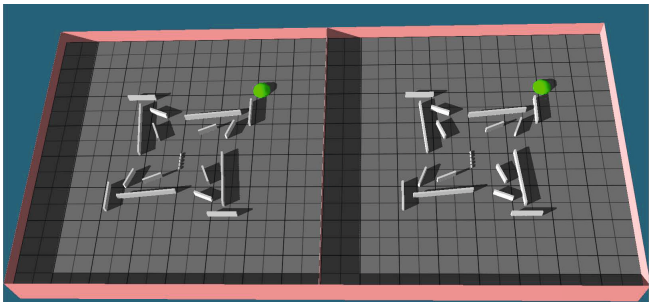


Fig. 8. Arena with obstacles. The starting position of the snake is the center of the arena. The green point depicts the target point.

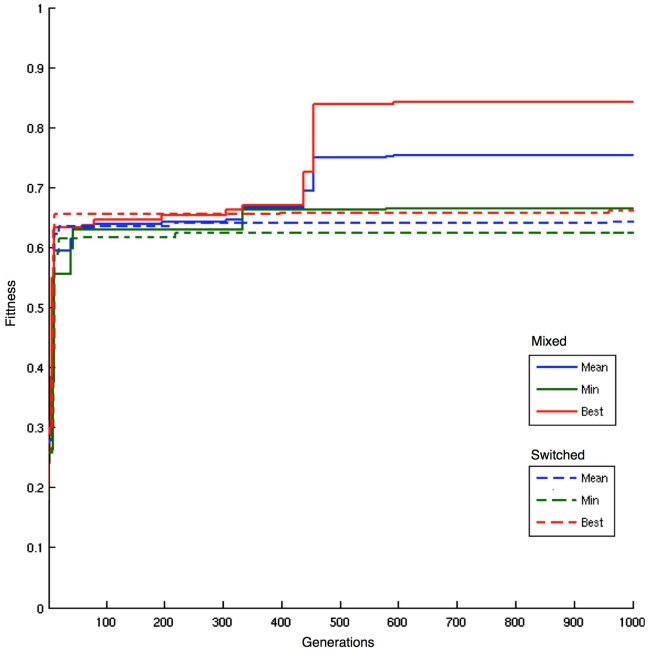


Fig. 9. Experimental results for the mixed and switched strategy

4.1 Experiments

We have performed tests in order to compare the two obstacle avoidance strategies.

In these experiments the snakes had to learn efficient locomotion, to go to a target point and to avoid obstacles. Therefore we used a similar configuration as in the previous set of tests, with targets placed at four different points, but adding obstacles in the arena (see Fig. 8).

Each controller is allowed running for 40 seconds at each iteration, since adding obstacles makes it harder to reach the target point, and therefore more time is needed.

The ability to reach all four targets was taken into account. Hence, each controller is tested for $40 \cdot 4 = 160\ seconds$.

The number of generations during which the controller is evolved has been increased in this experiment. Although in the previous task the snakes had managed to learn the task of going to a target point in a short time (less than 300 generations), this task is much more complex, and almost 1000 generations are needed to evolve good controllers. Experiments were performed only for the snake of the intermediate size of 5 modules.

Table 3. Summary of the results for the mixed and switched strategy

	Mixed strategy	Switched strategy
Best	0.84115	0.66055
Average	0.75281	0.64218
Std. dev.	0.12492	0.025981

The results can be seen in Table 3 and Figure 9. The results show that the mixed strategy achieved better results with respect to the switched strategy.

It also appears that the switched strategy get more easily stuck in local minima, since as it can be noticed in Fig. 9, the fitness increases at the beginning of the simulations but soon reaches a stable value.

5 Incremental Learning

In the las series of tests, we introduced the concept of *incremental learning*. Incremental learning consists on letting the robot learn one task at a time, building from what previously learned. In practice, this means that robot controllers are first evolved to produce efficient locomotion. Then, good controllers for locomotion are further evolved adding the new task of reaching a target location. Finally, good controllers are chosen and evolved for accomplishing the task of avoiding obstacles.

Clearly, this is different from what we have done before, where controllers for the "locomotion+reaching" tasks and for the "locomotion+reaching+avoidance" tasks where evolved from scratch, starting form chromosomes with random values. In the following we call this strategy "all-at-once" learning.

Hence, in each experiment, we choose the best controller from the previous experiment to learn the new skill.

When implementing this strategy, a series of issues must be dealt with. First, the size of the chromosome is different in each step. In the first experiment, which is the simple movement, chromosome needs $3 \cdot (n-1) + 1$ genes, which refer to the 3 variables needed by $n-1$ motors (amplitude, phase and frequency) plus one for the mutation rate μ that evolves with the rest of the chromosome. In the second experiment, reaching a target point, we had to add a variable to each motor, which is what allows the curved path. The same happens in the third experiment, avoid obstacles, which requires one more variable for each motor. New chromosomes were generated adding genes to the new ones and initializing these with the value 0.

The second issue is how to consider the old fitness in the new experiments. It was decided to maintain the previous fitness as initial value for the new experiments.

5.1 Experiments

The experiments performed were the same that was carried out done previously. A brief explanation is shown below.

First Step: Locomotion. In this step, each controller is checked for 10 seconds and the distance traveled was measured to calculate the fitness. In practice, this is exactly the experiment performed in [2].

Second Step: Reaching a Target Location. In this experiment, each snake had to get to four points at the four corners from an imaginary square, as described in Section 3. Fitness defined by Eq. (6) was adopted, since it shown to be the best option.

All the other simulation settings were kept as for the experiments of Section 3. As before, the snakes had 20 seconds to get to each point, always starting from the same starting point, so each controller is tested for $20 \cdot 4 = 80 seconds$. The only difference was that chromosomes were not initialised randomly, but were the best chromosomes resulting from the previous step.

Third Step: Obstacle Avoidance. In the third step, the best chromosomes resulting from the second step, i.e. controllers that were good at the locomotion and target reaching tasks, were used as starting point to evolve new chromosomes for controllers capable of avoiding obstacles. All the other simulation settings were kept as for the experiments of Section 4, and the "mixed" strategy was adopted.

The results obtained in the experiments are shown in Figures 10 and 11, which compare incremental learning to all-at-once learning described in the first part of the paper. Numerical data are summarized in Tables and 5.

The comparison between the two strategies shows that the controllers achieve similar results both as far as fitness and learning time is concerned (cf. 4), although incremental learning appears to be faster and has a slightly lower variance.

However, when increasing the difficulty, adding the obstacle avoidance skill, some important differences appears. First, on average, better controllers are produced with incremental learning. Second, it appears that the evolution process is more stable, in

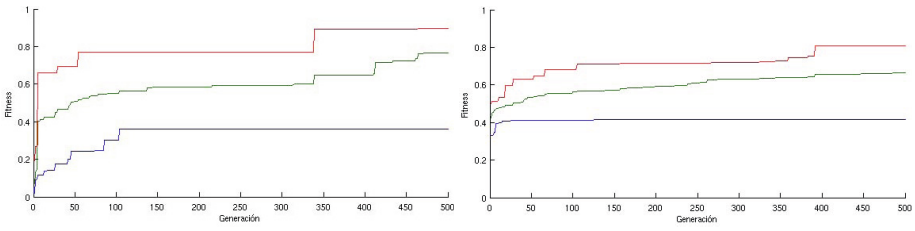


Fig. 10. Comparison between the minimum, maximum and average best fitness between all-at-once learning (left) and incremental learning (right) for steps 1 and 2

Table 4. Comparison between the minimum, maximum and average best fitness between all-at-once learning and incremental learning for steps 1 and 2

	Incremental	All-at-once
Best	0.8424	0.899
Average	0.6906	0.767
Std. dev.	0.1134	0.163

the sense that there is less difference between "good" and "bad" controllers (the standard deviation is notably smaller with respect to the all-at-once strategy). Finally and more important, evolution time is dramatically reduced. For instance, a fitness value of 0.7 (which denotes a good controller already) is achieved by incremental learning in approximately 50 generations, while learning from scratch took over 500. This last feature is extremely important since experiments are very time consuming due to the high computational cost of the physical simulations.

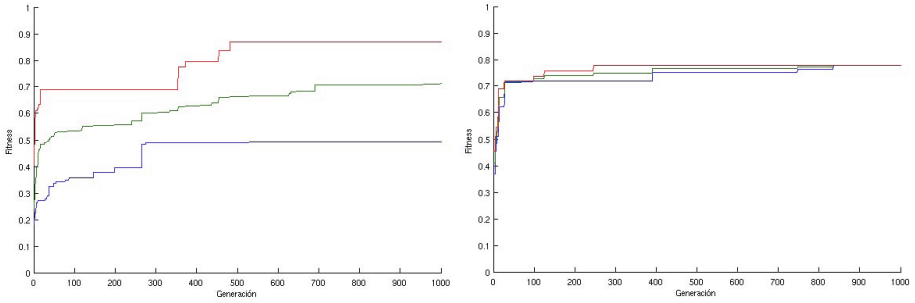


Fig. 11. Comparison between the minimum, maximum and average best fitness between all-at-once learning (left) and incremental learning (right) for steps 1, 2 and 3

Table 5. Comparison between the minimum, maximum and average best fitness between all-at-once learning and incremental learning for steps 1, 2 and 3

	Incremental	All-at-once
Best	0.77939	0.86782
Average	0.77839	0.71441
Std. dev.	0.00141	0.14309

6 Discussion and Conclusions

The ultimate goal of this research is to get a group of modular robots be able to learn basic survival tasks, such as moving, reach food, avoid obstacles and find a right partner. The range of possible combinations and design choices is extremely wide. For this reason, we are proceeding step by step, studying optimal design choices incrementally.

After the first set of experiments, aimed at studying good chromosome encodings for efficient locomotion (the very basic skill for survival), we moved to the next step, learning to go to a target point. Here, the issue of how evaluate individuals has been dealt with.

Next, we dealt with obstacle avoidance, which introduced the problem of how to deal with conflicting objectives.

We can summarise the results of these tests stating that for learning different -but related- tasks, the best option is to design fitness functions composed of various terms that reward each individual task, and good individuals are the one that perform better overall. We have also seen that a single behaviour taking into account all the goals is capable to be correctly tuned by evolution with no need of higher level behaviour controllers, although for more complex combinations a high level controller may be needed.

This has a further advantage. Besides global fitness, it is possible to know which particular skills the individuals are good at. Thus, when mating, individuals can be chosen taking into account an additional dimension: not just one number, but an array of values. In future work we plan to exploit such an additional dimension letting the individuals *choose* whom to mate with, according to its skills. For instance, one individual that is good in finding food, but not so good in avoiding obstacles, would want to mate with an individual with complementary skills, hoping that the offspring can inherit both good skills.

Finally, we have seen that incremental learning may provide an effective way to deal with learning different tasks in reasonable time, a key issue to perform extensive simulations. This is in fact similar to what children do at school, and in fact the term "robot school" has also been used to describe this learning strategy[5].

In the ecosystem we are implementing, individuals will be generated by asynchronous mating, and time will not be discrete, in form of "generations", as it is common in evolutionary algorithms, but continuous real time. New individuals will enter the world and will survive and reproduce according to their skills, part of which will be inherited, and part developed trough learning. The life span of the individuals will only be determined by their ability to find food, although both ageing mechanisms and grace period for newborn individuals shall be devised. Population size will thus be variable, only limited by the amount of resources available in the world.

The final and major future challenge of this work will be extending the proposed life cycle including evolvable morphologies for the individuals.

Acknowledgements. A significant part of this work has been carried out at Free University of Amsterdam, in collaboration with Prof. A.E. Eiben, whom we would like to thank.

References

1. Eiben, A., Kernbach, S., Haasdijk, E.: Embodied artificial evolution. Evolutionary Intelligence 5, 261–272 (2012)

[5] This term has been coined by Prof. A.E. Eiben.

2. Pérez-Moneo Suárez, D., Rossi, C.: Comparison between different evolutive configurations for learning basic functionalities. In: Proceedings of the 2013 European conference on Applications of Evolutionary Computation, EvoApplications 2013 (2013)
3. Pfeifer, R., Bongard, J.C.: How the Body Shapes the Way We Think. A New View of Intelligence. MIT Press (2007)
4. Ijspeert, A.: Central pattern generators for locomotion control in animals and robots: a review. Preprint of Neural Networks 21(4), 642–653 (2008)
5. Haasdijk, E., Rusu, A.A., Eiben, A.E.: HyperNEAT for locomotion control in modular robots. In: Tempesti, G., Tyrrell, A.M., Miller, J.F. (eds.) ICES 2010. LNCS, vol. 6274, pp. 169–180. Springer, Heidelberg (2010)
6. Clune, J., Beckmann, B.E., Ofria, C., Pennock, R.T.: Evolving coordinated quadruped gaits with the hyperNEAT generative encoding. In: Congress on Evolutionary Comutation (CEC) (May 2009)
7. Sims, K.: Evolving virtual creatures. Annual Conference Series, pp. 15–22 (July 1994)
8. Bongard, J.: Morphological change in machines accelerates the evolution of robust behavior. Proceedings of the National Academy of Sciences 108, 1234–1239 (2011)
9. Becerra, J.A., Bellas, F., Duro, R.J., Lope, J.D.: Snake-like behaviors using macroevolutionary algorithms and modulation based architectures
10. Weel, B., Haasdijk, E., Eiben, A.E.: The emergence of multi-cellular robot organisms through on-line on-board evolution. In: Di Chio, C., Agapitos, A., Cagnoni, S., Cotta, C., de Vega, F.F., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Langdon, W.B., Merelo-Guervós, J.J., Preuss, M., Richter, H., Silva, S., Simões, A., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Togelius, J., Urquhart, N., Uyar, A.Ş., Yannakakis, G.N. (eds.) EvoApplications 2012. LNCS, vol. 7248, pp. 124–134. Springer, Heidelberg (2012)
11. Lal, S., Yamada, K., Endo, S.: Evolving motion control for a modular robot. In: Ellis, R., Allen, T., Petridis, M. (eds.) Applications and Innovations in Intelligent Systems XV, pp. 245–258. Springer, London (2008)
12. Stanley, K., Bryant, B., Miikkulainen, R.: Real-time neuroevolution in the nero video game. IEEE Transactions on Evolutionary Computation 9(6), 653–668 (2005)
13. Eiben, A., Haasdijk, E., Bredeche, N.: Embodied, On-line, On-board Evolution for Autonomous Robotics. In: Levi, S.K.E.P. (ed.) Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution. Cognitive Systems Monographs, vol. 7, pp. 361–382. Springer (2010)
14. Bredeche, N., Haasdijk, E., Eiben, A.: On-line, on-board evolution of robot controllers. In: Collet, P., Monmarché, N., Legrand, P., Schoenauer, M., Lutton, E. (eds.) EA 2009. LNCS, vol. 5975, pp. 110–121. Springer, Heidelberg (2010)
15. Hirose, S., Morishima, A.: Design and control of a mobile robot with an articulated body. Int. J. Robot. Res. 9, 99–114 (1990)
16. Colorado, J., Barrientos, A., Rossi, C., Garzón, M., Galán, M., del Cerro, J.: Efficient locomotion on non-wheeled snake-like robots. In: Filipe, J., Andrade-Cetto, J., Ferrier, J.-L. (eds.) ICINCO (2), pp. 246–251. INSTICC Press (2010)

Multi-robot Operation System with Conflict Resolution

Eduardo Ferrera[1], Angel R. Castaño[2], Jesus Capitán[1], Pedro J. Marrón[1], and Aníbal Ollero[2]

[1] University of Duisburg-Essen,
Bismarckstr. 90, 47057, Duisburg, Germany
[2] University of Seville,
Av. de los Descubrimientos S/N, 41092, Seville, Spain

Abstract. Applications with large teams of robots are becoming more and more useful. If the scenario is very crowded or very dynamic, conflict resolution when using a shared workspace is a challenging problem. In this paper, an scalable, decentralized and reactive approach for collision avoidance is presented. The robots can navigate in a 2D environment avoiding each other and without high computational requirements. In addition to the conflict resolution algorithm, a multi-robot simulator is presented. The system is flexible and can be used to simulate different algorithms with realistic robots. Finally, an extension of the simulator is proposed in order to operate real robots in a multi-robot testbed. Results of the collision avoidance approach are shown with both real and simulated robots.

Keywords: Multi-robot coordination, conflict resolution, testbeds.

1 Introduction

The use of multi-robot teams shows a clear trend in the last decade. Advances in swarm robotics make it possible to use large teams of robots in different applications such air-traffic control, multi-UAV missions, logistics, etc. If the scenarios are highly dynamic or the number of robots is variable, the problem of conflict resolution becomes challenging. In particular, this paper focuses on multi-robot systems for conflict resolution in crowded environments. We consider teams that can vary dynamically during operation, adding new robots and removing others.

Although there are centralized algorithms that can tackle collision avoidance with multiple vehicles [4], they are not usually suitable for dynamic applications, since they take many resources when computing the solution (online recomputation becomes infeasible). This paper opts for a decentralized conflict resolution approach in which robots can reach their goals in a reactive manner, sharing as little information as possible and requiring low computational resources. In particular, robots detect potential collisions with local information and solve them assuming that the others will follow the same rules. Each robot only needs to

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_29,

know the position of its neighbouring robots, but not their velocities, orientations or goals. Therefore, the algorithm scales well with the number of robots and adapts when robots are missing.

This paper includes several contributions. First, the paper proposes a decentralized state machine to control the robots and avoid collisions. In particular, the state machine is designed for robots with differential drive which can rotate *in situ*. The approach is decentralized and scalable because each robot runs its own state machine accessing only local information. Second, a simulation framework with Matlab/Simulink has been developed in order to test multi-robot systems in realistic scenarios. The framework is general and flexible and allows us to deal with different kinds of robots and algorithms but, in this paper, it is illustrated to simulate the proposed conflict resolution approach. This tool is quite relevant in order to reduce the gap between theory and real multi-robot systems, since models of real robots and controllers can be used together with the conflict resolution scheme. Third, a framework to operate multiple robots in a real testbed has been developed. This framework allows users to run collision avoidance experiments with a team of Pioneer robots in a testbed scenario.

Regarding related work, many authors solved collision avoidance in the past by means of mathematical programming or control theory [4,1,8,11]. In particular, [1] provides a thorough review of these kind of approaches. Although these methods can obtain optimal solutions, they are complex and in many cases, do not allow for online re-planning. Navigation functions based on potential field can also be used to solve multi-robot collision avoidance [13,6]. If the navigation functions are decentralized, some approaches [12] propose to use *rules of the road* to assign priorities to the vehicles. Others achieve collision-free trajectories by reasoning with velocity profiles [5,2]. In [14], *Optimal Reciprocal Collision Avoidance* is introduced to select in a decentralized fashion velocities that guarantee collision-free movements. The algorithm is extended in [3] to cope with certain kinds of non-holonomic vehicles. Most approaches based on velocities assume that the velocities of the other robots can be shared or sensed. Finally, the work in this paper is similar to that in [10], but it offers advantages against their *Generalized Roundabout Policy* (GRP). Although the GRP can be applied to any kind of vehicles (it was originally developed for airplanes), it is not efficient for vehicles that can stop and turn at zero forward speed, like differential drive robots, helicopters or quad-rotors. Our method takes advantage of this kind of mobility and reduces the area required by the robots to navigate avoiding each other. Also, we present an implementation in a multi-robot testbed of our approach, showing the algorithm with real robots.

The paper is organized as follows: Section 2 presents the conflict resolution approach; Section 3 details the simulation framework with multiple robots; Section 4 explains the multi-robot system in a real testbed; and Section 5 includes conclusions and future work.

2 Collision Avoidance Approach

2.1 Preliminaries

This section presents a decentralized algorithm to avoid collisions among multiple robots working in a shared environment. The main objective is the following: *Given a team of robots in a 2-D workspace, and given their initial positions and their goal destinations, they should be able to navigate to the final configuration avoiding collisions.* Mathematically, there is a set of robots $\mathcal{N} = \{1, \ldots, n(t)\}$ with positions $\{p_i(t)|p_i(t) \in \mathbb{R}^2\}_{i=1,\ldots,n}$ and headings $\{\theta_i(t)|\theta_i(t) \in [0, 2\pi]\}_{i=1,\ldots,n}$. Besides, there is an initial configuration $\mathcal{I} = \{p_1(0), \ldots, p_n(0)\}$, a goal configuration $\mathcal{G} = \{g_1, \ldots, g_n\}$, and a set of static obstacles $\mathcal{S} = \{1, \ldots, m(t)\}$. Each robot has an omnidirectional perception system that allows it to detect obstacles (static or moving) in a circle of radius R_s. Note that the number of robots $n(t)$ can vary throughout time, since new robots could appear or some robots could stop working and become static obstacles. The algorithm makes the following assumptions:

1. Robots are able to localize themselves globally in the scenario, so each robot i has its position $p_i(t)$ available.
2. Robots have differential drive, so they can rotate *in situ* with zero linear velocity.
3. Obstacles around each robot can be sensed and also positioned. The sensing radius is R_s for each robot.
4. Each robot has access to the positions of its neighbouring robots (those within its sensing radius R_s). These positions could be transmitted or sensed with the local perception system.

In order to operate, each robot needs to define two regions around itself: a *Safety Disk* and a *Reserved Disk*.

Definition 1. *The Safety Disk is a circle of radius R_{sd} around each robot that determines the area in which that robot could stop safely (assuming maximum velocity) without hitting a hypothetical obstacle. For obstacles beyond this circle, the robot can assure a safe stop before colliding. R_{sd} can be defined depending on the robot dynamics and size. Therefore, a collision is considered to take place whenever two or more safety disks overlap.*

Definition 2. *The Reserved Disk is a circle of radius R_{rd} around each robot such that $R_{rd} > R_{sd}$, and it is used to detect conflicts. A conflict is considered to take place whenever two or more reserved disks overlap. Moreover, we define the conflictive neighbours of a robot as the subset of neighbours that are in conflict with it.*

It is important to note that R_s must be greater than $2R_{rd}$ in order to be able to detect all possible conflicts in the system. Otherwise, obstacles in conflict may be out of the sensing range. The idea of using a safety disk to define a collision allows the robots to operate safely even under uncertainties in their positions. Moreover, artificial safety and reserved zones can be defined for all the static obstacles, so that moving robots and other obstacles are treated seamlessly.

2.2 State Machine

This section describes the decentralized state machine that defines the behaviour of the robots. Each robot runs the state machine locally and tries to reach its goal reactively, without having information about others' goals, orientations or velocities. In order to detect and solve conflicts, only obstacles in the neighbourhood are considered. In the following, the behaviours for the different modes of the state machine (see Fig. 1) are described.

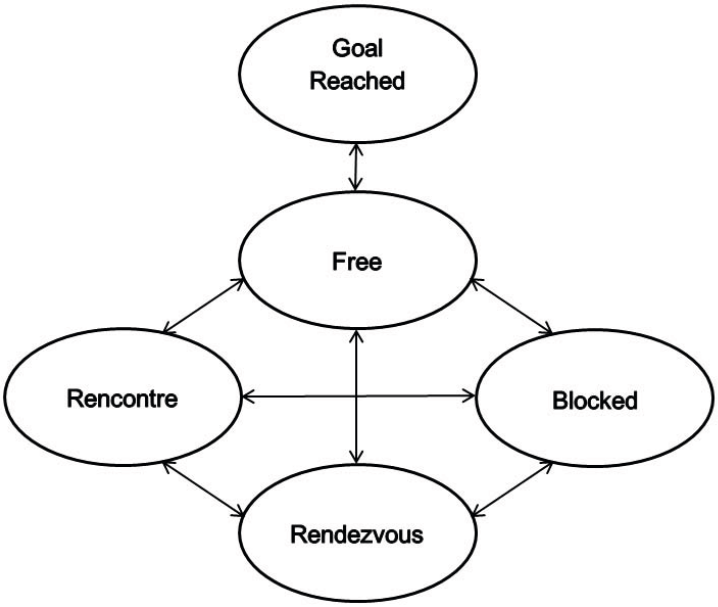


Fig. 1. The robot behaviour is defined by a state machine with five states

`Goal-reached.` When the robot gets to its destination point, the state switches to `Goal-reached`. In this state, the robot stops and waits until a new goal is assigned. Moreover, since the positioning systems of the robots have always some error, a distance threshold d_{th} is defined: a robot is considered to be at its goal if its distance to it is lower than d_{th}. Otherwise, robots may not converge to the destination due to minor inaccuracies. This threshold is usually lower than R_{sd} and its value depends on the accuracy of the positioning system of the robot.

`Rencontre.` In this state[1], the robot detects one or more conflicts that need to be solved, so it stops immediately its movement in order to prevent possible

[1] *Rencontre* is a word derived from French that means meeting someone unexpectedly.

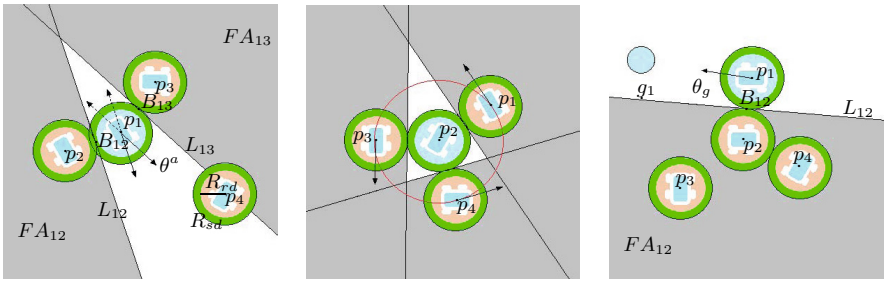


Fig. 2. Different conflictive situations. The safety and reserved disks are drawn around the robots. (a) `Rencontre` situation for robot 1, which has conflicts with robots 2 and 3. Its navigable area is shown in white. (b) Robot in the middle is in `Blocked` state. The other robots will surround it counter-clockwise until it is released. (c) Robot 1 is in `Free` state. Despite the conflicts, the robot can still drive towards the goal without collisions with heading θ_g.

collisions. Then, an *avoidance* heading is necessary in order to surround the obstacles. For that, all conflictive obstacles around the robot must be analysed.

For a given robot i and conflictive obstacle j, the space is split into two semi-planes defined by a separator line L_{ij}. If B_{ij} is the closest point to p_i of the intersection of the two reserved disks, L_{ij} is the line that contains B_{ij} and is perpendicular to the vector $\overrightarrow{p_i B_{ij}}$. The semi-plane defined by L_{ij} that contains the obstacle j is the *forbidden* area FA_{ij} for the robot, while the other is the *navigable* area NA_{ij}. After processing all the obstacles causing conflicts, the total forbidden area of the robot i is defined by the union of all the individual forbidden areas $FA_i = \bigcup_{j \in \mathcal{C}_i} FA_{ij}$. The remaining area is the total navigable area, since the robot can move in it without provoking collisions. This partition of the workspace is depicted in Fig. 2a. Note that only positions of the obstacles in the robot local coordinate frame are needed for the computation, but not their velocities nor orientations.

If the navigable area is an open space (infinite area), the robot can find an *avoidance* direction that allows it to surround the current obstacles preventing collisions. In particular, for each line L_{ij} two possible avoidance directions can be proposed, those parallel to the line. The robot only considers avoidance directions that drive it into the navigable area, and selects the one that will allow it to surround the associated obstacle counter-clockwise. For instance, in Fig. 2a each obstacle for robot 1 defines an avoidance direction moving the robot into the navigable area (vectors with solid lines) and one moving it into the forbidden area (vectors with dashed lines). In this case, θ^a is selected as avoidance heading because it also allows robot 1 to surround robot 3 counter-clockwise.

In the `Rencontre` state, the robot performs an *in-situ* turn in order to orientate towards its selected avoidance direction θ^a without moving into the forbidden area. Once it is oriented correctly, the avoidance manoeuvre is safe and the robot switches to the `Rendezvous` state.

`Rendezvous.` If the robot is performing an avoidance manoeuvre with any conflictive obstacle, it is in this state[2]. This means that a conflict has occurred but the navigable area is open, which allows the robot to look for an avoidance heading. Indeed, after the `Rencontre` state, the robot is oriented to an avoidance heading, so it can start the `Rendezvous` operation safely. The robot will move forward keeping the heading θ^a, which will vary in real time as the navigable area does.

`Blocked.` In case that the conflicts produce a navigable area that is a closed polygon (finite area), the robot is `Blocked` and surrounded by obstacles (see Fig. 2b). In this state, the robot stops and waits until any of the moving obstacles is gone. It is important to note that each robot assumes that the others will follow the same rules, so they will move away eventually if there is free space in the scenario.

`Free.` This is the normal operation mode, where the robot can go freely towards its goal. This situation will only happen if there are no conflicts or there are conflicts but the goal is still in the navigable area. Therefore, the robot will be commanded to go to the goal in a straight line without avoiding obstacles (see Fig. 2c). If the robot reaches its goal, it switches to the `Goal-reached` state, but if the goal gets out of the navigable area an obstacle avoidance will be required.

2.3 Discussion

All the static obstacles in the environment can be surrounded by a safety and a reserved zone as it was explained before. Therefore, collisions will always be avoided. Regarding the moving obstacles, in addition to the other robots in the scenario, there may be external agents that are not controlled by the system, such as other vehicles and people. In this case, the algorithm is no longer guaranteed to find the solution. The reason is that those external obstacles do not behave according to the system rules, so they may trap robots and lead them to deadlocks. However, robots stop whenever they detect collisions, so a safe navigation is assured, at least from our system's side.

Since we are considering robots that could fail and stop anywhere, there may be situations in which they occupy the goals of some other robots or block others' paths. To solve these situations, the inclusion of a higher level planner will be considered for future works. Deadlocks could be detected and solved by sending new destinations to the robots involved.

So far, only homogeneous robots with similar reserved and safety disks have been considered. However, the approach would work even in the case of heterogeneous robots. In that case, different robots would have different safety and

[2] *Rendezvous* is a word derived from French that means meeting someone at a specified time and place.

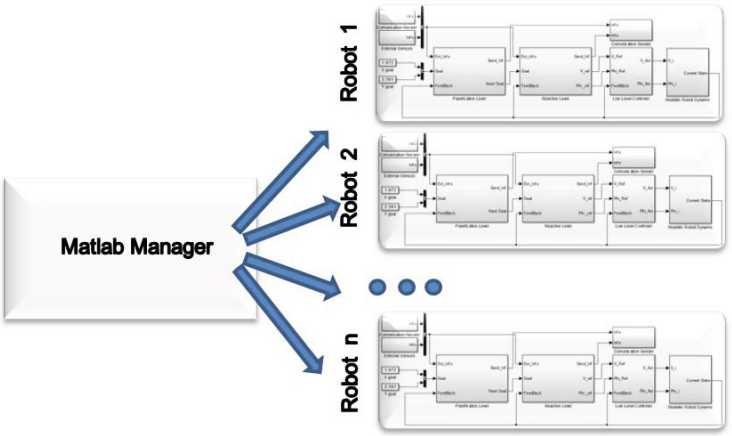


Fig. 3. Block diagram of the multi-robot simulator. A Matlab Manager is in charge of managing the individual Simulink blocks for each robot.

reserved disks, each of them according to their characteristics. Besides, neighbouring robots would also have to communicate that information, since others would need it to discover and solve possible conflicts.

3 Multi-robot Simulator

In order to test the capabilities of the system for different scenarios, a multi-robot simulator has been developed with Matlab/Simulink. The simulator is generic, so the algorithm proposed in this paper as well as alternative approaches can be tested easily. For the design, the following specifications were taken into account:

1. The simulator should consider large teams of robots coexisting in the same workspace.
2. All the simulated robots should have realistic dynamics.
3. The system should be flexible, allowing the user to configure new dynamic models or different control schemes.
4. The simulator should provide a friendly human-machine interface.

In the architecture, a Matlab Manager plays the relevant role of controlling the coordination among individual simulations for each robot (see Fig. 3), which are run in parallel. For a multi-robot experiment, the simulation is divided into time steps of T_{step} seconds and each robot is simulated individually with a Simulink model that is initialized with the last saved state. The simulator also implements a communication system that allows robots to share messages. In particular, a maximum delay of T_{step} in the reception of the messages is considered.

The Simulink model used to model each robot (see Fig. 4) adds some flexibility to the system, since it is prepared to include different dynamics. The only

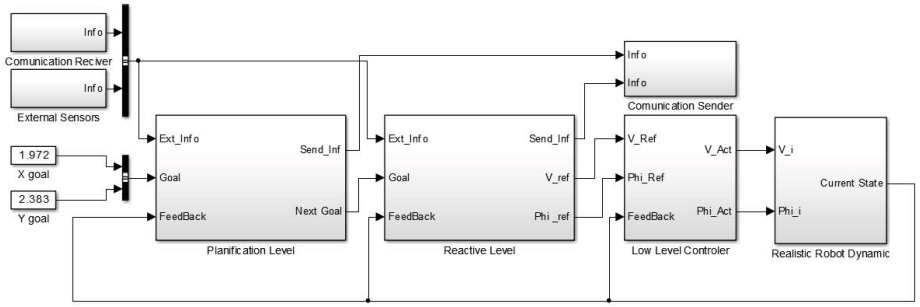


Fig. 4. Simulink block diagram of one of the simulated robots. Dynamics and low-level controllers for the motors are modelled, as well as the reactive state machine.

restriction is that the model should include the possibility of being initialized in a point p with an orientation θ and a speed v at any time. The architecture allows the user to change the whole configuration of the robot, giving to the system enough flexibility to emulate all kind of ground robots and control systems. Moreover, the Matlab Manager is in charge of allocating memory for all the Simulink modules and initializing them. The resulting state for each time step is saved in a logging file so that the experiment can be replayed later.

A friendly user interface was implemented to help the user to design fully controlled scenarios, challenging scenarios or randomly crowded scenarios. The user is also allowed to define a queue of simulations that can be combined and executed sequentially in a predefined or random order. This feature is quite useful for experiments with many trials, such as Monte-Carlo simulations.

In order to illustrate the simulator, an experiment evaluating the performance of the conflict resolution approach presented in this paper is shown. In particular, multiple simulations were run ($T_{step} = 0.1s$) with initial configurations drawn from a uniform distribution in a scenario of size $14 \times 14m$ (only initial configurations without conflicts were considered). The parameters of the algorithm for this experiment were $R_s = 5m$, $R_{sd} = 0.4m$, $R_{rd} = 0.6m$ and $d_{th} = 0.25m$.

The results of the simulations in terms of travelled distance are depicted in Fig. 5, where teams with different number of robots are considered. For each team 200 samples of distances are taken. All distances are normalized with respect to the unconstrained case, in which each robot can travel alone in the workspace straight to its goal. If the scenario is not very crowded, the travelled distances are close to the unconstrained case. When the number of robot increases the distances also do, but a nice degradation can be observed.

Finally, it is important to note that many other simulations were run with the system, testing complex initial configurations and crowded environments. They are not shown in the paper due to space constraints. Indeed, in a stress test for the simulator, it was able to run successfully simulations of up to 100 robots coexisting in a crowded workspace.

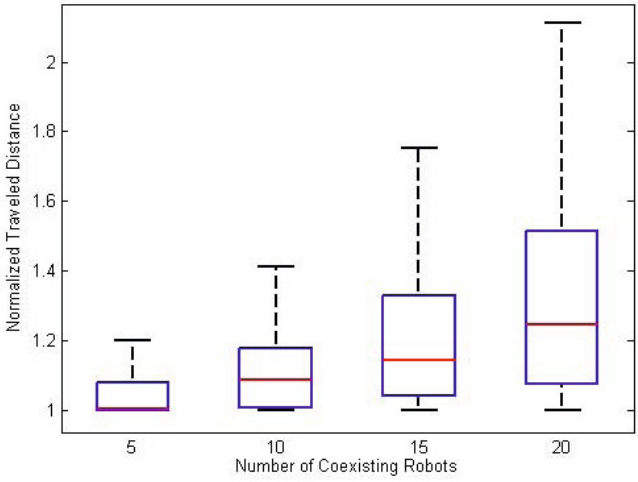


Fig. 5. Normalized travelled distances of simulations in a workspace of $14 \times 14m$, varying the number of robots. Each box plot represents the median (red line), and the first and third quartiles (blue box) for 200 simulations.

4 Multi-robot Testbed

In addition to the simulations, we have performed a number of real experiments using a testbed with multiple robots. For this purpose, an extension to the simulation tools has been developed in order to use the user interface of the simulator with the multi-robot integrated testbed of CONET[3] [9]. The CONET testbed is installed at the University of Seville (Spain) and was designed to test and validate collaborative algorithms with different levels of decentralization, including fully distributed and centralized systems. The testbed also allows users to operate experiments remotely by the Internet.

The testbed is set in a room of more than $500m^2$ ($22 \times 24m$) with three columns in the middle and an IP camera that provides remote users a general view. There are 5 skid-steered holonomic robots (Pioneer 3-AT). The robots are equipped with several sensors including an Hokuyo 2D laser and one Microsoft Kinect RGB-D sensor. A Netbook PC with an Intel Atom processor, 1,024 MB SDRAM and a IEEE 802.11 a/b/g/n Wireless bridge provides enough computational capacity at each robot to perform experiments.

The software architecture of the testbed is based on the open-source middleware Player [7]. The Matlab Manager in Section 3 is extended in order to interact with the testbed. In particular, a VPN direct connection is established with Player in order to command the Pioneers 3-AT. The same user

[3] Cooperating Objects Network of Excellence (INFSO-ICT-224053). Available online: `http://www.cooperating-objects.eu/`

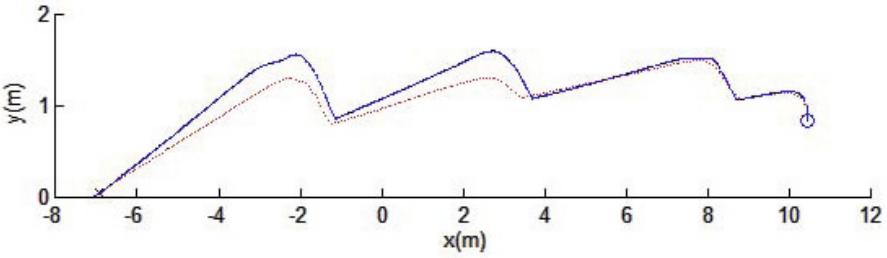


Fig. 6. In blue: Path of a Pioneer 3-AT using the proposed algorithm to avoid three static obstacles. In red: Path of a simulated robot initialized with the same conditions in the simulator. The circle is the initial position and the cross the goal.

interface as before can be used here to define the scenario and the experiments. Then, the Manager is in charge of managing the Player connections with the robots, initialize them and run the experiments autonomously. Moreover, after each experiments data are logged and the robots are sent to their idle positions. It is important to note that this architecture allows the user to perform real experiments in the testbed remotely, without the need for physical presence.

A test of the reactive conflict resolution algorithm presented in this paper was performed with a real robot in the testbed.[4] The experiment consists of a robot that has to cross the room avoiding the three columns in the middle, that are static obstacles. The same experiment was simulated with the system in Section 3 to compare the results, which are shown in Fig. 6. In both cases, the robot reaches the goal avoiding the obstacles, and it can be seen that the behaviour of the simulated robot is quite similar to the real one. The parameters of the algorithm for this experiment were $R_s = 5m$, $R_{sd} = 0.4m$, $R_{rd} = 0.6m$ and $d_{th} = 0.25m$.

A second test using five real robots was also performed. The experiment consists of five robots placed forming a circle in the free part of the room. The goal of each robot is placed at its antipodal position, enforcing a crowded confrontation in the center of the circle. The results of this experiment are shown in Fig. 7. It can be seen how each robot reaches its goal avoiding any collision with the others. This experiment uses the same parameters as above except for R_{rd}, that was increased to $R_{rd} = 0.7m$ in order to make the avoidance more conservative.

[4] In all the experiments, the robots were localized using a map of the testbed and the Hokuyo readings. Also, robots shared their positions to each other.

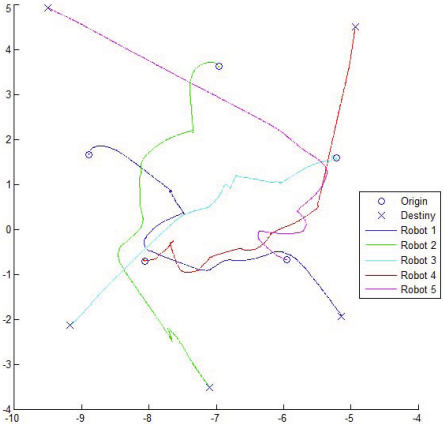
(a) Starting positions $t = 0s$.

(b) Going to the conflictive point $t = 4s$.

(c) Avoiding the confrontation $t = 29s$.

(d) Ending the confrontation $t = 47s$.

(e) Final positions $t = 59s$.



(f) Paths of the robots.

Fig. 7. Five Pioneer 3-AT using the proposed algorithm to solve a conflictive situation

5 Conclusions

This paper has presented a multi-robot system for conflict resolution. A decentralized and reactive algorithm for collision avoidance in crowded scenarios is proposed. The computational requirements for the robots are low, so the approach can be run in small robots in dynamic environments that require online recomputation. Besides, the algorithm is decentralized and scalable with the number of robots in the team. Although the paper focuses on robots with differential drive, future work will extend the method to Ackermann vehicles, including forward and backward manoeuvres.

A multi-robot simulator based on Matlab/Simulink is also presented. The idea is to exploit its flexibility in order to test different kinds of robots and controllers in a realistic manner. To complete the system, the simulator architecture is extended to be able to test algorithms in a real testbed. Experiments with real robots and simulations are shown to illustrate the performance of the collision avoidance method.

Future work will consider the adaptation of the multi-robot system to ROS[5], in order to be able to interact with other testbeds and make the system more portable. Moreover, using distributed computers to run the simulated robots in parallel can be of interest.

Acknowledgements. This work was partially supported by the DECOV project (P10-TIC-6220), the European Center for Ubiquitous Technologies and Smart Cities (UBICITEC) and the FP7 EC-SAFEMOBIL Project [288082].

References

1. Abichandani, P., Ford, G., Benson, H.Y., Kam, M.: Mathematical programming for multi-vehicle motion planning problems. In: IEEE International Conference on Robotics and Automation, pp. 3315–3322 (2012)
2. Alejo, D., Díaz-Báñez, J., Cobano, J., Pérez-Lantero, P., Ollero, A.: The velocity assignment problem for conflict resolution with multiple aerial vehicles sharing airspace. Journal of Intelligent & Robotic Systems 69(1-4), 331–346 (2013)
3. Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., Siegwart, R.: Optimal reciprocal collision avoidance for multiple non-holonomic robots. In: Martinoli, A., Mondada, F., Correll, N., Mermoud, G., Egerstedt, M., Hsieh, M.A., Parker, L.E., Sty, K. (eds.) Distributed Autonomous Robotic Systems. Springer Tracts in Advanced Robotics, vol. 83, pp. 203–216 (2013)
4. van den Berg, J., Snoeyink, J., Lin, M., Manocha, D.: Centralized path planning for multiple robots: Optimal decoupling into sequential plans 2 (2009)
5. Claes, D., Hennes, D., Tuyls, K., Meeussen, W.: Collision avoidance under bounded localization uncertainty. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1192–1198 (2012)
6. Dimarogonas, D.V., Loizou, S.G., Kyriakopoulos, K.J., Zavlanos, M.M.: A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. Automatica 42(2), 229–243 (2006)

[5] `http://www.ros.org`

7. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th International Conference on Advanced Robotics, pp. 317–323 (2003)
8. Hoy, M., Matveev, A.S., Savkin, A.V.: Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments. Robotics and Autonomous Systems 60(10), 1253–1266 (2012)
9. Jimenez-Gonzalez, A., Martinez-de Dios, J.R., Ollero, A.: An integrated testbed for cooperative perception with heterogeneous mobile and static sensors. Sensors 11(12), 11516–11543 (2011), http://www.mdpi.com/1424-8220/11/12/11516
10. Pallottino, L., Scordio, V., Bicchi, A., Frazzoli, E.: Decentralized cooperative policy for conflict resolution in multivehicle systems. IEEE Transactions on Robotics 23(6), 1170–1183 (2007)
11. Peng, J., Akella, S.: Coordinating multiple robots with kinodynamic constraints along specified paths. The International Journal of Robotics Research 24(4), 295–310 (2005)
12. Roussos, G., Kyriakopoulos, K.J.: Decentralized and prioritized navigation and collision avoidance for multiple mobile robots. In: Martinoli, A., Mondada, F., Correll, N., Mermoud, G., Egerstedt, M., Hsieh, M.A., Parker, L.E., Sty, K. (eds.) Distributed Autonomous Robotic Systems. Springer Tracts in Advanced Robotics, vol. 83, pp. 189–202 (2013)
13. Tanner, H.G., Kumar, A.: Formation stabilization of multiple agents using decentralized navigation functions. In: Robotics: Science and Systems, pp. 49–56 (2005)
14. Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Robotics Research, pp. 3–19. Springer (2011)

Path Planning in Service Robotics Considering Interaction Based on Augmented Reality

Francisco J. Rodríguez Lera*, Julián Orfo, Juan Felipe García Sierra, and Vicente Matellán

School of Industrial Engineering and Information Technology, University of León, Spain
{fjrodl,jorfop00,jfgars,vicente.matellan}@unileon.es
http://robotica.unileon.es

Abstract. This research presents how augmented reality (AR) can be used for assistance tasks in indoor environments. The approach of AR supported human-robot interaction arises two classical questions in robot navigation: What does the robot need to do to show the AR? (task planning) and, which are the best movements to reach the destination allowing the user to get the AR information? (path planning). In addition, a constraint related to assistance environments has to be considered: How can the robot perform its tasks in order to improve the interaction?. These problems along with how AR can enhance human-robot interaction, and therefore improve the human experience with the robot, are addressed along the paper. The first experiments developed with this system are performed using the gazebo simulator. The solutions proposed are analized using a low cost platform for elderly assistance, which already uses AR for some assistance tasks, developed in Universidad de Len.

Keywords: Navegacin, path planning, augmented reality, AR, Human-Robot Interaction.

1 Introduction

Human-robot interaction (HRI) is a branch of robotics which has been working for years to improve dialog between humans and robots. Based on some motivational features from human-human interaction (HHI) [1], HRI tries to improve human robot dialog: proxemics, movements and body position, postures, gaze or engagement, and gestures are some of the features in HHI defined by Hall.

Since, in some situations, robot navigation can be adapted to improve dialog as part of interaction with humans, this research focuses in movements and body position: a robot with a face should be looking at his human listener and an arm-equipped robot should be able to point objects. Since our robot interacts with

* This work has been partially funded by Telefónica SA through grant CATULE12-3 and by ESF RNP "CompCog" (www.compcog.org) (06-RNP-020).

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_30,

humans using augmented reality (AR), a valid display position has to be offered to the user.

Human gaze and proxemics have been taken into account for this work. Gaze has been proved to be useful for HRI in many works such as [3] or [2]. In Yoshikawa [5] works, human gaze control is utilized by the robot to improve human response; another example is the application of human gaze to improve robot gesture in ELIAS platform[4].

Proxemics is a HHI field which focuses in the space between individuals in the moment of interaction [6]. It can be considered as the personal space of the individual, defined by Hall in [7], or the individual territoriality [1]. This feature is relevant because the robot must remain inside a secure space while showing its display to the human.

In a regular HRI situation, dialog can be adapted for either explicit or implicit communication, as defined by Breazeal [11]: spoken dialogs (conversations, casual talk) or pointing tasks (Kanda [8]) can be defined as explicit communication. A robot taking decisions while paying attention to human gaze is an example of implicit communication.

This research introduces a method to improve human robot interaction using explicit communication models, specifically augmented reality. A system to define navigation paths has to be developed: it has to offer a safe path while the robot interacts with humans using AR and also maintain the human territoriality during the dialog.

During last years many researchers have been working in human robot dialog and robot pose to improve the interaction between humans and robots. For instance, Jorge Rios-Martinez presents a path planning system using an algorithm called Risk-RTT [9]; the system focuses path planning estimation in a proxemics safe way and avoids human disturbance or collisions problems following three rules proposed in Lam work [10]: collision free rule, interference free rule, and waiting rule.

If the tasks that the robot and the human can do together are considered, Krus[13] presents a way of path planning calculation focused in human requirements and improving the human aware navigation planner; they modified the HANP algorithm, oriented to build paths in environments, to have more human awareness functionality.

The basis for this work, like in others related researches, is: initiation, attention and position. The main differences with other research can be enumerate as:

- The platform. We have no physical embodiment or personification in the robot.
- The initiation point it is not casual, the robot is not searching the individual, and the robot only has a pre-defined cases of assistance.
- The attention. The view points of the robot and the view points of the individual are different and has to be defined working with the clasical deixis problem ([12]).

– The position and navigation. In our proposal path planning design has to be performed according to augmented reality. The navigation takes place in a defined position so that the human can reach the display, which means this position is related to robot morphology.

AR is beneficial for HRI in several aspects, for instance, an arm is not needed to point to things, regions, or interesting points, since a display can be used instead, avoiding the problems of deixis since a virtual signal is being overlapped over the goal and the user has the exact points that he needs to know. Also, extra virtual information can be offered to the user to ease task performance; for instance, environmental data or a map can be presented to the user for a task consisting of arriving at a given location where and object is located at.

This paper is organized as follows: section 2 defines the hardware and control software features of the plaform used for this research. Section 3 defines some traditional navigation strategies in general HRI situations. In section 4, interaction improving strategies available for robots using AR are defined. An experiment with gazebo is presented in section 5, and conclusions are given in section 6.

2 Overall Approach

In this section we define how does the robot work from two points of view: the hardware and the control system. From the hardware side, the whole platform is described. In regard to the control system, arquitecture deployed in the robot to control the system is analyzed.

2.1 Robot Configuration

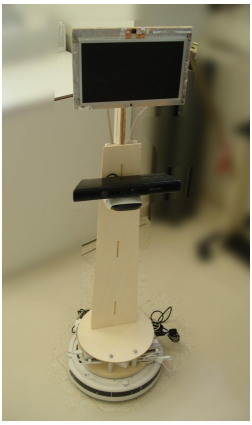
All the reasearch has been made around a platform based on iRobot base solution. Two versions of this platform are used, iCreate and Roomba, the iCreate version is used in turtlebot platform and the Roomba base is used with our platform MYRABot. The former differs from the latter in the parallel port available to plug more stuff.

These platforms have two independently powered wheels working in a differential way, which means that each wheel can run independently of the other. The two wheels are non steered and are situated in opposite position of the platform. In addition, the robot has a caster wheel.

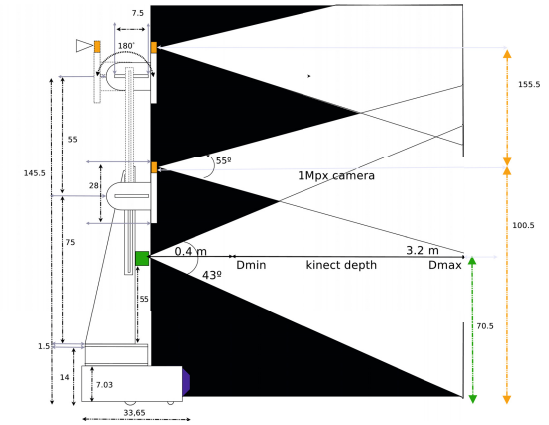
It is also worth noting that the robot is supposed to always move on a regular non-slippery floor without unexpected objects to simplify the experiments.

Figure 1 shows the theoretical robot configuration for this research. The components of our robot can be enumerated as:

1. Kinect Camera: The one in the front part of the platform is used.
2. iRobot Base: Represents the base that used for the research.
3. Display and webcam: the laptop display and camera in the top of the robot are used.



a) MYRABot platform



b) MYRABot morphology

Fig. 1. Robot Configuration

The location of these components is a limitation to our navigation model using HRI. In figure 1, a robot configuration with its display located at the robot front is presented. According to the display position, navigation has to be carried out in one way or another as it is explained in section 4.

All system are controlled using ROS.

2.2 Control Configuration

The robot control architecture is based on a hybrid tree layer architecture. Its high level representation is presented in figure 2.

The low level part represents the robot configuration defined previously. This control architecture defines four main components:

1. The Behavioral Subsystem: It is in charge of reactive behaviours and performs actions defined by the deliberative layer

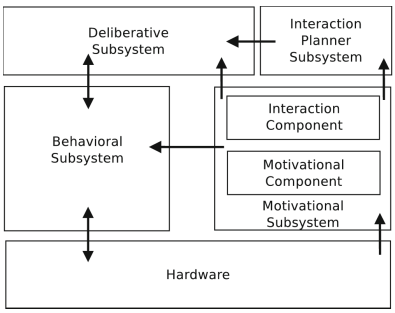


Fig. 2. Robot Control Architecture

2. The Deliberative Subsystem; It defines long term actives in the system.
3. The Motivational Subsystem: It has two components, one oriented to robot motivation and another oriented to HRI motivation.
4. The Interaction Planner Subsystem: It defines the basic controllers to manipulate the deliberative behaviours in a more effective interaction-oriented way.

3 Navigation Strategies

The navigation strategies are always restricted by robot morphology, sensors and actuators on the one hand and the environment on the other hand. Our robot restrictions are: the steering base, the sensors disposal and the display position.

In the first place, a robot navigation example is defined in order to better understand how to apply models of navigation with augmented reality. Then two proposal to improve the AR supported user experience are given.

3.1 Basic Navigation in Known Environment

This type of navigation takes place in a well known region, for instance, navigating using a map. The robot is placed in a given start position and then asked to reach a goal. The robot then makes different path planning and chooses the best path to try to reach the goal.

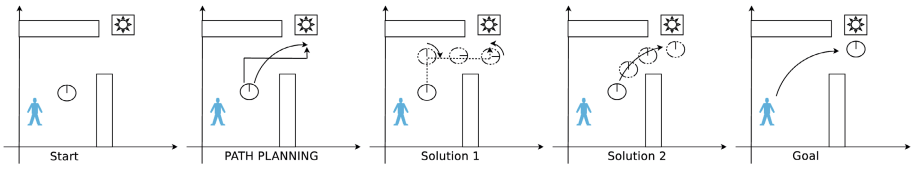


Fig. 3. Clasical navigation system

In figure 3, the classical navigation in well known environment can be defined in these five points, 1) star point, the user demands a goal, 2) path planning, the robot calculates the best paths to reach the goal. 3,4) the robot chooses the best of path to arrive at destination and goes there, 5) the robot is in the goal point and the user follows the robot.

In this research, we carried out this experiment using the turtlebot in the gazebo simulator, ROS and RVIZ. The map it is calculated a priori to be used in the test. The localization system used in our system is the adaptive Monte Carlo localization method proposed by Dieter Fox. The navigation planner used implements the Trajectory Rollout and Dynamic Window approaches to local robot navigation on a plane. Both are ROS packages defined in the navigation stack.

In picture 4, the proposed routes to a point near the table is depicted. In both cases the yellow line identifies the total route and the green line that overlaids the yellow one defines the local cost map to reach that point.

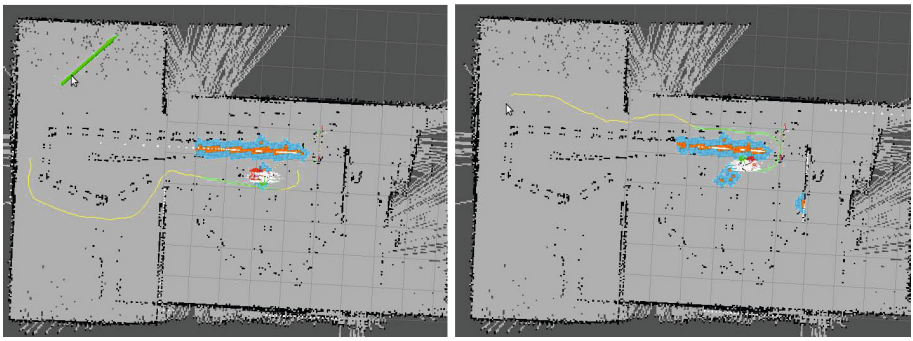
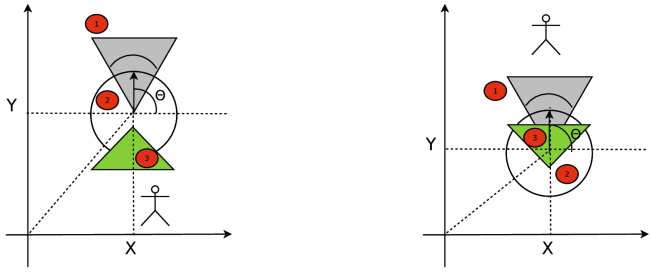


Fig. 4. Path planning

4 HRI Oriented Navigation Strategies Using AR

When an augmented reality system is to be used to improve the user experience, the robot's display location has to be taken into account. The reason for this is that the strategy to follow will be different according to each case.



a) The display is oriented backwards b) The display is oriented forward

Fig. 5. Display configurations in the platform

Two different cases are defined according to our robot morphology (see figure 5). The numbers identify the kinect sensor, the iRobot Base and the display plus webcam respectively. Picture 5 shows the display oriented to the back part of the robot. Picture 5 shows the display oriented to the front part of the robot. The human is represented in each picture with the needed position to get AR marks. These two scenarios result in two main strategies:

1. Back strategy
2. Frontal strategy

These strategies have to give an answer to the two questions defined at the beginning of the research: task planning and path planning.

4.1 Task Definition

The human needs something that is located somewhere in the house, and the robot knows the position of that object: it knows where the object is located in the map and it has to reach this position to advice the human. According to the chosen way (classical or with AR) the task plan differs:

The classical task plan consists of: a) the robot starts, b) the human asks about the object, c) the robot makes the path to the object, d) the robot starts the navigation e) the human follows the robot (optional) f) the robot reaches the object and g) notices the human.

The task with AR is defined as: a) the robot starts, b) the human defines the task, c) the robot shows a message in the display c) the robot makes the path to the object, d) the robot starts the navigation e) the robot shows virtual arrows in the display with the route to the goal f) the robot has to control the human position during navigation g) the robot reaches the object h) the robot chooses the best place to notice the human with AR. The main problems to take into account related with this task are:

- Problem oriented to proxemics (figure 6): task f) the robot has to control the human position during navigation. A PID has to be used to manage the human position while he is following the robot. The proxemics problem and the human territoriality space have to be managed this way to offer a safe AR solution.

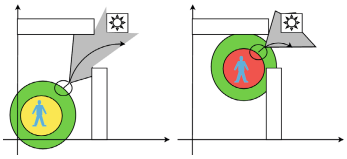


Fig. 6. Generic example of proxemics problem

- Problem oriented to deixis (figure 7): task h) the robot chooses the best place to notice the human with AR .
  A problem arises when the robot reaches the goal: the human has to be informed about the location of the object on the table, but there is no arm mounted on the robot to point to the object. Given that the deictic problem of "the object is on the table" or "the object is here" needs to be avoided since not all people has the same point of view of the scene, the augmented reality system has been added: when something has to be pointed to, a virtual marking signaling the goal is shown in the screen of the robot.

4.2 Back Strategy

This section explains the "Back Strategy". In this case the display and the webcam are located in the back position of the robot and the kinect is pointing to the front part.

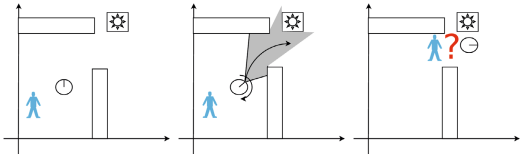


Fig. 7. Which is the best place to show the AR mark?

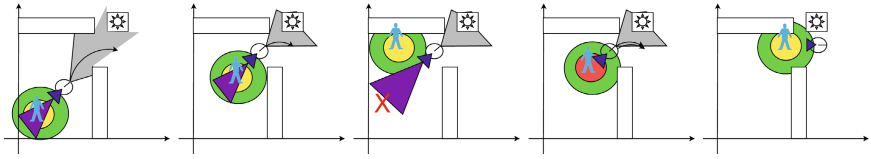


Fig. 8. Back Strategy navigation

For the proxemics problem the system has to use the webcam installed in the display. The control system has to take into account whether the human is too near or too far to regulate the robot velocity. If the robot get personal distance (less than 1.2 m), then an action like increase velocity, stop the platform or change the path plan has to be done.

The figure 8 presents this situation, 1) the robot recognized the individual and starts the navigation, 2) the robot continues the navigation 3) the human disappears, the robot stops or not depending of the interaction configuration, 4) the robot is in the private zone, the robot needs to increase velocity, 5) the robot reaches the goal.

The next strategy is oriented to the final position that the robot has to reach to be able to show the AR to the user. When the robot estimates the final position to be reachd, it takes into account the use of AR or not, depending of motivational and wether it is activate or not. The solution is fully human oriented and the location of the human in the scene has to be figured out.

The figure 9 shows the possible solutions presented in this research; two modes of AR presentation have been considered:

1. Mode 1: The robot is staring the object. The object is presented in the screen. The robot occludes the object.
2. Mode 2: The object is presented in the screen and it can be seen in front of the human.

The problem with mode 1 is that the kinect is being used and its height is is not enough to reach some tables. A solution could be taking a picture of the table during the path navigation and then show it to the human. The benefits to choose the mode 2 is that the webcam sensor can be used to get images from the table and show the AR in the display.

Therefore when the goal point is reached there are several possible solutions, the system only has to avoid the possible obstacles. An example can be seen in figure 10.

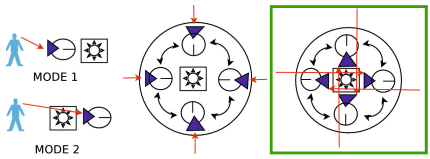


Fig. 9. Solutions available for BS

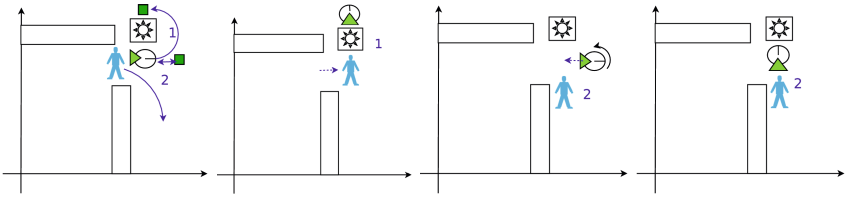


Fig. 10. Two possible solutions taking into account the AR and the environment

4.3 Frontal Strategy

For this strategy the display and the webcam are located in the front position; the kinect is also positioned in the front. Figure 11 shows the possibilities. Once again, if the robot sees the individual while navigating, the navigation plan continues or not according to the overall system; for instance, the system can be forced to wait until the human is in front of the robot.

The personal space with the human has to be watch over (so that the robot doesn't get nearer than 1.2 m). If the robot is too near the human, its velocity is increased to stay outside the personal zone. Again the deixis problem is solved in the same way it was with back strategy. The general modes are presented in figure 12, with the favorite mode to use being mode 2.

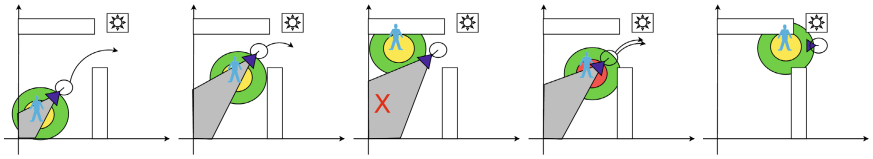


Fig. 11. Frontal Strategy navigation

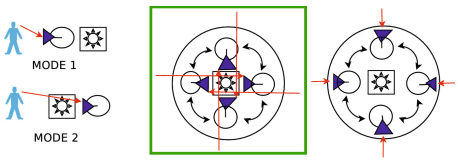


Fig. 12. Solutions available for Frontal Strategy

The big problem of this solution is the reactivity. The robot doesn't have any sensor in its back so the ability to reacts in a dynamic environment is not available. For this reason, this method should be revised with a different robot morphology to improve the backwards navigation.

5 Experiments

The experiments where developed using gazebo simulator, ROS and the turtlebot plataform assuming that the computer is open on top of the platform showing the display to its back. The experience constraints are:

- Task goal: the robot has to reach an object that is on the table. The robot has to navigate for the environment guiding the human until it reaches the object and alert the user about the object.
- Strategy: The strategy defined for this experiment is the Back Strategy.
- Description: The robot make the necessary signals to advise the user during the travel to reach the object. The human should be following it in safe conditions.
- Restriction: We haven't got human in this experiment. We are showing how the deictic problem is solved.

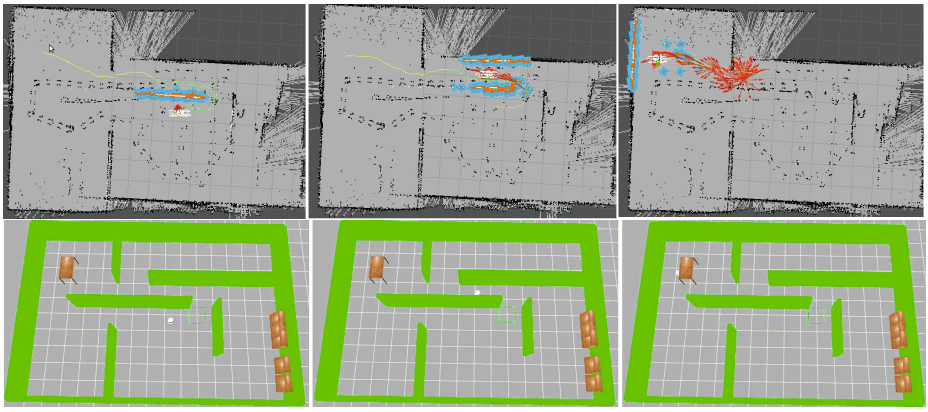


Fig. 13. Simple scenario simulator navigation

The left picture in figure 13 shows the initial path planning to reach the goal, the middle picture shows the robot route and the pose during its travel and finally a correction is applied to improve the experience and the position behind the table is reached. Considering the robot used and its defined morphology, the human should be seeing the display at the same height as the table.

A simple test with an actor was also performed in the simulator. These were problems that we found:

- The first one was the final position in the environment. Sometimes, the robot reaches approximately the final position, it sees the human and do not see the table, so the estimation agree with the goal of our algorithm.
- The second problem is related to proxemics, undefined human movements and recognition problems. If the human changes the velocity and the camera looses the human, the human can collide with the robot.

6 Conclusions

A novel approach to improve the explicit HRI dialog using augmented reality have been presented. Due to the robot morphology, the restrictions to tasks aiming to improve the dialog using AR are increased and path planning is modeled to offers the best point of view to human.

Two strategies were presented according to the robot restrictions: "Back Strategy" and "'Frontal Strategy". In both strategies proxemics and deixis problems are considered.

Navigation control to get a goal according the distance to the human and reaching a right position near the goal to show the augmented reality effectively is one of this research's contributions. The second contribution consists in the application of the augmented reality as a tool to mark objects in a environment in a explicit dialog.

This research is still in earlier phases, a mathematical approach has to be done. Experiments with the real platform should also be performed to validate the utility of augmented reality in guidance assistance task.

References

1. Knapp, M.L., Hall, J.A.: Nonverbal Communication in Human Interaction. Wadsworth, Cengage Learning, Boston, MA (2010)
2. Sciutti, A., et al.: Anticipatory gaze in human-robot interactions. In: 7th ACM/IEEE International Conference on Human-Robot Interaction, HRI 2012, Boston, Massachusetts, USA (2012)
3. Staudte, M., Crocker, M.W.: The utility of gaze in spoken human-robot interaction. In: Proceedings of the HRI Workshop on Metrics for Human-Robot Interaction (2008)
4. Kohlbecher, S., Wiese, E., Bartl, K., Blume, J., Bannat, A., Schneider, E.: Studying Gaze-based Human Robot Interaction: An Experimental Platform. In: 7th ACM/IEEE International Conference on Human-Robot Interaction, Boston, Massachusetts, USA (2012)
5. Yoshikawa, Y., Shinozawa, K., Ishiguro, H., Hagita, N., Miyamoto, T.: Responsive robot gaze to interaction partner. In: Proceedings of Robotics: Science and Systems (2006)
6. Edward, T.: A System for the Notation of Proxemic Behavior. American Anthropologist 65(5), 1003–1026 (1963)
7. Hall, J., Hall, E.T.: The Hidden Dimension. Anchor Books (1966) ISBN 0-385-08476-5

8. Kanda, T.: Toward Integrating Natural-HRI into Spoken Dialog. Fall Symposium Series (2010)
9. Rios-Martinez, J., Spalanzani, A., Laugier, C.: Understanding human interaction for probabilistic autonomous navigation using Risk-RRT approach. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, September 25-30 (2011)
10. Lam, C.-P., Chou, C.-T., Chiang, K.-H., Fu, L.-C.: Human-centered robot navigation,towards a harmoniously human-robot coexisting environment. IEEE Transactions on Robotics 27(1) (2011)
11. Breazeal, C., Kidd, C.D., Thomaz, A.L., Hoffman, G., Berlin, M.: Effects of Nonverbal Communication on Efficiency and Robustness in Human-Robot Teamwork. In: EEE/RSJ International Conference on Intelligent Robots and Systems IROS (2005)
12. Brooks, A.G., Breazeal, C.: Working with robots and objects: revisiting deictic reference for achieving spatial common ground. In: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction, pp. 297–304. ACM (2006)
13. Kruse, T., Kirsch, A., Sisbot, E.A., Alami, R.: Exploiting human cooperation in human-centered robot navigation. In: 2010 IEEE RO-MAN, September 13-15 (2010)

Leg Detection and Tracking for a Mobile Robot and Based on a Laser Device, Supervised Learning and Particle Filtering*

Eugenio Aguirre, Miguel Garcia-Silvente, and Javier Plata

Department of Computer Science and A.I., CITIC-UGR.
E.T.S. Ingenierías en Informática y en Telecomunicaciones
University of Granada. 18071 - Granada, Spain
{eaguirre,M.Garcia-Silvente,jplata}@decsai.ugr.es
http://decsai.ugr.es

Abstract. People detection and tracking is an essential skill to obtain social and interactive robots. Computer vision has been widely used to solve this task but images are affected by noise and illumination changes. Laser range finder is robust against illumination changes so that it can bring useful information to carry out the detection and tracking. In fact, multisensor approaches are showing the best results. In this work, we present a new method to detect and track people using a laser range finder. Patterns of leg are learnt from 2d laser data using supervised learning. Unlike others leg detection approaches, people can be still or moving at the surroundings of the robot. The method of leg detection is used as observation model in a particle filter to track the motion of a person. Experiments in a real indoor environment have been carried out to validate the proposal.

Keywords: Leg detection and tracking, laser range finder, supervised learning, particle filter, mobile robots.

1 Introduction

A social robot or a service robot should have located the people at its surroundings and to be aware of their presence. Therefore, people detection and tracking is an essential skill in order to achieve a social and interactive robot. This is a very challenging task as people is freely moving by the environment. Many researches have focused their efforts in this field in last years using different kind of sensors. Vision systems have been widely used to deal with this problem [10]. In our research group this problem has been attacked through the use of stereo vision [14], [15], [16]. Stereo vision systems and other kinds of devices as the Kinect [13], allow the acquisition of various kinds of information such as color and depth. However the depth information is not always reliable and

* This work have been partially supported by Andalusian Regional Government project P09-TIC-04813 and the Spanish Government project TIN2012-38969.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_31,

Fig. 1. Peoplebot robot equipped with laser range finder SICK LMS200

the images are affected by noise and illumination changes. Other kind of sensor used in people detection and tracking is the laser range finder (LRF) [2], [8], [17]. Compared with vision, the use of a LRF is advantageous since it is robust against illumination changes in the environment. Nevertheless, the LRF has also some limitations because we obtain only distance information in one 2d-plane. The best solutions are obtained fusing the information from different sensor systems. There are several authors who combine LRF and vision [1], [9]. Thus, the trend in this field is to use multisensor approaches [4] because integrating several sources of information the obtained system is more robust and efficient.

In this paper we focus our attention in the development of a system to detect and track people using a LRF. In future works we will integrate the information from the LRF system with previous stereo vision systems for people detection and tracking [16]. In the current work, first a new method to detect legs using LRF and supervised learning is proposed. This method analyses certain geometric features present in the laser data and then a classifier is trained using Support Vector Machines (SVM) [6]. The classifier will be able to identify patterns of legs in the surroundings of the robot both for static or moving people. After that, a particle filter is designed to track the motion of people. The classifier is used as an observation model to detect the legs of the people. In the current work the robot is immobile and observing the motion of people at its surroundings.

The rest of this paper is organized as follows. Section 2 describes the hardware of our system and presents the human leg detection algorithm. Section 3 shows the tracking algorithm based on particle filter. In Section 4, experimental verifications are described. Finally some concluding remarks and future works are commented in Section 5.

2 Leg Detection

Our hardware system comprises a PeopleBot mobile robot equipped with a laser range finder SICK LMS200. It scans 180º with a 1º resolution at 75 Hz. The maximum range of distance in the current operation mode is 8 meters. It is mounted at a height of 30 cm from the ground. Fig. 1 shows this robot.

In the detection stage, our aim is to detect legs when people are moving and static. It is a challenging objective because legs patterns are different in both situations. In our approach we separate the laser measurements into clusters and geometrical properties of these clusters are studied. Such as their width or depth, which let us to distinguish between legs and other objects such as waste baskets or table legs. In order to detect if a cluster is a leg or not a SVM classifier is used. It has been trained with a large number of positive and negative samples. Bellow we will explain deeply all the steps performed.

Our system is designed to be used in Human Robot Interaction (HRI) applications, so we are going to consider that the person who desire to interact with the robot is near. Thus we have defined a maximum interaction distance and all range measurements further than this distance are ignored. In the experimentation, we have estimated this distance as 3m.

The robot mostly operates in an indoor polygonal environment. Therefore, the following step is to remove the information associated to long lines, that correspond to walls. For this operation a Line Hough Transform [11] has been implemented. The Hough transform is a feature extraction technique used in computer vision. The purpose of the technique is to find lines by a voting procedure.

The next step is to cluster the remaining points by the application of the clustering threshold distance $D_{threshold}$ and the clustering minimum number of points C_{min}. We consider as a cluster a group with at least C_{min} points where the distance between two consecutive points is lower than $D_{threshold}$. As geometrical properties we have computed the same three attributes used in [8]. They are: the contour of the neighbor points in a cluster from P_1 to P_m, the width defined as the distance from P_1 to P_m and the depth as the maximum distance between a point P_i and the line $\overline{P_1 P_m}$. Figure 2 shows these attributes.

A large data set has been acquired for training the SVM. This dataset contains positives samples, which was registered with people walking and standing in front of the robot and negative samples corresponding to table legs, waste baskets, boxes and an extinguisher. Notice that some of this objects could have geometrical properties similar to an human leg, and they will increase the

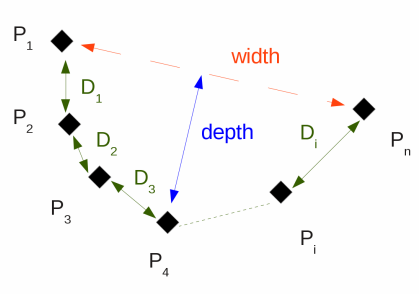


Fig. 2. Illustration of the defined geometrical parameters in a cluster

precision and robustness of the classifier. It is important to remark that positive samples were acquired while people were moving, so our system will be able to detect a person who is walking, not only people standing. We use a balanced dataset containing 1156 both positive and negative samples.

In order to apply the SVM classifier, the LibSVM was used [6]. Different kernels have been considered and the best precision is obtained with radial basis function (RBF). A wide grid-search using cross-validation has been performed in order to find the optimal value for these parameters [7] obtaining a precision about 91.5%.

Table 1 corresponds to contingency table for a 10-fold cross validation using the parameter values described before. Results are very promising since the rates of true positives and true negatives are very high.

Table 1. Contingency table

		Observation class	
		Positive	Negative
Predicted class	Positive	93.67 %	12.35 %
	Negative	6.33 %	87.65 %

3 Tracking

Particle filter is well known for its many applications in tracking. Target tracking problem is expressed by recursive Bayesian estimation. Essentially, two steps are given in each iteration: prediction and estimation. Both steps take into account the information of an observation model. Equations of particle filter are well known [3].

In this paper, a particle filter is used to estimate the position of the detected people. In the current work only one person will be tracked. The state in time t is defined as a pair of coordinates (x, y). The coordinates correspond to the middle point of the straight line which links the two points representing both detected legs of the person. That is, the people position S_t is represented by a pair $S_t = [x_t, y_t]$. The prediction is carried out by the model of the state transition. We do not consider in the model the motion of the people. Thus, a model of the movement of the target is intentionally not defined. The state transition is defined as $S_t = S_{t-1} + R_{t-1}$ where S_{t-1} is the previous state vector and R_{t-1} is the process noise. The noise is modeled using a Gaussian with average μ_R and standard deviation σ_R. Experimental data have been taken into account to establish the values of μ_R and σ_R in order to model the conditions of the real world.

Condensation algorithm [12] is used to generate a weighted set of particles $(s_i(t), \Pi_i(t))$ where $s_i(t)$ represents a possible position of the people, and $\Pi_i(t)$ is a factor called *importance weight* representing an estimation of the observation. At the beginning the algorithm is provided by an initial sample $(s_i(0), \Pi_i(0))$ of

N equally weighted particles. At each iteration, the algorithm uses the sample set $(s_i(t-1), \Pi_i(t-1))$ to create a new one. A resample mechanism is used to solve the divergence problem by eliminating particles having low importance weights. Afterwards, the model of state transition is used to predict the motion of the person obtaining the prediction of the state S'_t. The weight $\Pi_i(t)$ of each particle is computed based on the new observation $Z(t)$. Then the weights are normalized so that $\sum_{i=1}^{N} \Pi_i(t) = 1$.

The observation model is needed to carry out the update. As model of observation, the leg detection method explained in Section 2 is used. Thus, each laser reading set is analyzed and the positions of possible legs are obtained. To calculate the probability for each particle the nearest possible legs are taken into consideration. The probability of each particle is calculated as:

$$\Pi_i(t) = \frac{1}{\sigma_z \sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{dist-\mu_z}{\sigma_z}\right)^2} \tag{1}$$

being $dist$ the euclidean distance between $s_i(t)$ and the position of the nearest legs. The parameters μ_z and σ_z correspond to the average and standard deviation of a normal distribution, respectively. $\mu_z = 0$ and σ_z has been experimentally tuned. The final person position corresponds to the mean of the state $\mathcal{E}[S(t)]$, calculated as $\mathcal{E}[S(t)] = \sum_{i=1}^{N} \Pi_i(t) s_i(t)$.

4 Experimentation

Different experiments have been carried out in order to validate the proposal. The goal is to test the accuracy of the tracking algorithm in a real environment having several people. A set of seven paths on the floor were defined with different trajectories and including easy paths like a straight line as well as more complicated paths such as circles or curves in which motion direction continuously changes. Fig. 3 shows the acquisition scenario and the seven paths defined. Then, different samples of people following these trajectories have been acquired.

Fig. 3. Acquisition scenario for tracking system test

Four different people have participated in the experiments. It is important to acquire data from different people since each one has his or her own gait. Every person has walked on each trajectory six times with different speed walking. In total 168 trajectories have been analyzed.

In order to evaluate the tracking performance of each trajectory, we have calculated the distance of the predicted position by the tracker to the real trajectory, which has been manually computed. The MOPT measurement [5] has been used for this purpose. The algorithm has been evaluated with different numbers of particles: 50, 100, 150, 200 and 250. Table 2 shows the results obtained for each trajectory in mm, where column T indicates the trajectory. For each trajectory and each number of particles, mean error (MOTP) and standard deviation in mm are indicated. As it can be observed the error obtained is very low and it decreases when the number of particles increases. Fig. 4 shows as the tracking error decreases when number of particles increases. For a number of particles higher than 200 the error decreases in a lower rate. Because of that, we have used 200 particles. Using this number of particles, the average tracking error for all the trajectories is computed. This value is 24.11 mm and its standard deviation is 4.16 mm. The average run time obtained is very low, 2.4 ms, so we can claim that the system can operate in real time.

Table 2. Results in mm for particle filter with 50, 100, 150, 200 and 250 particles

	Number of particles									
	50		100		150		200		250	
T	MOTP	Std	MOTP	Std	MOTP	Std	MOTP	Std	MOTP	Std
1	30.78	17.31	28.92	16.70	27.41	16.20	26.71	16.62	26.66	16.47
2	34.39	42.08	31.30	40.28	30.47	40.23	30.44	40.53	30.24	41.30
3	24.12	17.24	21.87	15.71	20.53	15.33	20.31	15.92	19.21	15.46
4	28.77	18.68	27.13	17.84	25.44	17.67	25.05	18.22	24.65	18.17
5	28.40	17.25	26.23	16.85	25.42	17.16	25.49	18.10	24.64	17.37
6	26.78	18.33	23.77	17.72	23.74	17.41	22.90	17.18	22.83	17.52
7	21.47	16.90	19.39	17.20	18.58	16.93	17.90	17.07	17.19	16.76

5 Conclusions and Future Work

The contributions of this paper can be summarized in two. The first one is the method to detect the human legs. It does not assume any specific shape of legs. Also, it is not needed that the human is still and looking towards the robot. We use supervised learning to learn the different patterns of legs which can appear in the robot surroundings corresponding to moving or still people. These patterns are learned from a large number of sample data using SVM. The second contribution is the robust tracking of leg positions using the method for leg detection as an observation model within a particle filter. Experiments on leg detection and on people tracking have been carried out to validate the

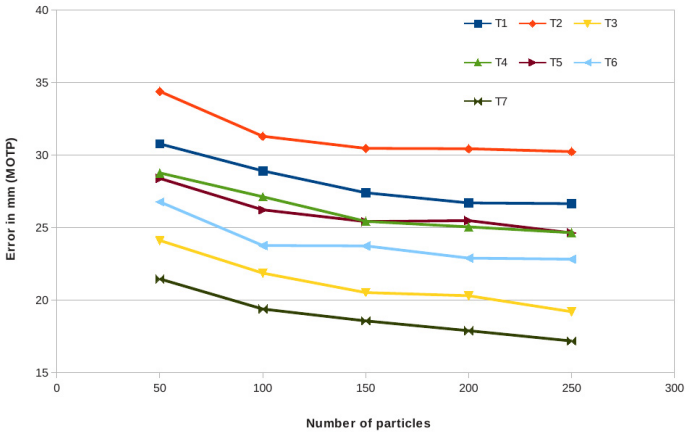


Fig. 4. MOTP error in mm vs number of particles for each trajectory

proposal. The experimental results show that human legs are properly detected in the most of the cases. Also, the experiments on the tracking show that our LRF based tracking system is working obtaining a low error tracking the human position.

As future work, the LRF based tracking system will be integrated with other previous stereo vision based systems in order to obtain a multisensor tracking system. The multisensor tracking system will be used to follow a person in an indoor environment.

References

1. Ansuategui, A., Ibarguren, A., Martínez-Otzeta, J.M., Tubío, C., Lazkano, E.: Particle filtering for people following behavior using laser scans and stereo vision. International Journal on Artificial Intelligence Tools 20(2), 313–326 (2011)
2. Arras, K.O., Grzonka, S., Luber, M., Burgard, W.: Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 1710–1715 (2008)
3. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing 50(2), 174–188 (2002)
4. Bellotto, N., Hu, H.: Multisensor-based human detection and tracking for mobile service robots. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39(1), 167–181 (2009)
5. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The clear mot metrics. Eurasip Journal on Image and Video Processing (2008)
6. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

7. Chang, C.-C., Hsu, C.-W., Lin, C.-J.: A Practical Guide to Support Vector Classification. National Taiwan University (2010)
8. Chung, W., Kim, H., Yoo, Y., Moon, C.-B., Park, J.: The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder. IEEE Transactions on Industrial Electronics 59(8), 3156–3166 (2012)
9. Fritsch, J., Kleinehagenbrock, M., Lang, S., Plötz, T., Fink, G.A., Sagerer, G.: Multi-modal anchoring for human-robot interaction. Robotics and Autonomous Systems 43(2-3), 133–147 (2003)
10. Gavrila, D.M.: The visual analysis of human movement: A survey. Computer Vision and Image Understanding: CVIU 73(1), 82–98 (1999)
11. Hough, V., Paul, C.: Method and means for recognizing complex patterns, U.S. Patent 3069654 (1962)
12. Isard, M., Blake, A.: Condensation-conditional density propagation for visual trackings. International Journal of Computer Vision 29, 5–28 (1998)
13. Microsoft. Kinect official webpage (2010)
14. Muñoz-Salinas, R., Aguirre, E., García-Silvente, M.: People detection and tracking using stereo vision and color. Image and Vision Computing 25, 995–1007 (2007)
15. Muñoz-Salinas, R., Aguirre, E., García-Silvente, M., Paúl, R.: A new person tracking method for human-robot interaction intended for mobile devices. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 747–757. Springer, Heidelberg (2007)
16. Paúl, R., Aguirre, E., García-Silvente, M., Muñoz-Salinas, R.: A new fuzzy based algorithm for solving stereo vagueness in detecting and tracking people. International Journal of Approximate Reasoning 53, 693–708 (2012)
17. Schenk, K., Eisenbach, M., Kolarow, A., Gross, H.: Comparison of laser-based person tracking at feet and upper-body height. In: Bach, J., Edelkamp, S. (eds.) KI 2011. LNCS, vol. 7006, pp. 277–288. Springer, Heidelberg (2011)

Part VI
Agricultural Robotics

Precision Humidity and Temperature Measuring in Farming Using Newer Ground Mobile Robots

L. Cancar, David Sanz, J.D. Hernández, Jaime del Cerro, and Antonio Barrientos

Center for Automation and Robotics UPM-CSIC,
C/Jose Gutierrez Abascal, 2,
28006 Madrid, Spain

Abstract. Precision farming goal is to maximize the productivity of the crops while minimizing the use of resources. Given that the agricultural technique has evolved along the millenniums, the only real option to keep increasing the agricultural productivity requires the use of technology: currently, many different robots are being used to harvest, plow and identify weeds. This work aims to reduce the environmental invasion the use of these implies, maximizing at the same time the knowledge of the crop status. We propose a new type of environmental-friendly ground mobile robot employed to monitor temperature and humidity in agricultural fields Used as a distributed sensor web. It takes measures of these environmental variables in any specific place required. It also allows to determine zones of the crop with bad irrigation or under germination risks and take corrective actions on time, providing better growing conditions: adequate germination time, faster growing and better efficiency. We present its architecture and a new external module used to measure the temperature and humidity. Beside, we also present two experiments carried out to validate the spherical robot approach: the results obtained first at a cornfield and the second in small orchard in CAR-UPM-CSIC facilities are exposed and analysed.

Keywords: Precision farming, mobile robots, temperature and moisture monitor, spherical robot.

1 Introduction

World population has reached 7 billion people and it is expected to reach 9 billion by 2050[1]. This excessive population increment has as a consequence the growth in humankind necessities (energy, food, health, etc.). Considering that the Earth's resources are now being overexploited, pleasing the increasing necessities requires and involves the use of new technologies. For example, it has been calculated that agriculture needs to increase a 25% its productivity to meet the production level needed by the year 2050 [Bergerman et al., 2013]. For this reason, robotics, automation and agriculture are fields that each day are more related. Automatic

[1] `http://esa.un.org/unpd/ppp/index.htm`

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_32,

sensing, handling, and automatic processing of products are now commonplace in agriculture [Billingsley et al., 2008]: The necessity of producing more food with less resources leads to, as consequence, the presence of more automated and robot-assisted crops. New technologies can be used in agriculture to plow the soil efficiently, seek for weed and only applied herbicides where needed. Efficient irrigation and intelligent harvesting are also fields were the technology has a significant impact.

Many robots have been used in these areas in many different ways. Nevertheless, there are still many challenges that have been not addressed yet: For example, the environmental impact or the efficient use of energy. This work proposes a different type of mobile robot to measure environment. It supposes a low environmental impact system to acquire temperature, soil temperature and environmental humidity data continuously, able to detect zones of the field with irrigation system problems or where new points of irrigation are needed. It would maximize the growing speed, the seed protection and crop health in general.

2 Precision Farming

Precision farming (PF) is an agricultural management concept based on observing and responding to intra-field variations: as explained by [Lu et al., 1997], nowadays PF goes further than precision agriculture (PA), since this concept concerns the whole farm management. Its goal is to optimize returns on inputs while preserving resources. Ideally, it can be considered a trend that allows a better use of natural resources preserving the environment and helping to reach each year the increasing production needs.

Focusing on this concept, the main targets of PF could involve: i) the reduction environmental loading by applying fertilizers and pesticides only where they are needed and when they are needed; ii) the minimization of losses from excess applications or due to nutrient imbalances; iii) the reduction of weed escapes and insect damage; iv) the development of techniques that allow to decrease of contamination of land and water; and, in general, the application of any sustainable technique that may provide an increase of the productivity and the efficiency [Bongiovanni et al., 2004]. Subsection 2.1 presents one of the physical evidences that possibilities the use of technology in a favorable way.

It is crystal clear that the technology is at the agriculture service. It can be applied in many topics: from sensing to automation of tasks (i.e. AI applied to the farming processes, etc.), it provides with great results in both environmental and economic terms. However, the main role is nowadays played by robotics for supporting on-site physical operations. Its employment in PF is still growing at a fast pace, providing every day new capabilities and advantages: for example, J.Valente uses UAVs with vision systems for detecting weeds in large fields [Valente et al., 2011]; the RHEA project [Gonzalez-de-Santos et al., 2011] employs different types of actuators for efficiently remove weeds, adequating the treatment to the specific problem and minimizing the use of pesticides. Another good example is the use of automated tractors equipped with sensors (lasers, cameras and others) to seed, harvest, weeding in a efficient manner reducing

the damage to the soil and the products used [Eaton et al., 2008]. Nevertheless, many challenges remains unsolved. Section 3 presents a study on this problematic and Section 4 digs into the solution adopted in this work: ROSPHERE

2.1 The Role of Humidity and Temperature in PA

No usually known, soil temperature and moisture are really important in crop management. Not during the adult life of the plant but during the seeding, planting and germination process, these variables are critical [Bierhuizen et al., 1974, Harper et al., 1965]. The soil temperature, for example, is a trustworthy indicator of the proper seeding timing: If the land is too cool, germination is delayed, which results in seed stress and uneven or inadequate seeding emergence. In contrast, if the temperature is unusually high, the seeds can be dried up and their dormancy period violated [Garcia-Huidobro et al., 1982, Roberts et al., 1988]. Table 1 presents some examples of the optimum (and minimum) germination temperatures for several different plans/crops.

On the other side, moisture should be considered as a relevant regulator of the own temperature. Wet soils take longer to warm up and cool down, being possible to use this effect for, for example, mitigating cold stresses during the planting [Doneen et al., 1943]. Furthermore, the adequate management of both variables supposes a huge difference in terms of production, growing speed and how quickly plants take up water and nutrients. Besides, soil temperature affects to the speed of biological reactions, speeding up the reactions the warmer temperatures and slowing them down the colder ones.

They are usually measured at 5cm under the land level and, considering the differences between day and afternoon/night, both variables have to be acquired (at least) twice per day [2]. Nevertheless, J.Dudhia proved a direct relation among the superficial temperature/moisture and the underground one: [Dudhia et al., 1996]. It allows to use a directional sensor as the SHT71 humidity sensor and extrapolate the soil temperature.

Even more, the relation between the environmental moisture and the irrigation necessities is clear: In every stage of the crop, an insufficient watering results in the crop dried up. However, an excessive irrigation can drown the seeds and drag the nutrients downwards. Thus, the humidity control provides with not only a good indication of the field status, but also with a reference for the irrigation system.

3 Mobile Robots in Agriculture

As presented before, a great amount of the mobile robots have been used in the last decades in PA: [Hollingum et al., 1999], [Lee et al., 1999] and [C.W. Bac et al., 2013] are a good example since they present different variations and capabilities. These last ones, together with the efficiency ratio and the general characteristics,

[2] The discrepancy between them could be up to 100%, but the average daily soil temperature usually lies between both readings.

Table 1. Germination temperatures for field crops

	Minimum (°)	**Preferred** (°)
Cereals and oilseed		
Wheat	4	20
Barley	3-5	20
Oats	5	20-24
Canola [3]	7-10	20
Forage Crops		
Alfalfa	1	25
Red clover	3	26
Sweet clover	1	18-25
White clover	5	18-20
Orchard	1	18-20
Vegetables		
Bean	8-10	16-30
Cabbage	4	7-35
Carrot	4	7-30
Corn	10	16-32
Lettuce	2	4-27
Pepper	16	18-35
Pumpkin	16	21-32
Spinach	2	7-24
Tomato	10	16-30

have been improved along the time, and their implementation (navigation, guidance, control, etc.) mastered. However, their typology have remained the same along the time: wheeled ground mobile robots. It implies that they continues presenting the same disadvantages: the use of these robots compacts the ground and hinders the nutrient penetration, damaging in many times the crop and slowing down the crop growing.

That is why a new generation of robots -depicted in figure 1- is emerging and becoming popular nowadays: from aerial robots [Valente et al., 2011] to mobile manipulators [De-An et al., 2011], they all try to minimize the environmental impact while maximizing the effectiveness. Nevertheless, they do not allow an *in situ* ground measurement, really important in many agricultural practices: The robot to be used in an specific crops have to be selected according to the field conditions, the fragility of the crops, its extension, etc. Is for those reasons that when is necessary to measure variables directly over the field, a compromise among capabilities, crop structure and field invasion has to be reached. The work presented in this paper proposes a new alternative that provides this local acquisition capability together with a environmental-friendly motion system: spherical robots, due to its way of induce movement, are much less invasive, being possible to use it as a ground sensor. The major drawback is referred to the necessity of new guidance and control techniques to allow precise movements in the field that guarantee a correct acquisition.

Fig. 1. Ground mobile robots in agriculture, (a) Aarhus University in Denmark (b) Wageningen University and Research Centre, (c) CAR-UPM-CSIC

3.1 Spherical Robots

Even though nowadays robotic spheres are not widely used as ground mobile robots, there are quite significant new references in the literature. They are mainly focused in operational concepts and new prototypes proposals: Initially, the research done aimed to validate concepts about its physics. In this regard, different authors have proposed diverse internal structures to generate movement, where the objective was to create a mechanical system that allows shifting its centre of mass inducing self-motion as consequence. There are five typical methods that can be found in literature used to achieve this goal, an explanation of the methods and examples could be found in [Amour et al., 2006 and Ylikorpi et al., 2007]. These typical methods are explained below.

- *Spring central member.* this method uses a central body that includes a driven wheel at one of its ends, and a passive wheel in the other, a fixed spring guarantees the contact between both wheels and the spherical shaped body. The main disadvantage of this design is the loss of energy due to friction between both wheels and the difficulty to orientated motion control.

- *Car driven.* It is one of the simpler concepts, the robot relies on a vehicle inside that induce motion in the sphere by its motion inside it. Even though it is easy to construct, the main drawback is the lack of contact when the robot moves on surfaces with depressions and bumps, implying loss of energy and difficulty to generate movement and overcome obstacles.
- *Ballast mass with fixed axis system.* This system uses an inner pendulum with two rotational degrees of freedom. The first DoF rotates around a fixed transverse axis and the second around a longitudinal axis. [Michaud et al., 2001 and Kayacan et al., 2011], has the advantages of better control to change the mass center and less energy loss. As disadvantage the robot is non-holonomic.

- *Ballast mass with moving axis.* It also has an inner pendulum, but in this case with an additional DoF that allows the main axis to move, its main

disadvantage is its complexity, requiring highly elaborated structures, as advantage it can generate movement in any direction.

– *Hemispherical wheels.* use two hemispheres as large encompassing wheels, the design of this type of robots is the simpler of all, but, it has as disadvantages the impossibility of a total enclosure due two the need of each side moving independently, difficulty to change direction, as well the need of an independent control of each of the hemispheres, as advantage the force is directly applied to the ground to generate movement implying a better efficiency.

4 ROSPHERE

ROSPHERE is a spherical-shaped robot with the capability to self-induce non-holonomic movements by varying its mass center, it uses the Ballast mass with fixed axis system explained previously. The robot has an inner two-degree-of-freedom pendulum. The robot includes a spherical shaped body (30 cm of diameter), fixed main axis, a central unit or ICU (Internal Control Unit) and the ballast or hanging mass. The first DOF rotates about the fixed axis, a hanging mass located at the bottom of the sphere is moved generating back and forward movements, while the second DOF has a mechanically limited range of rotation, and rotates about a perpendicular axis generating side movements, for more information see [J.D Hernández et al., 2013]. Current version of the robot was designed to get the Centre of Mass as far as possible from the geometrical center in order to induce the movements easily (see figure 2).

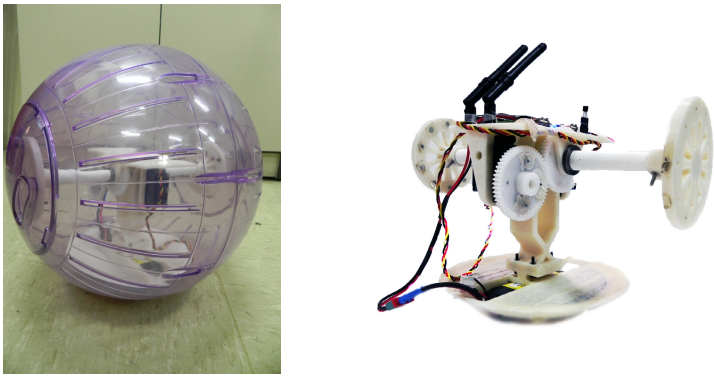


Fig. 2. ROSPHERE inner body and robot image

4.1 Hardware Architecture

ROSPHERE incorporates all necessary resources in order to behave as an autonomous vehicle. Thus, the equipment includes a embedded computing system

composed by a Robovero and an Overo Fire embedded computer. the communication capabilities of ROSPHERE includes Wi-Fi, Bluetooth and Xbee modules as communication mediums, to transmit and receive data not only to the base station but also with external sensors. with regard to the sensing capabilities, the system includes the equipment required for guidance, navigation and control (GNC) systems, such as a low cost inertial measurement unit (IMU) in order to measure angular velocities and accelerations, a magnetometer with pan-tilt correction capability and a single GPS.

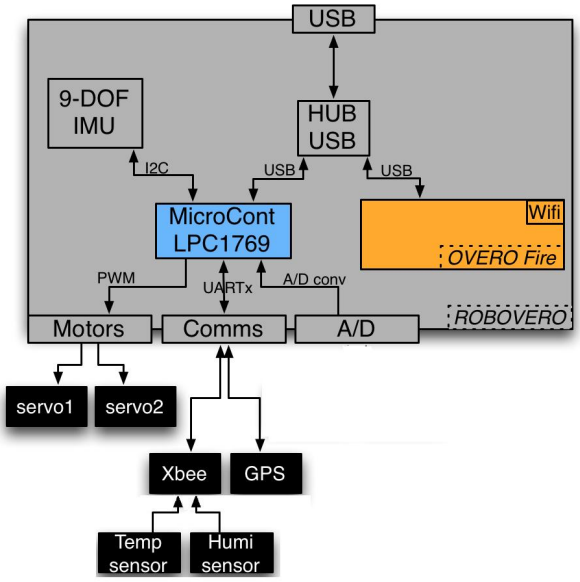


Fig. 3. ROSPHERE hardware

To allow visual inspection of how the pendulum works during the operation of the robot, the prototype has been made by using a commercial transparent ball. Although good results have been obtained by using this enclosure, a more suitable material should be used in its final version in order to increase traction. The slots in the case have been sealed using special rubber bands in order to keep the robot protected from dust and humidity and to increase traction. Different sensors can be placed on both of its sides, Nevertheless, if a heavy sensor (e.g. a camera) is used, a counterweight or different sensors should be added to the other side of the sphere in order to maintain the center of mass in the same position.

External sensors such as humidity and temperature are currently connected to an arduino external module that sends the collected data through an Xbee device to the central unit.

4.2 Software Architecture

Software architecture can be explained into two parts. The first part is responsible for the high-level processing and the second for the low level (sensors and actuators level, communications). The high level process interprets remote commands from the central station and performs basic GNC tasks by using information from the sensorial system (IMU, magnetometer). Therefore, this layer transfers teleoperation commands to low-level controller and calculates references to controllers to perform the maneuvers required. This part is executed on Overo board with Ubuntu version 12.0.4. further, a ROS-Core has been included in order to implement a standard ROS (Robot Operating System) architecture.

ROSPHERE aims to be a complete ROS compliance robot, in order to use a wide spread of algorithm and already developed tools and their drivers to be used with robotic simulators (e.g. Gazebo or V-REP).

The Overo communicates with Robovero by using a USB hub. Robovero is in charge of low-level control and sensing and serial communication. Therefore, it reads the serial message received by an Xbee module attached the humidity and temperature data and performs basic filtering in order to create complete messages for the high level controller. An important development was required in order to optimize Robovero firmware in order to improve the update frequencies in sensors readings and processing.

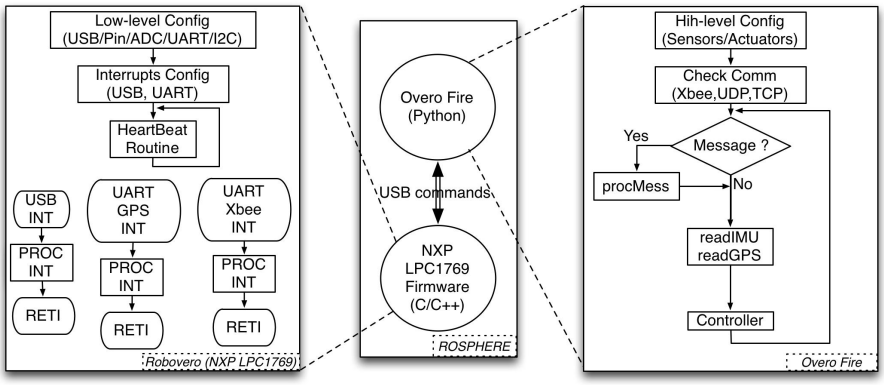


Fig. 4. ROSPHERE software

4.3 Temperature and Humidity Sensor Module

An arduino FIO device has been programmed to read and SHT71 which provide environmental humidity and temperature; It also reads a MLX90614 IR sensor, used to read soil or plants temperature. All the data is obtained from the sensors and transmitted to the arduino by and I2C channel and digital pins. Then the arduino process the data and transmit it via an XBee module to be received in

the main processor of the mobile robot used. The module could be used in any mobile robot only be attaching it to a convenient place, the robot only should have an XBee module on its main processor to receive the data string from the humidity and temperature module.

This module could be integrated easily in a mobile robot as ROSPHERE to obtain the temperature and humidity in crops

5 Test and Results

Two different sets of test were conducted in order to validate the behavior of the whole system. As far as the ROSPHERE mobility had been already verified, the experiments were designed to check the correct performance of the humidity sensor and the IR sensor implemented in ROSPHERE. The second goal of the tests conducted was to corroborate the utility of the moisture acquisition, in order to validate it as a bad-irrigation-zones and germination-risk-areas identifier.

5.1 Orchard Test

An experimental setup was made in the laboratory facilities in order to measure humidity and validate results: the irrigation system shown in figure 5 was implemented. It is composed by six irrigation nozzles, which have a separation 30cm among them. After a normal operation, the two nozzles located in the middle were obstructed to obtain a bad irrigation in order to check if difference in temperature and humidity in the surrounding area between followed the model studied in subsection 2.1. As table 2 shows, the hypothesis was validated, existing a relevant difference in the magnitudes between the broken and the normal irrigation nozzles.

Fig. 5. orchard at CAR-UPM-CSIC installations

In the orchard, humidity, environmental temperature and soil temperature were measured in 6 different points. Besides, they were measured at different times, contrasting the ROSPHERE measures with an external module located at 10cm above the ground. The acquisition started at 9:30am -just after the irrigation- measuring both the temperature and humidity every hour until 14:30pm, when the failure was simulated: In two consecutive points the irrigation system was obstructed and the data acquired proved the system's capability for detecting those failures only using the temperature and humidity measures. Table 2 presents the temperature and humidity log obtain every hour.

Table 2. Temperature and humidity in the orchard

	Irrigation points (°C , %)					
Time	**Point 1**	**Point 2**	**Point 3**	**Point 4**	**Point 5**	**Point 6**
9:30am	20.2 ,46.9	20.6 , 46.6	22.2 , 44.9	22.2 , 45.1	19.8 , 46.3	18.9 , 47.8
10:30am	21.2 ,40.9	21.0 , 41.1	23.0 , 39.7	23.4 , 39.8	20.5 , 40.5	20.3 , 41.2
11:30am	21.8 , 35.6	21.3 , 34.8	24.1 , 33.5	24.1 , 33.4	21.4 , 36.4	20.9 , 36.3
12:30am	29.6 , 30.1	29.5 , 30.0	33.7 , 29.7	33.2 , 28.6	29.3 , 32.7	29.8 , 32.5
13:30am	38.4 , 26.9	40.1 , 27.0	39.4 , 25.7	37.8 , 27.0	33.4 , 28.0	33.6 , 28.6
14:30am	43.0 , 20.9	45.6 , 22.5	44.9 , 19.4	43.4 , 19.4	38.8 , 20.4	39.3 , 20.4

Besides, the recorded humidity, environmental temperature and soil temperature in the orchard at 9:30am are plotted in figure 6 (a). There, it could be easily distinguished the zone with bad irrigation: highlighted in red, it is zone differentiated by an increased of the soil temperature and a decrease of humidity.

On the other hand, figure 6 (b) presents the plot obtained at 13:30 pm .In this occasion, a portion of the orchard (at the end of the crop row) was in the shade so its soil temperature was minor than the rest of the orchard. Even though, the humidity level was revealed as a good indicator of the proper irrigation.

5.2 CAR-UPM-CSIC Corn Crop

The mobile robot ROSPHERE was also used to measure the temperature and humidity of the corn crop located at CAR-UPM-CSIC facilities. This test was performed to verify the capacities of a mobile robot to measure directly in a real environment (see Figure 7). The sensor was located at the ROSPHERE's the side, 10cm high from the soil, pointing to the ground. The crop had a drip irrigation and it was not irrigated the day when the measures were taken.

Figure 8 presents the humidity, environmental temperature and soil temperature measured in a cornfield by ROSPHERE in the CAR-UPM-CSIC facilities' corn crop.

By using the ground mobile robot, it was possible to measure the environment temperature in any wanted position , soil temperature and humidity. It was also that the mobile robot ROSPHERE did not compact the soil, causing minimum alterations on it.

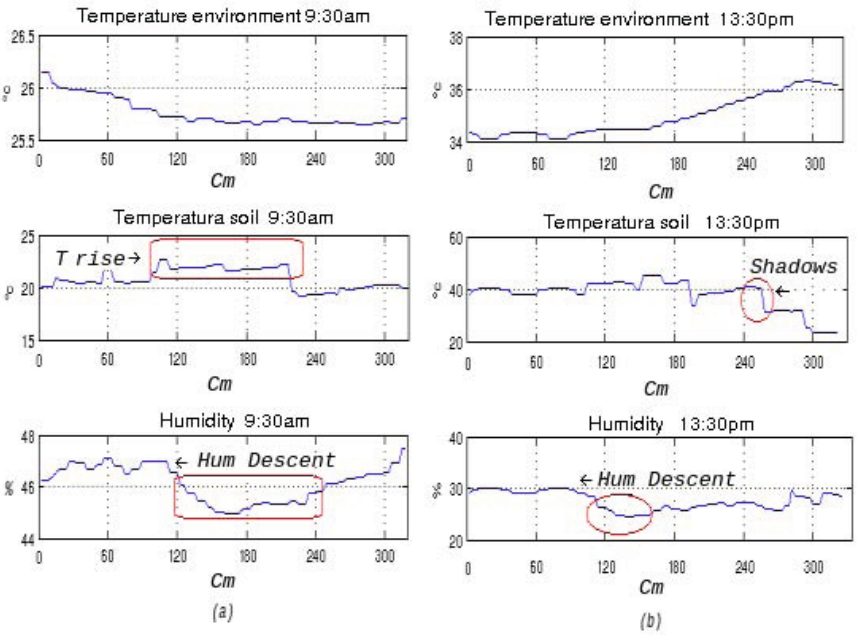


Fig. 6. Environmental variables in the orchard at 9:30am and 13:30pm

Fig. 7. ROSPHERE in the CSIC facilities' corn crop

Even though the temperature and humidity was measure properly, due to the irregular structure of the field (stones, holes, and other obstacles), the control of ROSPHERE was somewhat difficult and major problems were present to overcome medium size obstacles (i.e. 5cm height stones).

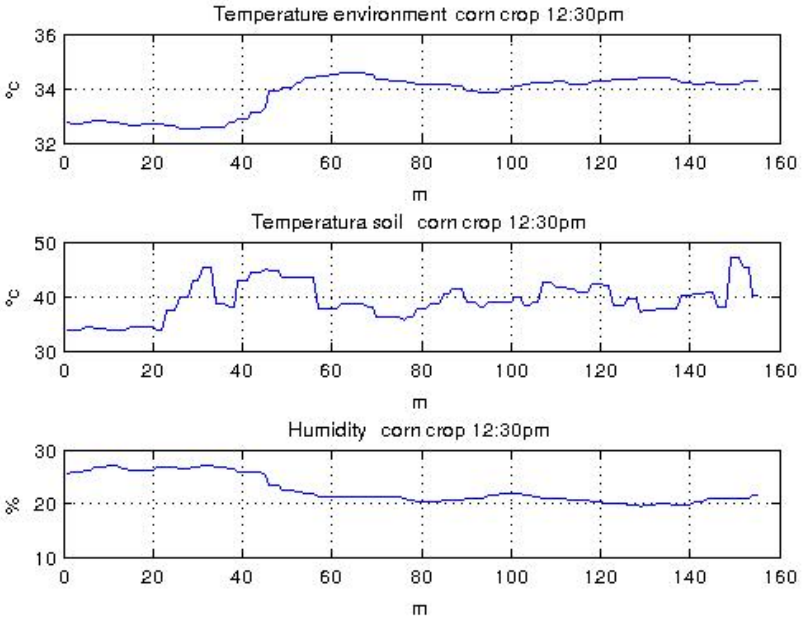


Fig. 8. Environmental variables, temperature and humidity obtained at corn crop CAR-UPM-CSIC

6 Conclusions

Precision agriculture is a valuable tool to increase crops productivity. It should use every new technology at the reach as new sort of ground mobile robots. However it implies a lot challenges that have to be addressed. For example, shadows affect how the soil temperature evolves: when the temperature falls drastically in the crop, only humidity measures should be taken into account to determine bad irrigation zones.

In this preliminary study, zones with bad irrigation could be detected by using a mobile robot due to they capability of measuring changes in temperature and humidity in every point along the the crop.

When a low impact methods are needed for monitoring the field, new types of mobile robots as ROSPHERE can be useful. It allows to measure environmental variables guaranteeing the crop integrity and the suitable nutrients and water absorption.

However, this types of robots requires new control algorithms and mechanisms in order to overcome obstacles in unstructured fields.

7 Future Work

Due to the difficulty to overcome obstacles (stones, holes, irregularities on the ground), new control control strategies and possibles mechanical changes should be developed to have the spherical robot ROSPHERE working in unstructured environments.

New experiments and algorithms to determine thresholds in the variation of humidity and soil temperature to determine fails in irrigation.

Determination of schemes to monitoring temperature and humidity of crops and cooperation with other robots to cover efficiently crops.

Acknowledgment. This work have been supported by the Robotics and Cybernetics Research Group at Technique University of Madrid (Spain), and funded under the projects ROTOS: Multi-Robot system for outdoor infrastructures protection, sponsored by Spain Ministry of Education and Science (DPI2010-17998); RHEA: Robot Fleets for Highly Effective Agriculture and Forestry Management, sponsored by the European Commissions Seventh Framework Programme (NMP-CP-IP 245986-2 RHEA); and ROBOCITY 2030 (S-0505/DPI/ 000235) from the CAM's Regional Plan for the Scientific Research and Technological Innovation (PRICIT)

References

Billingsley, J., Visala, A., Dunn, M.: Robotics in Agriculture and Forestry, pp. 1065–1077. Springer, Heidelberg (2008)

Bergerman, M., van Henten, E., Billingsley, J., Reid, J., Mingcong, D.: IEEE Robotics and Automation Society Technical Committee on Agricultural Robotics and Automation Robotics Automation. IEEE Magazine, 20–125 (2013)

Bongiovanni, R., Lowenberg-Deboer, J.: Precision Agriculture and Sustainability, pp. 359–387. Kluwer Academic Publishers (2004)

Eaton, R., Katupitiya, J., Siew, K.W., Dang, K.S.: Precision Guidance of Agricultural Tractors for Autonomous Farming. In: 2008 2nd Annual IEEE Systems Conference, pp. 1–8 (2008)

Valente, J., Cruz, A.B., del Cerro, J., Muñoz, D.S.: A waypoint-based mission planner for a farmland coverage with an aerial robot-a precision farming tool. Industriales (2011)

De-An, Z., Jidong, L., Wei, J., Ying, Z., Yu, C.: Design and control of an apple harvesting robot. Biosystems Engineering, 112–122 (2011)

Armour, R.H., Vincent, J.F.V.: Rolling in Nature and Robotics: A Review. Journal of Bionic Engineering, 195–208 (2006)

Ylikorpi, T., Suomela, J.: Ball-shaped robots. In: Climbing & Walking Robots, Towards New Applications, pp. 235–256. Itech, Vienna (2007)

Kayacan, E., Bayraktaroglu, Z., Saeys, W.: Modeling and control of a spherical rolling robot: a decoupled dynamics approach. Robotica, 110 (2011)

Michaud, F., de Lafontaine, J., Caron, S.A.: Spherical Robot for Planetary Surface Exploration. In: Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space (2001)

Hernández, J.D., Barrientos, J., del Cerro, J., Barrientos, A., Sanz, D.: Moisture measurement in crops using spherical robots. Industrial Robot: An International Journal, 59–66 (2013)

Bierhuizen, J.F., Wagenvoort, W.A.: Some aspects of seed germination in vegetables. 1. The determination and application of heat sums and minimum temperature for germination. Scientia Horticulturae, 213–219 (1974)

Garcia-Huidobro, J., Montheith, J.L., Squire, G.R.: Time, Temperature and Germination of Pearl Millet (Pennisetum typhoides S. & H.): I. Constant Temperature. Journal of Experimental Botany, 288–296 (1982)

Doneen, L.D., MacGillivray, J.H.: Germination (emergence) of vegetable seed as affected by different soil moisture conditions. American Society of Plant Biologists 524 (1943)

Harper, L., Williams, J.T., Sagar, G.R.: The behaviour of seeds in soil: I. The heterogeneity of soil surfaces and its role in determining the establishment of plants from seed. The Journal of Ecology, 273–286 (1965)

Roberts, E.H.: Temperature and seed germination. Symposia of the Society for Experimental Biology 109 (1988)

Dudhia, J.: A multi-layer soil temperature model for MM. Preprint from the Sixth PSU/NCAR Mesoscale Model Users' Workshop, 22–24 (1996)

Bac, C.W., Hemming, J., van Henten, E.J.: Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. Computers and Electronics in Agriculture, 148–162 (2013)

Hollingum, J.: Robots in agriculture. Industrial Robot: An International Journal, 438–446 (1999)

Sevila, F., Baylou, P.: The principles of robotics in agriculture and Horticulture. CAB International (1991)

Sistler, F.: Robotics and intelligent machines in agriculture. IEEE Journal of Robotics and Automation, 3–6 (1987)

Gerrish, J.B., Surbrook, T.C.: Mobile robots in agriculture. American Society of Agricultural Engineer (1984)

Lu, Y., Daughtry, C., Hart, G., Watkins, B.: The current state of precision farming. Food Reviews International, 141–162 (1997)

Valente, J., Sanz, D., Barrientos, A., del Cerro, J., Ribeiro, Á., Rossi, C.: An Air-Ground Wireless Sensor Network for Crop Monitoring. Sensors, 6088–6108 (2011)

Gonzalez-de-Santos, P., Ribeiro, A., Fernandez-Quintanilla, C.: The RHEA Project: using a robot fleet for a highly effective crop protection. In: 1st rhea workshop - RHEA Project - EU (2011)

Lee, W.S., Slaughter, D.C., Giles, D.K.: Robotic Weed Control System for Tomatoes. Precision Agriculture, 95–113 (1999)

Aerial Fleet in RHEA Project: A High Vantage Point Contributions to ROBOT 2013

Jaime del Cerro*, Antonio Barrientos, David Sanz, and João Valente

UPM- Robotics and Cybernet Research Group,
Centre for Automation and Robotics UPM-CSIC,
José Gutierrez Abascal 2, 28006 Madrid, España
{j.cerro,a.barrientos,d.sanz,joao.valente}@upm.es
http://www.robcib.etsii.upm.es

Abstract. The RHEA project is based on the cooperation among aerial and ground vehicles aiming at performing several precision agriculture tasks. This work is focused on the description of the aerial mission in the mentioned project. The work includes a description of the units, the goals and mission types, the requirements involved in the mission planning as well as their supervision and monitoring. The missions are performed by using a fleet of last-generation hex-rotors that rely on high payload and extraordinary stability. These features allow taking steady pictures with high quality cameras in large extension fields. As result, high-resolution images of the field to cover are obtained in order to provide with weeds pots positions to ground units for their removal.

Keywords: Precision agriculture, Unmanned Aerial Vehicles, Coverage Plath Planning, RHEA.

1 Introduction

The Robot Fleets for Highly Effective Agriculture and Forestry Management (RHEA) project [3] is being carried out under the 7th framework program, identified as NMP-CP-IP 245986-2.

RHEA is focused on the design, development, and testing of a new generation of robotic systems for both chemical and physical (mechanical and thermal) effective weed management. The project is focused on both agriculture and forestry. It covers a large variety of European products, including agriculture wide row crops (processing tomato, maize, strawberry, sunflower and cotton), close row crops (winter wheat and winter barley) and forestry woody perennials (walnut trees, almond trees, olive groves and multipurpose open woodland).

RHEA aims at reducing the use of agricultural chemical inputs in a 75%, improving crop quality, health and safety for humans, and reducing production

* Corresponding author.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_33,

costs by means of sustainable crop management using a fleet of small, heterogeneous robots –ground and aerial– equipped with advanced sensors, enhanced end-effectors and improved decision control algorithms. RHEA can be considered as a cooperative robotic system, falling within an emerging area of research and technology with a large number of applications.

As previously mentioned, the key elements of the project are both the ground and the aerial fleets. Currently, ground fleet is made up of three small tractors endowed with different systems so as to perform singular treatments (mechanical, chemical and thermal treatments). On the other hand, the aerial fleet consists of two hex-rotors that are in charge of carrying out aerial tasks.

The roll of the drones (aerial units) is focused on aerial photography. Therefore, they are required in order to take high resolution aerial pictures of the filed. The pictures taken from aerial units are used for creating geo-referred mosaicking.

Agricultural experts are in charge of obtaining the position of the weeds pods by using this mosaic. These positions are the inputs for the ground units mission planner. Moreover, considering the position of the targets (weeds) and the crop rows directions, the mission manager is in charge of determining the optimal path for the ground units in order to remove the weeds without damaging the crop.

All the process from obtaining digital images, mosaicking creation, weeds detection and ground vehicles path planning is performed in a Base Station computer, which is also in charge of sending the missions to tractors and the full mission monitoring.

The work presented is structured as follows. Next section is dedicated to description of the aerial units. Section 3 summarizes the definition of the aerial mission and the mission planner. Section 4 describes the full control structure as well as aerial mission monitoring. Finally, some conclusions are presented.

2 Aerial Vehicles Description

The aerial vehicles that are in charge of performing aerial missions in the RHEA projects are the AR200 units, designed by AirRobot company [1].

The six-rotors aerial units have sensor-payload up to 1.5 kg, big reach and long flying time (about 40 minutes). They rely on digital data transmission and control on a real-time basis. An important feature of the drones is their innovative folding system that allows a space-saving and comfortable transport to the field. Moreover, using six-rotors allows certain redundancy, reducing in this manner the accident risk in case of fail in one motor.

Two cameras (visible and near infrared spectrum) are shipped on the drone, mounted on a gimbal system that reduces vibrations and allows the cameras point down when the drones perform steady flights (i.e., hovering in a way point).

Drones allow way-point programming as well as some payload control commands. Thus, the main part of the program includes a list of ordered way-points where the drone has to take a picture. Drones are able to fly this programs in

Fig. 1. RHEA aerial fleet in CAR-CSIC facilities (Arganda del Rey, Madrid)

fully autonomous mode. Nevertheless, take off and landing are performed in a supervised mode. This means that after loading the mission into the drones, a supervisor pilot (required by the emerging UAV regulation) is in charge of commanding the drone during taking off maneuver by sending altitude commands. It is important to point out that a low level attitude control in charge of its stabilization is always running.

When the drone has reached some meters high, the drone operator is able to switch the system to fully-autonomous navigation mode, and then, the taking-pictures mission starts.

After finishing the missions, drones return to their home points and then, the pilot takes again the control so as to send the orders for descending in a safe way. Notice that drones do not rely on perception sensors in order to detect if landing point is free or occupied. Moreover, considering the small free distance from the cameras to the bottom base line of the drones, it is important to ensure that no stones are in the landing place.

Drones provide with telemetry information during the flight, including information required for supervision such as position estimation and battery level among other.

Cameras used in RHEA require a long time for image saving into the memory card. Thus, a minimum time is required between shootings, and therefore between consecutive way points, in order to ensure that all the pictures are correctly stored. After finishing the flights, operator has to remove the cameras cards in order to start with the mosaicking process in the base station computer.

Fig. 2. RHEA drone in flight with a detail of the cameras mounting

3 Aerial Mission Definition and Planning

The first step in mission definition has be performed by using the GUI (Graphical User Interface) provided by the base station computer. Thus, the user has to define the field of interest by using a polyline with the aid of graphical tool. Each point of the polyline contains their three UTM or Universal Transversal de Mercator coordinates. This polyline has to delimit an unshaped closed polygon.

Moreover, taking into consideration some safety parameters such as proximity of roads or forbidden areas to fly, two additional polylines have to be defined by the user so as to be consider them as safety borders. The inner border will be considered as the area that it is possible to fly in order to take pictures. Obviously, the field definition polyline has to be inside it. Nevertheless, the outside border, defines an area where drones are not allowed to fly in any way.

Taking into consideration that two drones will perform their task simultaneously, the next step in the process should be the area repartition. Several approaches have ben developed by the research group regarding area repartition that take into consideration different criteria, such as different endurance, battery level or just different features [4]. Nevertheless, the main criteria considered in the RHEA project is the safety. Thus, the goal is to maintain the drones as far as possible from each other during the mission. This issue has been crucial in the determination of the algorithm used for the mission planning. Therefore, the area repartition will be detailed at the end of this section.

On the other hand, nearly all the requirements of the mission arise from the spatial resolution required by the experts in detection for the final mosaic (i.e., 1 or 2 pixels/cm) and the camera to be used (Sigma DP2 Merrill Compact Digital Camera). Therefore, taking into consideration the resolution of the cameras (4800x3200 pixels), each image has a maximum size of 48x32 meters. Considering the focal length (30mm) and the sensor size (23.5 x 15.7 mm), heights H close to 60 meters are obtained by applying the following expresion to both sensor dimensions.

$$H = FocalLenght \frac{RealSize}{SensorSize} \tag{1}$$

In addition to this, in order to ensure that small errors in stabilization (i.e., attitude values different that zero for Roll and Pitch angles during the camera

shooting), errors in position estimation (mainly due to the use of single GPS instead of differential one) or errors in position control (i.e., difference between actual and goal position) do not affect to the mosaicking process, a typical value of 60% for overlapping between consecutive images has been stablished. Therefore, every point in the field is ensured to be in two consecutive images. As a result, the distance between waypoints should be 19 meters when 1 pixel/cm of spacial resolution is required.

Considering all these restrictions, a matrix of waypoints (i.e., points separated nineteen meters where the drone has to take a picture) is projected on the field. The selection of the main direction of the field, which is the main direction considered for performing this projection, is the longest segment of the polyline that defines the field. This direction is used for determining the yaw angle of the drones (that is the same that the cameras) so as to optimize the mosaicking process.

Since it has been stablished as a requirement for the planner that no shift between adjacent cells should happen (apart from the performance errors previously mentioned), no displacement between columns is considered in order to adapt the matrix to the shape of the field.

After projecting the complete waypoint matrix, the next step is to determine which of them are required (i.e., its image includes a part of the field) or not. It is important to highlight that if the field has not a rectangular shape, the matrix will be incomplete (some of the projected way points will be no longer required to fly over). Therefore, a complex algorithm has been implemented in order to decide whether the image taken from each point falls inside the field.

Once all the required waypoints have been obtained, a path for each aerial unit in charge of performing the coverage has to be designed by the planner. Therefore, the objective for the planner is to create optimal paths. This obviously involves criteria such as not repeat the same waypoint or minimizing the total mission time, that considering a fix number of waypoints means to minimize the path length.

Several techniques have been tested in order to design a optimal path for a fleet of drones in charge of performing aerial coverage [6] [5] during the project. Nevertheless these solutions have some shortcomings, mainly regarding their processing time and the difficulty to implement time dependent safety criteria in their cost functions. For these reasons, a basic solution based on back and forth motions has been finally selected. The proposed solution has very low computational cost, performing the planning in real time without the necessity of long optimization process, that became high time consuming.

Moreover, the structure of the solution obtained by using this type of algorithm allows easily splitting a global path into "n" parts, being "n" the number of aerial robots available for the mission. Neverheless, as previously mentioned, the main reason that led to this election was the safety.

The number of waypoints assigned to each drone could be modified depending on different factors, thus the initial and final distances to be travelled by the drones can be easily compensated by assigning less waypoints to the drone that

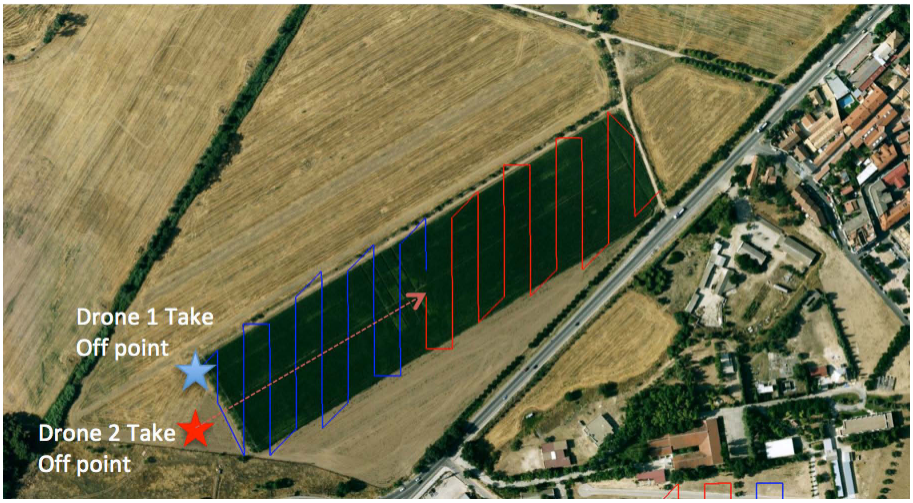


Fig. 3. Example of plan for two drones over maize field in CAR-CSIC facilities (Arganda del Rey, Madrid). No heading optimization applied.

should travel farther. Another safety criteria is to assign only full columns in the matrix to drones, thus two drones will never flight along the same column simultaneoulsy. Finally, some additional weighting procedure can be applied if some different among the drones exists. (e.g., different level of battery).

The planner has been coded by using C++ language. It is designed allowing multi-platform compilation. Nevertheless, it currently runs under windows 7 operative system that is the OS selected for all the ground station development.

4 Mission Execution and Monitoring

A overall structure and sequence for aerial mission managing is provided in Fig. 5. It also includes the main software blocks or modules that are involved in such process. This section is focused on the description of these modules, from the functionality point of view.

4.1 Graphical User Interface

The first block that should be invoked is the GUI or Graphical User Interface. It is been developed by Cyberbotics Ltd, by using Webots 7, a mobile robot simulator [2]. GUI not only has basic functionality such as mission definition and telemetry monitoring but also simulation capabilities. Thus, after defining the filed of interest by using polylines and geographical information systems tools, a mission definition file is created. This file also contains information regarding safety issues such as flight area and forbidden areas, according to user criteria. Moreover, cameras and resolution request parameters are also considered in mission definition. Cameras and hardware parameters are stored in a Data Base.

Fig. 4. Picture of the corner in maize field at ICA-CSIC facilities in Arganda del Rey (Madrid), 50 meters high

Once the mission is defined, the user is able to invoke the aerial planner in order to obtain the trajectories for the areal fleet according to the description provided in section 3. In this step, a mission plan file is created. This plan that includes path for one or two drones (according to user or mission preferences) should be displayed by GUI so as to allow the user to validate it. This validation is performed by the user in order to check if the full area is covered by the images and verify the overlapping stablished. In order to perform this visualization, the GUI should take into account the parameters of the cameras and the height of each way point.

After this validation, the mission plan file that has a generic format, which does not depend of the type of drone to be used, is coded according to specific Airrobot protocols. Thus, after this coding phase, one or two flight plans are created (depending on the number of drones to be used in the mission). This coding step requires the transformation between UTM coordinates that are used by all the RHEA system to terrestrial coordinates (latitude-longitude) used by drones. Moreover, a conversion to binary format that is used by the aerial units protocol in order to optimize the available bandwidth is also performed during this conversion.

As previously mentioned, the GUI includes an aerial drone simulator. The simulator is able to read the flight plan files in order to show on the screen of the GUI a 3D graphical representation of the drones performing the flight plans, taking into account distances and time between waypoints. This simulation is

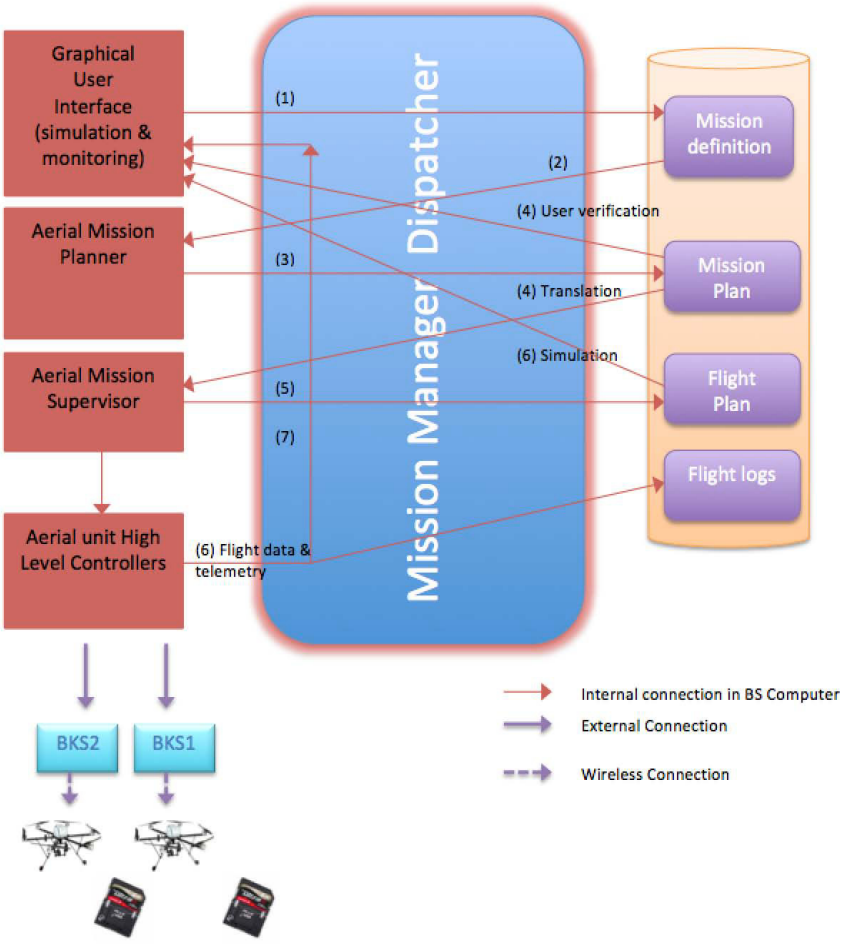


Fig. 5. Aerial Mission Managing Sequence

performed by using webots 7 simulator by Cyberbotics. This simulation also includes parameters of the cameras so as to allow visualizing their field of view as Fig. 6 shows.

4.2 Aerial Mission Supervisor

If the base station operator considers the simulation results as successful, the Aerial Mission Supervisor is in charge of downloading the flight plans to the drones trough their Aerial Unit High Level Controllers, as next section describes. Currently, both Aerial Mission Supervisor and the two Aerial Unit High Level Controllers are running in a separate laptop under windows 7 professional

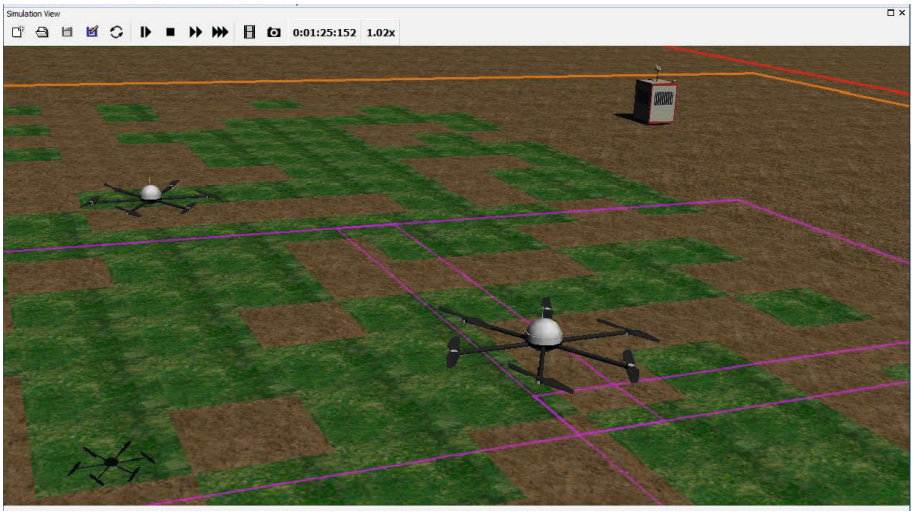


Fig. 6. Simulations of aerial mission by using Webots 7. Courtesy Cyberbotics.Ltd. Cameras field of view projection on the field corresponding to waypoints is shown in purple.

operative system. It is expected that in a near future, these modules will run into base station computer if its performance is good enough.

This module is in charge of analyzing the state of the vehicles and the mission. Thus, individual alarms from single vehicles are identified, processed and reported to user by using the GUI. It also controls the mission progress, taking into account some parameters and timeout previously stablished (e.g.., maximum time to reach a way point, or maximum time to complete de mission). Some warning levels have been also defined for the battery level.

On the other hand, the aerial mission supervisor receives the control mission commands from the GUI in order to start, pause, resume or abort the mission. Starting the mission only involves sending the full mission and parameters to drones so as to be able to take off. After loading the mission into the drones, the pilot has to send climbing commands to the drone by using the remote control in order to take off. Once the drone has reached certain high (stablished in the operational protocol of the project), the pilot is able to switch to fully autonomous navigation mode in order to start the flight to the first waypoint. The different stages during the missions are monitored trough the GUI in the base station.

Nevertheless, stopping the mission only requires that the drones have to remain in hovering maintaining their current positions, waiting for another command. If resume command is received in this state, the drone will continue the mission, going to the next waypoint in the list. On the other hand, aborting the mission not only requires to stop the flight plan immediately but also to return to home point. Anyway, the pilot is able to take the control of the drone and commanding velocity commands at any time just by using the manual controller.

It is also worth noting that drones do not rely on collision detection systems. Therefore, Aerial Mission Supervisor has the responsibility of commanding them in a safe way.

Due to security reasons, in some circumstances (i.e. when the battery level is under a certain threshold), the drone returns home autonomously with no previous control command. Moreover, if the security border list is breached, (e.g. due to wind guts) the drone will perform an emergency landing over that position with no intervention of the aerial mission supervisor.

4.3 Aerial Unit High Level Controller

The Aerial Unit High Level Controller (AUHLC) is a module that is in charge of low level commanding and communications with the drones, by using Airrobot protocols. It has been coded in C++ under windows 7 OS. A tread is created for managing each drone in a separate way.

The AUHLC handles the conversions between UTM coordinates from planner and Geographical (longitude-latitude) for de aerial units, the height references (relatives instead of absolutes) and the format of the communication packets.

Nevertheless, its main task is to convert the high level commands from supervisor (stop, pause, resume) to the adequate low level orders chain for the drones. Moreover is in charge of processing single UAV state information such as position, velocity, battery level, rotors speed, etc., and modulate it in order to inform to aerial mission manager according to the frequency required for performing high level mission supervision.

From the physical point of view, the AUHLC communicates to drones through a special hardware device developed by AirRobot, denoted as BKS shown in Fig. 7. There are two BKSs, one for each drone. Each BKS is connected to the AUHLC computer by using a USB port. BKS includes digital links for communicating both data and video streaming packets. The video (obtained from the cameras on board) is shown on the screen of the manual controller in real time to the drone operator . Additionally, it is possible to show in Base station computer by using the frame grabber output of the BKS.

On the other hand, the remote control required for supervised flight phases (take off and landing) as well as a safety controller, is also connected to the BKS, in order to transmit the manual commands to the drone (attitude and thrust commands).

4.4 Mission Dispacher

According to shown in Fig. 5, all the communications inside the RHEA system that involves some module in the base station are controlled by another module denoted Mission Dispatcher. It is in charge of performing the high level mission supervision by using a state machine implementation.

Fig. 7. BKS3 and manual controller (in black) for AirRobot drones

5 Conclusions

In this paper, a brief description of the roll of the aerial robots in the project RHEA has been presented. A description of the goal of their missions as well as the main requirements has been introduced. The work has been focused in the control and managing structure for aerial vehicles, from mission definition to monitoring and supervision, describing all the modules involved in this complex task.

Currently, the project is in the integration stage. Several flight plans have been obtained by the planer. Successful flights have been performed by using only one drone simultaneously. Neverthless, the complete control structure, mission definition and commanding have been tested. Final verification tests including two drones simultaneously will be performed on the last few days of August, 2013.

Acknowledgments. This work has been supported by the Robotics and Cybernetics Research Group at Technique University of Madrid (Spain), and funded under the project Robot Fleets for Highly Effective Agriculture and Forestry Management, (RHEA) of the European Unions Seventh Framework Programme [FP7/2007-2013] under Grant Agreement n 245986 in the Theme NMP-2009-3.4-1 (Automation and robotics for sustainable crop and forestry management)

Authors would like to thank all the RHEA partners, specially Cyberbotics Ltd and AirRobot for their collaboration in the project.

References

1. AIR ROBOT Website, `http://www.airrobot.de`
2. cyberbotics Website, `http://www.cyberbotics.com`
3. RHEA project Website, `http://www.rhea-project.eu`
4. Barrientos, A., Colorado, J., Cerro, J., Martinez, A., Rossi, C., Sanz, D., Valente, J.: Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. Journal of Field Robotics 28(5), 667–689 (2011)
5. Valente, J., Barrientos, A., Del Cerro, J.: Optimization of Aerial Surveys using an Algorithm Inspired in Musicians Improvisation. In: 23rd International Conference on Automated Planning and Scheduling, Rome (2013)
6. Valente, J., Sanz, D., Del Cerro, J., Barrientos, A., De Frutos, M.A.: Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields. Precision Agriculture 14(1), 115–132 (2013)

A New Combined Strategy for Discrimination between Types of Weed

P. Javier Herrera[1], José Dorado[2], and Ángela Ribeiro[1]

[1] Centre for Automation and Robotics, CSIC-UPM, 28500 Madrid, Spain
[2] Institute of Agricultural Sciences, CSIC, 28006 Madrid, Spain
{pj.herrera,angela.ribeiro}@csic.es, jose.dorado@ica.csic.es

Abstract. Specific weed management consists on adjusting herbicide treatments depending on the zone infested and the type of weed. In this context, the discrimination between grasses (monocots) and broad-leaved weeds (dicots) is an important objective mainly because the two weed groups can be appropriately controlled by different specific herbicides. This work proposes a method of discrimination between these types of weeds based on a combined strategy, the Sugeno Fuzzy Integral, where the final decision is taken by combining seven attributes, the Hu moments. The main challenge in terms of image analysis is to achieve an appropriate discrimination between both groups in outdoor field images under varying conditions of lighting as well as of soil background texture.

Keywords: Precision Agriculture, weed discrimination, monocots/dicots discrimination, Sugeno Fuzzy Integral, colour segmentation, Hu moments.

1 Introduction

Precision Agriculture is intended to adjust the use of resources and agronomic practices to the requirements of soil and crop seeking greater sustainability and efficiency. In other words, aims to reduce economic and environmental costs traditionally associated with agriculture.

The application of herbicides for agricultural purposes has been realised for decades with relevant gains in crop production. However, excessive or careless use of certain herbicides can have detrimental effects on the surface water and ground water quality. There is no doubt that the use of selective herbicide application on weed-infested areas of the field, rather than the entire field, would be of great interest [1]. As a result, efforts are being encouraged by concern over reducing agricultural chemical use without sacrificing crop yield.

Several authors show that the distribution of the most harmful weeds for a particular crop is not uniform, and it generally (about a 70% of the fields) affects less than 40% of the crop [2, 3]. Due to this fact, the selective application of herbicides only on infested areas can lead to significant reductions in the amount of product applied and provide both economic and ecological benefits [4 - 6]. From the point of

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_34,

view of herbicide treatment, efficiency is higher if selective treatment is performed for each type of weed instead of using a wide spectrum herbicide [7].

To adapt the treatment to the individual needs of each kind of land, it is essential to have accurate information on the state of the crop to be treated, i.e. where the weeds are located, the degree of coverage or type of infestation present. Traditionally, different methods have been used to obtain information of the field from the air and the ground. In the case of aerial images and data coming from satellites, information collection depends heavily on the weather (no clouds or fog) and, although remote sensing in agriculture has experienced a resurgence in recent years thanks to the use of cameras hyper and multi-spectral [8], yet the cost is high and the resolution is low. Ground sampling allows a lower resolution (centimetre) although in this case the information contained in each image covers small crop areas.

The development of methods of detection of weeds from images has always been an open field of great importance to Precision Agriculture [9 - 13]. The problem has no simple solution owing to the great diversity of crops and weeds, changes in exterior lighting, differences in the texture of the terrain (fundamentally due to humidity), different states of crop growth and infestation, great similarities between the crop and weeds that infest it, etc. [14, 15]. All this makes the discrimination between the crop, weed and soil a complex task, and the difficulty increases if the aim is to discriminate between groups of weeds or to apply herbicide in real time as the position of the infestation is detected [4, 16 - 20].

One of the key criteria for herbicide selectivity is based on differences between monocots and dicots. For this reason, the determination of the coverage percentages of both groups is essential to the development of an autonomous system of treatment able to adjust the type of herbicide and the dose to the dominant infestation.

The literature contains several examples of works that propose methods of image processing that combine colour, position, outline, texture, size, or spectrum to distinguish between weeds and crops [21, 22]. Other works present combinations of image processing techniques as Fuzzy Clustering and a fuzzy inference neural network to identify plants, based on leaves [23].

The success with which these aspects can be adapted to classification depends on the kind of crop, weeds, manner and moment that the images are gathered. In other words, early weed detection is an objective that can be planned according to criteria oriented to two different levels with an increasing requirement: 1) estimation of presence or absence of weeds by discrimination from bare soil and row crops, and 2) differentiation between groups of weeds e.g., monocots vs. dicots (since they can be controlled by different herbicides) according to one or several differential parameters (spectral characteristics, and other features as size or form). Previous works have faced this problem by means of techniques as neural networks or genetic algorithms [21, 24].

Changing the subject, moment invariants were firstly introduced to the pattern recognition community in 1962 by Hu [25], who employed the results of the theory of algebraic invariants and derived his seven famous invariants to rotation of 2-D objects. Since that time, numerous works have been devoted to various improvements and generalizations of Hu's invariants and also to its use in many application areas.

Invariance with respect to translation, rotation and scaling is required in almost all practical applications, because the object should be correctly recognized, regardless of its position and orientation in the scene and of the object-to-camera distance [26, 27].

Weed spatial distributions are unique, with monocot infestations more patchy than dicots [3]. Besides, monocots differ architecturally from dicots as one can see in Fig. 1. For these reasons, a strategy based on using Hu moments in shape recognition may be suitable. Moreover, these moments have been used successfully in previous works solving different problems but where images present similar features (outdoor images, changes in exterior lighting, varying conditions of lighting, differences in the textures, overlapping, etc) [28].

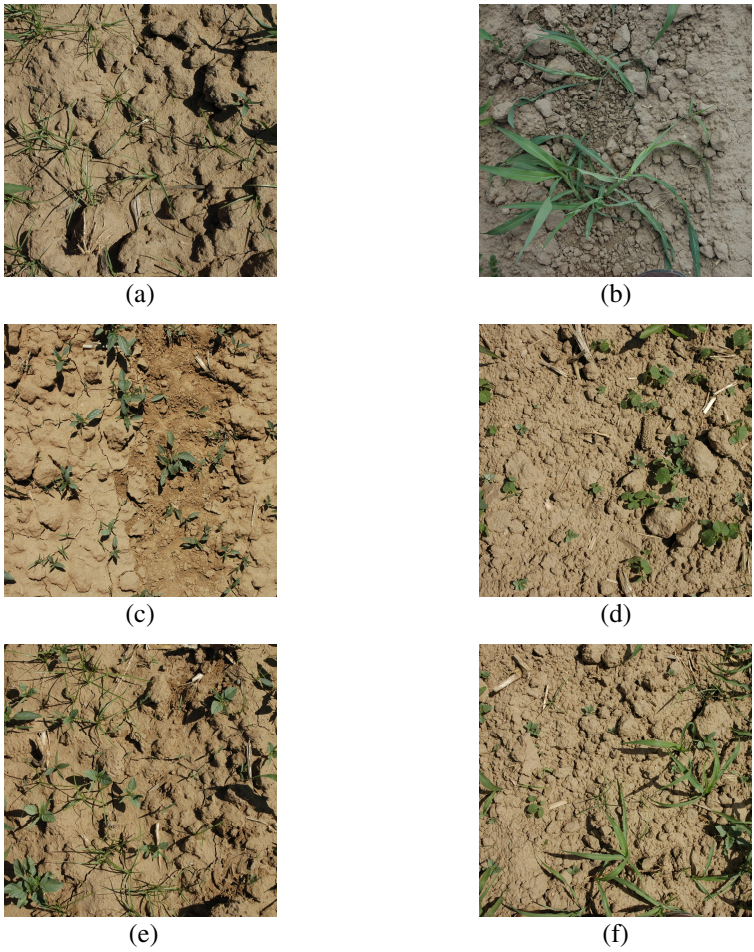


Fig. 1. Images (a) and (b) show monocots (long and slender leaf), whereas images (c) and (d) present dicots (broadleaf and short). Images (e) and (f) display a mixture of both kinds of weeds.

This work constitutes a new approach to discrimination among groups of weeds. In particular, this work presents a novel method of discrimination between monocots and dicots, based on the support that each region belonging to weed receives by combining attributes. Our hypothesis is that a region belonging to a weed species can be characterized by a set of seven attributes based on the seven invariant moments of Hu. Furthermore, the Sugeno Fuzzy Integral (SFI) approach is used for the combination, so the decision about the kind of weed for a region, can be made according to the support that each region receives by combining the attributes by means of the SFI.

SFI has been reported to give excellent results as a classifier combiner [29]. Moreover, based on the conclusions reported in [28, 30 - 32], the SFI appears as a suitable method for the combination of attributes. In fact, with a little adjusting it can be used for combining attributes in this proposal, so that a decision about a unique kind of weed (monocot or dicot) can be made for each region.

Summarizing, the combined SFI strategy makes possible an automatic way to distinguish between different kinds of weed in outdoors images. The final purpose is to estimate the coverage percentages of each type of weed in the image.

This work is organized as follows. Section 2 describes the proposed approach, including a brief overview of the SFI. Section 3 describes the results obtained by using the combined approach. Section 4 presents the conclusions and future work.

2 Proposed Approach

An essential issue in the field of pattern analysis is the recognition of objects and characteristics of these objects regardless of their position, size and orientation. The idea of using moments in shape recognition gained relevance when Hu [25], derived a set of invariants using algebraic invariants. In particular, Hu defined seven values, computed by normalizing central moments through order three that were invariant to object scale, position, and orientation.

Based on previous invariant moments the novel proposed approach consists of four stages: 1) image segmentation of vegetation cover and regions definition, 2) labelling of disconnected regions, 3) extraction of the Hu invariant moments for each region, and 4) discrimination of both monocots and dicots regions by means of the SFI.

The segmentation of the image is a two-steps process. First, it applies the equation (1) to each pixel of the original image with the aim of isolating the vegetation cover [33].

$$IS = r \cdot I(R) + g \cdot I(G) + b \cdot I(B) \tag{1}$$

where $r = -0.884$, $g = 1.262$, $b = -0.311$. The greyscale image resulting is then binarized by using a threshold that was set to 10 in this case (Fig. 2b).

After that, an opening morphologic operation is applied for enhancing the regions, avoiding the overlapping among regions belonging to different weed types, and for

removing in the image the pixels belonging to noise with the minimum alteration of those belonging to monocots and dicots.

The opening operation is accomplished with a structural element that symmetrically operates in all spatial directions, i.e. a 5×5 matrix known as *diamond*.

(a) (b)

Fig. 2. (a) Image with mixed monocots and dicots. (b) Segmentation of the vegetation cover of image (a).

In the second stage, the regions are labelled following the procedure described in [34], which basically finds the connected components (regions in this case) in a binary image. In this method, all the pixels in the same region are given the same level. The searching of the connected components is done in top-to-bottom scan order, i.e. all pixels in the first connected component are labelled as 1, those in the second as 2 and so on.

Once all regions have been labelled, the seven Hu invariant moments are computed for each region. Therefore, the attributes are the seven mentioned above, i.e. $\Omega \equiv \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7\}$, where $\phi_i \in [0,1]$ and it is associated to the seven Hu moments.

Then each region is matched with its class of weed. This is a decision that can be made through the combined SFI method, such as it is briefly described below.

The SFI requires the computation of the relevance assigned for each attribute, from which the so-called *fuzzy densities* can be computed. This is solved by computing the $\lambda - fuzzy$ measure [29]. In the proposed approach the calculation starts with selecting a set of seven fuzzy measures, that we will call g^1, g^2, g^3, g^4, g^5, g^6, g^7 according to [29], each one representing the individual relevance (strength or competence) of the associated attribute in Ω.

The value of λ needed for calculating g^i is obtained as the unique real root greater than -1 of the polynomial,

$$\lambda + 1 = \prod_{i \in \Omega} \left(1 + \lambda g^i\right), \quad \lambda \neq 0 \tag{2}$$

As it was previously mentioned, SFI is suitable for combining classifiers. Nevertheless, it requires a previous training stage to adjust some parameters. In that approach, SFI learns the relevance for each classifier, so that, during the combination,

every classifier intervenes with a specific different weight on the final decision. In our combined SFI approach, it is also computed the relevance of each attribute for determining its specific contribution to the decision through the fuzzy densities. As in [35], although with a different criterion, the relevance of each attribute is determined by considering a number of reliable true and false training examples obtained from a set of different regions. The process is as follows: for each region in an image, it is computed the grade of support for its class (monocot or dicot), but considering each one of the seven attributes separately. So, we compute the averaged percentage of error, p_1,... p_7, for the selected regions and for each attribute, based on the expert criterion. Thus, the relevance for an attribute i is computed by (3),

$$g^i = p_i \Big/ \sum_{j=1}^{7} p_j \tag{3}$$

Once the $g^1, \ldots g^7$ are obtained and λ is found, the SFI works as follows:

1. For a given region, it is obtained a vector as: $\left[\phi'_1, \phi'_2, \phi'_3, \phi'_4, \phi'_5, \phi'_6, \phi'_7\right]^T$; without lost of generality assume that ϕ'_1 is the highest value and ϕ'_7 the lowest. In this way, this vector is arranged under this criterion, i.e. $\phi'_1 > \phi'_2 > \phi'_3 > \phi'_4 > \phi'_5 > \phi'_6 > \phi'_7$.
2. Arrange the fuzzy densities correspondingly with the mentioned arrangement, i.e. $g^1, g^2, g^3, g^4, g^5, g^6, g^7$ and set the first fuzzy density $g(1) = g^1$.
3. For $t = 2$ to 7, $g(t)$ is calculated recursively by (4),

$$g(t) = g^i + g(t-1) + \lambda g^i g(t-1) \tag{4}$$

4. Calculate for each candidate region i, the final degree of support to be matched with each class l as,

$$\mu_i(l) = \max_{h \in \Omega} \{\min\{\phi_h, g(h)\}\} \tag{5}$$

5. The decision about the class a region belongs is made by selecting the maximum support $\mu_i(l)$ among all classes.

3 Results

The images used in this work were taken in maize crops sited in Madrid (Spain) on different days and therefore under varying lighting conditions. A conventional camera positioned atop a tripod was used and the images were acquired in vertical from heights below 1.5 meters, from a space of 50×50cm between two maize rows, with an original resolution of 1700×1700 pixels. Therefore the vegetation that appears in the images is always related to weeds, i.e. monocots, dicots or both kinds of weeds. In

short, given an RGB image the aim of the developed approach is to estimate in the image the cover percentages of each weed group as well as of soil.

From the sixty-six images available, twenty-eight presented a mixture of weeds. Nineteen images presented only monocots and nineteen only dicots. An important infestation was observed in the 14% of the images. In this work, fourteen images were selected because represented a wide range of situations. After applying steps 1 (vegetation cover segmentation) and 2 (labelling of disconnected regions) in the selected set of images, one-hundred different regions were extracted and analyzed. In general, the number of regions extracted per image ranged from five to twenty. In the cases where an important infestation was observed, it could be extracted less than five different regions.

Fig. 3 displays, as an example, the regions extracted by the application of steps 1 and 2 over the original image represented in Fig. 2a. Each region appears labelled with a unique label, represented as a colour in a scale for visualization purposes.

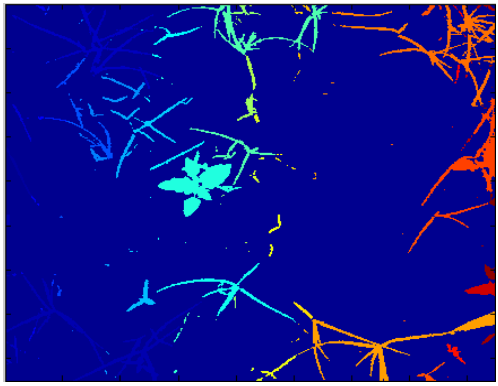


Fig. 3. Labeling regions. Each isolate region appears identified by a unique color.

The tests corresponding to the SFI strategy have been carried out with fourteen images including one-hundred different regions belonging to monocots and dicots. We use four of them (twenty-eight regions) for computing the relevance of each attribute for SFI, from which the fuzzy densities can be obtained. According to the explanation in section 2, the averaged percentage of error, $p_1,... p_7$, are: $p_1 = 8$ (ϕ_1), $p_2 = 10$ (ϕ_2), $p_3 = 15$ (ϕ_3), $p_4 = 25$ (ϕ_4), $p_5 = 21$ (ϕ_5) , $p_6 = 23$ (ϕ_6) and $p_7 = 19$ (ϕ_7). Based on (3), the fuzzy values are exactly the following: $g^1 = 0.066$, $g^2 = 0.083$, $g^3 = 0.124$, $g^4 = 0.207$, $g^5 = 0.173$, $g^6 = 0.190$ and $g^7 = 0.157$. As one can see, the most relevant attribute is the first one.

At a second stage, we apply the SFI approach region by region for the remainder ten images, as described in section 2.

Furthermore, it is available the information of class membership provided by the expert criterion. Thus, for each region in an image it is known its correct class according to the expert knowledge and this information can be used to compute the percentage of error of the proposed approach. For each one of the seventy-two regions

obtained from the remainder ten images used for testing, the error for each region and the average of these errors are computed.

According to the Hu moments extracted for each region, some preliminary conclusions can be obtained for each kind of weed. In monocot weeds, Hu moments tend to have negative values in the fifth, sixth and seventh moment and in the case of moments that give a positive value, they are close to 1 or 2. In dicot weeds, the values in the seven moments are close to zero and never reach 1. They may have negative values in the fifth, sixth and seventh moment, but these values are very close to zero. These assertions can be confirmed in Table 1, which shows the Hu invariant moments obtained from fifteen regions of each kind. These regions were randomly extracted from the images used in this work, belonging to the two types of weed under study, Fig. 4.

Table 1. Invariant moments of Hu obtained from fifteen regions of each kind

Weed	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
Monocot	1.7535	2.0493	2.2928	0.8794	1.1990	1.2571	−0.3485
	1.3380	1.3438	2.0149	1.3801	2.3014	1.5997	0.0134
	1.6944	0.9968	2.1177	0.1024	0.0359	−0.0080	−0.0314
	1.6345	1.0468	2.1471	0.4223	−0.3960	−0.4321	0.0702
	1.5036	0.9277	1.0881	0.3654	0.1670	0.2495	−0.1587
	1.0571	0.3035	1.2304	0.0342	−0.0023	−0.0046	0.0066
	1.2608	1.2070	0.2663	0.2907	0.0808	0.3193	−0.0043
	1.1625	0.5494	0.7047	0.0474	−0.0016	−0.0225	−0.0085
	1.7169	2.2861	2.0890	0.2256	−0.1469	−0.3326	0.0492
	1.4536	1.1760	2.0314	0.4380	0.3875	0.3890	−0.1433
	1.6496	2.5818	0.6256	0.4177	0.2127	0.6468	−0.0188
	1.1978	0.3617	1.4753	0.1781	−0.0877	−0.1067	0.0253
	1.6589	1.8999	2.2096	0.7830	1.0055	0.8419	−0.2225
	1.1070	0.2534	0.5144	0.0242	0.0013	0.0121	−0.0024
	1.3074	1.0152	1.6444	0.7993	0.8766	0.8044	−0.2670
Dicot	0.2982	0.0424	0.0030	247e−9	−41e−12	−29e−8	−21e−10
	0.2595	0.0189	0.0015	105e−9	66e−12	13e−10	−42e−11
	0.2605	0.0154	0.0078	0.0004	6e−7	42e−6	52e−9
	0.2799	0.0071	0.0177	0.0015	73e−7	12e−5	16e−7
	0.4753	0.1139	0.0585	0.0108	0.0003	0.0036	−11e−7
	0.2690	0.0382	0.0004	0.0002	43e−9	27e−6	69e−11
	0.2343	0.0032	0.0007	0.0001	25e−9	31e−7	−57e−10
	0.2971	0.0019	0.0261	0.0005	19e−7	99e−7	−57e−9
	0.1981	0.0056	0.0002	59e−7	19e−11	37e−8	6e−11
	0.1966	0.0066	0.0009	54e−6	12e−9	43e−7	−3e−9
	0.1926	0.0030	76e−5	3e−6	1e−10	1e−7	1e−10
	0.2094	0.0053	0.0018	63e−6	−1e−9	5e−7	2e−8
	0.2008	0.0029	0.0019	99e−6	37e−9	33e−7	2e−8
	0.2108	0.0036	0.0041	16e−6	37e−10	9e−7	2e−10
	0.1991	0.0086	58e−5	3e−5	26e−10	1e−6	3e−9

These conclusions allow to justify the relevance of Hu moments in this proposal to discriminate between monocots and dicots.

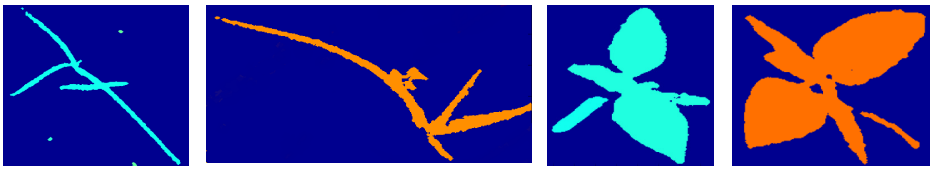


Fig. 4. (a),(b) Two regions belonging to monocots; (c),(d) two regions belonging to dicots

Through this initial analysis, a classifier based on the Hu moments was designed in order to discriminate between monocots and dicots as described in section 2. The average percentage of error (compared to the pre-classification done by an expert) obtained with SFI decision making approach is 14.9. The standard deviation (σ) obtained is 1.5. This means that the combination of attributes improve the results. The best individual criterion seems to be the first moment (ϕ_1). This implies that it is the most relevant attribute. This agrees with its relevance obtained above, as it has turned out to be the most relevant attribute.

4 Conclusions

In this paper we have proposed a strategy for discriminating between monocot and dicot. The method has proven effective and simple; furthermore it is based on colour segmentation, morphological operations and a well known strategy, common operations in image processing.

For each region in an image the seven Hu moments are obtained for determining its correspondence to monocot or dicot. Under the SFI method, the values among seven attributes are combined and a decision for choosing the unique class for each region is made.

The proposed combined strategy works properly when the weeds present an early stage of growth. This is the best moment for applying the herbicide. If the images present states of a higher crop growth, weeds will probably present overlapping and the segmentation process will get difficult mainly due to occlusions that do not allow to see the whole shape of each plant. Nevertheless, the proposed approach provides an useful methodology to discriminate between monocots and dicots in early growth stages.

Although the results achieved can be considered satisfactory, they could be improved by applying Machine Learning techniques to obtain other classifiers able to automate the classification, by means of some induced knowledge which implies the seven moments.

The proposed combined approach can be applied to any environment. In this context, site-specific weed management could mean a significant reduction in herbicide use, which saves the farmer money and benefits the environment. As it was described in the section 1, efficiency is higher if selective treatment is performed for each type of infection instead of using a wide spectrum herbicide. For this reason, this proposal can be essential in the future.

Acknowledgements. The Spanish Government and the European Union have provided full and continuing support for this research work through projects: PLAN NACIONAL - AGL2011 - 30442 - C02 – 02 (GroW) and UE - CP - IP245986 - 2 (RHEA). The authors wish to acknowledge to the Spanish National Research Council (CSIC) and the European Social Fund (ESF) for the JAE-Doc contract with the first author.

References

1. Thompson, J.F., Stafford, J.V., Miller, P.C.H.: Potential for automatic weed detection and selective herbicide application. Crop Production 10(4), 254–259 (1991)
2. Marshall, E.J.P.: Field-scale estimates of grass weed populations in arable land. Weed Research 28(3), 191–198 (1988)
3. Johnson, G.A., Mortensen, D.A., Martin, A.R.: A simulation of herbicide use based on weed spatial distribution. Weed Research 35(3), 197–205 (1995)
4. Tian, L., Reid, J.F., Hummel, J.W.: Development of a precision sprayer for site-specific weed management. Transactions of the American Society of Agricultural Engineers 42, 893–900 (1999)
5. Medlin, C.R., Shaw, D.R.: Economic comparison of broadcast and site-specific herbicide applications in nontransgenic and glyphosate-tolerant Glycine max. Weed Science 48(5), 653–661 (2000)
6. Timmermann, C., Gerhards, R., Kühbauch, W.: The economic impact of site-specific weed control. Precision Agriculture 4(3), 249–260 (2003)
7. Tang, L., Tian, L., Steward, B.L.: Classification of broadleaf and grass weeds using Gabor wavelets and an Artificial Neural Network. Transactions of the ASABE 46(4), 1247–1254 (2003)
8. López Granados, F., Jurado-Expósito, M., Atenciano, S., García-Ferrer, A., Sánchez de la Orden, M., García-Torres, L.: Spatial variability of agricultural soils in southern Spain. Plant and Soil 246, 97–105 (2002)
9. Onyango, C.M., Marchant, J.A.: Segmentation of row crop plants from weeds using colour and morphology. Computers and Electronics in Agriculture 39, 141–155 (2003)
10. Ribeiro, A., Fernández-Quintanilla, C., Barroso, J., García-Alegre, M.C.: Development of an image analysis system for estimation of weed. In: Stafford, J.V. (ed.) Proceedings 5th European Conf. On Precision Agriculture (5ECPA), pp. 169–174 (2005)
11. Tellaeche, A., Burgos-Artizzu, X., Pajares, G., Ribeiro, A., Fernández-Quintanilla, C.: A new vision-based approach to differential spraying in precision agriculture. Computers and Electronics in Agriculture 60(2), 144–155 (2008)
12. Tellaeche, A., Burgos-Artizzu, X.P., Pajares, G., Ribeiro, A.: A vision-based method for weeds identification through the Bayesian decision theory. Pattern Recognition 41, 521–530 (2008)
13. Burgos-Artizzu, X.P., Ribeiro, A., Tellaeche, A., Pajares, G., Fernández-Quintanilla, C.: Improving weed pressure assessment using digital images from an experience-based reasoning approach. Computers and Electronics in Agriculture 65, 176–185 (2009)
14. Tian, L.F., Slaughter, C.S.: Environmentally adaptive segmentation algorithm for outdoor image segmentation. Computers and Electronics in Agriculture 21, 153–168 (1998)

15. Brown, R.B., Noble, S.D.: Site-specific weed management: sensing requirements - what do we need to see? Weed Science 53, 252–258 (2005)
16. Lee, W.S., Slaughter, D.C., Giles, D.K.: Robotic weed control system for tomatoes. Precision Agriculture 1(1), 95–113 (1999)
17. Meyer, G.E., Mehta, T., Kocher, M.F., Mortensen, D.A., Samal, A.: Textural imaging and discriminant analysis for distinguishing weeds for spot spraying. Transactions of the ASABE 41(4), 1189–1197 (1998)
18. Ishak, A.J., Hussain, A., Mustafa, M.M.: Weed image classification using Gabor wavelet and gradient field distribution. Computers and Electronics in Agriculture 66, 53–61 (2009)
19. Hemming, J., Rath, T.: Precision agriculture: computer-vision-based weed identification under field conditions using controlled lighting. Journal of Agricultural Engineering Research 78(3), 233–243 (2001)
20. Burgos-Artizzu, X.P., Ribeiro, A., Guijarro, M., Pajares, G.: Real-time image processing for crop/weed discrimination in maize fields. Comput. Electron. Agr. 75, 337–346 (2011)
21. Burks, T.F., Shearer, S.A., Heath, J.R., Donohue, K.D.: Evaluation of Neural-network Classifiers for Weed Species Discrimination. Biosystems Engineering 91(3), 293–304 (2005)
22. Panneton, B., Guillaume, S., Samson, G., Roger, J.: Discrimination of Corn from Monocotyledonous Weeds with Ultraviolet (UV) Induced Fluorescence. Applied Spectroscopy 65(1), 10–19 (2011)
23. Camargo Neto, J., Meyer, G.E.: Crop species identification using machine vision of computer extracted individual leaves. In: Chen, Y.R., Meyer, G.E., Tu, S. (eds.) Optical Sensors and Sensing Systems for Natural Resources and Food Safety and Quality, Proc. SPIE, Bellingham WA, vol. 5996, pp. 64–74 (2005)
24. Sainz-Costa, N., Ribeiro, A., Andujar, D., Dorado, J.: Optimización evolutiva para la construcción de un método de estimación de porcentajes de cobertura de gramíneas y dicotiledóneas. In: Lozano, J.A., Gámez, J.A., Moreno Pérez, J.A. (eds.) Proceedings of the Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2011), vol. 1 (2011)
25. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Trans. Information Theory. 8, 179–187 (1962)
26. Mercimek, M., Gulez, K., Mumcu, T.K.: Real object recognition using moment invariants. Sadhana - Springer India 30(6), 765–775 (2005)
27. Flusser, J., Suk, T., Zitová, B.: Moments and Moment Invariants in Pattern Recognition. John Wiley & Sons, Ltd. (2009)
28. Herrera, P.J., Pajares, G., Guijarro, M., Ruz, J.J., Cruz, J.M., Montes, F.: A Featured-Based Strategy for Stereovision Matching in Sensors with Fish-Eye Lenses for Forest Environments. Sensors 9(12), 9468–9492 (2009)
29. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley (2004)
30. Klaus, A., Sormann, M., Karner, K.: Segmented-Based Stereo Matching Using Belief Propagation and Self-Adapting Dissimilarity Measure. In: Proc. of 18th Int. Conference on Pattern Recognition, vol. 3, pp. 15–18 (2006)
31. Herrera, P.J., Pajares, G., Guijarro, M., Ruz, J.J., Cruz, J.M.: A Stereovision Matching Strategy for Images Captured with Fish-Eye Lenses in Forest Environments. Sensors 11(2), 1756–1783 (2011)
32. Herrera, P.J., Pajares, G., Guijarro, M., Ruz, J.J., Cruz, J.M.: Segmentation and stereoscopic correspondence in images obtained with omnidirectional projection for forest environments. In: Torreao, J.R.A. (ed.) Stereo Vision, ch. 3, pp. 41–56. In-Tech (2011)

33. Burgos-Artizzu, X.P., Ribeiro, A., Tellaeche, A., Pajares, G., Fernández-Quintanilla, C.: Analysis of natural images processing for the extraction of agricultural elements. Image Vision Computing 28, 138–149 (2010)
34. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision, vol. I–II. Addison-Wesley, Reading (1992)
35. Herrera, P.J., Pajares, G., Guijarro, M., Ruz, J.J., De la Cruz, J.M.: Combination of attributes in stereovision matching for fish-eye lenses in forest analysis. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 277–287. Springer, Heidelberg (2009)

Generating Autonomous Behaviour for a Crop Inspection Robot

José M. Bengochea-Guevara, Jesús Conesa-Muñoz, and Ángela Ribeiro

Center for Automation and Robotics, CSIC-UPM, Arganda del Rey, 28500 Madrid, Spain
{jose.bengochea,jesus.conesa,angela.ribeiro}@csic.es

Abstract. This paper presents the main characteristics of a robot whose aim is to perform field inspection using autonomous navigation. The solution developed for crop row tracking is shown, which is a fundamental behaviour for crop inspection. For this purpose, an image processing method is implemented to determine the vehicle's relative position to the crop row in real time. This position is supplied to two fuzzy controllers, one for angular speed and the other for linear speed. To integrate crop row tracking and other skills that the robot needs, we propose generating the different behaviours of the robot using a network of nodes with different functions: perceptive nodes, cognitive nodes and actuator nodes. The actions of the robot emerge from this set of behaviours, depending on the goals and needs that must be met at each given moment in time.

Keywords: autonomous behaviour, crop inspection, nodes network, motivational constructs.

1 Introduction

Traditionally, farming practices have tended towards uniform management of the field while ignoring the spatial and temporal variability that appears in the crops. This presents basically two types of effect: a) air and soil pollution, with consequent pollution of groundwater and b) a marked increase in the economic cost of agricultural production.

Furthermore, in the next 25 years it will be necessary to double agricultural production, with increasingly less soil and water available, which leads us to a situation where technology is crucial to minimize production costs while performing safe management of the environment [1], [2].

In recent years, thanks to the development of technologies such as Global Positioning Systems (GPS), crop sensors, humidity and soil fertility sensors, multispectral sensors, remote sensing, Geographic Information Systems (GIS), and Decision Support Systems (DSS), the concept of Precision Agriculture (PA), which proposes farming management adapted to crop variability, has emerged.

PA techniques aimed at the selective treatment of weeds by applying herbicides only in infested crop areas, as opposed to the blanket methods used in traditional farming, are particularly important. Even the amount of treatment applied can vary, depending on the density and/or the type of weed.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_35,

The first step when performing a selective application is to estimate the herbicide needed for each crop unit [3]. This requires gathering data in the field to ascertain the distribution of weeds and the spatial variability of each species (Perception). With the information collected in the field, the next step is to determine the best action to apply to the crop (Decision Making). Finally, a set of field operations, consistent with the previous decisions, must be carried out to obtain the selective treatment of weeds (Action).

The perception stage is not required in conventional treatment where the same dose of herbicide is applied to the entire crop. Therefore, it is essential to automate this step as much as possible and make it as cheap as possible if the goal is economically viable Precision Agriculture. One technique that may be used to collect field information, autonomous vehicles equipped with onboard sensing elements (and of reduced dimensions so as to minimize impact on crop and soil compaction), displays considerable potential.

This paper presents a robot which is designed to conduct crop inspection using autonomous navigation. Based on ideas from the fields of etiology and cognitive psychology, we propose a network of nodes with different functions which, guided by the goals and needs of the robot, generates autonomous behaviour that allows the robot to inspect crops without human intervention.

2 Description

The robot used in this study is a commercial model (mBase-MR7 built by MoviRobotics). It has four wheels, with differential locomotion, has no steering wheel and can rotate on its vertical axis. The on-board camera used in this project is a Creative Live! Cam Notebook Ultra webcam, which has an 80° lens. The camera is placed at a distance of 80 cm from the soil and is connected to the computer inside the robot via a USB connector. The motherboard used is a Mini-ITX LV-67F with an Intel Core i7 Mobile Processor and 4 GB of DDR3 RAM.

In Fig.1a, the necessary equipment to conduct field inspection may also be seen, namely, a Panasonic Toughbook CF-19 laptop, a Canon EOS 7D camera and a Hemisphere R220 GPS receiver with RTK correction.

The typical layout of a crop field of maize may be seen in Fig. 1b. In order to inspect a crop row, the robot is positioned at either end of the latter so that the row is located between the two wheels. It begins to advance, tracking the crop row thanks to the onboard camera and continues until it reaches the far end of the row. Once it has covered one row, the robot performs the necessary manoeuvres to position itself at the head of the next row it wishes to inspect. The procedure is repeated until all the rows have been inspected.

For field inspection with full coverage, a set of individual behaviours is observed, such as: 1) the tracking of a crop row, 2) the detection of the end of a crop row, and 3) the correct positioning of the robot at the head of the next row. Additionally, the robot must also include behaviours such as 4) the detection and avoidance of obstacles to avoid collision and 5) monitoring the battery level, so that when it detects that the

power is about to run out, the robot can proceed to the nearest charging point. In order for the robot to travel to a given point along the edge of the crop field, such as a charging point or the starting point of an inspection task, additional abilities must be included so that it can guide itself with the help of the GPS from a point at the edge of the crop field (laterals and headers of the crop) to another point along the same edge.

(a) (b)

Fig. 1. (a) The mBase-MR7 robot covering a maize field. (b) A crop field of maize.

3 Crop Row Tracking Behaviour

An image processing method capable of extracting in real time the layout of the crop rows from images taken with a high-quality web camera situated at the front of the robot has been designed so that the robot is able to navigate visually guided by the crop rows. The aim of the image processing is to extract the vehicle's position in relation to the crop row (see Fig. 2), i.e. the motion direction angle (the crop rows are parallel to each other) and displacement (offset) between the robot centre and the closest point situated along the line that defines the crop row.

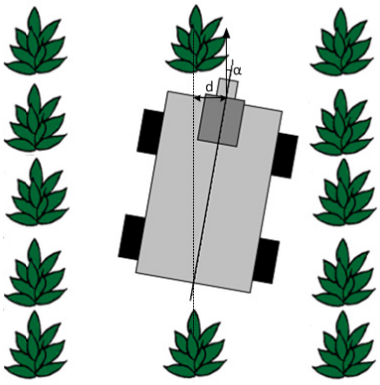


Fig. 2. Position of the robot in relation to the central crop row

The values of the vehicle's offset (d) and angle (α) are supplied to two fuzzy controllers, one for angular speed and the other for linear speed, which calculate the correction values for the steering, in order to generate row tracking behaviour in the robot.

3.1 Crop Row Detection

The image processing method used to extract the crop row in real time from the images captured (Fig. 3a) begins [4] with a segmentation step that segments the vegetation cover from the rest (soil, shadows, crop residues, etc.), obtaining a black and white image as a result (Fig. 3b). This segmentation makes use of the fact that pixels belonging to the vegetation cover have stronger green components than any other pixels of the images. The RGB image can be transformed into a grey scale image by means of a linear combination of the red, green and blue planes [5]. The grey image is then transformed into a black and white image using a threshold calculated by taking into account the maximum value of grey in the image. Subsequently, a morphological operation is applied over the image in order to remove the isolated white pixels (noise) and to enhance the pixels belonging to the crop that appears as an agglomeration of white pixels. In the next step, to deal with the perspective that appears in the image, the latter is divided into ten horizontal stripes, which are independently processed to obtain the ten points that define the crop row.

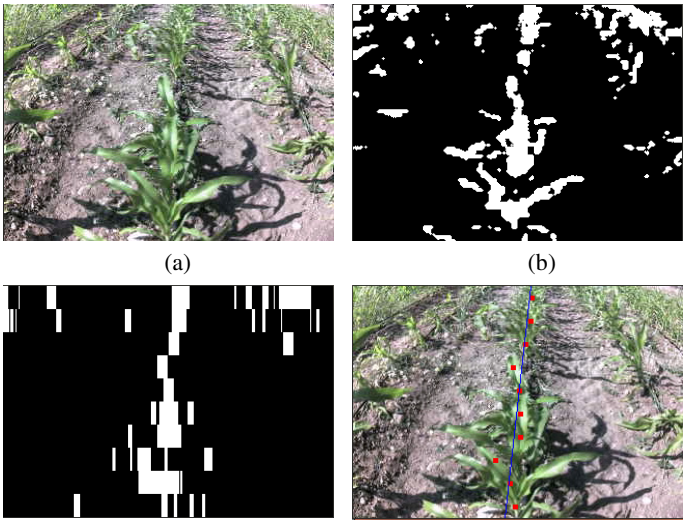


Fig. 3. Different stages of the image processing method developed

To obtain the ten points, the method labels as potential crop-rows those columns that have more white pixels than black. Then all pixels in columns labelled as crop rows are turned to white whereas the rest of the pixels are turned to black. In this way,

an image composed of black and white columns is obtained (Fig. 3c). To determine the points that define the crop row, in each strip the geometric centre of the block with the largest number of white columns together in the centre of the image are chosen. The method then estimates the line based on the average values of the coordinates of the points found, previously taking away the points that are false positives caused by weeds. These false positives can be easily identified because they normally tend to be located far from the other crop points detected. If the distance from the other points is greater than a certain threshold it may be classified as a false positive and eliminated from the set. After obtaining the straight line that defines the crop row (Fig. 3d), the angle (α) between the direction of motion of the robot and the line that defines the crop row and the displacement (d) between the centre of the vehicle and the line that defines the crop row may be calculated.

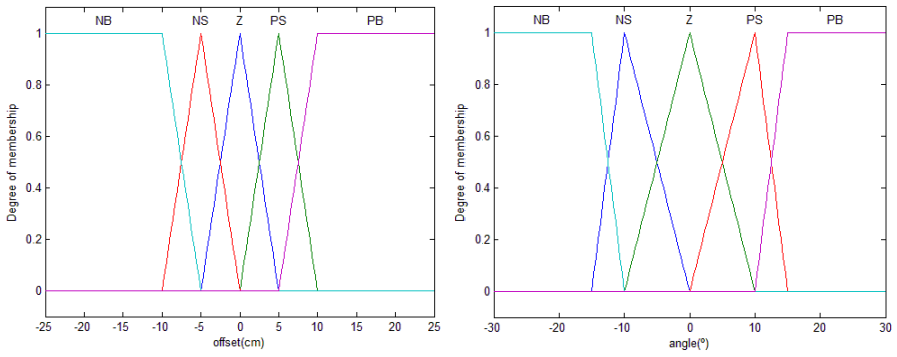
3.2 Navigation Control

The navigation control of the vehicle consists of two controllers, one for angular speed and the other for linear speed. Both controllers are fuzzy and therefore emulate the behaviour of an experienced driver; for example, if the vehicle is moved to one side of the crop row to be followed, the driver must turn the wheel the other way to correct the direction of the vehicle.

The controller acting on the angular speed of the vehicle has as inputs the displacement of the centre of the vehicle from the midpoint of the crop row (d in Fig. 2), and the angle of orientation of the robot (α in Fig. 2); and the controller produces as an output the angular speed of the vehicle. The rules governing the controller are described in Table 1, where NB (Negative Big), NS (Negative Small), Z (Zero), PS (Positive Small) and PB (Positive Big) are the fuzzy sets shown in Fig. 4, and which are defined according to the characteristics of equipment. Thus, in the input corresponding to the displacement of the robot with respect to the midpoint of the crop row (d), the value ranges of each set was chosen taking into account the distance between the wheels of the robot (41 cm). In the case of the angle of orientation (α), to choose the value ranges of each set it was considered that the robot will crush the crop row if it turns at an angle of either 30° or -30°.

Table 1. Fuzzy control rules of angular speed

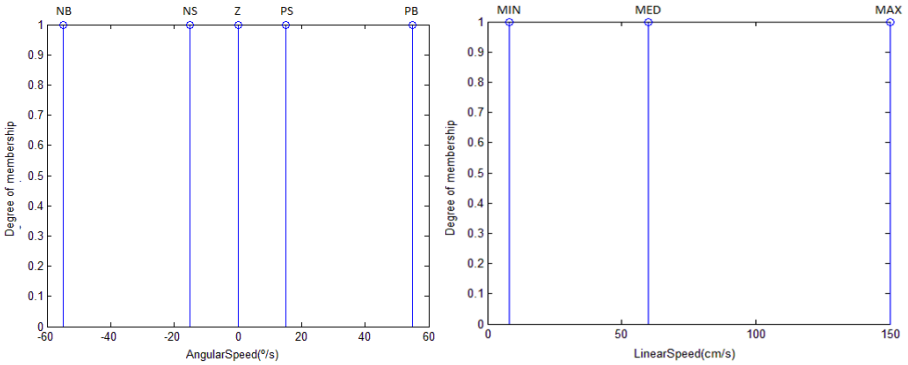
Offset d / Angle α	**NB**	**NS**	**Z**	**PS**	**PB**
NB	PB	PB	PB	PS	Z
NS	PB	PS	PS	Z	NS
Z	PB	PS	Z	NS	NB
PS	PS	Z	NS	NS	NB
PB	Z	NS	NB	NB	NB



(a) (b)

Fig. 4. Angular speed controller. Fuzzy sets of input variables: (a) offset and (b) angle.

The Takagi-Sugeno implication [6] was the method chosen, because it works well in applications that require on-time response. In this case, the output of the controller is the singleton type membership functions shown in Fig. 5a. The range of variation of the angular speed is -90° to 90° and the fuzzy sets have been chosen to cover the speed range that is considered necessary to be controlled by the robot.



(a) (b)

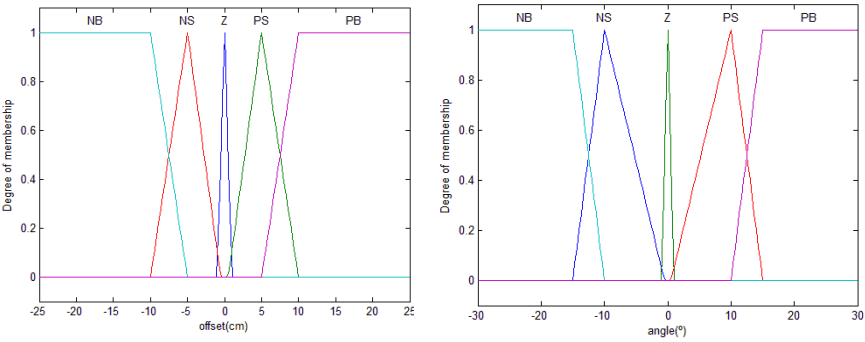
Fig. 5. Fuzzy sets for output variable: (a) angular speed and (b) linear speed

In the case of the controller acting on the linear speed of the robot, the input variables are the same as the previous case, i.e. offset and angle, and the fuzzy rules defined are summarized in Table 2.

The value ranges of fuzzy sets (Fig. 6) for both, displacement and angle, are chosen so that the vehicle moves at maximum speed (150 cm/s) provided that it is correctly positioned; otherwise, it slows down to avoid crushing the crop.

Table 2. Fuzzy control rules of linear speed

Angle α \ Offset *d*	**NB**	**NS**	**Z**	**PS**	**PB**
NB	MIN	MIN	MIN	MIN	MIN
NS	MIN	MIN	MED	MED	MED
Z	MIN	MED	MAX	MED	MIN
PS	MED	MED	MED	MIN	MIN
PB	MIN	MIN	MIN	MIN	MIN



(a) (b)

Fig. 6. Linear speed controller. Fuzzy sets of input variables: (a) offset and (b) angle.

The controller output is the linear speed applied to the vehicle and is characterized by three fuzzy sets (MIN: Minimum, MED: Medium, MAX: Maximum), whose values are chosen so that it covers the whole possible range of allowed speeds by the robot (Fig. 5b).

To test the robustness of the controllers they were subjected to the same experiments conducted in [7]. It was demonstrated that they functioned correctly.

4 Proposed Architecture

To integrate this crop row tracking behaviour and other actions that will be developed for the robot, we propose a network of nodes of different functionality that, guided by the goals and needs of the robot, generates an autonomous behaviour that allows the robot to inspect the crop without human intervention.

4.1 Nodes

This section introduces the concept of node, a fundamental component of the network that generates robot behaviours. A node is a basic unit with a specific functionality whose interconnection with other nodes forms behaviours. There are three types of node: perceptive nodes, cognitive nodes and actuator nodes. This division has been made based on the fundaments of cognitive psychology, rejecting the ideas of behaviourism, which ignores deliberative elements and explains behaviour as simple action-reaction reflexes [8]. Perceptive nodes are responsible for processing information from the world or from the robot itself. They are connected to the different sensors of the robot and process the information that other nodes use. For example, the crop row tracking behaviour begins with a perceptive node responsible for processing information from the camera detecting the position of the crop rows in the vegetation cover that appears in the image. This information is transmitted to the next cognitive node. Everything that is explained in the section 3.1 (crop row detection) will form this perceptive node. The cognitive nodes are deliberative elements, receive information from other nodes and decide how to act based on that information. Continuing with the above example, the cognitive node receives information regarding the position of the crop rows and decides how to act, correcting, if necessary, robot navigation by activating actuator nodes to avoid crushing the crop. The functionality of this cognitive node was outlined in the previous section 3.2 (navigation control). Actuator nodes materialize the behaviour and are connected to the actuators (motors, etc.) of the robot. They are usually simple nodes that perform the corresponding action when they are activated. To conclude the example, this crop row tracking behaviour has four actuator nodes: one that increases the speed of the robot, another that reduces it, another that acts by turning the robot to the left, and the last, which acts by turning the robot to the right. Depending which cognitive node is active it acts in one way or another. In Fig. 7 the nodes forming the crop row tracking behaviour may be seen; hereinafter, perceptive nodes are plotted with triangles, cognitive nodes with circles and actuator nodes with squares.

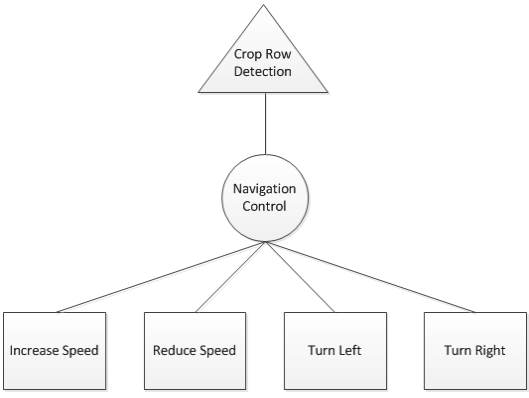


Fig. 7. Crop row tracking behaviour

4.2 Network of Nodes

To generate autonomous behaviour a network of nodes like those presented above is built. The goal is to generate this behaviour using this simple component structure, so that it is easy to interconnect with other nodes and use the same nodes for generating different behaviours. The structure of this network also allows affordable substitution of some nodes with others, thus responding to possible changes in the sensory equipment of the robot or the incorporation of other perception strategies. The simple replacement of nodes also responds well when the deliberative capacity of the cognitive nodes has to vary or when the actuators of the robot change and the actuator nodes have to be replaced. Another requirement of this network is to facilitate the incorporation of new behaviours to the robot's repertoire. With the type of network used, this is achieved simply by adding new connections to the network between existing or new nodes, and adding this new behaviour to the robot hierarchy of motivations, as will be discussed in section 4.3.

Behaviours are formed by the interconnection of nodes. The communication of information is unidirectional, perceptive nodes can provide information to cognitive nodes or actuator nodes, but not in the opposite direction, and cognitive nodes can send information to actuator nodes, but not backwards. The nodes can be active or inactive (so as not to lose processing capacity). Activation of the nodes occurs through the interconnections. The perceptive nodes are capable of activating the cognitive and actuator nodes that are connected to them. Likewise, the cognitive nodes, as neuralgic centres of behaviour, are capable of activating the perceptive, cognitive and actuator nodes connected to them. In the moment that an active perceptive node has information to transmit, it sends the data by all its connections, activating both receptor nodes at the same time so that they are capable of processing the information perceived. An active cognitive node, in contrast, has the ability to choose which nodes to activate or which to send information to out of all the nodes connected to it.

As already stated, behaviour is formed by the union of several nodes. Those same nodes, or a subset of them, may be part of other behaviours. Thus, the nodes are interconnected and the network is formed. Most behaviour nodes will be formed by perceptive, cognitive and actuator nodes, but with this structure reflex behaviours, i.e. action-reaction, may also be implemented. These are typical of reactive components and consist only of perceptive nodes and actuator nodes. It is also possible to construct purely deliberative behaviours, i.e. they do not take perception into account, and are thus formed by cognitive and actuator nodes. This type of behaviours can be used, for example, to propose a plan that actuator nodes can subsequently perform.

Under working conditions multiple active nodes run in parallel in the network. They may be part of the same behaviour, for example, perceptive nodes that are capturing information while previously perceived information is being deliberated, and actuator nodes are acting upon the world at the same time. But they may be part of different behaviours. This is because the robot can be alert, paying attention to different stimuli in the environment or its internal state in order to react appropriately in the event that the right stimulus is present. These behaviours on alert typically only

have the corresponding perceptive nodes active; the rest of the behaviours only become activated when appropriate stimuli are perceived. Fig. 8 shows a fragment of a network of the type explained, which includes crop row tracking and detection of the row end.

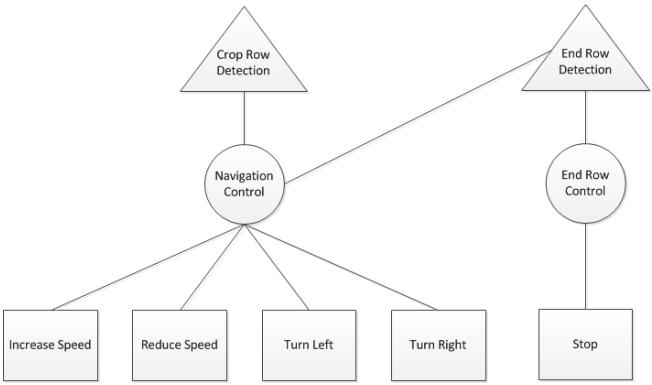


Fig. 8. Fragment of a network

4.3 Motives

This section aims to answer the question of how autonomous behaviour can emerge from the node network discussed above, with no conflicts and interferences between different behaviours. In other words, how is a coordinated, correct behaviour generated?

Motive may be defined as an inner driving force that activates the organism and directs its actions [9]. The most basic classification of motives is the distinction between primary and secondary motives. The primary motives are those related to the survival of individuals, while the secondary motives are considered as goals or reasons in themselves, mobilizing the individual into action. From this point of view, secondary motives are understood as the objectives or goals that originate inside the individual itself and that have themselves a meaning, a direction and an intention or purpose.

In animals, when a need associated with a primary motive appears, all the motivation of the individual is directed to immediately performing the behaviour to satisfy that need. Thus, when an animal is hungry, an appetitive behaviour destined to satisfy that need is elicited. This approach also applies to the robot itself. When the robot detects that power is running low on batteries, it stops whatever it is doing and will elicit a response destined to satisfy its "hunger", looking for the nearest charging point where it can charge its batteries.

In order for the robot to have a behaviour appropriate to its motivations and so that there are no conflicts between them, a hierarchy of motivational constructs is proposed (Fig. 9), with the primary motives in the lower levels and the secondary motives in the higher levels. All motives are organized from lower to higher levels so

that higher levels can only be satisfied when lower ones have been. Primary motives of the robot are the need to preserve its physical integrity and the need for power. The first of these is related to the need to react appropriately when some obstacle appears in the way, because if this need is not immediately handled, the robot will collide with the obstacle and it may be seriously damaged. The need for power is related to the need to find a point at which to recharge the batteries when its power is running low. The need to preserve physical integrity is at the lowest level of the hierarchy, since it is the most important need for the robot (if the robot is damaged by an obstacle, the other needs disappear), leaving the need for power in the next higher level. All primary motives have their behaviours on alert, to pay attention to environmental stimuli (obstacles) or its internal state (battery level) in order to react appropriately in the event that the right stimulus is present. As mentioned above, these behaviours on alert typically have only the corresponding perceptive nodes active, activating the other nodes only when appropriate stimuli are perceived.

Fig. 9. Hierarchy of motivational constructs

The robot's goals are part of the secondary motives. These are organized hierarchically so that the high-level goals are achieved through lower level goals. In our example, the maximum goal of the robot is to cover the entire crop field. For this it must successively cover the different rows that make up the crop, which are lower-level goals. In turn, covering a crop row means a series of lower-level goals, such as positioning itself correctly at the beginning of the row, tracking the crop without crushing it, and arriving at the end of the crop row (Fig. 10). The basis is to split the goals into increasingly simpler goals, so that as the goals of the lower levels are satisfied, the higher-level goals will gradually be satisfied. Within each level, the goals are pursued from left to right as they are located in the hierarchy.

The cognitive nodes of the behaviours that are being performed to satisfy the goals are responsible for putting the behaviour in charge to reach the next goal on alert, so it is ready when the current goal is met, and for disabling its behaviour when the goal has been achieved. When a behaviour to satisfy a goal is being performed and a primary need appears, the latter will be addressed and, subsequently, if possible, the goal that was put on hold will continue. For example, if an obstacle appears in the way, the robot will stop but if after a certain time the obstacle disappears the robot will continue with the previous goal. However, if the robot cannot continue with the goal, it will begin to satisfy again the goals of the same level belonging to the same goal of the next higher level. For example, if the robot while it is tracking the second

crop row is forced to interrupt inspection to find a charging point (low battery), the higher goal of covering the second crop row will be incomplete. Then, when the robot finishes recharging, it will have to satisfy the goal by positioning itself correctly at the beginning of the second row, in order to complete the tracking of it.

Thanks to this hierarchy of motivational constructs that guide the behaviour of the robot, an autonomous behaviour without conflicts in the different tasks is achieved.

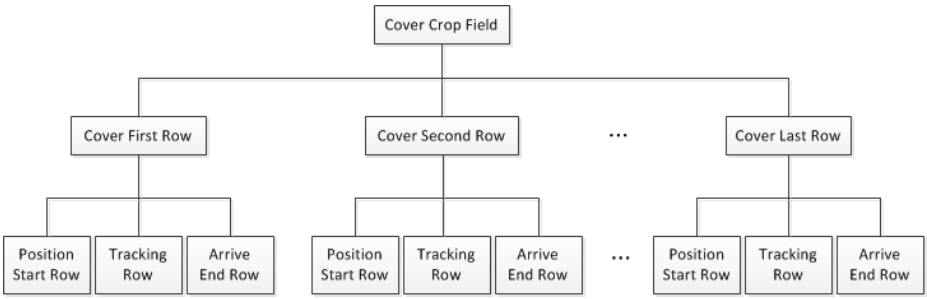


Fig. 10. Example of goals hierarchy

5 Conclusions and Future Work

In this paper we have established the different behaviours that a mobile robot for inspecting a crop field must present. Among these, the development devised for row-tracking behaviour has been shown. An image processing method capable of extracting the crop rows in real time from images taken with a high-quality web camera situated at the front of the robot has been implemented. Two fuzzy controllers have also been designed and developed to achieve vision-guided navigation for the robot. We have proposed a model for generating behaviours and we have suggested using this architecture to achieve autonomous behaviour in the robot. The division of behaviours into different components according to their function (nodes) facilitates the organization and implementation of behaviours, and facilitates reuse, replacement and addition of new nodes. Furthermore, thanks to the hierarchy of motivational constructs proposed, an appropriate behaviour for the robot emerges from its set of behaviours depending on the goals and needs that the robot must satisfy at each given moment.

Furthermore, to improve the image processing method, work is currently being carried out to detect 3 lines (central row and rows at the right and the left) which would serve to extract the correspondence between pixels and centimetres. This is needed to control the robot since the distance between crop rows is fixed and known. This would be done in such way that the robot could be precisely controlled without knowing certain camera parameters, such as the height at which it has been positioned.

Acknowledgements. The Spanish Government and the European Union have provided full and continuing support for this research work through projects: AGL201-30442-C02–02: Ground inspection systems in unmanned ground vehicles and their effective application in Weed detection for site-specific control (GroW) and UE-CP-IP245986–2: Robot Fleets for Highly Effective Agriculture and Forestry Management (RHEA).

References

1. Srinivasan, A.: Handbook of Precision Agriculture. Principles and Applications. The Haworth Press, New York (2006)
2. Stafford, J.V.: Implementing Precision Agriculture in the 21st Century. Journal of Agricultural Engineering Research 76, 267–275 (2000)
3. Senay, G.B., Ward, A.D., Lyon, J.G., Fausey, N.R., Nokes, S.E.: Manipulation of high spatial resolution aircraft remote sensing data for use in site specific Farming. Transactions of the American Society of Agricultural Engineers 41, 489–495 (1998)
4. Gottschalk, R., Burgos-Artizzu, X.P., Ribeiro, A.: Real-time image processing for the guidance of a small agricultural field inspection vehicle (2010)
5. Burgos-Artizzu, X.P., Ribeiro, A., Tellaeche, A., Pajares, G., Fernandez-Quintanilla, C.: Analysis of natural images processing for the extraction of agricultural elements. Image Vis. Comput. 28, 138–149 (2010)
6. Sugeno, M.: On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singletons Consequents. IEEE Trans. Fuzzy Systems 7(2), 201–224 (1999)
7. Bengochea-Guevara, J.M., Conesa-Muñoz, J., Ribeiro, A.: Developing a small autonomous robot for crop inspection (2012)
8. Bermúdez Moreno, J., Bermúdez, J., Pérez-García, A.M., Ruiz Caballero, J.A., Sanjuán, P., Rueda, B.: Psicología de la Personalidad. UNED, Madrid (2011)
9. Sanz Aparicio, M.T., Menéndez, F.J., Rivero, M.P., Conde, M.: Psicología de la Motivación. Sanz y Torres, Madrid (2011)

Validation of a Multisensory System for Fruit Harvesting Robots in Lab Conditions

Roemi Fernández[1], Carlota Salinas[1], Héctor Montes[1,2], Javier Sarria[1], and Manuel Armada[1]

[1] Centre for Automation and Robotics CSIC-UPM,
Ctra. Campo Real Km. 0,200 La Poveda, Arganda del Rey, 28500 Madrid, Spain
{roemi.fernandez,carlota.salinas,hector.montes,javier.sarria,
manuel.armada}car.upm-csic.es
[2] Faculty of Electrical Engineering, Technological University of Panama, Panama
hector.montes@utp.ac.pa

Abstract. This paper presents a multisensory system for the detection and localisation of fruits that are candidates to be harvested by a robotic manipulator. The devices that have been selected as primary sensors for this purpose are a high resolution colour camera, a multispectral imaging system that consists of a motorised filter wheel, and a Time-Of-Flight 3D camera. A controlled lighting system completes the set-up. The progressive RGB camera and the multispectral imaging system acquire the basic data inputs for the detection of areas of interest that belong to the fruits, whereas the Time-Of-Flight 3D camera provides fast acquisition of accurate distances enabling the localisation of the targets in the coordinate space. Several experimental tests have been carried out in laboratory conditions in order to evaluate the capabilities of the proposed multisensory system.

Keywords: fruit detection, multispectral system, optical filters, Time-Of-Flight camera, precision agriculture, image processing, classification.

1 Introduction

The first and most basic requirement for a harvesting robot is the efficient detection and localisation of fruits. This task is not trivial, since in natural scenes, most fruits are partially occluded by leaves, branches or other fruits. These occlusions remove the direct correspondence between visible areas of fruits and the fruits themselves by introducing ambiguity in the interpretation of the shape of the occluded fruit. In addition, colours of fruits cannot be rigidly defined because the high variability exhibited among the different cultivars within a same species and the different levels of ripeness. Moreover, fruits can be found in quite random positions and orientations in trees of various sizes, volumes and limb structures. Environmental conditions such as wind, rain, dust, moisture and lighting also increase the technical challenge imposed to the sensory system [1].

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_36,

In the last decades several studies aiming to provide automatic detection and localisation of fruits for robotic harvesting applications, have been reported in the literature. Parrish and Goksel [2] implemented the first computer vision system for detecting apples and guiding their harvesting robot. The proposed system was based on a monochrome camera and a red optical filter to increase the contrast between red apples and green-coloured leaves. Slaughter and Harrell [3-4] presented two approaches based on colour information to solve the fruit recognition problem for a Citrus Picking Robot. Whittaker [5] developed a system based on a monochrome camera to detect and located tomatoes in natural settings. Each acquired image was processed in order to find circular arcs that could correspond to tomato contours. Kassay [6] addressed the automatic detection of apples by using a stereo vision system which provided the 3D-dimensional position of each detected fruit. The vision system proposed by Buemi [7] for the tomato harvesting Agrobot robotic system was based on a single colour camera. Hue and saturation histograms were employed to perform thresholding to segment the image whereas the 3D information was obtained by stereo-matching of two different images of the same scene. Jiménez et al. [8] designed and implemented a sensory system based on an infrared laser range-finder sensor that provided range and reflectance images, capable of detecting spherical fruits in non-structured environments [8]. Some comprehensive surveys papers as [1] and [9] cover several aspects of these and other not-mentioned-systems.

This paper presents an automatic multisensory system that combines a high resolution colour camera, a multispectral imaging system, a Time-Of-Flight (TOF) 3D camera and a controlled lighting system for the detection, discrimination and localisation of fruits and other plant elements in natural environments. The proposed system is intended to be used in an autonomous harvesting robotic system, without requiring previous preparation of the crop. The rest of the paper is organised as follows: Section 2 describes the sensory rig that has been designed for the acquisition of images. Section 3 explains the perception cycle that has been proposed for this sensory system. Section 4 presents the results obtained from the experimental tests. Finally, major conclusions are summarised in Section 4.

2 Multisensory System Description

A sensory system designed for the detection, localisation and characterisation of fruits has to be capable not only of providing suitable data that can be analysed or arranged for a sensorial fusion procedure in order to extract the physical properties that identify the fruits, but also of managing the determinant factors of the different scenarios and environmental conditions.

The multisensory system proposed in this study combines a high resolution colour camera, a multispectral imaging system and a TOF 3D camera. The high resolution colour camera is not only utilised for the acquisition of RGB images, but also as part of the multispectral system, in which case it is set in the monochrome mode. The multispectral system is completed with a custom-made filter wheel and a servomotor that is responsible for the accurate positioning of the filter wheel. This positioning can

be achieved with at maximum angular velocity of 40 rpm and a position error if 0.001º. The filter wheel allows interchanging up to 5 optical filters, facilitating the adaptation of the system for the detection of different kinds of crops. Since correct illumination is critical to attain optimal image quality that is needed for best performance of image algorithms, the system also includes two different light sources, an array of xenon lamps and two halogen spots, located above and on both sides of the sensory system, respectively. This lighting system is connected to a control unit that enables the independent power on and off of the lamps, and the control of their intensities. A close-up of the proposed system is shown in Fig. 1.

RGB camera and multispectral imaging system will provide the input data required for the detection and characterisation of areas of interest that could belong to fruits, whereas the TOF 3D camera will supply simultaneously fast acquisition of accurate distances and intensity images of targets, enabling the localisation of fruits in the coordinate space.

In order to confer versatility to the set-up, the whole proposed multisensory system is installed on a pan-tilt unit that facilitates the data acquisition of different viewpoints. The tilt movement has a limited angular displacement of $\alpha=\pm 30°$ relative to the horizontal axis due to mechanical constraints. The yaw movement has no mechanical constraint, so it could rotate $360°$ around the vertical axis. However, for the stated application, the automatic yaw movement will be restricted for azimuthal angles out the range given by $0° \leq \beta \leq 180°$.

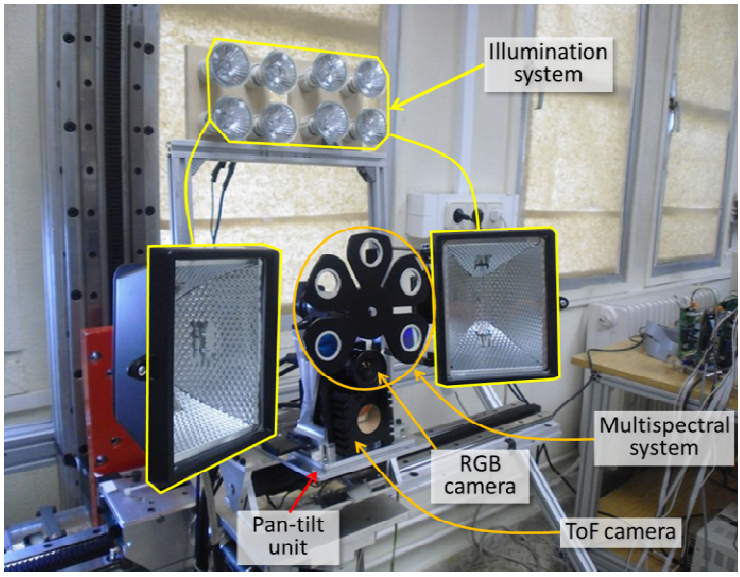


Fig. 1. Close-up of the multisensory system

3 Perception Cycle

Taking into account the proposed multisensory system, the goals of the harvesting tasks that the robotic platform is supposed to carry out, and the environment in which the whole system will operate, it is mandatory to determine what else can be modified in a controllable way in order to improve the process of extracting information and consequently increase the success rate in detecting and locating the fruits. Three different points have been identified as crucial for this process of enhancement: acquisition of multiple views of the region of interest, adaptation of some sensors parameters, and the control of the illumination system. The sequence of actions that will be carried out in order to deal with the three mentioned points will define the strategies that will maximise the perception capabilities. These steps, which constitute the proposed perception cycle, are outlined below.

1. At the start, the multisensory system will be located in front of the crop, and the scanning of the working area is planned to be carried out from the bottom part to the top part of the crop. This order is selected because it is more likely to find ripe fruits in the bottom of the crop than in the top. At the start, it must also be defined a reference model of the fruit that is going to be detected and localised for the harvesting. This model should include at least some features such as the colour, the average or the minimum size (minimum number of pixels onto the image plane) and the level of ripeness which must be the fruit to fulfil the desired quality standards. The multisensory system is then initialised.
2. The system proceeds with the data acquisition from the first window (effective field of view). For that, the integration time of the TOF camera is set according to the initial working distance, and the exposure time of the colour camera is set to the value defined for the RGB mode. Once depth information is acquired with the TOF camera, the illumination system is switched on and its intensity is adjusted to the level defined for the camera when it is operating in colour mode. An RGB image of the current window is then acquired. Next, the camera is set in monochrome mode, and filtered images are acquired for the same window. During the acquisition of the filtered images, the intensity of the illumination system, as well as the exposure time of the camera are adjusted in concordance with each of the selected filters. Finally, the illumination system is switched off.
3. Acquired colour, filtered and depth images are pre-processed and registered for the current working window. As a result, a classification map will be obtained, showing the possible zones with fruits onto the image plane. These zones will be marked as blobs.
4. If no blobs are marked in the previous step, the multisensory system will be moved to the next window, following the sequence defined in step 1, which goes from the bottom part to the top part of the crop.
5. If one or more blobs are marked in the step 3, the algorithm proceeds to compare the blobs with the defined reference model of the fruit that is intended to be detected. From this comparison, at least two results should be extracted: the segmentation of blobs which contain pixels belonging to different targets, and the number of pixels onto the image plane of each segmented blob.

6. The percentage of visibility of each blob is calculated based on its number of pixels and its comparison with the reference model defined initially. The minimum threshold for considering a fruit as visible was set to 50%. Therefore, all the blobs with a visibility equal to or greater than 50% will be marked as candidates for harvesting. The 3D location of these visible blobs is then extracted from the registered TOF data.
7. For those blobs with a visibility lesser than 50%, the active perception strategies will try to reduce the negative effect of occlusions by acquiring multiple viewpoints of the same region of interest in an effort to recover information that might be missing in a particular view. Two approaches are proposed:
   (a) The first approach involves computing the position where the sensory system should be located for acquiring an additional view. This could be carried out by selecting the less visible blob, calculating its approximate centroid, and estimating the normal vector to this point by utilising the reference model.
   (b) The second approach involves moving the sensory system to *N* predefined positions in order to acquire data from additional views. These positions should be selected after conducting a fruit detectability analysis for different camera positions, such as the ones presented in [10-11]. These positions will be defined with respect to the initial location of the multisensory system for the current working window.
8. Then a fusion algorithm should combine efficiently the information from different views and recalculate the percentage of visibility for each segmented blob. Those new blobs with a visibility equal to or greater than 50% will be marked as candidates for harvesting.
9. If after this procedure there are still blobs with a visibility lesser than 50%, and these blobs represent more than the 10% of the candidates that are labelled as ready for harvesting, the active perception strategies will repeat the step 7. On the contrary, if these blobs represent less than the 10% of the candidates that are labelled as ready for harvesting, they will be discarded, and the multisensory system will be located in the next acquisition window in front of the crop.

Fig. 2 summarises the main steps involved in the proposed perception cycle.

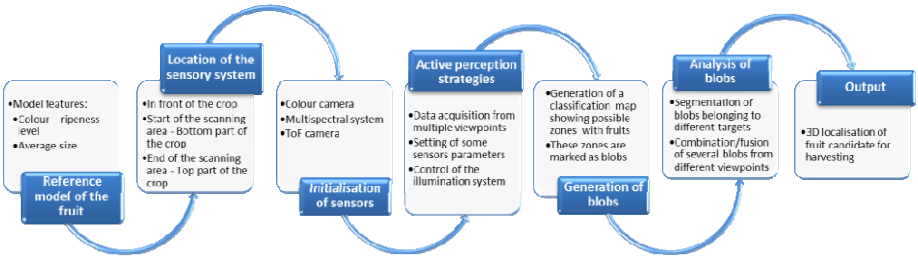


Fig. 2. Overview of the main steps involved in the perception cycle

4 Experimental Results

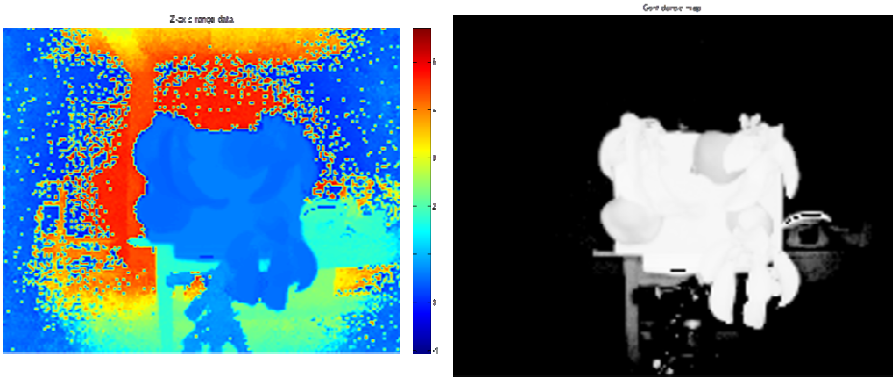
In order to validate the proposed multisensory system, several experiments have been conducted in laboratory field conditions. For these experiments, the whole multisensory system (including the lighting system and the pan-tilt unit) was installed on an electrically driven platform with two prismatic degrees-of-freedom. The two prismatic joints provide the movements in the horizontal and the vertical axes, or in other words, provide the positioning in the Cartesian plane.

For the laboratory experiments a simulated scenario, consisting on leaves and artificial apples, was located in front of the robotic platform, at a distance of 640mm. After starting the active perception cycle, RGB, multispectral and TOF data were acquired for the first window of the scenario, corresponding to the first viewpoint of the sensory system. Figs. 3 and 4 show the images acquired with the RGB and the TOF camera, while Fig. 5 displays the images acquired with the multispectral system by using two band-pass filters with centre wavelengths of 635 nm and 880 nm. Note that the TOF camera provides amplitude, depth and confidence data simultaneously for each pixel of the image captured. The amplitude represents the greyscale information, the depth is the distance value calculated within the camera and the confidence is the strength of the reflected signal, which means the quality of the depth measurements.

Once the scene information has been acquired, the images are pre-processed and registered, and a classification map is obtained, showing the possible zones with fruits (see Fig. 6). Red and green colours are utilised to visualise pixels classified as pertaining to red and green apples. For each blob that satisfies the predefined requisites to be a fruit candidate for harvesting, the centroid and the area is calculated. In the example shown in Fig. 7 it is possible to appreciate four blobs with areas that range from 781 to 31266 pixels.

(a) (b)

Fig. 3. Images acquired with the RGB and the TOF camera. (a) RGB image. (b) TOF amplitude data.



(a) (b)

Fig. 4. Images acquired with the TOF camera. (a) Z-axis range data acquired with the TOF camera. (b) TOF camera confidence map.

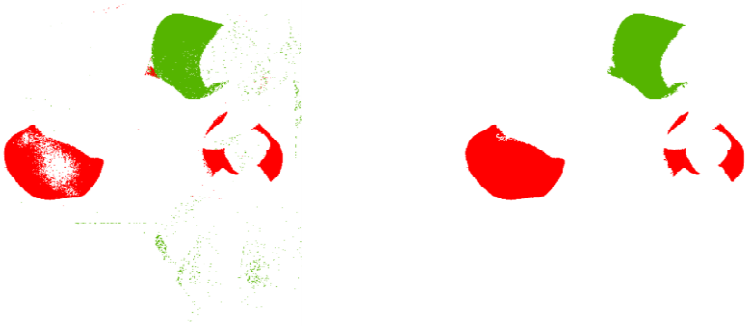
(a) (b)

Fig. 5. Images acquired with the multispectral system. (a) 635 nm image. (b) 880 nm image.

Then the percentage of visibility of each blob is calculated based on its number of pixels and its comparison with the reference model defined initially. In this case, the reference model was a red circular area with about 37377 pixels. That's mean that the percentages of visibility of the four blobs are: 83.7%, 2.16%, 9.4%, 12.5%. Therefore, only the first blob satisfies the minimum threshold for being considering as a visible fruit and consequently can be marked as candidate for harvesting. The rest of the blobs require further inspection.

Taking into account that the blob that has an area of 781 pixels is the less visible blob with a percentage of visibility of 2.16%, the next position where the sensory system should be positioned is calculated by following the first approach described in the step 7 of the cycle defined in Section 3. Then the calculated coordinates are sent to the control unit, who will be in charge of moving the joints of the robotic platform in

order to position and orientate automatically the sensory system for the next viewpoint. Fig. 8 shows the results obtained with the images acquired in the second viewpoint. In this case only two blobs are detected. The first blob corresponds to the one that had been detected in the previous step, so it is disregarded. The second blob that is detected has a percentage of visibility of 53.1%, and consequently, after just one movement, it can be marked as candidate for harvesting. In this way, the process would be completed, following the steps described in the previous section.



(a) (b)

Fig. 6. Enlarged view of the classification map – possible zones with fruits. (a) Before applying morphological operations. (b) After applying morphological operations.

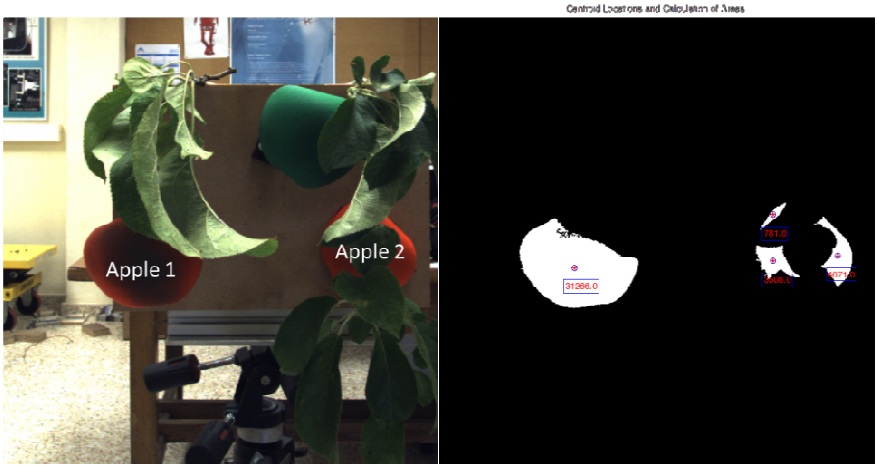


Fig. 7. Centroid and area calculation for the blobs obtained from the initial viewpoint

Fig. 8. Centroid and area calculation for each blob detected in the second viewpoint

5 Conclusions

This paper proposes a multisensory system for the detection and localisation of fruits that are candidates to be harvested by a robotic manipulator. The solution includes a colour camera and a multispectral system for acquiring reflectance measurements in the visible and NIR region that will be used for finding areas of interest that belong to the fruits, and a TOF camera that provides fast acquisition of accurate distances enabling the localisation of the targets in the coordinate space. A perception cycle that defines a sequence of actions aimed to improve the detection and localisation of fruits has also been defined in this paper for the proposed multisensory system. The action lines addressed are the acquisition of RGB, filtered and TOF data of the region of interest from several viewpoints, the setting of some sensors parameters, and the control of the illumination system.

For the acquisition of data from multiple views, two approaches have been presented. The first approach involves computing the position where the sensory system should be located, whereas the second involves moving the sensory system to *N* predefined positions. First experimental results concerning this point show that with these approaches it could be possible to increase the percentage of visibility of the detected fruits. While the latter approach seems to be a priori more robust than the former, acquiring images from *N* viewpoints could be more time consuming and less efficient from an energetic point of view. However, this last statement may not be entirely true if it is possible to resolve the occlusion of several fruits simultaneously. Therefore, one or other strategy could be selected depending on the goals and the constraints of the application.

On the other hand, the setting of the exposure time will avoid acquiring underexposed or overexposed images, and consequently the loss of image information.

This will be especially important during the acquisition of filtered images, which require very different exposure times between them and with respect to the RGB image.

The perception cycle will also be in charge of adjusting the integration time of the TOF camera if the distance between the sensory system and the target is varied significantly during the acquisition process. In this way, the error in the computed depth will be minimised, ensuring more precise distance measurements. To provide a correct illumination, which is crucial for attaining optimal image quality, the perception cycle will also take care of switching on and switching off the illumination system, as well as controlling its positioning and intensity.

Acknowledgments. The authors acknowledge funding from the European commission in the 7th Framework Programme (CROPS Grant Agreement Nº 246252) and partial funding under ROBOCITY2030 S-0505/DPI-0176 and FORTUNA A1/039883/11 (Agencia Española de Cooperación Internacional para el Desarrollo AECID). Héctor Montes also acknowledges support from Universidad Tecnológica de Panamá.

References

1. Sarig, Y.: Robotics of Fruit Harvesting. Journal of Agricultural Engineering Research 54, 265–280 (1990)
2. Parrish, E., Goksel, A.: Pictorial Pattern Recognition Applied to Fruit Harvesting. Trans. ASAE 20, 8222–8827 (1977)
3. Slaughter, D., Harrel, R.C.: Color vision in robotic fruit harvesting. Transactions of the ASAE 30(4), 1144–1148 (1987)
4. Slaughter, D., Harrel, R.C.: Discriminating fruit for robotic harvest using color in natural outdoor scenes. Transactions of the ASAE 32(2), 757–763 (1989)
5. Whittaker, D., Miles, G.E., Mitchell, O.R., Gaultney, L.D.: Fruit location in a partially occluded image. Transactions of the ASAE 30, 591–597 (1987)
6. Kassay, L.: Hungarian robotic apple harvester. Paper ASAE Nº 92-7042 St. Joseph, MI 49085, 1–14 (1992)
7. Buemi, F., Massa, M., Sandini, G.: Agrobot: a robotic system for greenhouse operations. In: 4th Workshop on Robotics in Agriculture, IARP, Toulouse, pp. 172–184 (1985)
8. Jiménez, A., Ceres, R., Pons, J.: A vision system based on a laser range-finder applied to robotic fruit harvesting. Machine Vision and Applications 11(6), 321–329 (2000)
9. Jiménez, A., Ceres, R., Pons, J.: A survey of computer vision methods for locating fruits on trees. Transactions of the ASAE 43(6), 1911–1920 (2000)
10. Ruizendaal, J.: Fruit visibility analysis for different camera positions in sweet peppers. Thesis Farm Technology, MSc Agricultural and Bioresource Engineering, Wageningen University (2011)
11. Hemming, J., Bac, W., van Tuijl, B.: Fruit detectability analysis for different camera positions in sweet pepper harvesting. In: Proceedings of the International Conference of Agricultural Engineering, CIGR-AgEng 2012, Valencia Spain, July 8-12 (2012)

Configuring a Fleet of Ground Robots for Agricultural Tasks

Luis Emmi, Mariano Gonzalez-de-Soto, and Pablo Gonzalez-de-Santos

Centre for Automation and Robotics (UPM-CSIC),
28500 Arganda del Rey, Madrid, Spain
{luis.emmi,mariano.gonzalez,pablo.gonzalez}@car.upm-csic.es

Abstract. Lately, many computer-based sensors and actuators have been incorporated in agricultural equipment with the main goal of configuring agricultural robots capable of achieving different tasks autonomously. However, the incorporation of different electronic systems in a robot impairs its reliability and increases its cost. Thus, hardware minimization and ease of software integration is mandatory to obtain feasible robotic systems. This paper strives to find a hardware architecture for both individual robots and robots working in fleets to improve reliability, decrease development costs and allow the integration of software from different developers.

Keywords: Sensors Integration, Hardware architecture, Autonomous robots, Agricultural vehicles, Fleets of robots.

1 Introduction

At the present time, there exists commercially a huge number of specific devices such as machine vision, global positioning system (GPS) real time kinematics (RTK), laser-based equipment, inertial measurement units, embedded computers, etc., that have allowed the development of many autonomous vehicles [19]. These autonomous systems provide accurate positioning and steering in outdoors scenarios, which make them able to accomplish many industrial and service missions if equipped with the proper tools.

There are currently many researchers working on the basis of autonomy for vehicles but there are many others trying to apply the current knowledge to industrial and/or service applications, specifically in agriculture [16, 12, 13]. One step forward, in which this article is focused on, is the development of fleets of robots that collaborate to accomplish a certain mission. Their theoretical foundations have been investigated in recent years [1, 3], but the first developments will appear in the near future [5, 18].

Fleets of robots grant many benefits [2, 14] but also exhibit a number of disadvantages. In a real robotic application, as those devoted to agriculture, for instance, lots of information must be processed and a large number of signals have to be handled to coordinate tools and devices. And this is an important limitation

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_37,

because the number of total devices (sensors, actuators, computers/controllers, etc.) rises with the number of units in the fleet. As a consequence, the mean time between failures decreases significantly because of a failure in one robot component produces the entire fleet to malfunction. The reliability of a fleet is influenced significantly by that reduction in the time between failures, which is vital in robots for real missions and especially for those devoted to agriculture where a fault in a mobile robot can ruin a crop.

To achieve a flexible, reliable and maintainable fleet of autonomous mobile robots, the hardware architecture design, involving sensors and actuators, for both the vehicle navigation system and the operation of the implements must be robust, simple and modular. These were the features sought in the configuration of the mobile robots in which our fleet is based on. Thus, this article attempts to define topologies of controllers for fleet of robots allowing shearing resources in a holistic way. To accomplish this objective, this article first presents the project RHEA, funded by the European Commission through the Seventh Framework Programme; thus, a brief description of the RHEA ground mobile units (GMU) based on a wheeled robot is presented in Section 2. Section 3 focuses on the description of the GMU controllers. The different topologies envisaged for fleets of robots working in agriculture are stated in Section 4. Finally, some conclusions are pointed out in Section 5.

2 The RHEA Fleet

RHEA came out as an attempt to provide robotic solutions for crop and pest scouting and management avoiding the shortcomings of current huge agricultural vehicles (dangerous situation if the vehicle goes out of control, terrain compaction, cost, etc.) by configuring a group of medium-sized mobile robots endowed with specific sensors and actuators capable of achieving a number of field tasks. The following benefits of this proposal over the conventional agricultural machinery can be pointed out as:

1. A number of medium-sized robots can attain better efficiency than one huge vehicle
2. Fleets of robots permit re-planning the mission in case of failure of one unit; whereas, a failure of one big vehicle stops the entire mission until it is mended
3. medium-sized robot (or a fleet of them) is intrinsically safer than a huge vehicle running autonomously
4. fleet of robots can be supervised by a single operator, while large machines have one each

RHEA is thus focused on different scientific and technical objectives related with (a) perception and discrimination of crops and weeds, (b) actuation systems for weed and pest control, (c) human-machine interfaces and (d) mobile robots both aerial and terrestrial. Specifically, the objectives related with terrestrial vehicles, the robots in which this article is focused on, are:

1. To devise a fleet of ground mobile units capable of getting images of the mission field and applying mechanical and/or chemical methods for management of crops and weeds. The ground mobile units will be based on wheeled platforms capable of carrying the associated tools and equipment. The ground units should be small, lightweight, and reliable, exhibiting communication and location features
2. To steer the ground mobile units to follow the crop rows at about 6 km/hour with a positioning accuracy of about ±2 cm. Robot guidance is to be based on devices and algorithms relying on machine vision to detect and track the crop rows and avoid obstacles
3. To develop human-machine interfaces for monitoring automated vehicle outdoors

To accomplish these basic features the whole system is broken down into the following modules and subsystems (See Fig. 1):

Mission Manager

This module allows the operator to describe the whole mission by indicating (a) the type of task to be carried out; (b) the features of the working field including prior information on the field obtained by remote sensing, sampling, etc.; (c) the number and features of the available robots; etc. The mission manager based on this information and on decision-making algorithms computes a sequence of proper mobile unit actions to complete the task [6]. In case of system failure, the operator is allowed to re-plan the mission in order to accomplish the initial task. The Mission Manager runs in the Operator Station computer and takes into account different aspects such as coordination, cooperation and collaboration among units in the fleet. This computer is located in the Operator Station, a movable cabin located at the edge of the mission field where the DGPS base antenna and communication antennas are placed. This Operator Station is moved along with the fleet from mission field to mission field (See Fig. 2).

Perception System

The Perception System relies on advanced vision devices and image processing techniques and algorithms and aims at precise localization of natural elements and discrimination between crops and weeds. This system is based on cameras operating in the visible spectrum and lasers installed onboard the ground mobile units. These devices provide the data and particularly the images coming from the crops [11]. Those images are the inputs for the row detection system in charge of steering the ground mobile units in wide-row crops (maize, garlic, onion, etc). Objects in the robot's path are detected by a combination of vision and laser data. The technical aspects of the object detection system are out of the scope of this article

Actuation System

The Actuation System provides direct actuation to vehicles and crops. It is divided hierarchical into the device system, the low-level actuation system and the high-level decision making system:

The Device System is the lower module in the hierarchy and the ultimate system in the RHEA fleet. It generates the required actions for the actuation equipment, which

consists of a number of end-effectors for weed or pest control (chemical and physical application tools). The chemical tools apply the treatment through spray nozzles positioned in traditional or special spray bars [4, 20]. The physical tools act directly on the weeds applying mechanical and thermal processes with physical devices [15].

The Low-level Actuation System consists of a number of electronic drivers and algorithms acting directly on the Device System actuators: spray valves, electrical motors, the thermal-weed removal devices and the mobile unit actuators (steering, throttle, three-point hitch, power take-off, and the hydraulic pumps for providing power to the implements).

The High-Level Decision Making System (HLDMS) is the heart of the ground mobile units and it is responsible for deciding what process to apply, where to apply it, the precise dose and the application instant. This system is based on decision techniques that take information of (a) the general features of the field (dimensions, irregularities, etc), (b) the a-priori knowledge on crops and weeds of the specific field and (c) data from the perception system as inputs and elaborate the pertinent actions for the Low-level Actuation System.

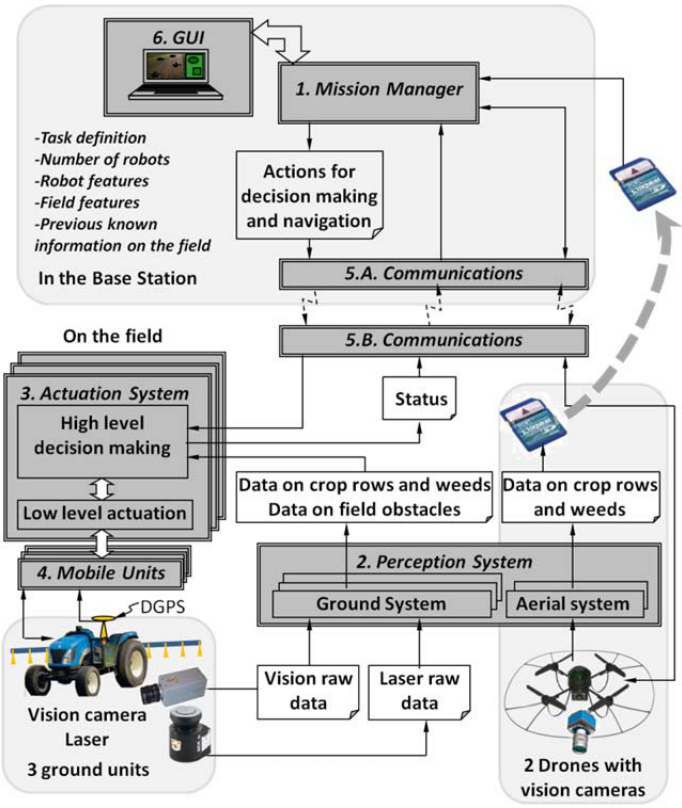


Fig. 1. RHEA system breakdown

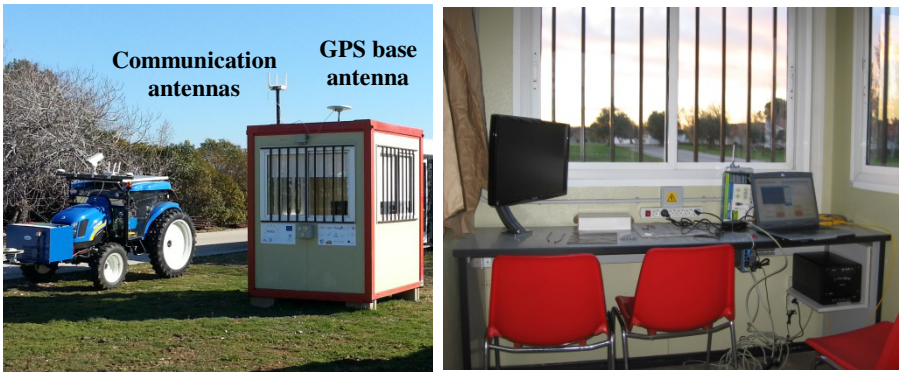


Fig. 2. Operator station: exterior and interior views

Mobile Units

RHEA uses a fleet of heterogeneous –ground and aerial– robots to carry the perception and actuation system over the mission field. The ground units are medium-sized vehicles based on a commercial tractor chassis with onboard equipment for autonomous navigation and communications. These robots are capable of towing implements where the actuation devices for crop treatments are placed.

In addition, a team of aerial robots relying on small commercial hex-rotors endowed with vision systems is operative to provide aerial images for off-line weed patch detection. This part of the RHEA robotic fleet is out of the scope of this present article.

Communication and Location System

Communication and location components are installed in the mobile units, the operator station, and personal mobile terminals that provide flexible service-like interfaces for wireless communication and data collection/dissemination among the elements of the RHEA system [9]]. The communication facilities at the operator station make available the Internet connectivity for the management of the system remotely (See Figs 2 and 4).

Operator Station and Graphic User Interfaces

The operator station is the point where the operator interacts with the whole system. It consists of a Graphic User Interface (GUI) for monitoring tasks, information record-keeping and operation optimization. The GUI system communicates in real-time with the real system and displays the actual general status of every mobile unit (position, orientations, speed, and so on). The GUI allows the operator to send commands to the robots of the fleet and includes a simulation system to let the user test different missions in advance.

The group of ground mobile units that configures the RHEA fleet for testing the concept is shown in Fig. 3; where each robot is towing a different implement.

3 Ground Mobile Unit Configuration

The concept of ground mobile unit in RHEA evolved from tailored-made vehicles to the adaptation of small commercial tractors (See Fig. 4(a) and (b)). This allowed the consortium to advance system integration and testing while avoiding other high-time consuming activities such as chassis design, manual assembling, testing, and vehicle homologation, for instance. Furthermore, these modifications drastically increased vehicle reliability by using a long-term tested items (engine, braking, steering and transmission systems, housing, etc.) while decreasing time for availability.

The selected vehicle was a CNH Boomer-3050 (51 hp – 37.3 KW, 1200 kg) with a restructured and empty cabin used to place the computing equipment of the perception, communication, location, safety and actuation systems. In addition, some systems need to place specific elements outside the cabin: vision camera, laser, antennas (GPS and communications) and emergency bottoms. Fig. 4(c) and (d) illustrates the placement of elements and computers on both the vehicles and in the cabin.

The hardware architecture is devised to control these ground mobile units, which contain the following subsystems:

- A weed detection system to detect weed patches that relies on machine vision
- A crop row detection system to help steer the vehicle based on machine vision
- A laser range finder to detect obstacles in front of the mobile units
- Communication equipment linking the operator station, the mobile units, and the user portable devices
- A Global Positioning System to locate the vehicle in the mission field
- An inertial measurement unit (IMU) for complementing the GPS data and allowing the vehicle positioning to be improved
- A vehicle controller in charge of computing the steering control law, throttle and braking for path tracking purposes
- A central controller as decision making system in charge of gathering information from all perception system and computing the actions to be ordered to the actuation components
- An additional energy power supply based on a fuel cell, which is monitored by the central controller

In addition, the hardware has to take into account the needs of the low level actuation system described above.

At the beginning of the project, all aforementioned subsystems, developed by different research groups, run in different computers working under different operating systems (Windows, Linux, QNX, etc.). In addition, the software modules were developed in dissimilar languages (C++, Python, NET, etc) making the entire system difficult to manage (See Fig. 5).

A first attempt was made to integrate all subsystems in a single computer as a central controller. The initial clue was given by the fact that the vision camera is GigE vision standard compliant (global camera interface standard developed using the

Gigabit Ethernet communication protocol framework for transmitting high-speed video and related control data); therefore, selecting a powerful computer with GigE ports would ease the integration of the vision systems in the Main Controller. As a prospective solution, we consider National Instruments compactRIO series that, apart from integrating two GigE ports, allows the programmers to call external libraries in the form of DLLs (Dynamically Linked Library) and code interface nodes (CINs). This facilitates the integration of software developed under different programming languages. Furthermore, this equipment also integrates a Field Programmable Gate Array (FPGA), which is also an interesting feature for executing some pieces of code with highly demanded time restrictions. Finally, this solution permits a direct interface between the camera and the Main Controller using the functionalities provided by LabVIEW for both configuration and acquisition, avoiding the development of new drivers.

This is an example of possible system centralization of complex subsystems; nevertheless, NI compactRIO allows other subsystems to be centralized in a direct manner, for example:

- Ethernet communications: NI WLAN modules and Ethernet switch are commercially available
- Industrial communication buses: a CAN bus and an ISO bus can be integrated through National Instruments CAN interfaces
- General I/O modules: National Instruments analog and digital I/O modules
- Laser systems: connected through Ethernet or NI specific modules
- Inertial Measurement Units: connected through National Instruments serial modules

Fig. 3. RHEA fleet: three equal robots towing different implements

Fig. 4. (a) Initial commercial tractor; (b) final RHEA GMU; (c) external equipment onboard the GMUs and (d) internal equipment distribution inside the GMU's cabin

Fig. 6 shows the final system that is fully integrated into the ground mobile units, while its final arrangement is shown in Fig. 4.

Autonomous vehicle architectures for outdoors are normally made up of four main modules [10]: navigation sensors, vehicle motion models, navigation planner and steering controller. In RHEA, navigation sensors are based on Differential Global Positioning Systems (DGPS) that consist of a base antenna located on the operator station and a receiver placed in every mobile unit with two antennas to get both robot's position and heading (See Figs. 2 and 4). In some applications, when the system has to follow a crop row, a ground perception system onboard de mobile unit is used. Some vehicle motion models are required for autonomous navigation. They consist of mathematical models describing how a vehicle behaves according to different aspects such as kinematics, essential for autonomous motion; dynamics, to be taken into account when moving a heavy mass at medium/high speed; and dead-reckoning which is essentially a motion memory system based on internal sensors (encoders, compass, etc.) [17]. Then, the navigation planner is in charge of following a path (straight line or curve) avoiding obstacles in the course. It takes data from the GPS, machine vision, laser, etc. and generates the commands for the module in charge of vehicle steering. Thus, the steering controller is basically a traditional positioning controller, which tries to correct the lateral vehicle displacement with respect to the desired path. Several control algorithms for correcting lateral error for outdoors vehicles can be found in the literature; the most popular are: PID, feed-forward PID, fuzzy logic controllers, etc [21, 22]. In this case, we have used an algorithm capable of following paths, both straight and curves, with the same rule base, and in both cases, obtaining high accurate [7].

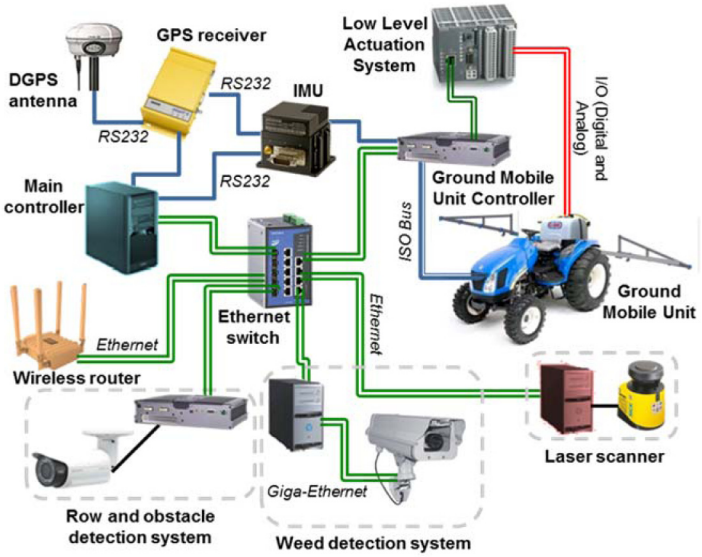


Fig. 5. Initial hardware configuration for the RHEA ground mobile units

4 Topologies for Fleets of Robots

After minimizing the hardware of the individual mobile units, the next step is to minimize the hardware of the whole fleet. Basically, a fleet is a number of independent mobile units that must be coordinated by a computer and interfaced with an operator at given instants. The procedure of minimizing the hardware of a fleet of robots relies on these two items. The operator must be present at some instants: (a) mission configuration, (b) mission start, (c) mission stop or suspension, etc. Thus, an operator interface is essential, which can be provided in the form of a base station (computer monitor and keyboard) or a portable device (e.g., tablet, smartphone) that allows the operator to move close to the mobile units. Nevertheless, the computing system can be reduced to a minimum depending on the selected topology. Revising typical communication techniques, three topologies can be envisaged regarding the central computer:

1. **Central-external controller:** a master external computer connected to the fleet units through a wireless communication system runs a mission manager (mission planner and mission supervisor) that sends commands to (and receives data from) the fleet mobile units. A failure on this master computer stops the mission of the fleet. This is the topology implemented in the current RHEA fleet, in which the master controller is the one located at the base station (see Fig. 7.a).

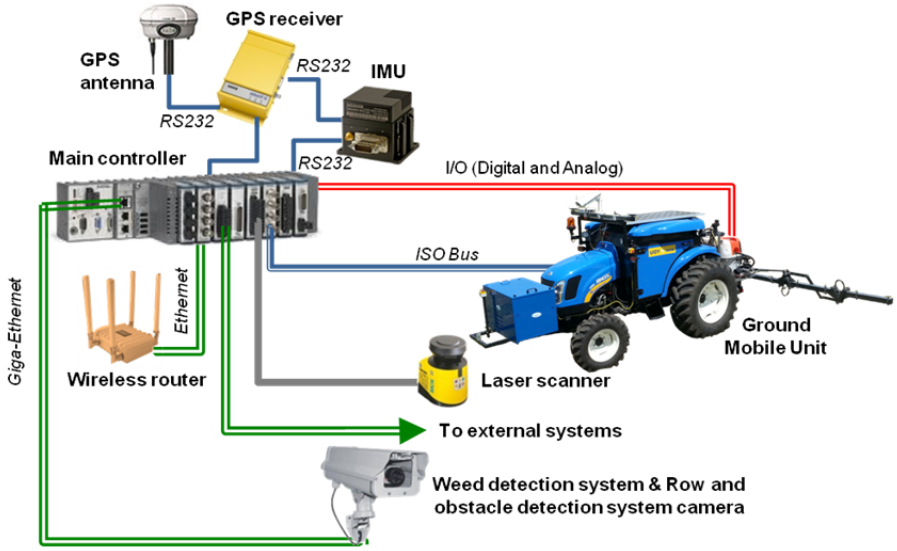


Fig. 6. Hardware configuration for the RHEA ground mobile units

2. **Central-internal controller**: one fleet unit controller acts as a master running the mission manager while the rest act as slaves receiving commands and returning data. A failure in this master controller also stops the mission of the fleet, but the likelihood of failure decreases because the whole fleet has one less computer and communication system (see Fig. 7.b). Adaptation to this topology is straight forward: the mission manager running on the Base Station computer is packaged into a DLL and included, with minor software modifications, in a GMU controller, which will act as the fleet master controller. This process is the one explained in Fig. 5.
3. **Immerse controller:** the mission manager is copied in all mobile unit controllers so that a failure in one unit means that unit stops, while the rest can reconfigure the mission plan to accomplish the task. Note the hardware is not increased and the same mission manager algorithms run on every unit controller (see Fig. 7.c), while the unit statuses are shared among all the robots by broadcasting a few status data in a sampling period basis. When a unit goes out of order the others received that information in the status or by a sampling period timeout; in such a case, the still active units will compute the mission manager taking into account that incidence.

For the last two topologies, there is not a clear gain of reducing the amount of hardware in the general architecture but the immerse topology increases the robustness of the system by using a mirrored mission planer, located on each mobile unit controller. This immerse controller allows each mobile unit to supervise (mission supervisor) the execution of the plan and monitoring the status (position, speed, etc) of the others mobile units while is able to adapt the missions of individual units to meet the goal of the fleet. This topology, which the authors are currently working out, is well suited to the hardware architecture presented in section 4 for the ground

mobile unit, where the use of a cRIO system, as Main Controller, allows the communication with others cRIO systems to be direct and without the development of drivers and communication protocols, thanks to the LabVIEW utilities for sharing information between NI systems.

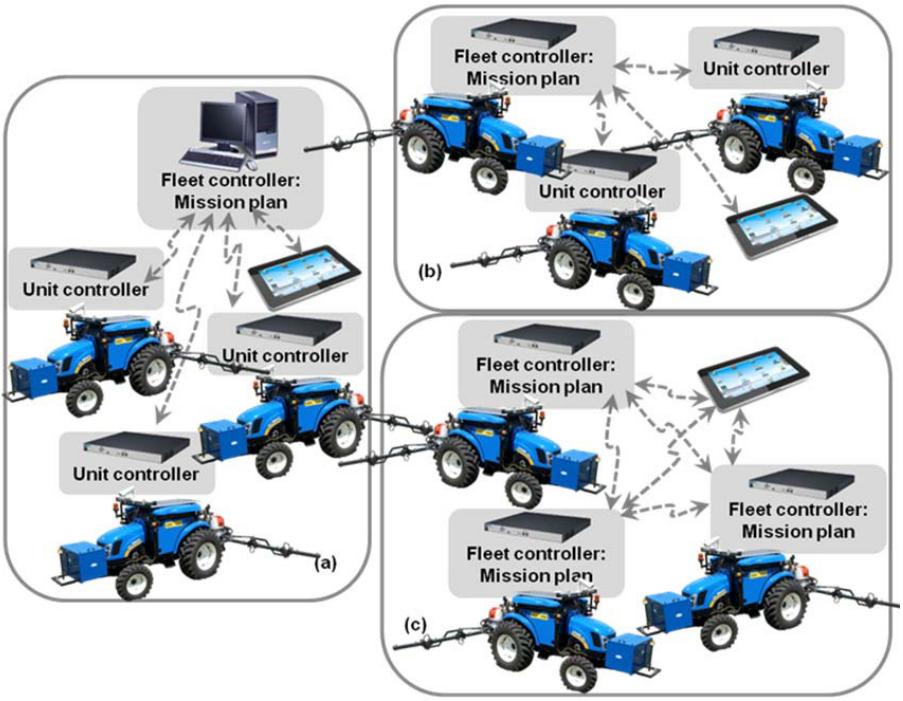


Fig. 7. Different hardware topologies for fleets of robots

This topology is well suited to the hardware architecture presented in Section 2 for the ground mobile unit, where the use of a compactRIO system, as main controller, allows the communication with others similar systems to be direct and without the development of drivers and communication protocols, thanks to the LabVIEW utilities for sharing information between National Instruments systems.

5 Conclusions

The hardware architecture presented in this article pursues to configure a well-organized fleet of robots for efficient agricultural tasks minimizing the hardware of devices and maximizing deterministic performances. The result is a simple, powerful, robust and reliable system in charge of the navigation control of a fleet of autonomous mobile robots as well as the control of the associated agricultural machinery elements.

The proposed architecture for the centralization of the Main Controller and the principal sensory systems provides some advantages for a future sensor fusion. In addition, in an autonomous agricultural application, when the environment exhibits changing characteristics of light, soil and crop, among others, the sensory system is required to achieve more complex tasks, which leads to the problem of overcharging the Main Controller due to both the execution of multiple tasks in the same controller and the high use of resources for sensory fusion tasks. Nevertheless, in the proposed solution, this overuse is compensated by the Main controller characteristics and its ability of executing many processes in parallel and in real-time as well as the possibility of implementing very specific and time-critical operations in a FPGA device.

This study is an attempt to increase the robustness of autonomous agriculture robots and fleets of robots by reducing the hardware of the equipment onboard the mobile units and easing the integration of different sensors, devices and software modules developed by professionals of different areas and skills.

Acknowledgments. The research leading to these results has received funding from the European Union's Seventh Framework Programme [FP7/2007-2013] under Grant Agreement nº 245986.

References

1. Bautin, A., Simonin, O., Charpillet, F.: Towards a communication free coordination for multi-robot exploration. In: Proceedings of the 6th National Conference on Control Architectures of Robots, Grenoble, France, pp. 8–14 (May 2011)
2. Blackmore, B.S., Stout, W., Wang, M., Runov, B.: Robotic agriculture – the future of agricultural mechanisation? In: Proceedings of the 5th European Conference on Precision Agriculture, Uppsala, Sweden, June 9-12, pp. 621–628 (2005)
3. Bouraqadi, N., Fabresse, L., Doniec, A.: On Fleet Size Optimization for Multi-Robot Frontier-Based Exploration. In: Proceedings of the 7th National Conference on Control Architectures of Robots, Nancy, France, May 10-11 (2012)
4. Carballido, J., Perez-Ruiz, M., Agüera, J.: Design, development and lab evaluation of a weed control sprayer to be used in robotic system. In: Proceedings of the First International Conference on Robotics and Associated High Technologies and Equipment for Agriculture (RHEA 2012), Pisa, Italy, September 19-21, pp. 23–30 (2012)
5. Cartade, P., Lenain, R., Thuilot, B., Benet, B., Berducat, M.: Motion Control of a heterogeneous fleet of mobile robots: Formation control for achieving agriculture task. In: Proceedings of the International Conference on Agricultural Engineering (CIGR-AgEng 2012), Valencia, Spain, July 8-12 (2012)
6. Conesa-Muñoz, J., Ribeiro, A., Andujar, D., Fernandez-Quintanilla, C., Dorado, J.: Exploring the effect of turning maneuvers on the multi-path planning of a robot fleet in agricultural tasks. In: Proceedings of the First International Conference on Robotics and Associated High Technologies and Equipment for Agriculture (RHEA 2012), Pisa, Italy, September 19-21, pp. 55–60 (2012)

7. Emmi, L., Pajares, G., Gonzalez-de-Santos, P.: Integrating robot positioning controllers in the SEARFS simulation environment. In: Proceedings of the First International Conference on Robotics and Associated High Technologies and Equipment for Agriculture (RHEA 2012), Pisa, Italy, September 19-21, pp. 151–156 (2012)
8. Emmi, L., Paredes-Madrid, L., Ribeiro, A., Pajares, G.: Pablo Gonzalez-de-Santos. Fleets of Robots for Precision Agriculture: a Simulation Environment 40(1), 41–58 (2013)
9. Hinterhofer, T., Tomic, S.: Wireless QoS-enabled Multi-Technology Communication for the RHEA Robotic Fleet. In: International Workshop on Robotics and Associated High Technologies and Equipment for Agriculture (RHEA 2011), Montpellier, France, pp. 173–186 (September 9, 2011)
10. Li, M., Imou, K., Wakabayashi, K., Yokoyama, S.: Review of research on agricultural vehicle autonomous guidance. International Journal of Agricultural and Biological Engineering 2, 1–16 (2009)
11. Montalvo, M., Guerrero, J.M., Romeo, J., Emmi, L., Guijarro, M., Pajares, G.: Automatic expert system for weeds/crops identification in images from maize fields. Expert Systems with Applications 40, 75–82 (2013)
12. Nagasaka, Y., Saito, H., Tamaki, K., Seki, M., Kobayashi, K., Taniwaki, K.: An Autonomous Rice Transplanter Guided by Global Positioning System and Inertial Measurement Unit. Journal of Field Robotics 26, 537–548 (2009)
13. Nørremark, M., Griepentrog, H.W., Nielsen, J., Søgaard, H.T.: The development and assessment of the accuracy of an autonomous GPS-based system for intra-row mechanical weed control in row crops. Biosystems Engineering 101, 396–410 (2008)
14. Peleg, D.: Distributed Coordination Algorithms for Mobile Robot Swarms: New Directions and Challenges. In: Pal, A., Kshemkalyani, A.D., Kumar, R., Gupta, A. (eds.) IWDC 2005. LNCS, vol. 3741, pp. 1–12. Springer, Heidelberg (2005)
15. Peruzzi, A., Frasconi, C., Martelloni, L., Fontanelli, M., Raffaelli, M.: Application of precision farming to maize and garlic in the RHEA project. In: Proceedings of the First International Conference on Robotics and Associated High Technologies and Equipment for Agriculture (RHEA 2012), Pisa, Italy, September 19-21, pp. 55–60 (2012)
16. Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K., Stentz, A.: The Demeter system for automated harvesting. Autonomous Robots 13, 9–20 (2002)
17. Reid, J.F., Zhang, Q., Noguchi, N., Dickson, M.: Agricultural automatic guidance research in North America. Computers and Electronics in Agriculture 25, 155–167 (2000)
18. RHEA: A robot fleet for highly effective agriculture and forestry management, http://www.rhea-project.eu/(accessed on January 16, 2013)
19. Urmson, C., et al.: Autonomous Driving in Urban Environments: Boss and the Urban Challenge. Journal of Field Robotics 25(8), 425–466 (2008)
20. Vieri, M., Lisci, R., Rimediotti, M., Sarri, D.: The innovative RHEA airblast sprayer for tree crop treatment. In: Proceedings of the First International Conference on Robotics and Associated High Technologies and Equipment for Agriculture (RHEA 2012), Pisa, Italy, September 19-21, pp. 93–100 (2012)
21. Wu, D., Zhang, Q., Reid, J.F.: Adaptive steering controller using a Kalman estimator for wheel-type agricultural tractors. Robotica 19, 527–533 (2001)
22. Xue, J., Zhang, L., Grift, T.E.: Variable field-of-view machine vision based row guidance of an agricultural robot. Computers and Electronics in Agriculture 84, 85–91 (2012)

Part VII
Humanoid Robots

REEM: A Humanoid Service Robot

Francesco Ferro and Luca Marchionni

Pal Robotics S.L.,
Barcelona, Spain
info@pal-robotics.com

Abstract. REEM is a humanoid service robot designed to provide useful applications in public spaces and to assist people in domestic environments. REEM is the result of several years of research in real scenarios. Its main functionalities are autonomous navigation, person detection and recognition, speech synthesis and recognition.

1 Introduction

Service robots are an extensive category of robots which perform services useful to the well-being of humans, which include all non-industrial applications. The main restriction which a service robot must accomplish is human safety. For this reason, REEM, a humanoid service robot developed by Pal Robotics, has been designed with extensive network of sensors which allow to detect humans and its environment and react consequently, giving priority to human safety, afterwards to environment protection and finally task accomplishment. Safety restrictions affected REEM's design to include passive safety by means of soft material covers, real-time actuator control, object and people detection, environment awareness, effective HRI, safe wireless communications, etc. REEM is designed putting special attention in its aesthetics and its functionalities to make it suitable for application in real human environments. Several experiments in shopping malls, hospitals, museums and trade fairs demonstrated that REEM could provide useful applications assisting and entertaining people in public spaces.

2 REEM

REEM has been designed with a humanoid shape and a nice appearance. It has a compact wheeled base that allows it to move through irregular terrains, a torso and two arms with hands that it uses to interact with its surroundings. The robot is equipped with 26 actuators and numerous sensors. Laser range finders are installed in its base and torso allowing it to navigate and localize with state of the art SLAM techniques robustly, even in crowded places. Additionally REEM is surrounded by ultrasound and infrared sensors that in conjunction with the lasers allow the robot to navigate safely among people. On the head there is a stereo system that gives the robot the functionality to detect and identify places, objects and people. Thanks to stereo microphones also placed in the head the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_38,

robot can be operated by voice. The microphone array also allows to localize the source of commands helping the robot to filter external disturbances and focus its attention on the users. The humanoid appearence is really important because for being engaged in social scenarios which are compelling and familiar to humans, the robot has to provide a social communicative functionality that is natural and intuitive. This can be achieved if appearance and functionality fit the robots tasks and robots are as self-explaining as possible [2].

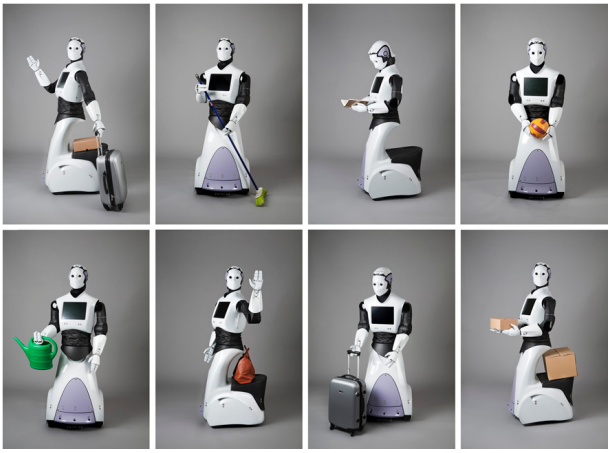


Fig. 1. REEM service robot

3 Autonomous Behaviour

REEM service robot performs autonomous mapping, localization, planning and obstacle avoidance in large public spaces as shopping malls or trade fairs, where the environment could be crowded and affected by significant modifications over short periods of time [1].

Safety is a critical feature when the robot moves autonomously. REEM is equipped with several sensors that enable the detection of both static and dynamic obstacles. When guiding people or when wandering around the environment the robot adapts its speed based on the distance to the closest obstacle. To prevent the robot from moving towards dangerous location (stairs),virtual barriers could be specified on the map used by the robot. Furthermore, to help users in predicting its movements, REEM uses LED indicators and speech synthesis to provide visual and sound feedback of what it is doing. When for example REEM is going to turn left, it moves the head towards that direction and its left ear's LED starts blinking, to show to the users its intentions. REEM has been used in several shopping malls and exhibition centers as guide and dynamic information point:

Fig. 2. REEM acting as guide in a crowded event at La Caixa Science Museum Barcelona 2012

- GLORIES: Shopping Mall, Barcelona, Spain (∼36.000 sqm)
- L'ILLA: Shopping Mall, Barcelona, Spain (∼50.000 sqm)
- ADNEC: Exhibition Center, Abu Dhabi, UAE (∼90.000 sqm)

4 HRI

A service robot should be easy to use and the user should perceive that the robot is aware of his presence. To achieve these goals REEM has an intuitive touchscreen interface, can detect and recognize people and can communicate through speech.

4.1 Touchscreen

REEM has a multimedia touchscreen from where the user can change language and access to a variety of applications. The screen could be used to show pictures or videos when the robot is working as entertainer or information point. REEM touchscreen contents could be updated remotely through a secure wireless connection and touchscreen usage statistics are collected in a database to allow developers to improve the user experience of the interface.

4.2 Speech Recognition and Synthesis

Speech is the most natural means for human communication and for interact with humans, a robot has to be able to generate and understand speech. REEM has a system that gives it the capability of speech synthesis in several languages. Speaking allows REEM to give help, indicate its intention, warn people around him or give information, in a really natural way. In order to remove the inner-noise generated by the robot we are currently applying MCRA-based spectral

Fig. 3. REEM touchscreen home page

subtraction [5] and for interference removal we are able to remove signals coming from a different direction than the one we are interested in (i.e., user speech). We are basing our algorithms on the work of [6]. Our recent findings indicate that by combining these techniques it is possible to reduce the recognition error by 25% in extreme noisy conditions (i.e., $SNR < 0dB$).

4.3 People Detection and Recognition

Person detection provides the input data to higher level processes like person tracking or behaviors to interact with people. REEM exploits its base laser and its stereo camera in order to detect persons, by mean of several algorithms: a legs detector [3], a full body detector [4] and a face detector. REEM is able to recognize a person by means of matching its face to a database. Therefore, person recognition is restricted to the range of distances in which the face detector is effective and in REEM this distance is up to 1.5 meters.

5 Conclusions

REEM robot has been used in several real scenarios and the robot was required to provide information, interact with people and attract them while moving safely around.

We conclude that service robots like REEM should have the following features:

– Humanoid shape: people do prefer interact with a robot that shows a (kind of) human appearance. Every time REEM shows on public places, people starts taking picture and interacting with it. REEM acts as an attractor because of its shape and lively appearance.

- Autonomous behaviour: people prefer robots that are autonomous and decide by themselves. When we use the remote control to control REEM to perform a show in an event, people show deception when they discover the human behind. Instead, people gets amazed when they observe how the robot moves by itself through the crowd or how it follows a face.
- Human robot interaction: the robot should be easy to use and the user should perceive that the robot is aware of his presence. Face tracking and hand shaking applications are successful because they give the user the perception of the robot as a living machine.

We think that the future of Service Robots goes through the successful implementation of those three characteristics: humanoid body, autonomous behaviour and human robot interaction. The race to improve those features will mark the agenda of the Service Robots industry for the next years.

References

1. Marchionni, L., Tellez, R.: Current challenges in humanoid navigation in dynamic environment. In: Proc. of the IEEE-RAS Proc. Conf. on Humanoid Robots, HUMANOIDS (2011)
2. Goetz, J., Kiesler, S., Powers, A.: Matching robot appearance and behavior to tasks to improve human-robot cooperation. In: Proc.of the 12th IEEE Workshop on Robot and Human Interactive Communication, Roman (2003)
3. Arras, K.O., Mozos, O.M., Burgard, W.: Using boosted features for the detection of people in 2D range data. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, ICRA (2007)
4. Paisitkriangkrai, S., Shen, C., Zhang, J.: Real-time pedestrian detection using a boosted multi-layer classifier. In: The 8th Workshop on Visual Surveillance, Marseille, France (2008)
5. Cohen, I., Berdugo, B.: Noise Estimation by Minima Controlled Recursive Averaging for Robust Speech Enhancement. IEEE Signal Processing Letters 9(1), 12–15 (2002)
6. Kim, C., Kumar, K., Stern, R.M.: Binaural Sound Source Separation Motivated by Auditory Processing. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 4 (2011)

Locomotion Control of a Biped Robot through a Feedback CPG Network

Julián Cristiano[1], Domènec Puig[1], and Miguel Angel García[2]

[1] Department of Computer Science and Mathematics, Rovira i Virgili University, Tarragona 43007, Spain
julian11495@yahoo.com, domenec.puig@urv.cat
[2] Department of Electronic and Communications Technology, Autonomous University of Madrid, Madrid 28049, Spain
miguelangel.garcia@uam.es

Abstract. This paper proposes a locomotion control system for biped robots by using a network of Central Pattern Generators (CPGs) implemented with Matsuoka's oscillators. The proposed control system is able to control the system behaviour with a few parameters by using simple rhythmical signals. A network topology is proposed in order to control the generation of trajectories for a biped robot in the joint-space both in the sagittal and coronal planes. The feedback signals are directly fed into the network for controlling the robot's posture and resetting the phase of the locomotion pattern in order to prevent the robot from falling down whenever a risk situation arises. A Genetic Algorithm is used to find optimal parameters for the system in open-loop. The system behaviour in closed-loop has been studied and analysed through extensive simulations. Finally, a real NAO humanoid robot has been used in order to validate the proposed control scheme.

Keywords: Biped locomotion, CPGs, Humanoid robots.

1 Introduction

Biped locomotion is a complex problem that requires a robust control system that deals with several unexpected situations that can arise during the robot's motion. Thus, the locomotion system should allow a natural integration of three fundamental aspects: the robot's body, the environment and the control system. This control system basically consists of two stages. The first stage is responsible for the generation of the locomotion pattern, whereas the second stage is responsible for introducing feedback into the system in order to modulate the current locomotion pattern such that stability may be recovered whenever unexpected situations appear while the robot walks.

The generation and control of locomotion patterns have been addressed in several ways for humanoid robots that use electric motors for controlling the motion of their joints. Basically, the main objective is the generation of reference signals for the robot's joints, either as an angular displacement or as a torque in

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_39,

order to describe a stable motion. These signals are mainly generated through either pre-computed trajectories or trajectories computed in real-time through diverse control schemes. For example, the most common methodology to generate biped locomotion is by using simplified dynamic models of the humanoid robot's body in order to describe the trajectories for the robot's arms and legs in the task-space. These approaches are mainly based on a controller of the position of the Zero Moment Point (ZMP), which is used to ensure the robot's stability [1].

Alternatively, biological studies have shown that neural networks control the locomotion of vertebrate and invertebrate animals. These networks are called Central Pattern Generators. They are constituted by a set of neurons that can generate complex multi-dimensional rhythmic signals with simple input signals for controlling coordinated periodic movements. Currently, there is a growing interest in CPGs for controlling different types of robots [2]. Thus, some approaches recently used to solve the biped locomotion problem have been inspired by CPGs. The first studies about biped locomotion with CPGs were conducted by Taga, who investigated the use of CPGs for controlling the walking of a simulated humanoid robot on a 2D plane [3], [4].

The control systems for biped locomotion based on CPGs can be classified into CPG-joint control methods (e.g. [5]) and CPG-task-space control methods (e.g. [6], [7]). The first one aims at the appropriate generation and control of the multidimensional control signals in the joint-space through a CPG network that generates the coordinated locomotion. In the second approach, CPGs are used in the task-space to describe the Cartesian-space trajectories for the robot's joints. In that case, it is necessary to solve the inverse kinematics to determine the joint signals in order to control the biped locomotion.

In this work, a control system based on CPG-joint control methods is proposed in order to generate simple rhythmical signals for locomotion control of biped robots. The system behaviour is characterized by a reduced number of parameters. In an off-line stage, the parameters that characterize the walking pattern are determined by means of dynamics simulations. Afterwards, in an on-line stage, those parameters are directly tested on the humanoid robot in order to validate the obtained locomotion pattern. This paper is organized as follows. Section 2 describes the proposed controller based on a CPG network for generation of biped locomotion patterns. Section 3 presents some automatic feedback strategies implemented in the proposed CPG network. Section 4 discussed the obtained experimental results. Finally, conclusions and future work are given in Section 5.

2 Generation of Biped Locomotion Patterns

The proposed biped motion controller is based on a CPG network. The goal is to generate the signals for the angular displacement of each of the robot's joints in order to describe a valid locomotion pattern. The combination of parameters of the proposed network is found by evaluating the locomotion performance through dynamics simulations by applying a genetic algorithm.

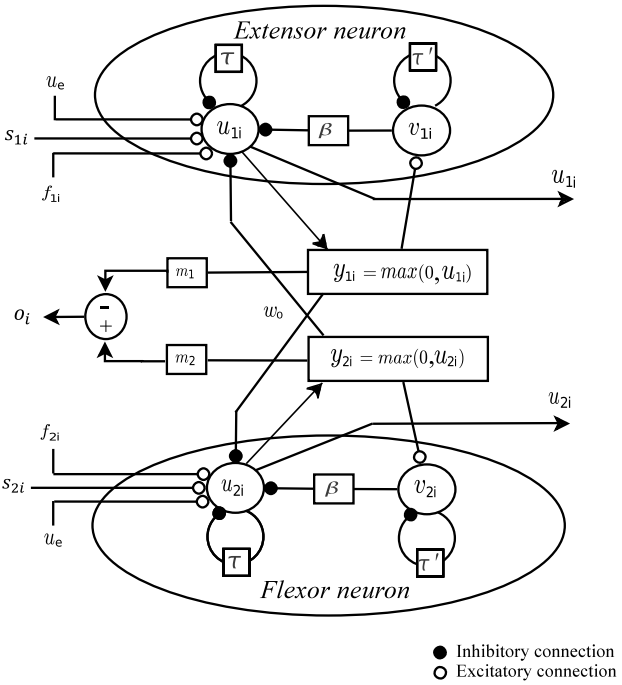


Fig. 1. Matsuoka's non-linear oscillator generalized to be applicable to a system with multiple degrees of freedom, as proposed in [10]

2.1 Basic Oscillator

The proposed CPG network is modelled by means of a set of interconnected non-linear oscillators previously proposed by Matsuoka in [8], [9]. The formulation of the Matsuoka's oscillator has been generalized to be applicable to a system with multiple degrees of freedom, as proposed in [10]. In the present work, the connections between oscillators are established in a single direction in order to control the desired output phases corresponding to the outputs of the CPG network. Each non-linear oscillator consists of two tonically excited neurons with a self-inhibition effect, which are reciprocally linked via inhibitory connections, as shown in Fig. 1. Each oscillator has four state variables $(u_{1i}, v_{1i}, u_{2i}, v_{2i})$. Its behaviour is described by (1) and (2). In this mathematical model, subscript 1 corresponds to the extensor neuron and subscript 2 to the flexor neuron, with num representing the number of oscillators.

$$\begin{cases} \tau \dot{u}_{1i} = -u_{1i} - w_0 y_{2i} - \beta v_{1i} + u_e + f_{1i} + a s_{1i} \\ \tau' \dot{v}_{1i} = -v_{1i} + y_{1i} \end{cases}$$

$$y_{1i} = max(0, u_{1i}), \qquad i = 1, ..., num \tag{1}$$

$$\begin{cases} \tau \dot{u}_{2i} = -u_{2i} - w_0 y_{1i} - \beta v_{2i} + u_e + f_{2i} + a s_{2i} \\ \tau' \dot{v}_{2i} = -v_{2i} + y_{2i} \end{cases}$$

$$y_{2i} = max(0, u_{2i}), \quad i = 1, ..., num \tag{2}$$

The external input u_e affects the amplitude of the oscillator output, whereas its frequency is determined by the time constants τ and τ' [11]. The set of parameters must satisfy some requirements in order to yield stable oscillations [12]. The terms f_{1i} and f_{2i} are feedback variables. They can be used to control the output amplitude and phase. Parameters s_1 and s_2 represent the interaction between neighbouring oscillators within the CPGs. They are defined as (3), where w_{ij} represents the unidirectional connection weight between a master oscillator, O_j, and a slave oscillator, O_i. Only two types of connections between oscillators are possible:

1) Excitatory connection: The weight w_{ij} is set to 1 in this case.
2) Inhibitory connection: In this case, the weight w_{ij} is set to -1.

$$\begin{aligned} s_{1i} &= w_{ij} u_{1j} \\ s_{2i} &= w_{ij} u_{2j} \end{aligned} \tag{3}$$

The output signals of the oscillators are calculated using (4). They are used to directly control the joints of the humanoid robot.

$$o_i = -m_1 y_{1i} + m_2 y_{2i} \tag{4}$$

In order to modulate the frequency of the oscillator, an additional parameter, k_f, is introduced as proposed in [13]. The time constants in (1) and (2) are thus reformulated as:

$$\begin{aligned} \tau &= \tau_o k_f \\ \tau' &= \tau'_o k_f, \end{aligned} \tag{5}$$

where τ_o and τ'_o are the original time constants of the Matsuoka's oscillator. For certain values of its parameters, an oscillator will generate a periodic oscillation by itself. However, when a network of oscillators is set, they all oscillate together according to the network connections, converging to a specific pattern and limit cycle. Table 1 shows the internal values used for each non-linear oscillator. Figure 2 shows the output signals for the aforementioned parameters. Basically, the objective is to generate a periodic output signal with a stable limit cycle.

2.2 Proposed CPG Network

The main goal of the proposed CPG network is to generate stable patterns for controlling the motion of the humanoid robot. This network generates the motion of the robot arms and synchronizes them with the motion of its legs in order to yield a more stable walking pattern. The proposed CPG network follows a master-slave topology, in which a central non-linear oscillator is used to drive

Table 1. Oscillator parameters

Parameter	Value	Parameter	Value
τ_o	0.2800	u_e	0.4111
τ_o'	0.4977	$m1, m2$	1
β	2.5000	a	1
w_0	2.2829	k_f	0.2193

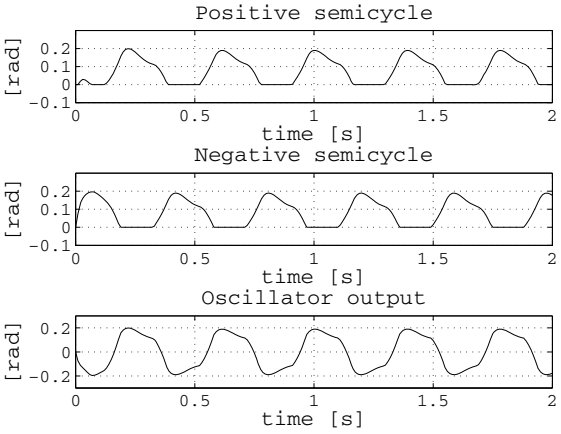


Fig. 2. Oscillator outputs for the fixed internal oscillator parameters defined in Table 1

the different joints of the robot through slave oscillators associated with each of those joints.

The central oscillator is referred to as a pacemaker oscillator, since it generates the first electrical impulse or master signal that is propagated across the other oscillators in the different chains of the robot (left arm in the sagittal plane, right arm in the sagittal plane, left leg in the sagittal and coronal planes and right leg in the sagittal and coronal planes). The concept of pacemaker oscillator was introduced in [14]. However, they did not utilize it to directly control the angular displacement of joints with the CPG outputs. If the pacemaker oscillator does not have any input or feedback signal, it exclusively oscillates according to its internal parameters.

The proposed CPG network has been designed in order to imitate the human gait, which has the following features:

1. The arms' motion in the sagittal plane is in anti-phase.
2. The legs' motion in the sagittal plane is in anti-phase.
3. The two motions mentioned above are in anti-phase, that is, the motion of the right arm is synchronized with the motion of the left leg, whereas the motion of the left arm is synchronized with the motion of the right leg.

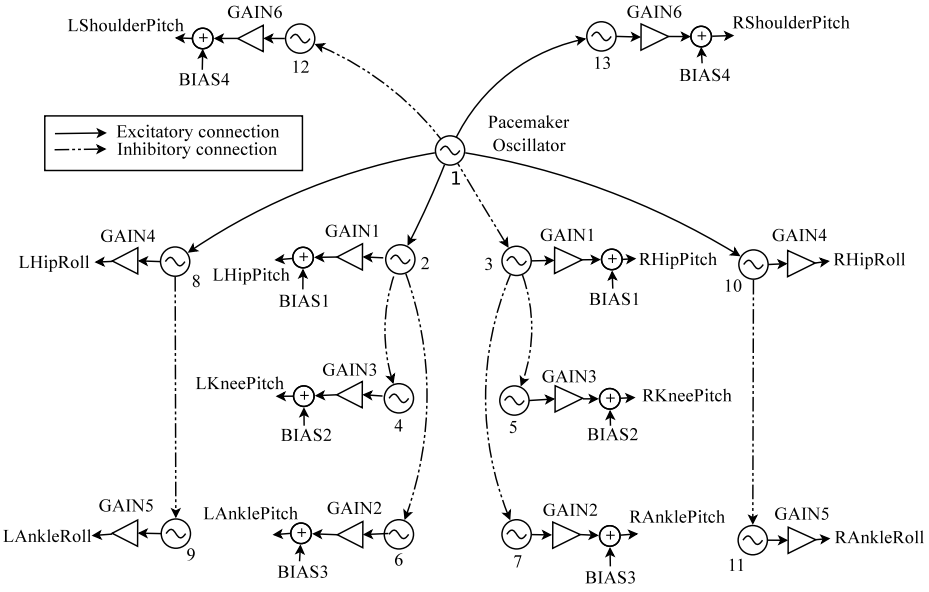


Fig. 3. Proposed topology for the CPG network

4. The coronal motion of legs is synchronized with the sagittal motion of legs in order to generate the periodic motion by changing the location of the center of masses.

The large number of parameters is one of the main drawbacks when dealing with CPG networks. In order to minimize this problem, all oscillators have been configured with the same parameters presented in Table 1. The parameters of the CPG network shown in Fig. 3, which characterize the locomotion pattern of a biped robot are: seven gains (k_f, GAIN1 ,GAIN2, GAIN3, GAIN4, GAIN5, GAIN6) and four offsets (BIAS1, BIAS2, BIAS3, BIAS4). Parameter k_f can be utilized for changing the walking speed, since it controls the locomotion pattern's frequency, achieving an extra control on the robot's velocity. This is an important feature, since it is possible to modulate the locomotion pattern's frequency with a single parameter.

2.3 Automatic Estimation of CPG Network Parameters through Genetic Algorithms

Evolutionary algorithms are typically used to solve high-dimensional optimization problems. These algorithms have proven to be a good approach for finding the parameters associated with CPG networks. A genetic algorithm has been applied off-line in order to calculate the CPG network parameters with the objective of describing an optimal locomotion pattern in straight line. Each individual of the genetic algorithm represents a different combination of parameters for the

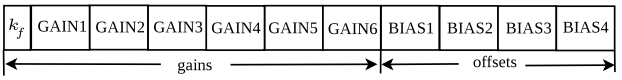


Fig. 4. Chromosome structure

proposed CPG network. In particular, each individual is modelled by a chromosome with 11 traits that represent the combination of parameters of the CPG network, as shown in Fig. 4. The genetic algorithm has been initialized in this work with a randomly selected initial population of 200 individuals. This number was heuristically set after a large amount of simulations. The locomotion pattern associated with every individual has been evaluated during 16 seconds using a dynamics simulator and a 3D synthetic model of the robot in order to allow the robot to walk a significant distance for calculating its fitness value. If the robot falls down or strongly oscillates during that evaluation period, the fitness value is set to zero for the current individual. Otherwise, its fitness value is calculated as follows. The individual with the best fitness score proceeds to the next generation. The other individuals are chosen based on the roulette wheel selection rule. In particular, each individual is given a probability of being selected that is proportional to its fitness value. This rule makes it possible to select individuals with a low score, thus preventing a fast convergence of the genetic algorithm to a local maximum. The individual with the best fitness score and the selected individuals from the current generation constitute a mating pool. Crossover and mutation procedures are then applied in order to obtain the individuals of the next generation. The simulation stops when a maximum number of generations is reached or the variation of the fitness function is lower than a given threshold ϵ ($\epsilon = 0.01$ in this work). The fitness function used for sorting the individuals in each generation has been formulated as (6), where *vel* represents the average straight line velocity, *dev* is the lateral deviation distance with respect to the straight line, α and γ are weighting coefficients.

$$f = \alpha(vel) - \gamma(dev) \tag{6}$$

3 Introduction of Feedback Strategies into the System

This section presents some automatic feedback strategies implemented in the proposed CPG network in order to modulate the locomotion patterns obtained with the methodology explained above. This modulation is performed according to the analysis of the available feedback signals in order to compensate for internal mismatches or external perturbations. It takes advantage of the ability of the CPGs for adapting their behaviour through external feedback variables introduced in the CPG model. The feedback strategies are introduced into the system as explained below.

3.1 Phase Resetting Controller

Whenever a humanoid robot walks on flat terrain, it interacts with the floor through its feet soles. If the robot is walking at a constant speed on flat terrain, the elapsed times between the transition from the left to the right foot and from the right to the left foot should be approximately equal. This indicates that the walking pattern is symmetric and the robot is correctly interacting with the floor.

However, if the robot is walking on irregular terrain, the interaction times with the floor can be modified. For instance, if the robot steps on an obstacle or if it is experimenting an external perturbation, the interaction time could be smaller or higher for one or both interaction times. The information extracted from these time measures is used to synchronize the interaction between the feet and the floor by means of phase resetting. It aims to synchronize the pattern generation with the current interaction between the robot and the environment.

As explained above, the model proposed by Matsuoka for the non-linear oscillator is used in this work. To our knowledge, the phase resetting of the oscillator's output has not been implemented in Matsuoka's oscillator to control the biped locomotion. In the study presented in [15] by the same author, some initial studies about the phase resetting behaviour in the oscillator are presented. In that work, two phase response curves of the oscillator for slow and fast dynamics are presented. These curves show that the phase shift introduced in the oscillator output depends of the oscillator internal parameters and also on the current phase of the output signal. These phase response curves have a non-linear behaviour. In order to adequately control the phase resetting through the current interaction times between the robot and the floor, it is necessary to exactly impose the needed phase difference. By using the phase response curves, the phase resetting cannot be easily controlled since it can happen in whatever instant of the gait due to irregularities in the walking terrain or perturbations that can be applied to the robot.

Therefore, an alternative strategy for controlling the phase resetting is proposed in this work by using the feedback signals of the original Matsuoka's oscillator model in order to adequately control the phase of the oscillator's output signal. This is done by controlling the current state of the oscillator variables through the feedback signals in order to reset the phase of the output signal. The introduction of this feedback in the oscillator model is presented below.

The mathematical model of each oscillator has two feedback variables, referred to as f_1 and f_2. If these variables have a positive value, they can be used to control the oscillator's output amplitude in real-time. Otherwise, if they are negative and with specific values, it is possible to control the time during which the output signal is disabled and then turn it on again in order to impose the required phase for the output signal. If these variables are set to zero, there is no effect on the oscillator output.

In this work, the phase resetting mechanism is implemented in the pacemaker oscillator. The output signal imposed by the central oscillator is propagated along the slave oscillators in order to synchronize the interaction between the

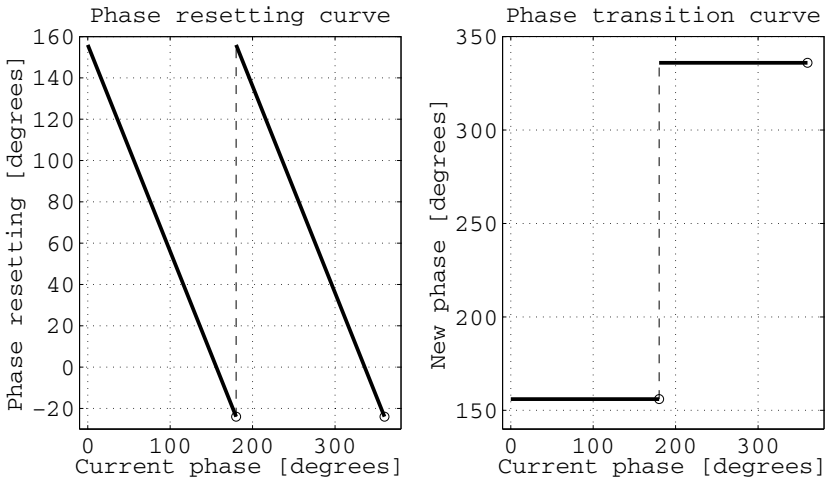


Fig. 5. Phase resetting

robot and the floor in real-time, aiming to prevent the robot from falling down due to an unstable situation.

The exact moment of impact between each robot's foot sole and the floor is detected through the analysis of the signals provided by the force sensors located in the feet soles. These force measures are used to calculate the elapsed time between the change in the robot's support leg. If this time is smaller or higher than some fixed limits determined by the current walking frequency, the phase resetting mechanism is activated in order to synchronize the interaction between the robot feet and the floor. The phase resetting mechanism implemented for the pacemaker oscillator is described in Fig 5. The zero phase for the pacemaker oscillator signal is defined as the point of transition of the pacemaker oscillator amplitude from a negative value to a positive value.

3.2 Pose Controller

In a bipedal gait, the robot's trunk pose during motion must be controlled since it has a higher concentration of mass and, therefore, it plays an important role in the robot's stability. A feedback strategy to control the robot's pose through the control of the pose in the sagittal plane is implemented. In order to maintain the robot's trunk in a correct pose, parameter BIAS1 is modified for controlling the pitch angle of the trunk according to:

$$BIAS1_{new} = BIAS1 + K_1(\theta_{trunk} - \hat{\theta}_{trunk}), \tag{7}$$

where $BIAS1_{new}$ is the new value for the parameter BIAS1, $\hat{\theta}_{trunk}$ is the reference value for the trunk inclination in the sagittal plane and θ_{trunk} is the current inclination in the sagittal plane for the trunk provided by the robot sensors.

Table 2. Fixed angles for other joints

Joint name	Angle [radians]	Joint name	Angle [radians]
HeadPitch	0	LShoulderRoll	0.23
HeadYaw	0	LElbowYaw	-1.51
RShoulderRoll	-0.23	LElbowRoll	-0.5
RElbowYaw	1.51	HipYawPitch	0
RElbowRoll	0.5		

4 Experimental Results

In order to validate the proposed controller, a NAO humanoid robot has been used [16]. The proposed CPG network, shown in Fig. 3, aims at imitating the main features of the human gait in order to describe a biped motion by specifying the phase differences for 12 joints of the sagittal and coronal planes. This network is able to generate a 3D motion in both simulated and real environments. The motion control of the arms is performed through position control of the shoulders in the sagittal plane. The other joints in the arms, hip and head have constant angles that aim to mimic an upright walking pose. These angles are shown in Table 2.

The robot simulations have been performed with Webots [17], a commercial dynamics simulator that includes the physical models of different real robots and is able to simulate the dynamic interaction between the robots and the simulated environment.

The genetic algorithm works in coordination with the CPG network and the robot simulator in order to estimate the best combination of CPG parameters within the predefined search space. The aim is to control the signals of the robot joints through the CPG network in order to generate an optimal locomotion pattern. The genetic algorithm was initialized in this work with an initial population of 200 chromosomes. The crossover probability was set to 80 percent and the probability of mutation to 10 percent. Those values were heuristically set after a large number of simulations.

When the frequency of the locomotion pattern is higher than 2.28 Hz, the simulator is unable to reproduce the real interaction between the feet soles and the floor, leading to a noticeable deviation between the simulated motion pattern and the real one. Therefore, the maximum frequency allowed for the simulated locomotion pattern has been set to 2.28 Hz, which is reached with $k_f = 0.2$.

Table 3 shows the different limits that have heuristically been defined for the gains and offsets. Those limits define the search space of the genetic algorithm. Figure 6 shows the results obtained with the simulator in order to find out the optimal locomotion pattern in straight line. The weighting coefficients α and γ have heuristically been set to in 80 and 100 respectively. The CPG parameters shown in Table 4 were finally obtained through the proposed methodology. The total simulation time was 132 minutes on an Intel Core 2 Quad Q9400 processor at 2.66 GHz with 8GB of RAM.

Table 3. Genetic algorithm search space

CPG network parameters	Parameter range	CPG network parameters	Parameter range
GAIN1	0.01 - 1.00	BIAS1	-0.90 - 0.00
GAIN2	0.01 - 1.00	BIAS2	0.5 - 1.2
GAIN3	0.01 - 0.50	BIAS3	-0.70 - 0.00
GAIN4	0.01 - 0.50	BIAS4	1 - 2
GAIN5	0.01 - 0.50	k_f	0.2 - 0.5
GAIN6	0.01 - 1.00		

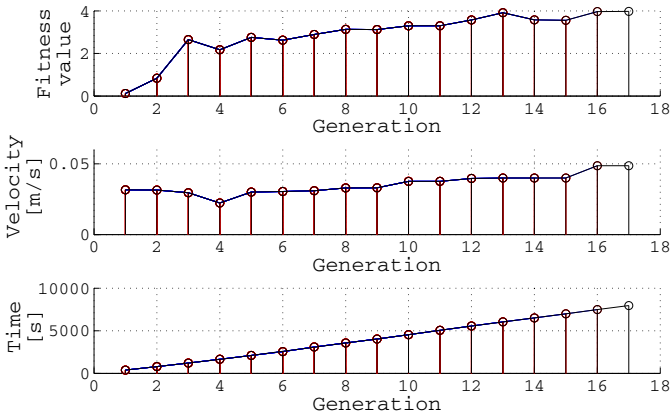


Fig. 6. Simulation results obtained with the genetic algorithm

Table 4. CPG network parameters found by the GA

CPG network parameters	Value	CPG network parameters	Value
GAIN1	0.24555	BIAS1	-0.60598
GAIN2	0.18665	BIAS2	0.70700
GAIN3	0.11685	BIAS3	-0.30590
GAIN4	0.40850	BIAS4	1.4797
GAIN5	0.43000	k_f	0.462
GAIN6	1.00000		

In order to feedback the system, the phase resetting mechanism described above was implemented in the pacemaker oscillator. One example of the phase resetting mechanism is shown in Fig. 7. The value for the parameter K1 in (7) was experimental set to 10 in order to control the robot's vertical pose.

For testing the robustness of the proposed closed-loop system, an environment with diverse classes of obstacles on the walking terrain was designed. Those

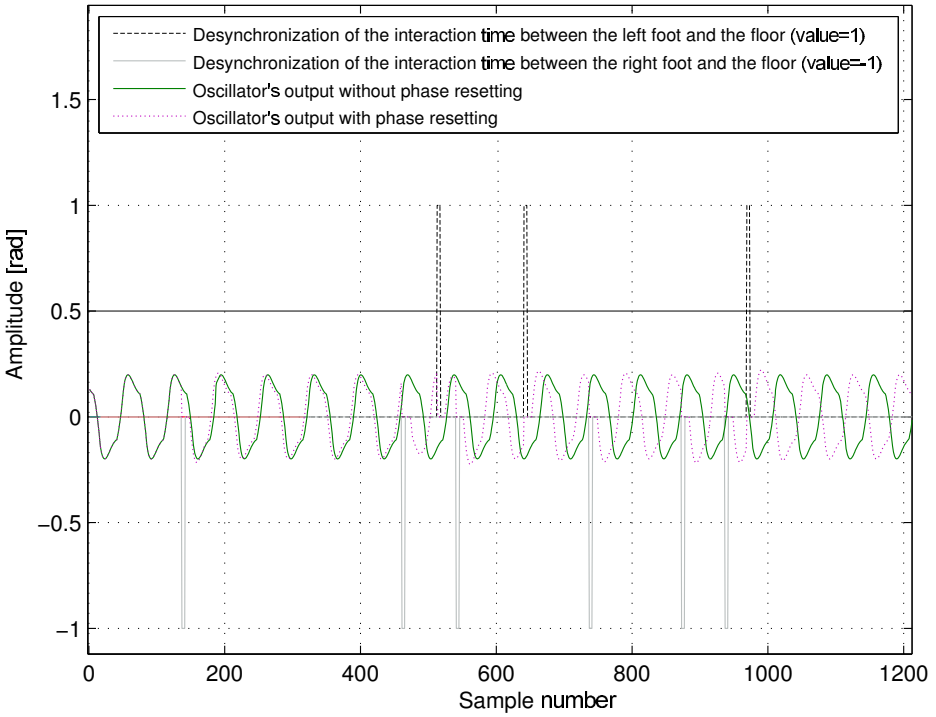


Fig. 7. Phase resetting example for the pacemaker oscillator

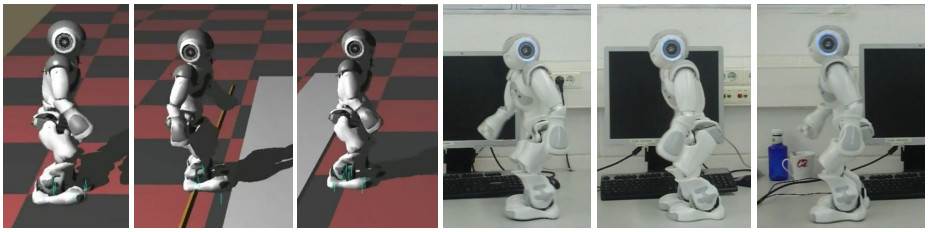


Fig. 8. Snapshots of the simulation and real experiments

obstacles have round, triangular and rectangular shapes such as is shown in the simulation experiments. Due to their shape, those obstacles produce forces in different directions in the robot's feet soles while it is walking on those obstacles and as a consequence, different types of perturbations are presented. These perturbations are used to test the robustness of the feedback system in order to mainly prevent the robot from falling down while it is exposed to unexpected situations. The system provides feedback to the locomotion pattern that is being generated on-line in order to modify the joint signals for preventing the robot from falling down. The proposed methodology has been validated on a real NAO

humanoid robot. Comparative videos between the results obtained in simulation and the real environments have been included in the companion website[1]. Some snapshots with simulated and real experiments are shown in Fig. 8.

5 Conclusions

A control architecture has been proposed to control the locomotion of a biped robot by using a CPG network based on Matsuoka's oscillators. A new CPG network has been proposed whose parameters are determined through evolutionary computation in order to generate optimal biped locomotion patterns. These patterns are generated in the joint-space by controlling the angular displacement of the robot's joints both in the sagittal and coronal planes. A feedback mechanism is introduced in the network to modulate the locomotion patterns according to the current sensory feedback in order to compensate for internal mismatches and external perturbations. The feedback is performed by controlling the robot's pose and the interaction between the robot and the floor through phase resetting of the pacemaker oscillator. The system behaviour is characterized with a few parameters using simple rhythmical signals.

Experimental results have validated the proposed methodology on a NAO humanoid robot. It can be easily adapted to other biped robots with a similar structure of joints after performing the corresponding changes (robot model, limits for the genetic algorithm search space, etc.).

Future work will include the study of the parameters that characterize the CPG network in order to find out a mathematical description that automatically finds the optimal values for the variables of the model.

Acknowledgment. The authors would like to thank Fundación Carolina and Rovira i Virgili University by the doctoral scholarship supporting Julián Cristiano.

References

1. Vukobratović, M., Borovac, B.: Zero-moment point - thirty five years of its life. International Journal of Humanoid Robots 1(1), 157–173 (2004)
2. Ijspeert, A.: Central pattern generators for locomotion control in animals and robots: A review. Neural Netw. 21(4), 642–653 (2008)
3. Taga, G., Yamaguchi, Y., Shimizu, H.: Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. Biol. Cybern. 65(3), 147–159 (1991)
4. Taga, G.: A model of the neuro-musculo-skeletal system for human locomotion. Biol. Cybern. 73, 113–121 (1995)
5. Righetti, L., Ijspeert, A.: Programmable central pattern generators: an application to biped locomotion control. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 1585–1590 (May 2006)

[1] Companion website: `http://deim.urv.cat/%7Erivi/NAO%5FROBOT2013.html`

6. Park, C.-S., Hong, Y.-D., Kim, J.-H.: Full-body joint trajectory generation using an evolutionary central pattern generator for stable bipedal walking. In: Proc. IEEE Int. Conf. Intell. Robot. Syst., pp. 160–165 (October 2010)
7. Liu, C., Chen, Q.: Walking control strategy for biped robots based on central pattern generator. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 57–62 (May 2012)
8. Matsuoka, K.: Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biol. Cybern. 52(6), 367–376 (1985)
9. Matsuoka, K.: Mechanisms of frequency and pattern control in the neural rhythm generators. Biol. Cybern. 56(5), 345–353 (1987)
10. Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Tomita, K., Kokaji, S.: Automatic locomotion design and experiments for a modular robotic system. IEEE/ASME Trans. Mechatronics 10(3), 314–325 (2005)
11. Williamson, M.: Robot arm control exploiting natural dynamics. Ph.D. Thesis, Massachusetts Institute of Technology (1999)
12. Matsuoka, K.: Analysis of a neural oscillator. Biol. Cybern. 104(4-5), 297–304 (2011)
13. Zhang, D., Poignet, P., Widjaja, F., Ang, W.T.: Neural oscillator based control for pathological tremor suppression via functional electrical stimulation. Control Eng. Pract. 19(1), 74–88 (2011)
14. Aoi, S., Tsuchiya, K.: Locomotion control of a biped robot using nonlinear oscillators. Auton. Robots 19, 219–232 (2005)
15. Nakada, K., Sato, Y.D., Matsuoka, K.: Tuning time scale parameter of piecewise linear oscillators for phase resetting control. In: The 21st Anual Conference of the Japanese Neural Network Society (December 2011)
16. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Mariner, B., Serre, J., Maisonnier, B.: The NAO humanoid: a combination of performance and affordability. In: IEEE - International Conference on Robotics and Automation (2009)
17. Michel, O.: Cyberbotics ltd. webots tm: Professional mobile robot simulation. Int. J. Adv. Rob. Syst. 1, 39–42 (2004)

Multi-modal Active Visual Perception System for SPL Player Humanoid Robot

Francisco Martín, Carlos E. Agüero, José M. Cañas, and Eduardo Perdices

Robotics Lab (GSyC), Rey Juan Carlos University,
Fuenlabrada, Madrid 28923, Spain
{fmartin,caguero,jmplaza,eperdices}@gsyc.es

Abstract. Robots detect and keep track of relevant objects in their environment to accomplish some tasks. Many of them are equipped with mobile cameras as the main sensors, process the images and maintain an internal representation of the detected objects. We propose a novel active visual memory that moves the camera to detect objects in robot's surroundings and tracks their positions. This visual memory is based on a combination of multi-modal filters that efficiently integrates partial information. The visual attention subsystem is distributed among the software components in charge of detecting relevant objects. We demonstrate the efficiency and robustness of this perception system in a real humanoid robot participating in the RoboCup SPL competition.

Keywords: robot soccer, active vision, multitarget tracker, humanoid, attention.

1 Introduction

Several international robot competitions and challenges have emerged in the last years. For instance the DARPA Grand Challenge and Urban Challenge[6], RoboCup[10], the FIRA Robot World Cup[13], and the DARPA Robotics Challenge. They aim to foster AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. The motivation of competition makes them attractive, and they serve as proof of concept of current technological limits, pushing them further. In particular, RoboCup chose soccer as a central topic, which is a complex and challenging scenario for robotics research as it is dynamic, with opponents, and allows the cooperation of several robots inside a team. It has the long term goal to develop a team of fully autonomous humanoid robots that can beat the human world champion team by 2050.

The RoboCup competition is organized into leagues according to the type of robots. In the Standard Platform League (SPL) [7] all teams use the same robot, and changes in hardware are not allowed, so the effort focuses on software. A fully featured SPL soccer player has to get information from the environment, mainly using its on board camera. It provides high volume data about the environment, and task relevant information must be extracted from them. Robot

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_40,

control decisions may be based exclusively on current image, but this suffers from some limitations. The camera scope is limited to a 60° field of view, and it is common to have occlusions of the objects and even false positives. Robots must identify and locate the ball, goals, lines and other robots. Having this information, the robot has to self-localise and plan the next action: move, kick, search another object, etc. The robot must perform all these tasks very fast in order to be reactive enough to be competitive in a soccer match.

This paper presents the visual perception system of our humanoid robots participating in the SPL (SPiTeam[21]), composed by a visual memory and an integrated attention subsystem. The aim is to improve perception in order to make good decisions and so unfold good behaviors. The visual memory is composed of a collection of Joint Probabilistic Data Association Filters (JPDAF) [1] [4] that store and update the relative position of objects, like goals or the ball, relative to the robot. This approach has been designed to cope with partial and ambiguous observations. Partial observations occur when an object is occluded or when the camera field of view is limited. The observations can also be ambiguous, if the perceived feature is not unique due to symmetries or similarities with other objects. We also propose a novel attention subsystem that controls the head movement and continually shifts the focus of attention so the camera looks at different areas of the scene providing new images to feed the visual memory and to update the object state estimates. This visual perception system has been integrated in the behavior-based architecture of the humanoid robot, named BICA [11].

Next, we review the state of the art in world modeling, attention and multi-object estimation, with emphasis on the other teams of the RoboCup SPL. The two main components of the proposed perceptive system, the visual memory and the visual attention module, are described in detail in sections 3 and 4, respectively. The analysis of the experiments conducted is detailed in section 5, while discussion generated by this work is presented in section 6.

2 Related Works

Researchers within the RoboCup community typically maintain an object representation known as the world model containing the position of relevant stimuli: ball, goals, robots, etc. The world model is updated using the instantaneous output of the detection algorithms or by running an extra layer that implements some filtering. The most commonly used filters are Kalman filter[22], its nonlinear variants Extended Kalman filter (EKF) or unscented Kalman filter (UKF)[20], particle filters[9], hybrid techniques, or even multi-modal algorithms[19].

A topic closely related to visual memory is the control of camera gaze. In the RoboCup environment, policies to decide when and how to direct the gaze to a particular point can be divided into three groups. First, those that delegate to each behavior the decision on the positioning of the head. Second, those which continuously move the robot's camera in a fixed pattern to cover the entire search

space of the robot. Its main drawback is that it does not allow tracking a detected stimulus. In addition, much time is wasted on exploring areas where a priori there is no object. A third group includes those using a specific component responsible for making this decision based on the requirements of active behaviors. There are attention mechanisms guided by utility functions based on the task the robot is currently doing[8], or salience-based schemes which increase with time[16] or time-sharing mechanisms, among others.

One SPL team [16] associates a tuple $< \rho, \theta, anchor >$ with each stimulus that the robot can detect. The $\rho[0, max\_dist]$ and $\theta[-\pi, \pi]$ indicate the relative distance and orientation to the stimulus, respectively. The $anchor[0, 1]$ indicates the confidence in the information of the stimulus. In this system, the behaviors define the importance of each stimulus. Depending on the importance defined for each stimulus and the values for *anchor*, the active vision system decides which stimulus to focus on at any time. The behaviors themselves establish which stimuli should be observed. This approach does not tolerate observations with occlusions and partial observations. In this approach that only one of the objects receives the feature's update, while the other objects remain unchanged. This association is carried out according to a specific function, like Euclidean distance. In this work the search for new objects uses the same fixed pattern for head positions, independently of the type of the object to search.

One interesting approach [2] shares information among all the robots of a team. Each stimulus is labeled as valid, suspicious and invalid, depending on the confidence on the stimulus. This label is set to valid when an object is detected in the image of the robot, or suspicious if this information comes from a teammate. The stimuli which are labeled as valid do not need to be attended urgently. Those labeled as suspicious have to be inspected to check whether they are still valid. There are behaviors that define the importance of a stimulus. All stimuli are attended at all times. Regarding the visual memory, a mixed approach is used to estimate the goal positions. The estimate of the goal extracted from the self-localization system is mixed with the one captured in the last scan.

The approach proposed in [14] uses a multi-modal algorithm to estimate the positions of other robots. The observations are compared with the positions maintained by the world model using the Euclidean distance criterion. A similar approach was used in [15] but using the Mahalanobis distance between observations and objects in memory. The disadvantage of these algorithms is their full association between the observation and one of the objects.

In [18] and [5], the state of the robot is modeled using Monte Carlo Localization (MCL). The state includes both the robot's position and the ball's position. The aim of the active vision system is to minimize the entropy of the robot's state. Here the active vision is associated with a utility (self-localization and detection of the ball), and that the utility of turning our gaze towards one place or another is quantifiable depending on how it decreases the entropy of system. Behaviors do not define the importance of the stimuli and do not modulate the active vision system in any way.

3 Multi-object Visual Memory

In BICA the humanoid robot intelligence is decomposed in perceptive components, which provide information and behavior components, that make control decisions taking into account such information. Some behavior components for the SPL are `SearchBall`, `GoToPoint`, and several shooting movements. Some perception components are `BallDetector`, `GoalDetector` and `LineDetector` which detect the relevant stimuli in the RoboCup scenario. These components are responsible for detecting the stimulus in the image, calculating its 3D position, updating the visual memory with this position, and providing information to the attention system.

Beyond the instantaneous detection in current image a visual memory is built and updated. The visual memory is formed by the composition of various independent JPDAF filters running in parallel, one for each object to be tracked. In order to avoid the additional uncertainty the self-localization algorithms and to be fast, the visual memory records the position and uncertainty of the objects in a coordinate system relative to the robot. For each object the last time it received a new observation is also stored. For the SPL humnanoid robot there are three JPDAF filters running in parallel: one for the ball, one of the team's goal, and another for the opponent's goal.

All filters are updated regularly based on the odometry generated by the robot and the detected features from perceptual components. In the initialization phase of each filter it is possible to set some features like the object motion model. For the ball, as can be kicked by other robots, the uncertainty of its estimate must increases if the robot stays still. As time passes without generating new observations, it becomes more likely that the ball changes its position. However, the goals will not move, so it makes sense not to increase their uncertainty if the robot remains motionless.

The general working environment for the JPDAF algorithms consists of several objects and several observations. These objects may have the same appearance, and therefore cannot be assigned to each estimator straightforward. The problem is to find out how we can associate each observation to each object. Using the notation of [17], we have a set of objects $X^k = x_1^k, \ldots, x_T^k$ at time k. The set of observations at that instant k is defined as $Z(k) = z_1(k), \ldots, z_{m_k}$. The entire set of possible associations between an observation j and object i, is defined as a joint association event θ to a specific set containing couples $(j, i) \in \{0, \ldots, m_k\} \times \{1, \ldots, T\}$. The set θ uniquely associates each observation with each object. The special empty observation $z_0(k)$ means that the object has not been perceived. The term Θ_{ji} expresses the set of all joint association events that pair j with the object i.

The core of the JPDAF algorithm is based on calculating the parameter β_j, whose mission is to measure the likelihood that the observation j belongs to object i. Equation (1) shows how to perform the calculation of the posterior probability.

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} P(\theta|Z^k) \tag{1}$$

The probability $P(\theta|Z^k)$ can be expanded according to equation 2 assuming the problem is Markovian and using the theorem of total probability. The full derivation of this formulation can be found in [17]. $P(\theta|X^k)$ evaluates the probability of a specific θ set given the current state of the tracked objects.

$$P(\theta|Z^k) = \alpha \int P(Z(k)|\theta, X^k) P(\theta|X^k) p(X^k|Z^{k-1}) dX^k \tag{2}$$

$P(Z(k)|\theta, X^k)$ of equation (2) specifies how likely the set of observations obtained with the current state of objects and a particular set of observation-object associations is. To derive this term, it is necessary to consider the case of a false positive. We call γ the probability that an observation is a false positive and the number of false positives on a θ is expressed as $(m_k - |\theta|)$. In turn, the probability associated with all false positives in at time k and a specific θ is $\gamma^{(m_k - |\theta|)}$.

For $P(\theta|X^k)$ it is assumed that all sets of pairs of associations are equally likely and, therefore, this term can be approached by a constant (as can be seen in [3]).

Taking into account the derivation of the previous two terms and assuming independence between observations, we have:

$$P(Z(k)|\theta, X^k) = \gamma^{(m_k - |\theta|)} \prod_{(ji) \in \theta} \int p(z_j(k)|x_i^k) p(x_i^k|Z^{k-1}) dx_i^k \tag{3}$$

Combining equations (2), (3) and (1) we get:

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} \left[\alpha \gamma^{(m_k - |\theta|)} \prod_{(ji) \in \theta} \int p(z_j(k)|x_i^k) p(x_i^k|Z^{k-1}) dx_i^k \right] \tag{4}$$

Once we know how to compute β_{ji} to weight each observation with each object, all we need is to describe how to update the Kalman filters to estimate each object. The prediction phase is performed by equation (5) and the correction phase is described by the equation (6).

$$p(x_i^k|Z^{k-1}) = \int p(x_i^k|x_i^{k-1}, t) p(x_i^{k-1}|Z^{k-1}) dx_i^{k-1} \tag{5}$$

$$p(x_i^k|Z^k) = \alpha p(Z(k)|x_i^k) p(x_i^k|Z^{k-1}) \tag{6}$$

This is where we introduce the factor β_{ji}, integrating over all the observations obtained and its association with the object i.

$$p(x_i^k|Z^k) = \alpha \sum_{j=0}^{m_k} \beta_{ji} p(z_j(k)|x_i^k) p(x_i^k|Z^{k-1}) \tag{7}$$

3.1 Ball Object

The ball is one of the objects stored in visual memory. Although there is only one ball, in practice and due to calibration problems or lighting, it is possible to detect several balls in the same frame in addition to the correct ball. The JPDAF ball algorithm is configured as a single object, i.e., you can not create or destroy new estimates, only one remains. This causes the correct observation to be weighted with a high value of β. The comparison between the correct observation and false positives greatly benefits the correct observation and, therefore, the rest have less influence on the estimate.

3.2 Goal Objects

Unlike the ball filter, the JPDAFs for each goal are set to hold two independent objects simultaneously, one for each post. Although the tracking algorithm maintains an independent estimation for each post, there is a restriction that can be applied and improves the accuracy of the estimate. This constraint imposes the posts are always at the same distance from each other, in our case 1400mm.

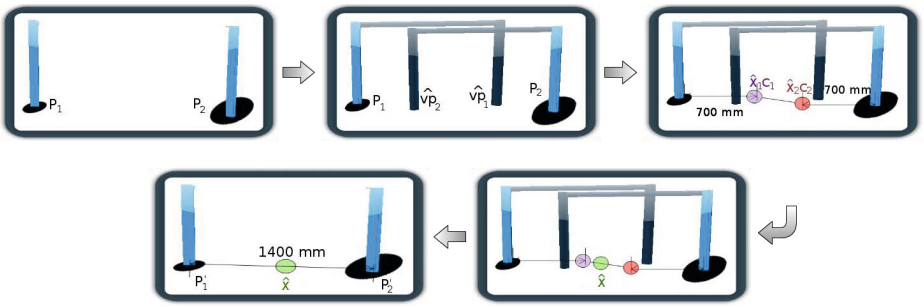


Fig. 1. Diagram of the algorithm for the calculation of goal's center

The idea of the algorithm is to assume that each post is correctly estimated and therefore we can infer the opposite virtual post position $v\hat{p}_i$. Along with the position of each post and the opposite ones virtually generated, two hypotheses with centers of the goals at $\hat{X}_i$ are created. The end point where you estimate the center of the goal would be in a straight line connecting these two hypotheses. Although the midpoint of the two hypotheses could be a first approximation to the solution, it does not take into account that the original estimations for each post can have different uncertainties. Therefore, it is preferable to weight the uncertainty as each of the hypotheses. Equation 8 calculates the desired point $\hat{X}$ assuming $\hat{X}_i$ is the position of the hypothesis i and its covariance associated with C_i. Finally, the original estimations of each post P_i are adjusted according to the calculated midpoint and the restriction of known distance from the midpoint of each post obtaining P'_i. Algorithm 1 summarizes the steps for calculating the

Algorithm 1. Optimization to improve the calculation of the goal center

1: Update posts estimations P_i using JPDAF algorithm
2: **for** $i \leftarrow 1, 2$ **do**
3: $\hat{vp}_i$ = Generate the virtual opposite post
4: $\hat{X}_i$ Calculate the hypothesis between P_i and the vp_i
5: **end for**
6: **for** $i \leftarrow 1, 2$ **do**
7: $\hat{X}$ = Obtain the corrected center applying weighting the two Gaussians $(\hat{X}_1, C1))$ and $(\hat{X}_2, C2))$
8: Using $\hat{X}$ as the center of the net, generate P_i' according with the width constraint
9: **end for**

central point of the goal. Furthermore, Figure 1 shows an example of the whole adjustment process made to the position of each object.

$$\hat{X} = \hat{X}_1 + C_1[C_1 + C_2]^{-}1(\hat{X}_2 - \hat{X}_1) \tag{8}$$

4 Visual Attention

Robots equipped with cameras have a limited field of view. Visual stimuli may not all be present simultaneously in the image perceived by the camera of a robot. For this reason, the robot has to search in the scene by varying the orientation of the camera. The visual stimuli are incorporated into the visual memory, as described in section 3, and they should be checked periodically to update their position. The visual attention system is responsible for performing the scanning for visual stimuli in a scene, and it verifies and updates the already collected stimuli.

Some actuation components have perceptual requirements, i.e., a set of visual stimuli to be aware of. These perceptual requirements are met by activating the perception components responsible for detecting each one of these stimuli and by setting the importance, in the range (0.1], for each visual stimulus.

Our visual attention system receives attention requests from each of the components activated with interest in some objects. Each of these components have different requirements that may conflict with each other, so the visual attention component acts as a referee assigning control of the attention fairly among all visual stimuli. Figure 2 (left) shows how some actuation components send their attention requirements about the visual stimuli in the scene. For each visual stimulus received, the visual attention attention system chooses the highest value received and then it normalizes this value by the sum of all importances. At the left side of figure 2 (left), the value of importance for the goals is 0.5 $(max(0.5, 0.25))$, for the ball is 1.0 $(max(0.75, 1.0, 0.5))$ and 1.0 for the lines.

If an actuation component sets a need for a visual stimulus, and then becomes inactive, the visual attention system must adapt to this new situation. The components are iterative, and their mechanism of deactivation is simply stop running

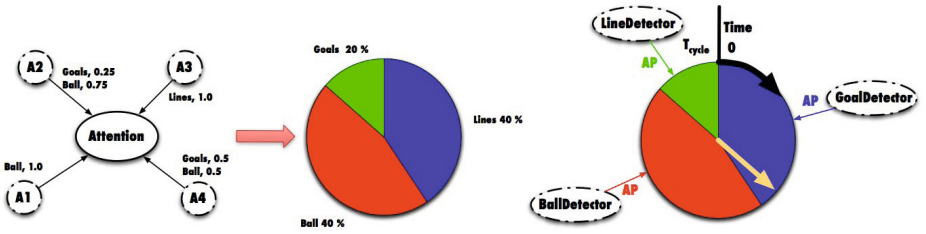


Fig. 2. Example of attention Setup. left: Actuation components (A1, A2, A3 and A4) send their perception requirements to the Attention component, which calculates the importance relationship among the object. Right: The attention system receives desired points to focus the camera. When it is the ball's turn, the attention system asks the component in charge of detecting the ball (red), and when it is the goal's turn, the attention system asks to the component in charge of detecting the goal (blue).

them silently. Because of this, each component in each iteration should resend their perceptual requirements. The visual attention system discards those stimuli that are not refreshed frequently, recalculating the importance relationship among visual stimuli.

The visual attention system implements a time-sharing policy, a time slot is assigned to each visual stimuli. This time slot is dedicated to seeking and tracking the corresponding stimulus. Figure 2 (right) shows how the turns are managed. On each turn, the visual attention system asks the component responsible for the stimulus what to do. The answers come in the form of attention points (AP). These attention points are three dimensions coordinates where the camera should focus. The only feedback sent to each detector is when the camera is pointing at the requested attention point.

It is remarkable that the visual attention system does not decide where to focus the camera, how to seek or track an object or when it is considered that a stimulus has been found or when it is lost. The novelty of this system is that these tasks are delegated to the corresponding perceptive component (detector):

- The attention component notifies to the detector components when the camera is focusing to the desired attention point. If the object is not at the position where it should be, the detector is responsible to guess if it is lost, using the last detection time. Some detectors could me more robust to occlusions, depending on the object characteristics.
- If an object is considered lost, or its position is not known, the detector generates attention points sequentially in the positions where the object can be found. Whenever a requested position is reached, the attention system reports this event, and the detector generates the next attention point. It is important that the focal points generated depend on which object you are looking for: the ball can only be at a point on the ground, but it is more effective to find the goals at the points on the horizon. Actually, it is

even possible to use the self-location information to decide where to generate search points. In any case, the selection of these attention points is done by each detector, which is specialized to look for the object.
– When an object is detected and the current slot corresponds to its detector, decisions may be different. In the case of the ball, being a very dynamic object, the remaining time slot can be dedicated to track the object.

More details and a comparison between several developed attention mechanisms can be found at [12].

5 Experiments

An extensive experimentation has been carried out to validate the system described in this article. We have used the real NAO robot in the real SPL environment as shown in Figure 3. The Nao robot is a medium-sized humanoid robot with 58 cm. of height, with 21 degrees of freedom, and a built-in x86 AMD Geode cpu at 500 MHz running GNU/Linux. The Nao features two CMOS 640x480 cameras, Ethernet, Wi-Fi, an inertial unit, force sensitive resistors, sonars, bumpers, four microphones, two hi-fi speakers and a complete set of leds.

Fig. 3. Experimental setup. The environment is the real SSL field and the robot is equipped with a pattern in order to be detected by a ground truth system.

During the experiments, we collected a great amount of data used to analyze the visual attention subsystem and the visual memory. The data collected during the execution of the experiments is stored in a log file for offline processing. Meanwhile, we have adapted the SSL [23] ground-truth system that captures the correct position of the dynamic elements of the field (robots and ball). This system is composed of two ceiling cameras mounted above the center of each half field. During the experiments, the robot is equipped with a visual pattern easily detected by the cameras. The error of the ground-truth system is less than 3 cm. in position and less than 5 degrees in orientation. Post-processing of robot log data and the ground truth data led us to accurately calculate the error in the perception of the robot.

Visual Memory Accuracy Experiments. The first experiment makes a statistical analysis of the accuracy obtained in estimating the current stimuli in the visual memory. The stimuli analyzed are the ball and one of the goals. For each stimulus, we calculate the error (measured as the Euclidean distance between the real and the estimated position) and the standard deviation of the estimate over time. There have been two different tests, one with a static robot and a second with the robot in motion. The attention settings for each test are labeled in the key of the graph. Figure 4 shows the data extracted from the experiment, where data analyzed from the static robot are located in the upper image and the data extracted from the robot in motion are displayed in the lower image. Besides the error in the estimate and standard deviation, Figure 4 also shows on a horizontal bar the intervals where each object of attention has the attention turn.

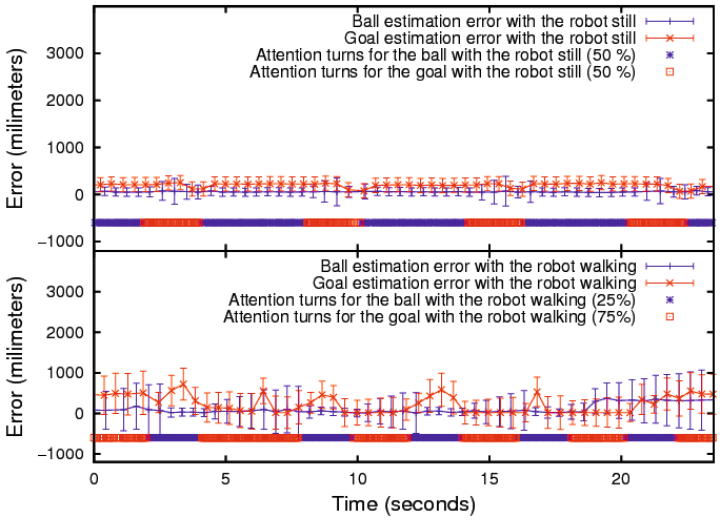


Fig. 4. Accuracy and standard deviation of the visual memory estimation

One interesting result is how the standard deviations vary while the robot remains static (Figure 4 upper). In the case of the goal (displayed in red), the uncertainty remains constant even when there is lack of attention. However, in the ball case (displayed in blue), the standard deviations increase during intervals in which no features are received from the ball. When the robot moves (Figure 4 lower), the dynamic of the stimulus is automatically configured with different parameters and this affects how the uncertainty grows when the head does not look at the goal and, consequently, no features are received. If we pay attention at the lower blue curve on Figure 4, we notice that the standard deviations are larger and grow faster when there are no features perceived, compared with the static robot experiment.

Attention Experiments. The next experiment analyzes how the visual attention system simultaneously detects, confirms and tracks two different elements: the ball and goal. In some tests that are part of this experiment, the robot is stationary, and in others the robot is moving. Perceptual configurations are different, giving in some cases more attention to the ball, and in some cases greater attention to the goal.

As described in Section 4, attention is distributed in the stimuli detectors. The search is different in the case of the ball (it looks at the ground) and the goal (it searches the horizon line). Likewise, the ball tracking is different in the case of the ball (all the time available is used to track and update the ball position) and the goal (when the goal is detected and its position is updated, it yields the rest of this time). Finally, the ball is a manipulable element, so its uncertainty increases over time as the robot moves. Instead, the goal is static, and its uncertainty only grows when the robot moves.

Figure 5 (right) shows the evolution of the uncertainty (standard deviation of estimate) in the detection of the ball and the goal when the robot is stopped. In further experiments we will analyze the accuracy of the system, but in this case we want to show how the attention system keeps low uncertainty of the estimates. This graph shows how in the configurations with the ball importance set to 100%, the average uncertainty remains very low, as the robot performs a constant tracking of the ball, without paying attention to other stimuli. The standard deviation of ball estimate does not increase too much as the goal importance increases. In the case of the goal, the uncertainty is high when the robot is not actively looking for this element, decreasing as the goal detector increases its importance.

Figure 5 (left)shows how the uncertainty is much greater when the robot walks. When the importance of the ball is less than 50%, the uncertainty increases to 750mm. The goal uncertainty, however, remains adequate when its importance is greater than 50%.

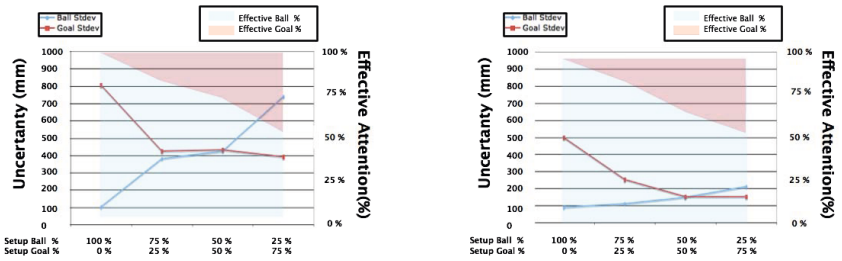


Fig. 5. Attention experiment with the robot **walking** (left) and the **static** robot (right). The graph shows the standard deviation of the ball and goal estimates, depending on the attention modulation. The colored area shows the actual attention distribution.

Figure 5 show (colored areas) the real value of time spent on each item. The percentage of time spent on the ball is always greater than on the goal due to its internal implementation. The ball always take the most of all the available time, while the goal detector yields the extra time.

Robustness Experiments. The next set of experiments measures the performance of visual memory under undesirable situations, such as false positives or false negatives observations. During the tests the robot has remained static. The first experiment consisted in estimating the position of the ball in the presence of false positives. A second ball located inside the robot's field of view was used to generate false positives. In the second experiment we used a single ball that was periodically occluded. The third experiment evaluates the behavior of the goal estimator against false positives. We used an extra goal post, which was perceived by the robot's camera during certain frames. Finally, a fourth experiment analyzes the behavior of the estimator of goals to false negatives caused artificially occluding one of the goal posts.

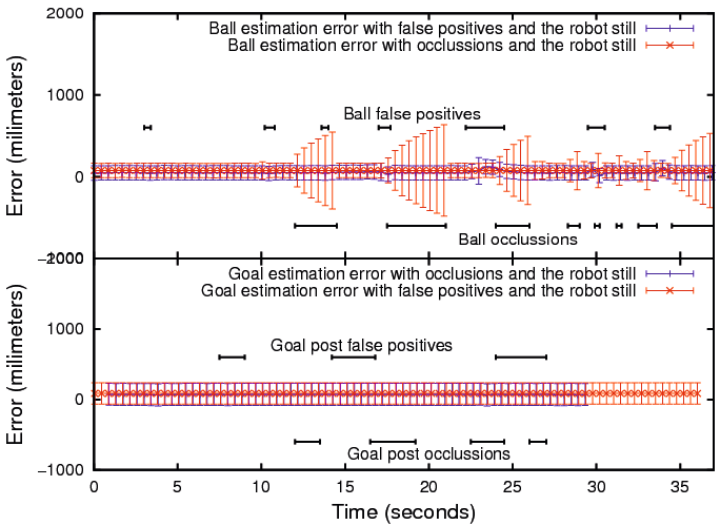


Fig. 6. Robustness against false positives and false negatives for the ball and goal

Figure 6 shows a summary of data extracted from the experiments. In Figure 6 (upper) it is displayed the error of the ball estimator under false positives and false negatives. In Figure 6 (lower) we show the same information for the goal. In both graphs we have labeled the time intervals for each experiment in which false positives and false negatives appeared.

The graphs show that the estimators have high robustness, due to low error variation when the false positives or false negatives arise. A rapid increase in ball

position uncertainty is noticeable during ball occlusions due to the movement mode associated with the ball.

Eficiency Experiments. The next experiment measures the visual memory and the attention system CPU consumption. In this experiment we run the main behavior for playing soccer (`Striker` component). Moreover, `Striker` executes the attention (`Attention` component) and visual memory behaviors (`VisualMemoryBall` and `VisualMemoryGoal`), along with several other components that analyze images (`Perception`), self-localized the robot itself (`Localization`), among others. During the experiment a log file stores information about the average execution time of each component. Thus, we can see the average execution time of the visual memory and attention components and we can compare their values with the time required by other components. Figure 7 shows the results obtained.

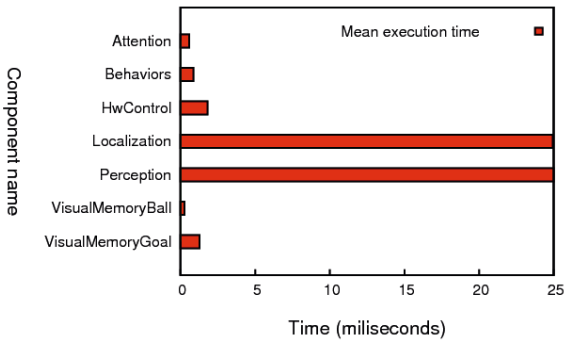


Fig. 7. CPU consumption of the `Attention`, `VisualMemoryBall` and `VisualMemoryGoal` components compared with the rest of components running on the robot

The average execution time of the component that implements the visual memory for the ball is 0.21 ms. In turn, the visual memory required for the goal is 1.5 ms due to greater number of features received compares with the ball. In turn, the attention component consumes 0.47 ms. to execute one of its iterations. Looking at Figure 7 we can see how the results are comparable to the fastest components contained in our architecture and thus they are appropriate to ensure real time execution.

6 Discussion and Future Lines

In this work we have presented the visual perception system developed for a humanoid robot participating in the Standard Platform League of RoboCup.

This system is composed of a visual memory and an attention subsystem. The visual memory has been designed to store the relative position of the objects, like goals or the ball, around the robot. We use several multi-modal JPDAFs to update the object features from the camera images. The images coming from the robot camera fed this visual memory. The attention mechanism moves the robot's head in order to visually explore the environment for new objects, to reobserve the existing ones or to cover all robot surroundings. Following a time sharing approach this attention component combines several perceptive needs of the soccer application like seeing the ball or seeing the goals to self-localize. A time slot is reserved for each relevant stimuli. Inside it, the head control is distributed among the active perception components for detecting and tracking the different stimuli.

Advantages of this visual memory system include time persistence and better integration of partial information, due to a multi-modal algorithm. It has broader scope than the single current camera image and the object estimations are more reliable than instantaneous estimations for occlusions, false negatives and false positives which usually appear in the images. The visual memory provides more and more reliable information about the robot surroundings than the current camera image alone. Taking both into account, memory and current image, the robot can take better behavior decisions.

The advantages of the attention subsystem include the convenient combination of perception requirements, usually contradictory, and the delegation of control commands to the components specialized in each object. The perception requirements are combined in a fair time sharing distribution among different stimuli. This solves the need of looking at the ball and looking at the goals from time to time. This organization allows the independent development of different behaviors, because their perceptive requirements can be met regardless other behaviors, so there is no need to consider interferences at this level.

The experimentation confirms the advances provided by this work. The visual memory system keeps object estimates robust to errors in perception, such as false positives or false negatives. The different tests carried out on the visual attention subsystem have shown how perceptual behavior requirements are met. During the experiments, we have shown how this system shares the turns between stimuli and their uncertainty is kept low. In addition to laboratory experimentation, this system was also used in the RoboCup-2011 by the SPiTeam during real games. It is difficult to provide quantitative results of the participation. Instead, from a qualitative point of view, the results of the perceptual system were very satisfactory, since the software operated as expected.

We are extending this work in several directions. First of all, an occlusion detection mechanism might be developed. When a new object has been detected, we could check whether it is lined up with another object estimated before. In the positive case, we could modulate the growth of uncertainty of the occluded object in a much slower way, probably because the object remains in its original position but it has been occluded by the new object. Other future directions to extend the system is the introduction of other robots (teammates or opponents)

into the visual memory and the use of sensor information from other teammates to update the robot visual memory.

Acknowledgements. This research has been partially sponsored by Community of Madrid through the RoboCity2030 project (S2009/DPI-1559). The authors also would like to thank all the members of the URJC Robotics Group, SPiTeam and CORAL Robot Lab who have collaborated in this work.

References

1. Bar-Shalom, Y., Fortmann, T.E.: Tracking and data association. Mathematics in Science and Engineering, vol. 179. Academic Press Professional, Inc., San Diego (1987)
2. Coltin, B., Liemhetcharat, S., Mericli, C., Tay, J., Veloso, M.: Multi-humanoid world modeling in standard platform robot soccer. In: Proceedings of 2010 IEEE-RAS International Conference on Humanoid Robots, Nashville, TN, USA, December 6-8 (2010)
3. Cox, I.J., Hingorani, S.L.: An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(2), 138–150 (1996)
4. Cox, I.J.: A review of statistical data association techniques for motion correspondence. International Journal of Computer Vision 10, 53–66 (1993)
5. Czarnetzki, S., Kerner, S., Kruse, M.: Real-time active vision by entropy minimization applied to localization. In: Ruiz-del-Solar, J. (ed.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 266–277. Springer, Heidelberg (2010)
6. DARPA. DARPA Urban Challenge, `http://archive.darpa.mil/grandchallenge/`
7. The Robocup Federation. Robocup Standard Platform League (2010), `http://www.tzi.de/4legged/bin/view/Website/WebHome`
8. Guerrero, P., Ruiz-del-Solar, J., Romero, M.: Explicitly Task Oriented Probabilistic Active Vision for a Mobile Robot. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 85–96. Springer, Heidelberg (2009)
9. Isard, M., Blake, A.: Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In: Proc. 5th European Conf. Computer Vision, vol. 1, pp. 893–908 (1998)
10. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E.: Robocup: The robot world cup initiative. In: ICJAI 1995 - Workshop on Entertainment and AI/ALIFE (1995)
11. Martín, F., Agüero, C., Cañas, J.M., Perdices, E.: Humanoid soccer player design. In: Papic, V. (ed.) Robot Soccer, pp. 67–100. INTECH (2010)
12. Martín, F., Rubio, L., Agüero, C., Cañas, J.M.: Effective visual attention for behavior-based robotic applications. In: Proceedings of the 2013 Workshop on Agentes Físicos (2013)
13. Federation of International Robot-soccer Association FIRA (2011), `http://www.fira.net`
14. Ratter, A., Hengst, B., Hall, B., White, B., Vance, B., Sammut, C., Claridge, D., Nguyen, H., Ashar, J., Pagnucco, M., Robinson, S., Zhu, Y.: rUNSWift Team Report 2010. Technical report, University of New Wales (2010)

15. Röfer, T., Laue, T., Müller, J., Fabisch, A., Feldpausch, F., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Humann, A., Honsel, D., Kastner, P., Kastner, T., Könemann, C., Markowsky, B., Riemann, O.J.L., Wenk, F.: B-human team report and code release (2011), http://www.b-human.de/downloads/bhuman11_coderelease.pdf
16. Saffiotti, A., LeBlanc, K.: Active perceptual anchoring of robot behavior in a dynamic environment. In: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), San Francisco, CA, pp. 3796–3802 (2000), http://www.aass.oru.se/~asaffio/
17. Schulz, D., Burgard, W., Fox, D., Cremers, A.B.: People tracking with mobile robots using sample-based joint probabilistic data association filters. I. J. Robotic Res. 22(2), 99–116 (2003)
18. Seekircher, A., Laue, T., Röfer, T.: Entropy-based active vision for a humanoid soccer robot. In: Ruiz-del-Solar, J., Chown, E., Plöger, P.G. (eds.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 1–12. Springer, Heidelberg (2010)
19. Stone, L.D., Corwin, T.L., Barlow, C.A.: Bayesian Multiple Target Tracking, 1st edn. Artech House, Inc., Norwood (1999)
20. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). MIT Press (September 2005)
21. Universidad de León y Universidad Rovira i Virgili Universidad Rey Juan Carlos, SPITeam's Application for qualification to RoboCup-2011 (2011)
22. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill (2004)
23. Zickler, S., Laue, T., Birbach, O., Wongphati, M., Veloso, M.: SSL-Vision: The Shared Vision System for the RoboCup Small Size League. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS, vol. 5949, pp. 425–436. Springer, Heidelberg (2010)

Framework for Learning and Adaptation of Humanoid Robot Skills to Task Constraints

Daniel Hernández García, Concepción A. Monje, and Carlos Balaguer

Universidad Carlos III, Madrid, Spain
{dhgarcia,cmonje,balaguer}@ing.uc3m.es

Abstract. Humanoid robots are expected to work and collaborate with humans performing in changing environments. Developing this kind of robots requires them to display intelligent behaviors. For behaviours to be considered as intelligent they must at least present the ability to learn skills, represent skill's knowledge, and adapt and generate new skills. In this work a framework is proposed for the generation and adaptation of learned models of robot skills for complying with task constraints. The proposed framework is meant to allow: for an operator to teach and demonstrate to the robot the motion of a task skill it must reproduce; to build a knowledge base of the learned skills knowledge allowing for its storage, classification and retrieval; to adapt and generate learned models of a skill, to new context, for compliance with the current task constraints. A learning from demonstration approach is employ to learn robot skill by means of probabilistic methods, encoding the motion dynamics in a *Gaussian Mixture Model*. We propose that this models of the skill can be operate and combine to represent and adapt the robot skills.

Keywords: humanoid robots, learning from demonstration, skill adaptation.

1 Introduction

A major goal in robotics research is to develop human-like robotic systems capable of interacting and collaborating with humans. The ultimate goal is for a robotic platform capable of performing, autonomously, in the unstructured scenario of humans natural environment. Humanoid robots must realize any number of tasks which could be reasonably expected from them by its human operators, during the normal development of a typical workday. Working alongside humans means dealing with continuously changing environments and a huge variability of tasks, thus the robots should have the ability to continuously learn new skills and adapt the existing skills to new contexts. Future humanoid robots would need to execute a wide range of movements in a natural human-like manner, process information from multiple sensors into a reliable representation of the world in order to understand and react to their environment. Also, they would need to provide means for a meaningful interaction with their human partners; they must be engaging and responsive. And they must present intelligent, natural, predictable and reasonable behaviors.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_41,

Therefore, humanoid robot need to be provided with intelligence, that is, the ability to sense the environment, make decisions and take actions, to recognize objects and events, represent knowledge, reason and plan for the future, and to act successfully under a large variety of circumstances. An intelligent agent is one that is flexible to changing environments and changing goals, learns from experience, and makes appropriate choices given perceptual limitations and finite computation [13]. Intelligence requires an interconnecting system architecture that enables the various system elements to interact and communicate with each other, integrating perception, reason, learning and behaviour generation [1]. Figure 1 shows a model an intelligent agent architecture with systems for perception, action, interaction, reasoning, world knowledge and learning.

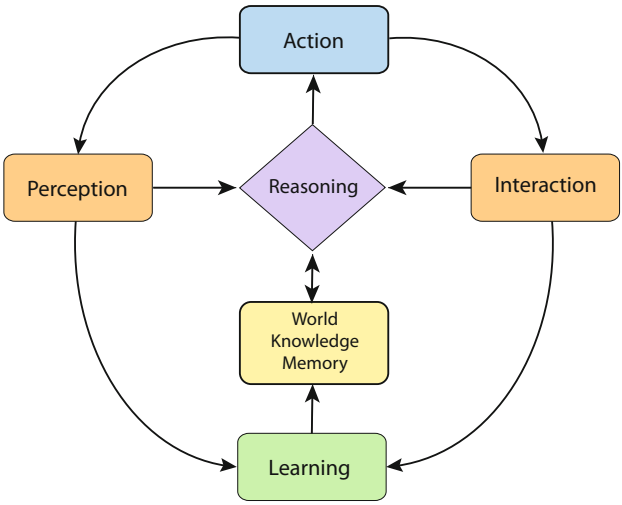


Fig. 1. Model for an architecture of intelligent agents. For an intelligent agent, with the needs of a humanoid robot, it is necessary to have systems for perception, action, interaction, reasoning, world knowledge and learning.

It is necessary that the humanoid robots display a sufficient level of intelligent behaviour, this must include the capacity to perceive and understand, to choose wisely, and to act successfully under a large variety of circumstances [1]. Humanoid robots, in order to cope with working in continuously changing environments and performing a huge variability of tasks, must be provided with systems that allow them to continuously learn new skills, represent their skill's knowledge, and adapt their existing skills to new contexts, as well as to robustly reproduce new behaviours in a dynamical environment.

A skill in our context is defined as a motor trajectory motion learned by the agent, an acquire ability for the execution of a task. A robot skill is a complex action reproducible when appropriate, and generalize to different contexts.

Learning systems are required to acquired skills and developed task knowledge of how to act. Algorithms for learning, and extracting important features of task actions are fundamental in order to build intelligent behaviours. *Robot Programming by Demonstration (RPbD)* [4], also known as *Imitation Learning* or *Learning from Demonstrations (LfD)*, has appeared as one way to respond to this growing need for intuitive control methods.

The *Imitation Learning* approaches focuses on the development of algorithms that are generic in their representation of the skills and in the way they are generated. *LfD* formulates user-friendly methods by which a human user can teach to a robot how to accomplish a given task, simply by demonstrating this task [8], and generalizing the demonstrated movements across a set of demonstrations. The goal is on exploiting the variability inherent to the various demonstrations and to extract the essential components of the task. Figure 2 illustrate this process. Even though *LfD* offer natural, fast and implicit means of teaching a robot new skills some very important questions are frame in [15]: Is there a basic set of primitives? How can new primitives be learned, and old primitives be combined to form higher level movement primitives? How can sequencing and recognition of sequences of movement primitives be accomplished?

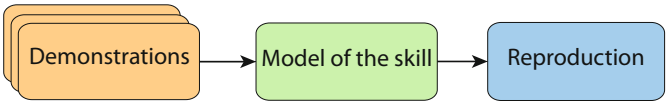


Fig. 2. Generalization of a skill in Robot PbD by extracting the statistical model across multiple observations. Adapted from [4].

The robotic systems must be able to store and latter retrieve and use their knowledge of learned skills. The knowledge base needs to hold all necessary information for reproduction of the skills in the environment. Knowledge of the task would be distributed among the representation of objects, actions and events of the task and the state of the world. A behaviour could be represented by the phrase ***"Do an Action (A), To an Object (O), For achieving Goal (G), When State of the World is (W)"***. Therefore, the tuple formed by $\langle$ *Do = Action(A), To = Object(O), For = Goal(G), When = World State(W)*$\rangle$ holds all necessary information for the reproduction of the task behaviour.

Despite the *LfD* approaches clear advantages, it would still be impractical for the human operator to teach the robot the skills for every needed task and for every foreseen situation. Therefore it is necessary to extend the classical *LfD* approach of learning a skill model in a way that allows the adaptation of a robot previously learned motion skills to new unseen contexts.

To reproduce a task adapted for an unseen context the robot must be given knowledge of the state of the environment and the constraints of the task. Using both, the already learned model of a skill, and the extracted constraints information of the current task, the model of the skill can be adapted to reproduce the task. Figure 3, illustrates the process for enhancing classical *LfD* approach to generalize a skill to allow adapting a robot previously learn skills models. The traditional approach in *LfD* from [4] as represented by the Figure 2 presents a somewhat static control scheme, a kind to an open loop controller, and it won't be sufficient to reproduce a task when the state of the environment is too dissimilar from how it was when the demonstrations were given to encode the model of the skill. By adding environmental and task knowledge as input to the scheme, in Figure 3, the control diagram could be thought as a close loop controller, with a feedback signal from the constraints of the task and the environment, which allow the model of a skill to be adapted accordingly to its context.

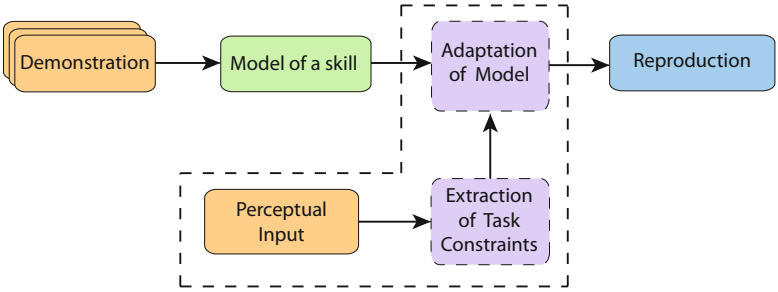


Fig. 3. Augmenting the *LfD* approach for the generalization of a skill to allow adapting a robot previously learn skills models. Adaptation of a learn skill to new context by extracting the task constraints with a new observation.

This work is centred around the major idea of future robotic systems, more specifically humanoid robots, that are capable of interacting with humans in their homes, workplaces, and communities, providing support in several areas, and collaborating with humans in the same unstructured working environments. Our focus is on topics concerning the learning, representation, generation and adaptation, and reproduction of robot skills. In this work a framework is proposed for the learning, generation and adaptation of skill models to comply with task constraints. The following sections described the framework presented in this work. Section 2 presents the proposed framework followed in the rest of this work. Section 3 address the learning of the skill models. Section 4 discuss the knowledge representation and organization of the knowledge base. Section 5 address the adaptation of the robot skills. Section 6 describes the preliminary experimental validation. Discussion of the work presented in this paper is provided in Section 7.

2 Framework for Learning and Adaptation of Robot Skills

From the discussion in the previous section it is clear that humanoid robots, in order to cope with working in continuously changing environments and performing a huge variability of tasks, must be provided with systems that allow them to continuously learn new skills and adapt their existing skills to new contexts, as well as to robustly reproduce new behaviours in a dynamical environment.

To deal with those challenges in this work a framework is proposed for a cognitive model for the generation and adaptation of learned models of robot skills for complying with task constraints, as illustrated in Figure 4. The framework is meant to allow:

1. For an operator to teach, and demonstrate, to the robot the motion of a task skill it must reproduce.
2. To build a knowledge base of the learned skill models allowing for its storage, classification and retrieval.
3. To adapt and generate learned models of a skill, to new context, for compliance with the current task constraints.

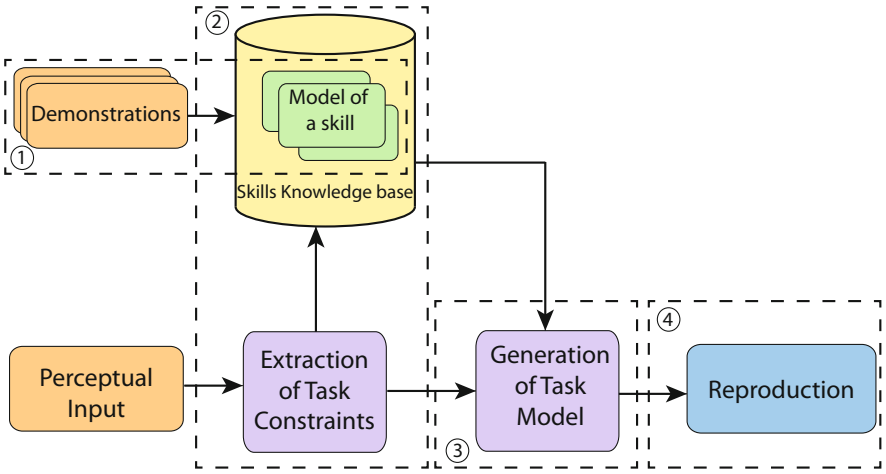


Fig. 4. Framework for learning and adaptation of robot skills. A knowledge base (2) is built with the models of the skills learned through demonstrations (1). The constraints of a requested task are extracted from the perception of the world state. With the current task constraints and the models of a skill retrieve from the knowledge base a new adapted task model (3) is generated for reproduction (4).

The rest of this work would describe the modules for learning the robot skills models, representing the robot skills knowledge and generating and adapting robot skills.

3 Learning Robot Skills Models

The *LfD* approaches focus on development of algorithms that are generic in their representation of the skills and in the way they are generated. One common approach creates model of the skill based on sets of demonstrations performed in slightly different conditions generalizing overt the inherent variability to extract the essential components of the skill [5]. Employing statistical learning techniques is a popular trend for dealing with the high variability inherent to the demonstrations. One of most promising approaches are those that encapsulate the dynamics of the movement into the encoding, [4]. Autonomous dynamical systems (*DS*) has been proposed representing movements as mixtures of nonlinear differential equations with well-defined attractor dynamics [9].

3.1 Learning Motion Dynamics as Multivariate Gaussian Mixtures

The *DS* framework provides an effective mean to encode trajectories through time-independent functions that define the temporal evolution of the motions. First it is assume that the state of the robot system can be unambiguously describe using a state variable define as ξ (end-effector positions, velocities, etc.), and further assume that the motion is governed by a first order autonomous ordinary differential equation,

$$\dot{\xi} = f(\xi, \theta), \tag{1}$$

A probabilistic framework is employed to build an estimate $\hat{f}$, of the nonlinear state transition map f, based on the set of demonstrations. *Gaussian Mixture Models (GMM)* are used to directly embed the multi-variate dynamics of a motion through the encoding of the demonstrated data. A mixture model of $\mathbf{K}$ components is defined by a probability density function,

$$p(\xi) = \sum_{k=1}^{\mathbf{K}} p(k) p(\xi \mid k) \tag{2}$$

where ξ is a data point, $p(k)$ is the prior and $p(\xi \mid k)$ is the conditional probability.

The *GMM* define a joint probability distribution $p(\xi^i, \dot{\xi}^i)$ of the training set of demonstrations as a mixture of the $\mathbf{K}$ Gaussian multivariate distributions,

$$\begin{aligned} p(\xi, \dot{\xi}; \theta) &= \frac{1}{K} \sum_{k=1}^{\mathbf{K}} \pi^k \mathcal{N}^k(\xi, \dot{\xi}; \mu^k, \Sigma^k) \\ \text{with} \quad \mu^k = \{\mu^k_{\xi}; \mu^k_{\dot{\xi}}\} \quad &\text{and} \quad \Sigma^k = \begin{bmatrix} \Sigma^k_{\xi} & \Sigma^k_{\xi\dot{\xi}} \\ \Sigma^k_{\dot{\xi}\xi} & \Sigma^k_{\dot{\xi}} \end{bmatrix} \end{aligned} \tag{3}$$

The *GMM* estimates the function f, so the unknown parameters θ are the prior, π^k, the mean, μ^k, and the covariance matrix, Σ^k, of the $\mathbf{K}$ Gaussian, such that $\theta^k = (\pi^k, \mu^k, \Sigma^k)$, as in Eq. 3, define the robot skills models.

To recover the expected output variable $\hat{\dot{\xi}}$, given the observed input in ξ^*. one then can sample from the probability distribution function $p(\xi, \dot{\xi})$ in Eq. 3. This process is called *Gaussian Mixture Regression (GMR)*, taking the conditional mean estimate of $p(\dot{\xi} \mid \xi^*)$ the estimate of our function $\hat{\dot{\xi}} = \hat{f}(\xi^*)$ is,

$$\hat{\dot{\xi}} = \sum_{k=1}^{K} h^k(\xi^*)(\Sigma_{\dot{\xi}\xi}^k(\Sigma_{\xi}^k)^{-1}(\xi^* - \mu_{\xi}^k) + \mu_{\dot{\xi}}^k) \tag{4}$$

$$\text{where,} \quad h^k(\xi) = \frac{p(\xi; \mu_{\xi}^k, \Sigma_{\xi}^k)}{\sum_{k=1}^{K} p(\xi; \mu_{\xi}^k, \Sigma_{\xi}^k)} \text{with} \quad h^k(\xi) > 0 \quad \text{and} \quad \sum_{k=1}^{K} h^k(\xi) = 1$$

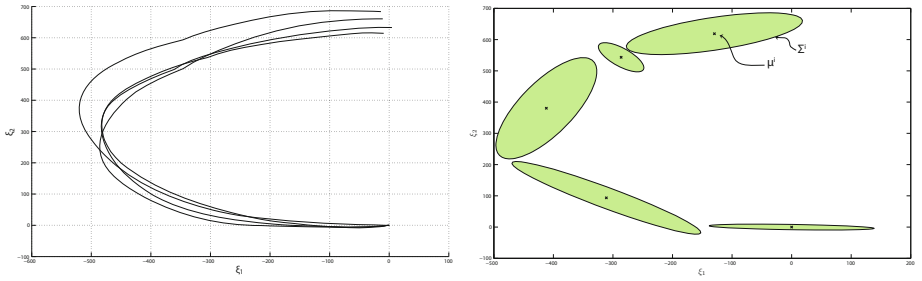


Fig. 5. *GMM-GMR* Learning process: (left) Recorded data of the demonstrated trajectories. (right) The *GMM* model represented by ellipses centre at μ^i, magnitude and direction of the ellipses are given by the eigenvectors and eigenvalues of Σ^i.

Figure 5 illustrates the learning process. Theoretical analysis of *GMM* and *GMR* can be found on [6], [16].

3.2 Stable Estimator of Dynamical Systems

The non-linear *DS* are susceptible to instabilities. Guaranteeing the estimates $\hat{f}$ result in an asymptotically stable trajectory is one key requirement to provide useful robot skills. The unknown parameters, θ, must be determined such that by starting the motion from any point the energy of the system decreases until it reaches the target.

To build a globally asymptotically stable *DS* [10] proposed a learning method, called *Stable Estimator of Dynamical Systems (SEDS)*, establishing a set of sufficient stability conditions. Learning the parameters of the *GMM* proceeds as a constraint optimization problem, ensuring that the model satisfy global asymptotic stability of the *DS* at the target [11]. By being time-invariant and globally asymptotically stable at the target, the *DS* estimated with *SEDS* are able to respond immediately and appropriately to perturbations that could be encountered during reproduction of the motion [11].

4 Representation of Robot Skills Knowledge

An important challenge for robots acting on unstructured dynamic environments is dealing with internal representation and understanding of the world. Decisions must be made on which aspects of the world to focus on and which aspects of the world to ignore, and how to structured the knowledge about the world. Despite claims against the use of internal representation, abstractions are necessary because no system can possible manage a world model that includes the whole of the world. However, representations must be limited and physically grounded to the environment, good representations must be selective and oriented to a particular use by a particular agent [2].

The central task of a knowledge representation is capturing the complexity of the real world. Representations thus perform as functional abstractions of the perceived environment, encoding an agents knowledge about its world, objects, actions, events, etc., into manageable internal structures.

4.1 Representing Objects Knowledge

Organizing objects is a vital part of knowledge representation. The organization of the knowledge of objects in the world towards a manageable structure of objects knowledge is a critical aspect of the design of a knowledge base.

A system dealing with objects in the real world must deal with various forms and types of knowledge. [12] suggested the idea of using object-oriented groups of procedures, which where called frames. The frame knowledge structure can be seem as instances of an object-oriented representation analogous to the development in an object-oriented programming. [12] pictured a great collection of frames systems stored in permanent memory, when the perception evidence suggest one will fit a frame is evoked to working memory. In this work the data structure of frames is used to store the knowledge about the objects.

4.2 Representing Actions Knowledge

The interrelation between objects and actions representations is fundamental when executing tasks upon the world. For actions representations the concept of affordances is essential, affordances relate the representation of objects and actions. The affordances are proprieties of the objects and of what kinds of interactions they can support.

To acquire an internal representation of an affordance, and agent must carry out a complex encoding of the sensory stimulus; to reproduce the action an agent must decode the encoded representation into proper signals. The representations ought be encoded in the body and not in the head [2]. *DS* theory of cognition provides a way to overcome the separation between mind and the world largely prevalent in most work in *AI* [3]. The *DS* theory provides an alternative to the traditional formats of representations, yet, despite their differences the approaches can be complementary [3]. Section 3 presented the learning of robot skills as *DS*. The robot skills are modelled by the parameters θ of $\hat{f}$. *DS* can store knowledge and have this stored knowledge influence their behaviour [7].

4.3 Representing Events Knowledge

To Focus only on objects and actions would not be enough for the knowledge representation needed by the humanoid robots. Representational attributes need to also take into account the state of the world, grounding the representations to the environment, the task at hand and present events.

The frames in [12] are a data-structure intended for representing a stereotyped situation. The idea is that when encountering a new situation one selects a frame structure. When a proper frame is retrieved its slots are fill with available information. The matching process is controlled by the system's goals and by information attached to the frame [14]. Representations of events concentrates in two frames, one of the system tasks knowledge, and one representing the state of the world knowledge. Task frames would hold knowledge for the requested execution of a task. The representation of a world frame would try to maintain an accurate model of the agent's environment.

When working in dynamic environments it is not possible to cope with many details at once. At each moment one must work within simpler frameworks. One does not process the whole of a scene but constantly discriminates information. A worker focus all its attention into a small region of features. We propose building from the two event frames, representing the task and world knowledge, a single frame of what constitutes the relevant aspects of the scene. This active view frame, consist of knowledge from objects and relationships in the environment taken from the world frame according to what the task frame requires.

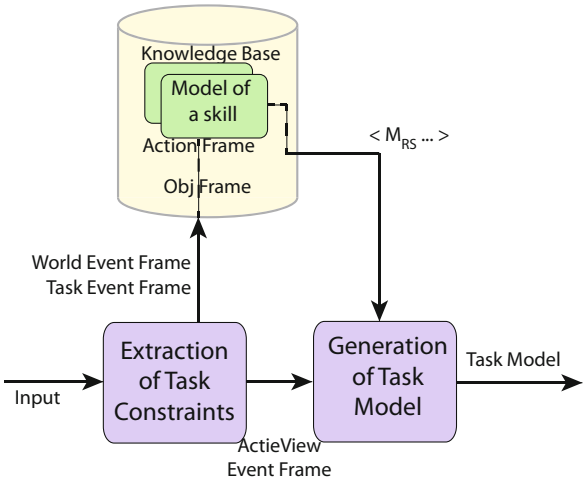


Fig. 6. Knowledge base control flow. World Event Frame and Task Event Frame are instantiated, and an Active View Event Frame is built from them. From Object and Action Frames the models of the skill are taken for building the task model.

4.4 Knowledge Base Structure

The process for using the representations begin by instantiating the appropriate frames. The system representations include objects, actions, and events as stated in subsections 4.1, 4.2 and 4.3. From a given scene the system instantiate frames, generally governed by the precedence of visual evidence. From the perceived given input the first step for extracting a task constraints is the matching of the world to an instance of the World Event Frame and the instantiation of the Task Event. From information collected in the World and Task event frames, which in turned are made up by other object and action frames, the system would have information about its current goals and situation of the environment, yet this is not enough to ground the representation in order to effectively use them for supporting its performance. [12] contend that any problem that a person can solve at all is worked out at each moment in a small context and that the key operations in problem solving are concerned with finding or constructing these working environments. For an agent's working in an unstructured environment the focus of its perception must be directed towards its executing action. Knowledge of its environment and task would be collected into their appropriated frames and a focused active view frame would be built taken from their global knowledge and breaking it down into a simpler framework from which computations and knowledge takes place. Figure 6 presents the control data flow for the process of using the representations in the knowledge base and Figure 7 presents the organization of the knowledge base.

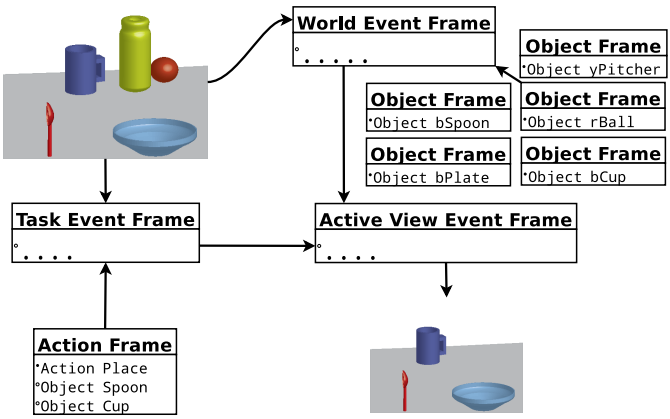


Fig. 7. Knowledge base structure and organization of the knowledge representations. World Event Frame and Task Event Frames represent the knowledge of the state of the environment, with Object and Action Frames representing the available objects and actions. From the knowledge of this frames an Active View Event Frame is built of the focused knowledge required to drive the agent execution.

5 Adaptation of Robot Skills

Reproduction of the robot skills if they are to be general enough need to present the capacity for adaptation and to generate new skills when the current situation of the world and its constraints of the task demand it.

The robot skills learned with the methodology described in Section 3 would present stable trajectories that accurately reproduce the demonstrated motion dynamics. The robot skills models were learned by employing a *DS* approach. By learning the robot skills under a probabilistic approach employing *GMM* the parameters governing the attractor dynamics of the motion are fully encoded into the parameters defining the Gaussian functions. The learned robot skills models would form a set of basic primitives of action from which a knowledge base of robot skills was built in Section 4.

An approach based on movement primitives relies on possessing available sequences of motor commands, executed in a certain order, to accomplish a given motor task. It is generally regarded that humans employ basic motor primitives as an underlying mechanism of biological motor control. Evidence exist from human and animal experiments supporting the believe that sets of motor primitives are used to build a basis for voluntary motor control [15]. By working under a theory based on the existence of basic primitives, from which full human motions are made off, it seen clear that following the issue of how to learn this basic units of action primitives the next question would be focused on how can this primitives be manipulated?

To generate complex motions from a learned set of basic primitive skills and be able to reproduce various complex task behaviours, methods for operating and manipulating upon the primitives are needed. The robot skills must be adaptable to conditions of its operating environment. Also, the models of a skill must be updatable, when given new information for the representation of a skill the system must allow for the models to be improved. Additionally, the action primitives approach must be able to generate new skills by merging two or more primitives into a new skill, multiple desired robot skills may be composed from superposition of various primitives. Another important property is the combination of the robot skills models to generate new models that encompass a larger spectrum of the attractor dynamics. A final desirable operation over the basic set of primitives skills consist on the sequencing and transition between models of robot skills in order to generate complex behaviors with smooth transformation among the reproduction of different skill motions.

Methods for model combination can be found in the field of machine learning and pattern recognition. One method involves the learning of different models and then use the average of the predictions made by each model. An alternative form chose one model to make the prediction as a function of the input variables, so different models become responsible for in different regions. Probabilistic framework for combining models, known as mixtures of experts, can be viewed as mixture distributions conditioned on the input variables. A mixture of experts can be given as:

$$p(t|x) = \sum_{k=1}^{K} \pi_k(x) p_k(t|x) \tag{5}$$

In which the mixing coefficients $\pi_k(x)$ are known as gating functions and the individual component densities $p_k(t|x)$ are called experts. The idea behind is that different components can model the distribution in different regions and the gating functions determine which components are dominant in which region.

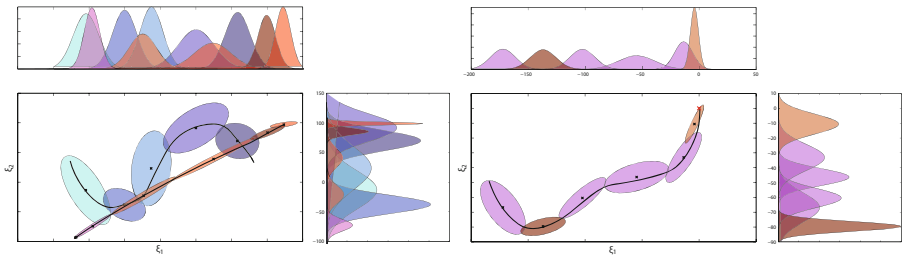


Fig. 8. Generation of a new model of a skill by combining previously learned skill models. Two or more basic models of a skill can be combine (top) to generate a new complex model (bottom) for complying with the constraints of the current task.

The robot would receive from the different modules of perception and interaction the required appropriate commands ordering the reproduction of a skill and extract the constraints of the task and its environment to instantiated the appropriate knowledge frames as for Section 4. With this information taken from the knowledge base, together with the robot skills models for the requested task, the module for the generation of task models is called to adapt the skills accordingly and generate the task models for the robot reproduction. Figure 8 illustrates the process.

6 Preliminary Experiments

To validate the proposed methods for generating new skills from previously learned models by updating or combining robot skills models we choose a very simple scenario in which the robot is required to grasp a plastic cup, see Figure 9.

The contemplated task requires for the robot to grasp the cup at any possible placement from a "cupboard" made up of two shelf, a bottom and a top shelf. The *HOAP-3* robot must be able to grasp the cup, as long as it is inside the robot's arm workspace, in any of the six possible placements, three at the bottom shelf and three at the top shelf, i.e., left-, right- or center-, top or bottom, respectively. Initially only the skills for learning to grasp the cup placed in the bottom shelf are taught to the robot by the methods described in 3. The complete task would

be unachievable with the robot skills learned so far, since the skill reproduction would not generalize well to the target new position at the top shelf. To grasp the cup, placed in the top shelf, at either side of the robot the skills learned to grasp the cup at the bottom shelf must be updated to generate the required new robot skill models. Finally, to generalize across the whole working space the three models of the robot skill, for right-, left- and center-, top or bottom shelf grasping, are combined into a single model of the attractor dynamics. Figure 9 illustrates the scenario.

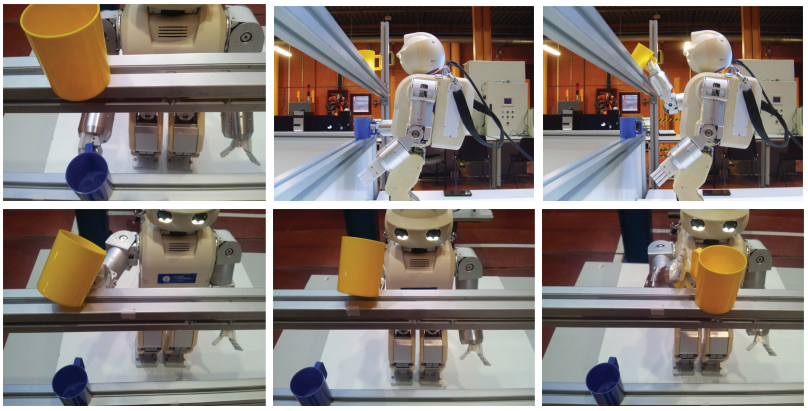


Fig. 9. Description for experiment B in skill generation and adaptation scenario. *HOAP-3* robot grasp a cup placed at different positions (top-row) Robot is taught a grasp skill motion for the cup place in the bottom shelf. By the update of the robot skill a new model is generate to allow the *HOAP-3* robot to grasp the cup place at the top shelf. (bottom-row) By the combination of various robot skills the *HOAP-3* robot can grasp the cup place at its right, center or left, using a single model of the skill.

This scenario is a demonstrator for the combination of the robot skills method presented in 5. First demonstrations are given, recorded by kinaesthetic teaching, to the learning module to encode the models of the robot skills for grasping trajectories at the possible locations, left, right and in front of the robot. Then the learned robot skills are feed to the skill combination algorithm to generate a new robot skill model from the combination of the previous skills. Finally, the model of the skill is given to the *HOAP-3* robot controller for execution of the skill.

A second experiment would have an agent and the *HOAP-3* robot interacting to complete a simple task. The task in this case requires the robot to pick up a cup and a spoon in each hand and then to put the spoon inside the cup, then finally it would put down the cup in front of it. The agent would provide the robot with the cup and spoon objects so it can pick them up, also the agent would indicate the robot where to put down the cup, see Figure 10.

Execution of the demonstration could vary depending on the actions of both the human agent and the *HOAP-3* robot. At the start of the demonstrator the robot is given the task event frame knowledge for the desire behaviour containing the knowledge of the 4 action skills needed to complete the action, pick spoon, pick cup, place spoon in cup, place cup down. Extracting the adequate action would depend on the agent interaction and the content of the rest of the knowledge base. The purpose of this demonstrator is to validate the performance of the developed knowledge base in a dynamic interaction with an agent where the invocation of an action skill is controlled by the representations in the knowledge base as is been described in 4.

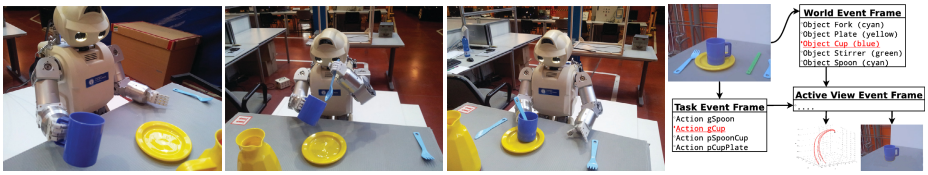


Fig. 10. Description for experiment B in the knowledge base scenario. At the beginning the task can be started by either picking the cup or the spoon. (left) The *HOAP-3* robot start the task by grasping the *blue cup*. (center-left) With the cup and the spoon in its hands the robot performs the action skill to put the spoon inside the cup. (center-right) Finally, the *HOAP-3* robot places the *blue cup* on the saucer plate. (right) State of the knowledge base at some step during execution. From the Task and World Events the Active View Event is built to drive the action execution.

This demonstrator highlights the operation of the knowledge base and how the representations in it of object, action, task event, world event and active view event frames are used to command the robot execution of the desired task.

The perception system handles the interaction with the user and the detection of objects in the environment. The knowledge base system would receive this information from the perception system and would instantiate the frames and built the knowledge representation of the scene in the knowledge base. The knowledge base system would select and activate an action skill when the conditions in the knowledge representation afford such action. Once an action is selected the *HOAP-3* robot controller would execute the robot commands require for the skill reproduction. This demonstrator scenario is mean to provide a proof of concept of how action execution is invoke by the state of the representation frames present in the knowledge base.

7 Summary and Conclusion

This work is centred on the aspiration of building humanoid robots capable of interacting with humans in their homes, workplaces, and communities, providing

support in several areas, and collaborating with humans in the same unstructured working environments. The aspiration is to have humanoid robots acting as robot companions and co-workers sharing the same space, tools, and activities. This paper focus is on topics concerning the learning, representation, generation and adaptation, and reproduction of robot skills.

The main contribution of this work is the proposition of a framework for the generation and adaptation of learned models of a skill for complying with task constraints, as in Figure 4. In which, a knowledge base of robot skills is built with the models of the skills learned through demonstrations. During execution the constraints of a requested task are extracted from the perception of the world state and the models of an appropriate skill are retrieved from the skills knowledge base. With all available representations a new adapted task model is generated for reproduction.

A *Learning from Demonstration* algorithm have been studied and implemented in teaching and learning with the robot the different sets of skills employed in the rest of the framework. The robot skills were learned in a *Dynamical System* approach.

A knowledge base of skills has been developed and implemented. Representations for robotics must include information about objects and actions, the world and situations, events and goals, for effective situated performance. A structure build on frames has been adopted in this work. The knowledge of the environment and goals is represented in terms of World Event Frame and Task Event Frames, with Object and Action Frames representing knowledge about available objects and actions respectively. From their knowledge an Active View Event Frame is built of the focused knowledge promoting the agent's execution.

Also, methods for model combination has been presented. Probabilistic frameworks for combining models can be viewed as mixture distributions conditioned on the input variables. The idea behind is that different components can model different regions and a gating function determines which components are dominant in which region. The manipulation of the skills must allow for the adaptation, update, merger, combination, and transition between the robot skills models as necessarily.

A framework for humanoid robots need to provide a minimum degree of intelligent behaviour, this is, the ability to sense the environment, learn, and adapt its actions to perform successfully under a set of circumstances. Efforts to generate robotic skills can only have a real implementation value for developing humanoid robotic systems if the models of the skill can be operated upon to generate new behaviours of increasing levels of complexity.

References

1. Albus, J.: Outline for a theory of intelligence. IEEE Transactions on Systems, Man and Cybernetics 21(3), 473–509 (1991)
2. Anderson, M.L.: Embodied cognition: a field guide. Artif. Intell. 149(1), 91–130 (2003), http://dx.doi.org/10.1016/S0004-3702(03)00054-7

3. Bechtel, W.: Representations and cognitive explanations: Assessing the dynamicistś challenge in cognitive science. Cognitive Science 22(3), 295–318 (1998)
4. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot programming by demonstration. In: Siciliano, B., Khatib, O. (eds.) Handbook of Robotics, pp. 1371–1394. Springer, Secaucus (2008)
5. Calinon, S.: Robot Programming by Demonstration: A Probabilistic Approach. EPFL/CRC Press (2009)
6. Dasgupta, S., Schulman, L.J.: A two-round variant of em for gaussian mixtures. In: UAI 2000: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, pp. 152–159. Morgan Kaufmann Publishers Inc., San Francisco (2000)
7. van Gelder, T., Port, R.F.: It's about time: an overview of the dynamical approach to cognition. In: Mind as Motion, pp. 1–43. Massachusetts Institute of Technology, Cambridge (1995), http://dl.acm.org/citation.cfm?id=225765.225768
8. Gribovskaya, E., Zadeh, K., Mohammad, S., Billard, A.: Learning Nonlinear Multivariate Dynamics of Motion in Robotic Manipulators. International Journal of Robotics Research (2010)
9. Ijspeert, A., Nakanishi, J., Schaal, S.: Trajectory formation for imitation with nonlinear dynamical systems. In: IEEE International Conference on Intelligent Robots and Systems, IROS 2001 (2001)
10. Khansari-Zadeh, S., Billard, A.: Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2676–2683 (2010)
11. Khansari-Zadeh, S., Billard, A.: Learning stable nonlinear dynamical systems with gaussian mixture models. IEEE Transactions on Robotics 27(5), 943–957 (2011)
12. Minsky, M.: A framework for representing knowledge. In: Winston, P. (ed.) The Psychology of Computer Vision, pp. 211–277. McGraw-Hill, New York (1975)
13. Poole, D., Mackworth, A., Goebel, R.: Computational Intelligence: A Logical Approach. Oxford University Press, USA (1998), http://books.google.es/books?id=3p6KZSHjD4YC
14. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence. Prentice Hall (2010), http://books.google.es/books?id=8jZBksh-bUMC
15. Schaal, S.: Is imitation learning the route to humanoid robots? (6), 233–242 (1999), http://www-clmc.usc.edu/publications/S/schaal-TICS1999.pdf, http://www-clmc.usc.edu/publications/S/schaal-TICS1999-rep.pdf
16. Sung, H.G.: Gaussian mixture regression and classification. Ph.D. thesis, Rice University (2004)

AVASTT: A New Variable Stiffness Actuator with Torque Threshold

J. López Martínez[1], J.L. Blanco[1], D. García Vallejo[2], J.L. Torres[1], and A. Giménez Fernández[1]

[1] Department of Engineering, University of Almería, Spain
javier.lopez@ual.es
[2] Department of Mechanical Engineering and Materials, University of Seville, Spain

Abstract. Variable Stiffness Actuators (VSAs) emerged as an alternative to conventional actuators in a variety of applications, such as walking robots and service robotics. New requirements, which were obviated in the design of rigid actuators, must be accounted for during the mechanical design of such new devices. Among them, we find the possibility of tunning the natural frequency of the mechanical system or damage reduction in case of impacts A multitude of solutions have been already proposed in the literature, each characterized by the kind of mechanism in charge of implementing the equilibrium position and the stiffness of the joint. With this work, we introduce a new actuator design based on tendon transmission, where a main motor controls the equilibrium position of the link while a secondary motor is in charge of modifying the joint stiffness. Unlike existing actuators, our proposal achieves a wide range of stiffness values, from close to zero up to completely rigid. Another distinguishing feature of the new design is the existence of a torque threshold, such that variable stiffness only becomes effective once the load is above a certain predefined threshold.

1 Introduction

In recent years, advance in the variable stiffness actuators (VSA) has been notable in the robotic field. Several designs of VSA have been proposed and studied the performance in different applications.

The concept of a VSA is to elastically decouple the link inertia and the motor rotor inertia, where the stiffness of the coupling can be modified and controlled. This mechanical implementation has two main applications. On the one hand, the elastic component of the variable stiffness mechanism allows to store and release energy, reducing the energy consumption and maximizing the velocity peak torque achievable with a classical "stiff" actuator. Here, the capability of modify the stiffness of the joint is needed to adjust the natural frequency of the system. The most relevant application is in walking and running robots and in robotic prosthesis, where by varying the stiffness of the mechanism, the natural frequency is modified and adjusted to different step frequency [1–4]. Also, the capability of store and release energy can be used to achieve faster movements

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_42,

of a robot arm. In a rigid actuator the link speed is the same of the motor, but in a flexible joint, the potential energy can be used to accelerate the link with respect the output shaft of the motor. Albu-Shäffer et al. [5] have experimentally confirmed a speed gain of 265% for the link velocity between a rigid and an elastic joint.

On the other hand, the VSA can be used as a safety mechanism. Since the VSA decouples the rotor inertia from the link inertia, the rotor inertia and the inertia of the rest of the links of the kinematic chain do not take part in case of an unexpected impact with the environment. This mean lower impact forces for the safety of the environment, or for the human safety in the case of a service or an assistive robot, and lower joint peak torque for joint protection. For a safety robot, the VSA can be act stiff during precise tasks at low speeds and compliant in fast movements when positioning is not important [6].

As a result of works carried out in different robotic fields where VSAs find applications, a large number of designs have been proposed. According to the functional concept and disposition of the flexible components of the VSA, they can be classified into antagonistic configuration or serial configuration. In the antagonistic configuration, two elastic elements act in opposite directions at the joint, and the equilibrium position and the stiffness of the joint are controlled by two motors. In the majority of antagonist configurations, motors only act over one of the two elastic components, so that motors also act in opposite directions jointly changing both the position and the stiffness of the joint [7, 8]. Nevertheless, under the point of view of energy efficiency it becomes desirable to use each motor for one independent task [9]; in those cases, a first motor controls the link position while another one changes the joint stiffness [10, 11]. In the serial configuration only one flexible element is required between the main motor, in charge of settling the equilibrium position, and the link; a second, smaller motor, modifies the stiffness [12–15, 4].

The present work introduces the design of a novel VSA, named *AVASTT* (Actuator with VAriable Stiffness and Torque Threshold), whose working principles rely on tendon transmission, and set it apart from previous similar devices due to the wide stiffness range, from close to zero up to ideally rigid; and the existence of a torque threshold, where the compliance of the actuator only acts when the external torque reach a predefined minimun value. In section 2 we firstly review the desiderata regarding design criteria of this kind of actuators. Next, our proposed design is introduced and a mathematical modeling is derived in section 3, while the next section describes our practical implementation of a physical prototype. We end providing some conclusions in section 6.

2 Design Criteria

The most relevant design criteria of a rigid actuator are the maximum torque that the driving motor should provide and the maximum rotational speed. In contrast, new important characteristic must be taken into account when dealing with VSA:

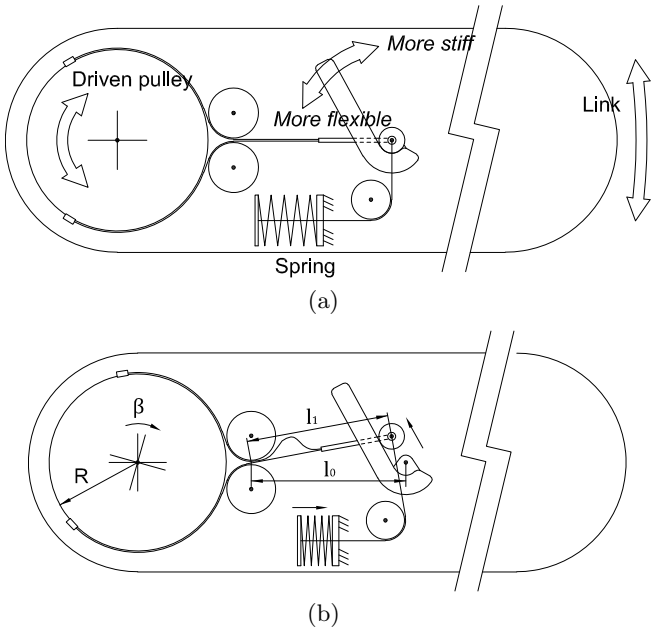


Fig. 1. The conceptual design of our VSA, (a) in its rest configuration and (b) deformed under an external force

- Allowed stiffness range.
- Permitted range for the angular deviation.
- Dependence between stiffness and angular deviation.
- Energy storage capability.

The characteristics of the different kinds of VSAs are strongly conditioned by the specific type of mechanism employed to control the position and stiffness of the actuator. Apart from the classification into antagonist versus serial configuration, another key feature of VSAs is the way in which stiffness is actually modified. The most frequently-adopted solution is the pretension of the flexible component, usually a spring, which by means of the appropriate mechanism becomes a variation of the actuator stiffness. However, an important drawback of this approach is that the motor must operate working against the flexible element [7, 8, 10, 11, 13, 14, 4]. Different proposals that can be found in the literature rely on the modification of the length of a lever, thus maintaining the pretension of the elastic component constant during operation [12, 15].

Next we address the main aspects of the above mentioned VSA properties:

- **Allowed stiffness range.** It is defined by the minimum and maximum stiffness values allowed by the device. Although this range may depend on the particular application, in principle it would be desirable to have the largest possible range. Most previously-proposed VSAs have a limited stiffness range

and only a few proposals (e.g. those based on the variation of the lever length) can reach a totally rigid configuration. Among those ones, AWAS-II [15] can also achieve, at least in theory, a null stiffness.

- **Range of angular deviation.** This is the range of the potential angles between the equilibrium position and that of the link under external loads. The mechanism should allow a range of angles wide enough such that, during normal operation, it does not reach its mechanical limit positions. A deviation of 15° is considered enough for humanoid robot arms [11].
- **Dependence between stiffness and angular deviation** (stiffness curve), that is, how the stiffness varies as the angular deviation increases. In applications where the goal is reducing the damage caused by a potential impact between the robot and a human, increasing stiffness is preferable [11, 13–15]. An stiffness that increases with deformation has been also found to be desirable in active prostheses and in hopping robots [4].
- **Energy storage capability.** This property is closely related to the range of angular deviation and the stiffness curve. The overall elastic potential energy could be employed to absorb kinetic energy in an impact or to intentionally accelerate the link motion by means of a proper control strategy [11].

3 Mathematical Modeling

In this section we first describe the working principles of our new AVASTT actuator, then derive a suitable mathematical model for it. We must remark that during the design phase we took in mind all the aspects mentioned in the previous section.

The proposed mechanism comprises a serial configuration, with the driving rotor determining the equilibrium position of the output link and a secondary motor to modify the stiffness of the joint. Fig. 1 illustrates the conceptual design, whose main characteristic is the usage of a tendon-pulley transmission. The output shaft of the motor is attached to the driving pulley, which has two wires attached each with its own tensioner. In turn, the cables are attached to an internal rigid rod whose end is pinned to a cylinder that can freely roll over a lever. A third tendon is attached between the cylinder axis and a spring that is attached to the link frame.

When a torque is applied by the motor shaft, the spring is compressed. This elastic deformation leads to the angular deviation β between the shaft motor position and the output link –refer to Fig. 1(b). The value of this angle clearly reflects the degree of stiffness of our VSA, and is determined by the angular position of the lever actuated by the secondary motor. When the lever is exactly in the vertical position, according to Fig. 1(a), the actuator becomes an ideal rigid actuator since no angular offset is allowed between the input and the output link. On the other hand, when the lever is horizontal we achieve maximum flexibility. Notice that Fig. 1(a)–(b) show two different states of our design for the same configuration of the internal lever, i.e. constant stiffness, but with a zero and a nonzero input torque, respectively.

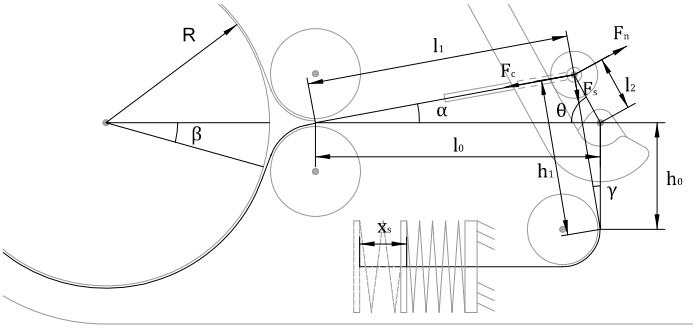


Fig. 2. Schematic description of all the variables involved in the analysis of our VSA

In order to properly characterize our actuator we must obtain the relationship between the actuating torque T and the resulting angle β. Please, refer to Fig. 2 for the physical meaning of all the variables that we present next while searching for the relationship between the angular deviation β and the torque T and the stiffness K.

When the driving pulley rotates an angle β with respect to the equilibrium position, and neglecting the elastic elongation of the tendons, it is clear that part of the initial length l_0 will move towards the pulley (in particular, a length of βR) while the rest must match the final length of the cable (l_1 in the figure). Therefore,

$$l_0 = l_1 + \beta R \tag{1}$$

In this new position, the cylinder has rolled over the lever a distance l_2, which depends on the angular position of the lever. Let θ be this parameter that controls the stiffness of our actuator. Then, the angle α can be obtained from the triangle with sides l_0, l_1 and l_2 as:

$$\sin\alpha = \frac{l_2 \sin\theta}{l_1} \tag{2}$$

and regarding the rolling distance l_2, it can also be obtained from the cosine rule applied to the same triangle, which leads to:

$$\begin{aligned} & l_1^2 = (l_2 \sin\theta)^2 + (l_0 - l_2 \cos\theta)^2 \\ \longrightarrow \quad & l_2 = l_0 \cos\theta - \sqrt{l_1^2 - l_0^2 \sin^2\theta} \end{aligned} \tag{3}$$

The displacement of the cylinder generates a spring compression of:

$$x_s = h_1 - h_0 \tag{4}$$

with h_0 and h_1 the initial and final cable lengths as depicted in Fig. 2. It can be then shown that:

$$h_1 = \sqrt{(l_1 \sin\alpha + h_0)^2 + \left(\frac{l_1 \sin\alpha}{\tan\theta}\right)^2} \tag{5}$$

By now substituting (1), (2) and (3) into (5), then the resulting expression into (4), we arrive at the expression that relates the spring compression x_s with the angular torsion β,

$$x_s = \sqrt{A^2 + h_0^2 - 2\mathbf{h}_0 A \sin\theta} - h_0 \tag{6}$$

$$\text{with: } A = \sqrt{(l_0 - R\beta)^2 - l_0^2 \sin^2\theta} - l_0 \cos\theta$$

Next we address the evaluation of the torque T from the spring elastic force. The tension of the spring tendon is simply its stiffness K_s times the deformation, i.e.

$$F_s = K_s x_s \tag{7}$$

Three different forces act on the rolling cylinder: (i) the force F_s above, (ii) a reaction force F_n, normal to the lever, and (iii) the tension F_c from the tendon attached to the driving pulley. Solving for force equilibrium we arrive at:

$$F_c = F_s \frac{\sin(\theta + \gamma)}{\cos(\theta + \alpha)} \tag{8}$$

with the angle γ given by:

$$\gamma = \tan^{-1} \frac{l_2 \cos\theta}{h_0 + l_2 \sin\theta} \tag{9}$$

Finally, the torque T can be written as (the complete expression is not shown for the sake of conciseness):

$$T = F_c R = T(R, l_0, h_0, K_s, \theta, \beta), \tag{10}$$

and taking derivatives of (11) with respect to β we obtain the torsional stiffness of our actuator:

$$K = \frac{dT}{d\beta} = K(R, l_0, h_0, K_s, \theta, \beta), \tag{11}$$

Thus, both the torque T and the stiffness K depend on a number of parameters: mechanism dimensions (R,l_0,h_0), the linear stiffness of the spring (K_s), the lever angle (θ) and the angular deviation (β). For any given actuator the first four parameters will remain fixed, while the lever position θ will be controlled with the secondary motor. As previosuly introduced, when the lever is in the vertical position (i.e. (θ)=0 deg) no angular deviation is mechanically allowed, then the actuator is completely rigid. In the opposite behaviour, if the lever is horizontal (i.e. (θ)=0 deg), no force is requiered at the beginning of the angular deviation (as derived from (8)), hence null stiffness is showed. Others main properties of the proposed VSA are summarized besides the mechanical implementation in 5.

4 Torque Threshold

A singular feature of the AVASTT is the possibility of adjust a torque threshold. In this configuration, the joint act as a rigid one since the external force applied in the link do not overtakes a selected threshold. Once this value is exceeded the compliance of the mechanism comes into operation. With this feature, the robot can work as a rigid robot in precise positioning tasks assuring the joint protection in case of an external overload. Also, in a compliant configuration of the VSA, a slight torque threshold makes the joint more stable without losing its benefits of store and release energy.

In AVASTT the threshold torque can be adjusted with the tensioner of the wires. If the wires have an initial pretension, in the equilibrium position of the joint the two wires have the same value of tension. When an external force is applied at the link, the tension of one of the two wires start increasing, while is decreasing in the other wire (Fig. 3). Until the first wire has doubled the initial tension and the second one is loosed the spring do not start compression. Then, adjusting the wires initial tension it can be modified the torque threshold at which the compliant actuator acts.

5 Mechanical Implementation

Fig. 4 illustrates the mechanical implementation of the AVASTT. The driving (main) motor is a Maxon DC motor with a nominal torque of 0.4 Nm, coupled to a planetary gearhead with a gear ratio of 100:1. As a secondary motor we employ a Maxon DC motor with a peak torque of 0.14 Nm. The latter motor acts on the lever that controls the system stiffness by means of a non-reversible worm drive with a ratio of 60:1. The linear spring has a stiffness of 80.5 kN/m and a maximum length compression of 34 mm, which limits the angular deviation (β) of the mechanism to a maximum of 54°, when used in its minimum-stiffness configuration.

We present in Fig. 5(a) a number of torque vs. angular deviation curves for different configurations of our VSA (i.e. different positions of the inner lever), which have been computed with the nominal dimensions of our prototype ($R = 7$ cm, $l_0 = 151$ mm, $h_0 = 47$ mm). It can be seen how the curves present an increasing slope as the lever angle increases, becoming totally vertical when the mechanism limit is reached, that is, when the pulley cable becomes perpendicular to the lever. Fig. 5(b) shows the stiffness curves, which also exhibit the same behavior of increasing with the angular deviation. A comparative between the main characteristics of our prototype and other representative VSA can be found in Table 1. Notice that the dimensions of this first prototype have been largely overestimated in order to make it resistant to severe impact tests. A lighter and more compact design is feasible in order to make it suitable for integration into humanoid robots.

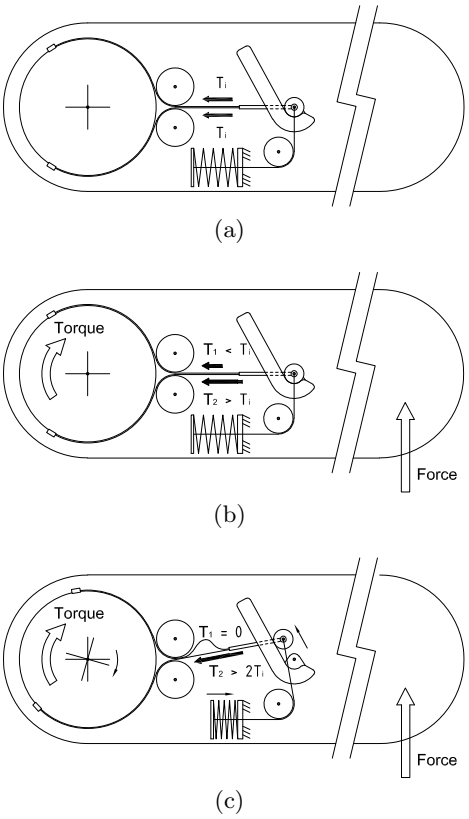


Fig. 3. Illustration of how pretensioning of the inner cables can be exploited to achieve a torque threshold in the behavior of the actuator. a) Equilibrium position. Wires initial pretension. b) Applied external force below the threshold limit. c) Applied external force over the threshold limit, VSA actuation.

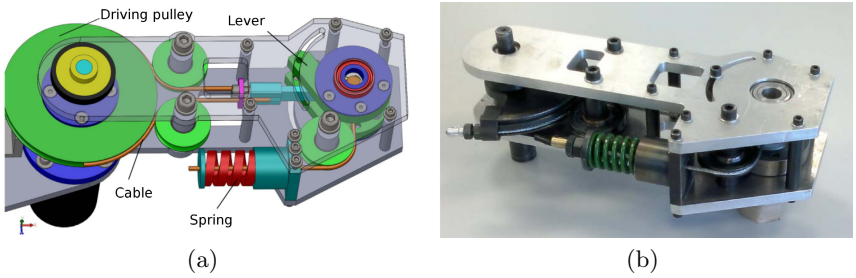
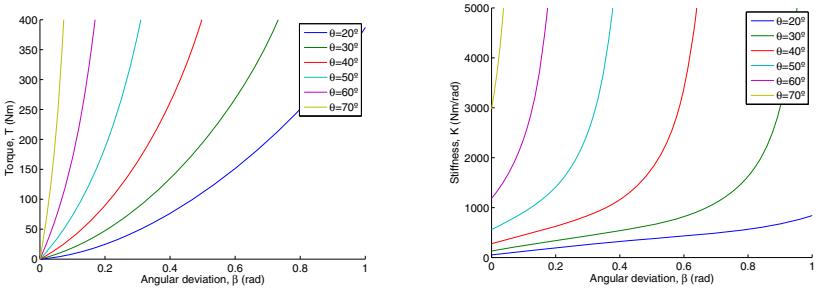


Fig. 4. (a) CAD model and (b) real picture of our VSA design, without the motors

Table 1. Comparison of the AVASTT prototype with other VSAs

	AVASTT (This work)	AwAS [16]	AwAS-II [15]	FSJ [16]	MACCEPA [2]	VSA-CUBE [17]	VSA-HD [18]
Nominal torque (Nm)	40	10.75	10.75	31.3	50	1.1	10
Maximum stiffness (Nm/rad)	∞	1300	∞	826	110	14	8360
Minimum stiffness (Nm/rad)	0	30	0	52.4	5	3	0.38
Maximum elastic energy (J)	46	3.5	5.8	5.3	27.9	0.047	0.12
Maximum deflection ($^{\circ}$)	54	14	17	15	60	16	60
Active rotation angle ($^{\circ}$)	360	120	150	180	150	120	360
Mass (kg)	4.5	1.8	1.4	1.41	2.4	0.26	1.7



(a) (b)

Fig. 5. (a) Torque-angle and (b) stiffness-angle curves for our prototype

6 Conclusions

With this work we have introduced a new VSA design. We opted for a serial configuration design, where a main motor drives the actuator while a secondary one is in charge of modifying the stiffness by means of rotating an inner lever. Our design allows fulfilling with the imposed design requisites, with a wide range of achievable stiffness, even allowing a complete rigidity. One particularity of the presented model is the possibility of adjust a torque threshold, with this feature the compliance of the joint only acts when the external force overtakes a certain level. Future works will address an extensive experimental validation of the physical prototype in order to verify its predicted properties.

Acknowledgements. This work has been partially funded by the Spanish "Ministerio de Ciencia e Innovación" under the contract "DAVARBOT" (DPI 2011-22513) and the grant program JDC-MICINN 2011, as well as by the Andalusian Regional Government grant programs FPDU 2008 and FPDU 2009, cofinanced by the European Union through the European Regional Development Fund (ERDF).

References

1. Hurst, J.W., Chestnutt, J.E., Rizzi, A.A.: An actuator with physically variable stiffness for highly dynamic legged locomotion. In: IEEE International Conference on Robotics and Automation (ICRA), vol. 5, pp. 4662–4667. IEEE (2004)

2. Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B., Lefeber, D.: Maccepa, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. Robotics and Autonomous Systems 55(10), 761–768 (2007)
3. Sugar, T.G., Hollander, K.W., Hitt, J.K.: Walking with springs. In: SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring, International Society for Optics and Photonics, pp. 797602–797602 (2011)
4. Vanderborght, B., Tsagarakis, N.G., Semini, C., Van Ham, R., Caldwell, D.G.: Maccepa 2.0: Adjustable compliant actuator with stiffening characteristic for energy efficient hopping. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 544–549. IEEE (2009)
5. Albu-Schaffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimbock, T., Wolf, S., Hirzinger, G.: Soft robotics. IEEE Robotics & Automation Magazine 15(3), 20–30 (2008)
6. Bicchi, A., Tonietti, G.: Fast and "soft-arm" tactics. IEEE Robotics & Automation Magazine, 22–33 (2004)
7. Migliore, S.A., Brown, E.A., DeWeerth, S.P.: Biologically inspired joint stiffness control. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4508–4513. IEEE (2005)
8. Schiavi, R., Grioli, G., Sen, S., Bicchi, A.: Vsa-ii: A novel prototype of variable stiffness actuator for safe and performing robots interacting with humans. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2171–2176. IEEE (2008)
9. Vanderborght, B., Van Ham, R., Lefeber, D., Sugar, T.G., Hollander, K.W.: Comparison of mechanical design and energy consumption of adaptable, passive-compliant actuators. The International Journal of Robotics Research 28(1), 90–103 (2009)
10. Hurst, J.W., Chestnutt, J.E., Rizzi, A.A.: The actuator with mechanically adjustable series compliance. IEEE Transactions on Robotics 26(4), 597–606 (2010)
11. Eiberger, O., Haddadin, S., Weis, M., Albu-Schaffer, A., Hirzinger, G.: On joint design with intrinsic variable compliance: Derivation of the dlr qa-joint. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1687–1694. IEEE (2010)
12. Kim, B.S., Song, J.B.: Hybrid dual actuator unit: A design of a variable stiffness actuator based on an adjustable moment arm mechanism. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1655–1660. IEEE (2010)
13. Hyun, D., Yang, H.S., Park, J., Shim, Y.: Variable stiffness mechanism for human-friendly robots. Mechanism and Machine Theory 45(6), 880–897 (2010)
14. Wolf, S., Hirzinger, G.: A new variable stiffness design: Matching requirements of the next robot generation. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1741–1746. IEEE (2008)
15. Jafari, A., Tsagarakis, N.G., Caldwell, D.G.: Awas-ii: A new actuator with adjustable stiffness based on the novel principle of adaptable pivot point and variable lever ratio. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4638–4643. IEEE (2011)
16. Wolf, S., Eiberger, O., Hirzinger, G.: The dlr fsj: Energy based design of a variable stiffness joint. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 5082–5089. IEEE (2011)

17. Catalano, M.G., Grioli, G., Garabini, M., Bonomo, F., Mancinit, M., Tsagarakis, N., Bicchi, A.: Vsa-cubebot: A modular variable stiffness platform for multiple degrees of freedom robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 5090–5095. IEEE (2011)
18. Catalano, M.G., Grioli, G., Bonomo, F., Schiavi, R., Bicchi, A.: Vsa-hd: From the enumeration analysis to the prototypical implementation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3676–3681. IEEE (2010)

A Kinect-Based Motion Capture System for Robotic Gesture Imitation

José Rosado[1], Filipe Silva[2], and Vítor Santos[3]

[1] Department of Computer Science and Systems Engineering, Coimbra Institute of Engineering, IPC, Coimbra, Portugal
jfr@isec.pt
[2] Institute of Electronics Engineering and Telematics of Aveiro, Department of Electronics, Telecommunications and Informatics, University of Aveiro, Aveiro, Portugal
fmsilva@ua.pt
[3] Institute of Electronics Engineering and Telematics of Aveiro, Department of Mechanical Engineering, University of Aveiro, Aveiro, Portugal
vitor@ua.pt

Abstract. Exploring the full potential of humanoid robots requires their ability to learn, generalize and reproduce complex tasks that will be faced in dynamic environments. In recent years, significant attention has been devoted to recovering kinematic information from the human motion using a motion capture system. This paper demonstrates and evaluates the use of a Kinect-based capture system that estimates the 3D human poses and converts them into gestures imitation in a robot. The main objectives are twofold: (1) to improve the initially estimated poses through a correction method based on constraint optimization, and (2) to present a method for computing the joint angles for the upper limbs corresponding to motion data from a human demonstrator. The feasibility of the approach is demonstrated by experimental results showing the upper-limb imitation of human actions by a robot model.

Keywords: 3D pose estimation, constraint optimization, articulated structures, inverse kinematics, gesture imitation.

1 Introduction

Programming robots to perform complex tasks and extends its repertoire can be extremely tedious and time consuming. Learning from demonstration is a promising methodology that offers a more intuitive approach to teach a robot how to generate its own motor skills [1-2]. To this end, the robot should be able to estimate human poses when performing a desired task, as well as to translate the skeleton data into appropriate motor commands. In the last years, a large body of work has studied the use of computer vision-based human motion capture for extracting poses as input to robotic applications [3-7]. Despite much research progress, these systems are usually expensive, require careful calibration and its application is limited to rigid

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_43,

environments. To overcome these limitations, the main challenge with the imitation learning paradigm is to develop accurate methods for extracting 3D human poses from image sequences using low-cost systems as a valid alternative.

Recently, the field of markerless motion capture has experienced a strong evolution with the development of high-speed and cheap depth cameras. In particular, the depth data provided by the PrimeSense sensor opened up new opportunities for extracting gesture-based interactions with a more portable and less costly system. The publication of the tracking algorithm of the Kinect Software Development Kit [8] and the availability of several development environments (*e.g.*, OpenNI, Microsoft SDK) have contributed for a growing interest in model-free approaches. However, the success of these alternatives depends on the accuracy and robustness required in each specific area of application.

This paper addresses the main problem associated with the use of a Kinect-based human motion capture in robotics: the lack of a kinematic model to assure coherence in the provided poses. The main objective is to demonstrate and evaluate both a human action pose correction method and an inverse kinematics technique. The former aims to assure constant limb lengths over an entire sequence of poses. The later converts each of the 3D poses into the corresponding angles for the upper-body joints, including a validation test to deal with physical limits (*e.g.*, joint velocity limits). The motivation of this work is to create a database of classified motions to learning control in robotics. In line with this, the remainder of the paper is organized as follows: Section 2 presents the motion capture system based on a single Kinect camera and the experimental conditions. Section 3 describes the pose correction method based on constraint optimization. Section 4 focuses on the kinematic mapping from 3D poses to joint angles. Section 5 discusses the results achieved to validate the proposed solutions. Finally, Section 6 concludes the paper and proposes future extensions.

2 Human Motion Capture

The Kinect sensor provides a 640×480 depth image, at 30 frames per second, for the skeleton-based pose estimation with depth resolution of a few centimeters. The human skeleton estimated from the depth image includes a total of 20 body joints that will be the input for our approach. These captured data consists of a set of Cartesian points in the 3D volume for each human pose, which will be called raw-data hereinafter. Several studies have assessed the accuracy of the depth reconstruction and joint positions from the Kinect pose estimation, including comparisons with ground truth motion capture data [9-11]. In general, these studies highlight the potential of the Kinect skeleton in controlled body postures whenever self-occlusions are avoided.

In our experiments, we have used a single Kinect camera positioned at about 3 meters from the human subject to capture the whole body standing upright. In this study the attention is dedicated to the upper limbs, including the shoulder, elbow and wrist joints of both right and left arms. In order to ensure the most convenient acquisition conditions, the human subject was asked to prevent lower trunk movements and to perform controlled scapular motions. Precautions were also taken to avoid occlusions of the upper limb parts.

Besides the accuracy and robustness of the skeletal poses, a critical element is the stability of the estimated frame-to-frame body geometry. As mentioned before, a characteristic of the human body skeletonization with the Kinect sensor is that the limb lengths are not kept constant through the entire sequence and differ between the two arms. Fig. 1 illustrates the variations of the limb lengths, from frame-to-frame, for a static posture and a reaching arm movement. In the static case, the mean value rounds 268 mm for the arm and 233mm for the forearm, while the standard deviation is around 3.65 mm and 1.51 mm, respectively. These measures are significantly different during the execution of a reaching movement: a mean of 265 mm for the arm and 216 mm for the forearm with a standard deviation of 15.9 mm and 8.8 mm, respectively.

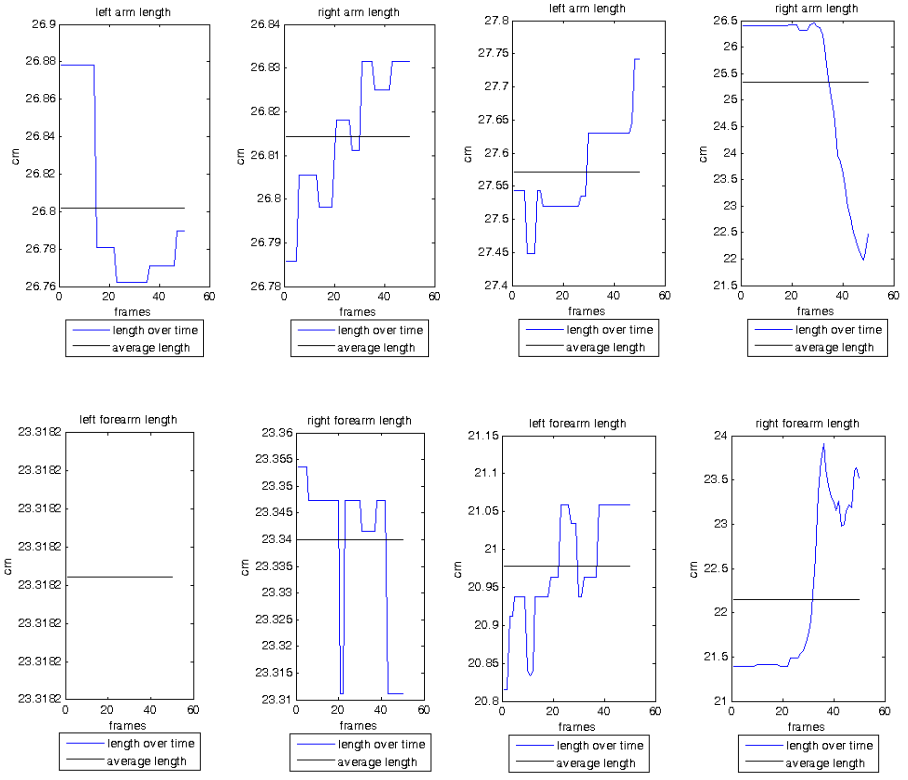


Fig. 1. Frame-to-frame variation of the limb lengths for a static posture (left) and a reaching arm movement (right)

3 Constrained-Based Motion Filtering

The pose correction method aims to convert the motion of a source human subject into a new motion, while satisfying a given set of kinematic constraints. These

kinematic constraints are formulated in order to assure a kinematic model with constant limb lengths. The proposed method, applied to each individual frame, can be divided into two main steps:

- Static calibration: the first step is a static calibration of the arms, prior to each data collection, to define the reference model of the subject anthropometry. Concretely, the human subject was told to hold their arms full extended aligned with the trunk (fundamental standing position), while several frames are acquired. A distance vector among consecutive joints (shoulder-elbow and elbow-wrist) is calculated as the mean value taken over all these frames for both arms. It should be pointed out that this arm calibration is the basis for the joint-angle calculations in Section 4: all joints angles are defined as zero degrees at this calibration posture.
- Pose correction: the basic problem is to find the closest configuration $X = (x_1, x_2, ..., x_n) \in \Re^{3 \times n} \quad (x_1, ... x_n \in \Re^3)$ to the measurements that are observed over time $\hat{X}$, such that the distance between consecutive points (*i.e.*, link lengths) remains constant.

In line with this, we deal with the following optimization problem:

$$\min_X \sum_{i=1}^{n} \left\| X - \hat{X} \right\| \tag{1}$$

where $\|\cdot\|$ is an appropriate matrix norm which measures goodness of fit. Here, we admit the Euclidean norm as measure of closeness. The goal is to minimize the objective function (1) by selecting a value of X that satisfies all equality quadratic constraints defined by:

$$\left\| x_i - x_{i+1} \right\| = d_{i,i+1} \tag{2}$$

where the left part is the Euclidean distance between two consecutive points and the right part is the link lengths in the reference model.

Since the main goal is to generate large sets of human motion data for robotic imitation, all the computations at this point are performed offline using the Matlab environment. The constrained minimization problem was solved with the OPTI toolbox that can solve this problem of optimizing a quadratic function of several variables subject to quadratic constraints. In this context, the execution time is around 15 seconds per frame on an Intel Core2 Duo CPU T9400 @ 2,53GHz. The comparison of the human skeletons obtained with the Kinect raw-data and those after the pose correction are illustrated in Fig. 2. Different poses are represented for a movement sequence involving both the right and the left arm. Table 1 presents some statistical measurements applied for quantifying the error between the Kinect raw-data and the filtered data after the pose correction.

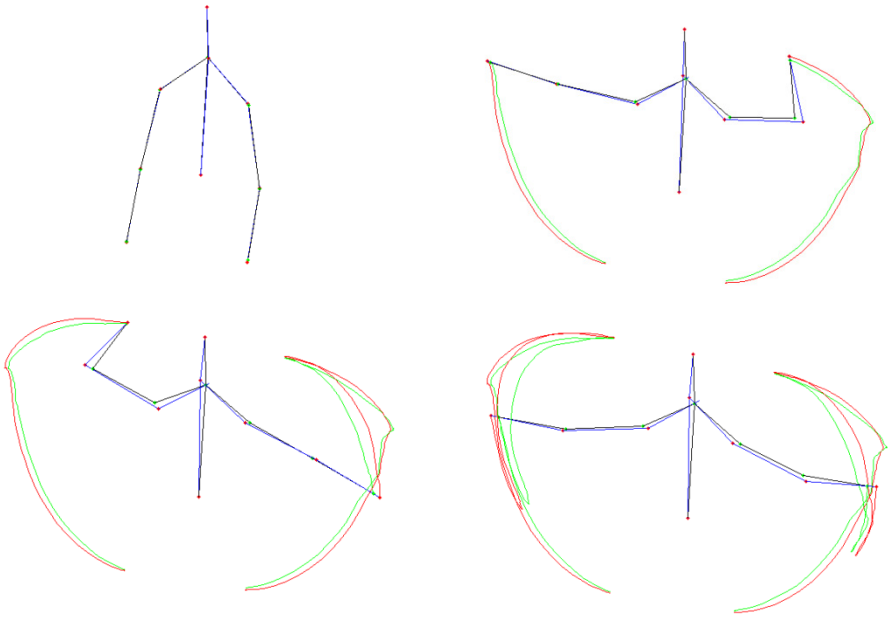


Fig. 2. Overlap of the human skeletons extracted from the Kinect raw-data (green points and black lines) and those after the constraint-based optimization (red points and blue lines) at different frames (green and red lines represent the end-effectors' paths)

Table 1. Error quantification between the Kinect raw-data and the filtered data

Joint	Minimum error (mm)	Maximum error (mm)	Mean error (mm)	Standard deviation
Left Shoulder	0.5	25.7	15.1	7.0
Left Elbow	8.6	89.5	31.7	14.7
Left Wrist	5.2	75.2	30.9	16.0
Right Shoulder	0.8	56.0	21.8	13.5
Right Elbow	2.4	86.5	27.2	14.7
Right Wrist	4.2	82.6	25.0	18.6

4 Kinematic Mapping

One of the main issues in using motion capture data for training robots is to convert the 3D joint positions into joint angles relative to a robot model. In this context, the human skeleton is replaced by two 4 degree-of-freedom (DOF) robot arms of the same dimensions. Then, an inverse kinematics algorithm generates the corresponding

joint angles of the robot for each pose. The problem is decomposed into a per-frame inverse kinematics algorithm, followed by motion filtering and interpolation.

4.1 Inverse Kinematics

The filtered movement data is the input for the inverse kinematics module in which the human arms are modeled as two independent 4-dof serial chains consisting of a 3-DOF shoulder (rotations joints with intersecting axis) and a 1-DOF elbow joint. The implementation of the inverse kinematics follows some basic assumptions. First, the robot model was defined to match the anthropometric measures of the human subject, avoiding the retargeting problem (*i.e.*, compensate for body differences). Second, the perturbations in the movement data caused by the movement of the subject's shoulder are ignored. Concretely, we consider that all joint positions are uniformly affected by the perturbations and the shoulders are at the origin of the reference system with fixed coordinate frames. Third, the inverse kinematics considers mechanical constraints on the joints, such as physical limits both on the range of joint motions (*e.g.*, the elbow cannot invert the motion when full-stretched) and on the maximum joint velocities.

Given the 3D positions of the shoulder, elbow and wrist, the inverse kinematics algorithm is simplified: two degrees of freedom completely describe the elbow when the position of the shoulder is known (the elbow lies on the surface of a sphere centred at the shoulder). Similarly, the wrist can only lie on the surface of a sphere centred at the elbow. Thus, the configuration of the arm is completely represented by four variables (the joint angles). Attention was devoted to avoid discontinuous jumps near $\pm 180^{o}$ associated with the use of inverse tangent functions.

Additionally, the implemented algorithm includes a validation test since there may be motions where the robot's joints are not able to approximate the human pose in a reasonable way due to physical limitations. The proposed strategy to properly cope with the joint velocity limits is to slowing down the task-space trajectory whenever the limits are encountered. Thus, whenever the generated joint velocities violate the speed limits of the joint actuators, the trajectory is scaled in time by an appropriate constant that simultaneously assures tracking of the desired arm path and the fulfillment of the velocity constraints.

4.2 Filtering and Interpolation

The frame rate of the Kinect sensor and high frequency components in the movement data imposed a post-processing stage to refine results. The exact procedure combines basic interpolation and smoothing techniques. On the one hand, the joint-angle trajectories are filtered using a moving average algorithm to smooth out short-term fluctuations based on predefined trail onset and termination times. On the other hand, the strategy adopted to provide a more detailed description of the action performed by the human subject is to use spline interpolation over the set of observations to satisfy the requirements of differentiability. To evaluate the different steps of post-processing, we use a measure based on jerk, the third time derivative of position, to

quantify smoothness at the level of the joint-angles trajectories. Concretely, the particular jerk metric used to quantify movement smoothness is the integrated squared jerk [12] defined by:

$$\eta_{isj} = \int_{t_1}^{t_2} \dddot{x}(t)dt \tag{3}$$

A comparison of movement smoothness measures among the original signal (after pose correction), the moving average filtered signal, the cubic spline interpolation and the fifth-order spline interpolation was performed (Fig. 3). The exact procedure to be followed depends on the ultimate goal. Anyway, the previous considerations may be of importance in determining what strategies are appropriate to the problem in hand.

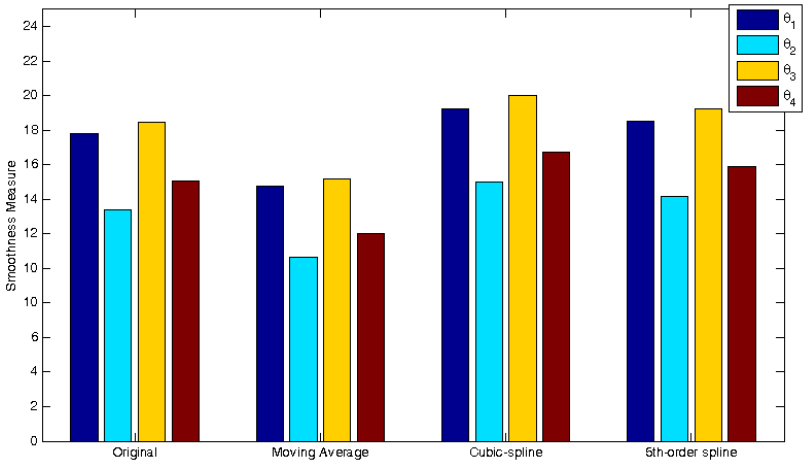


Fig. 3. Comparison of the smoothness measure for different motion post-processing methods applied on the joint-angle trajectories (for graphical presentation, a log was applied to results from (3) before plotting)

5 Gestures Imitation in a Robot

Several real-time movements executed by a human subject were captured using the Kinect sensor to provide validation for our algorithms. Two different movements were chosen to illustrate the results: a rhythmic motion repeated many times and a discrete sequence of upper-limb movements. In the first experiment, the human subject is asked to repeat a circular path with the right arm trying to keep, as much as possible, a constant speed across all trials. Fig. 4 and Fig. 5 show the variability always present in human movements in task- and joint-spaces, respectively. Since the details vary, it seems necessary to consolidate the demonstrated movements having in mind the desired final result (*i.e.*, the extent to which the motor goal is reached).

The second experiment consists of a gesture imitation task using the two arms in different configurations around the T-pose. Fig. 6 compares the positions of the right and left wrists as seen by the filtered data and the robot simulation. The consistency between the two curves suggests the efficacy of the human motion reconstruction algorithm proposed.

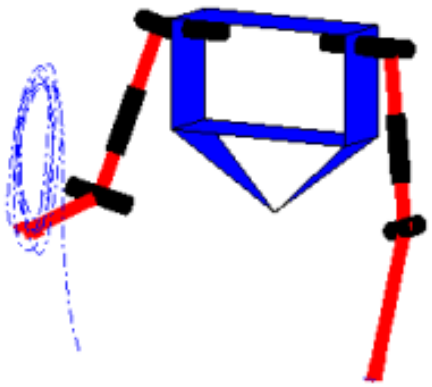


Fig. 4. Variability of the human movements in the task-space during the execution of a circular path repeated many times

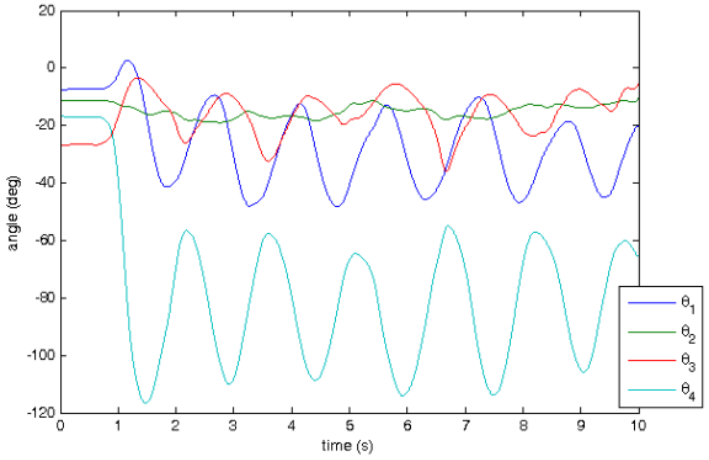


Fig. 5. Time course of the joint trajectories joint-space (right) for the execution of a circular path repeated many times

For this particular movement, the difference between the motion capture data (after pose correction) and the gestures replicated by the robot are quantified using the Euclidean distance. The time courses of the error measure for the elbow and wrist of the left and right arm are shown in Fig. 7.

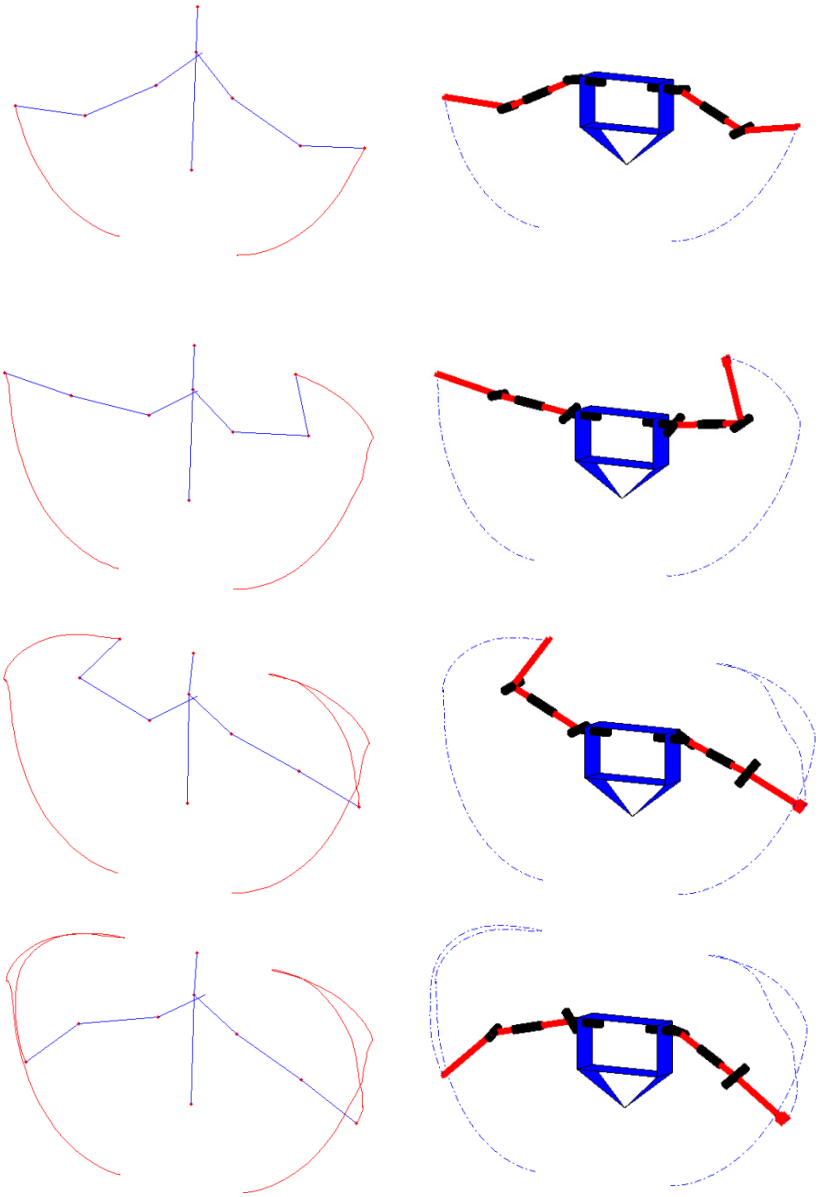


Fig. 6. Comparison of the motion capture data (left) with the corresponding gestures replicated by the robot (the end-effector path is represented in both cases)

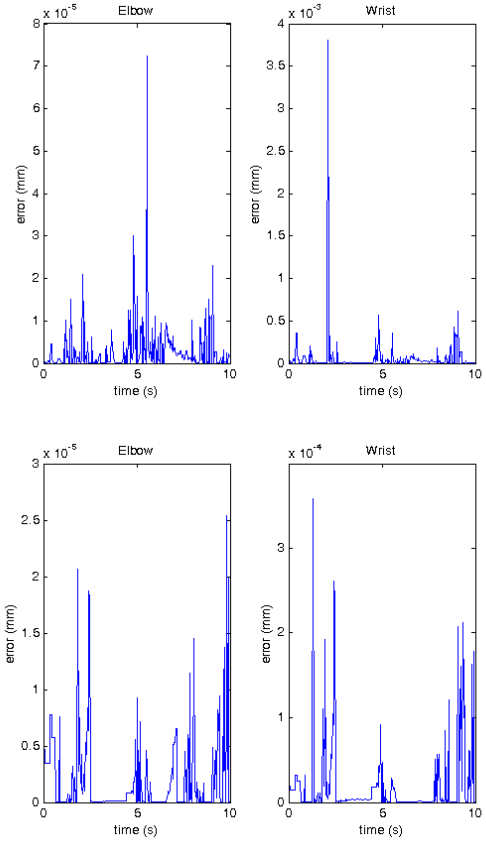


Fig. 7. Difference between the motion capture data and the gestures replicated by the robot for the left arm (top) and the right arm (bottom)

6 Conclusions

In this paper, we have described the potential of the Kinect sensor for gestures imitation of an upper-body robot from demonstrations of a human teacher. First, the pro-posed pose correction method proved to be effective for creating real data sets of human actions by data filtering on a per-frame basis. Second, the implementation of the proposed ideas on a 4-DOF robot model and the experiments on various types of motion show that human-demonstrated gestures are well reproduced by the robot. In this context, the approach is useful for providing a natural and intuitive interface for a human subject to teach complex movements to a robot.

The main goal is to generate large motion capture datasets that can later be used for learning a compact representation of a given task. More concretely, these motion datasets will assist in developing learning techniques for manipulation/locomotion behaviours based on examples of human demonstrations using neural networks or

similar tools of soft-computing. Ongoing work is devoted to the analysis of motions where the robot's joints cannot approximate the human pose in a reasonable way (e.g., closeness to the physical limits at different speeds). At the same time, the application of a retargeting filter to deal with the different dimensions of the human subject's skeleton and the robot body is also being considered.

References

1. Billard, A., Callinon, S., Dillmann, R., Schaal, S.: Robot Programming by Demonstration. In: Siciliano, B., Khatib, O. (eds.) Handbook of Robotics, Springer, New York (2008)
2. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A Survey of Robot Learning from Demonstration. Robotics and Autonomous Systems 57(5), 469–483 (2009)
3. Dasgupta, A., Nakamura, Y.: Making Feasible Walking Motion of Humanoid Robots from Human Motion Capture Data. In: IEEE International Conference on Robotics and Automation, pp. 1044–1049 (1999)
4. Elgammal, A., Lee, C.-S.: Tracking People on a Torus. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(3), 520–538 (2009)
5. Inamura, T., Toshima, I., Tanie, H., Nakamura, Y.: Embodied Symbol Emergence Based on Mimesis Theory. International Journal of Robotics Research 23(4-5), 363–377 (2004)
6. Kulic, D., Takano, J.W., Nakamura, Y.: Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains. International Journal of Robotics Research 27(7), 761–784 (2008)
7. Shon, A.P., Grochow, K., Hertzmann, A.: Rao. R.P.: Learning Shared Latent Structure for Image Synthesis and Robotic Imitation. In: Weiss, Y., Schlkopf, B., Platt, J.C. (eds.) Advances in Neural Information Processing Systems. MIT Press, Cambridge (2005)
8. Shotton, J., Fitzgibbon, A.W., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: "Real-time Human Pose Recognition in Parts from Single Depth Images. In: IEEE Computer Vision and Pattern Recognition. Colorado Springs, USA (2011)
9. Khoshelham, K., Elberink, S.O.: Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. Sensors 12(2), 1437–1454 (2012)
10. Smisek, J., Jancosek, M., Pajdla, T.: 3D with Kinect. In: International Conference on Computer Vision Workshops, Barcelona, Spain, pp. 1154–1160 (2011)
11. Obdržálek, S., Kurillo, G., Ofli, F., Bajcsy, R., Seto, E., Jimison, H., Pavel, M.: Accuracy and Robustness of Kinect Pose Estimation in the Context of Coaching of Elderly Population. In: International Conference of the IEEE Engineering in Medicine and Biology Society, California, USA, pp. 1188–1193 (2012)
12. Platz, T., Denzler, P., Kaden, B., Mauritz, K.-H.: Motor Learning After Recovery from Hemiparesis. Neuropsychologia 32, 1209–1223 (1994)

Humanoid Robots Play Theater

Javier Orcoyen Chaves and Daniel Borrajo

Universidad Carlos III de Madrid,
Avenida de la Universidad, 28911 Leganés, Madrid, Spain
100072854@alumnos.uc3m.es, dborrajo@ia.uc3m.es

Abstract. In this paper we present a tool designed to provide entertainment by using social robots, specifically NAO humanoid robots. The tool takes as input a declarative representation of a theater piece that includes both text and movements to be said (performed) by robots. It can involve a one-robot monologue or a dialogue between two robots. Given the declarative representation of the input, it allows for anyone to create new pieces and easily test how they are represented by real or by virtual NAO robots.

1 Introduction

Often humanoid robots resemble humans in appearance and movements and, actually, they are in continuous development and research. One of its various utilities is entertainment, which this paper focuses on. More specifically, we focus on robots that can play theater [1]. Currently, in several cities, robots are representing theater plays originally designed in Japan.[1] These representations are being very well received by spectators, and tickets are even sold out in some of the cities. So, more and more projects involve the use of social humanoid robots, such as the NAOs that we will use in this paper [2].

An example of such efforts is the European FEELIX [3], developed in collaboration between eight universities and robotics companies in the European Union. One of its objectives is to improve even more the interaction of NAOs with people by the use of nonverbal signals. So far, this project has succeeded in designing NAOs to mimic the emotions of a one-year child and to be able to create bonds with people who treat it kindly. It can also detect human emotions through a series of clues like body language and facial expressions. Furthermore, this NAO is able to memorize faces and remember iterations with people.

Another project using the NAO platform is called ASK NAO (Autism Solution for Kids) [4], also created by Aldebaran Robotics and that will begin to be used to help autistic children. In the first phase this version has been tested in a school in the United Kingdom. The robot comes with games and applications designed by specialists to promote autism children attention and help them improve their state.

[1] http://www.japansociety.org/event/seinendan-theater-company-osaka-universityrobot-theater-project-sayonara-i-worker

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_44,

Like the NAO robots used in this project, ASK NAO is equipped with sensors, two cameras, four microphones, voice synthesizer, two speakers and LED lights.

In this paper, the objective is to build a system capable of providing entertainment by the interpretation of theater plays using robotic agents, specifically NAO robots. These dialogues may be performed by one or two robots, which can say text and combine it with pre-programmed movements. The system uses an XML text file as input, which includes the phrases and types of movements that must be performed. The robots coordinate (in case of two robots plays) to provide a small representation based in the contents of the XML file.

The paper is structured as follows: Section 2 presents the architecture of the system. Section 3 describes the format of the input in the form of an XML file. Section 4 defines how two robots coordinate to interpret two-robots plays. Section 5 shows some short plays represented by our robots. Finally, Section 6 draws some conclusions and presents some future work.

2 Architecture

In Figure 1 we show the architecture of the system, and its main components. Next, we present a high-level description of each component:

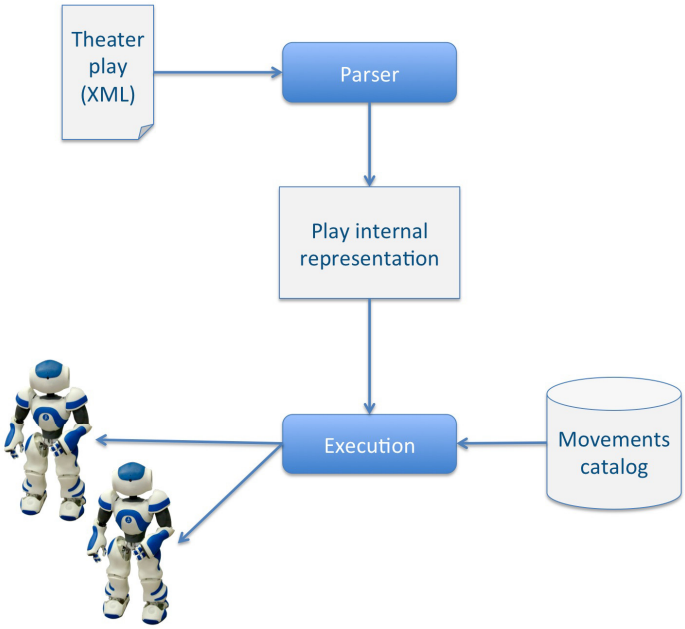


Fig. 1. High-level architecture

– XML file: contains the text instructions and movement types. This file can have actions that could be played by one or two robots. We describe it in more detail in the next section.
– Parser: analyzes the XML file, generating an internal representation of the contents of the XML file. It takes the structure of a tree.
– Execution: analyzes the tags stored in the internal representation and decides, in each case, which robot plays (one or both) and what action it/they must perform (only speak, only move or speak and move). This module is responsible for deciding, from the movement's database, the specific movements that must be performed by the robots.
– Movements database: includes the list of available movements. Each of these movements corresponds to a function that will be called in the Execution component, depending on the movement to be performed by the robot.
– Robots: we use two NAOs. They combine text and movements to interpret the plays.

3 XML File Structure

It must contain a list of `<action>` labels. In addition, each action can contain the following attributes that can take different values:

– `name`: it may be "robot1", "robot2" or "robot1androbot2". It indicates which robot will perform the actions defined in the label. These actions can be done by only one of the robots or both, respectively. In case `name` = "robot1" or "robot2" the following attributes may be included:

- "text": that the robot indicated in the attribute "name" must say.
- "move": type of move that the robot indicated in the attribute "name" must perform. The values that this attribute can take are shown in Table 1.

In Table 1, the movements corresponding to different ways of walking (`move` = "walk", "walkTalk1" or "walkTalk2") and different ways to turn (`move` = "turn180", "turn90Left" or "turn90Right") are based on the use of the function `walkTo (float x, float y, float theta)` included in the robot's library. The rest of the movements have been programmed from scratch for this work and are available for any other theater play. There are several alternatives to program the movements such as: direct programming, programming by demonstration, using NAO's programming toolts (such as Choregraphe) or learning. We used a mixture of the first three.

Table 1 also shows, in the last row, that the value of the "move" attribute can be "null". This will be necessary in case the desired robot must only speak without moving. For actions that include some movement, as "walk", "walkTalk1" or "walkTalk2", the XML description of the action should also inlcude the distance to walk. So, "distance" indicates the distance that the robot must walk, in meters. In case `name` = "robot1androbot2", the following attributes may be included:

– “move1” and “move2”: contain the type of move each of the robots must perform.
– “text1” y “text2”: contain the text each of the robots must say.
– “distance1” and “distance2”: indicate the distance the robots must walk, in meters. These attributes will be used only in cases where attributes “move1” and “move2” involve walking. Specifically, “distance1” will be used for “move1” and “distance2” for “move2”.

Table 1. Table with the description of the different values for the attribute “move”

Robot action	“move” values
Movements associated to actions	”speak”, “no”, “yes”, ”ask”, “anger”, “hello”, “shout”
Different ways of pointing	“point1” … “point6”
Different ways of speaking and pointing	“speakpoint1” … “speakpoint6”
Deny with arms and head	“noArms”
Bow	“reverence”
Encourages to go somewhere	“come”
Moves an arm and checks the time	“clock”
Moves the hand to the chest	“chest”
Does a gesture of being cold	“cold”
Moves an arm to the eyes or ears	“see”, “listen”
Walks without talking	“walk”
Walks speaking in two different ways	“walkTalk1”, “walkTalk2”
Turns 180° and 90° to left and right	“turn180”,“turn90Left”, “turn90Right”
Turns head to left and right	“turnHeadLeft”, “turnHeadRight”
No move	“null”

4 Robots Coordination

One of the tasks that required more effort was robot coordination when they had to perform some action in parallel, or they should wait for the other robot to say some text. After several failed attempts, we succeeded by using the attribute “name” in the XML file. This attribute indicates which robot must play next. So, the complete play is sent to only one robot who will be responsible of coordinating the conversation. Each time the coordinating robot analyzes an instruction, it checks the attribute “name”. Depending on its value, it changes the values of the “ip” and “port” of which robot must intervene.

5 Results

In the following three links, three examples of pieces from Shakespeare plays can be found. We show the combination of talk and a variety of movements. And, in Figure 2 and 3 we show two movements of the play.

– https://www.youtube.com/watch?v=S5Fa5zS3Q4c
– https://www.youtube.com/watch?v=E2pGcJWReEk
– https://www.youtube.com/watch?v=cJu1VweV3BQ

Fig. 2. A moment where the two robots are interpreting Hamlet

Fig. 3. A moment where the two robots are interpreting Macbeth and walk together and talk

6 Results

We believe one of the strengths of this work is the declarative representation of the input. In [1], the authors describe the first robots theater play, where combine some robotics and artificial intelligence techniques to provide a complete program. The let the roots read some text dynamically generated by the public, performed small pieces and even sketch some drawing of someone from the audience. In their paper, it is not clear how the plays are interpreted, but it seems everything is programmed from start to end. In our case, instead of programming the whole play, we rely on a declarative representation of the play. This approach allows for the dynamic generation and modification of the play by other software. Also, artists can define their own plays in a simple way and the system, including the robots, will perform them. This opens a wide variety of applications of this system, as short comedy plays, product presentations, or companion robots. As future work, we would like to link this system with tools for automatic narrative compositions, in order to automatically generate plays and interpret them.

Acknowledgements. This work has been partially supported by Spanish MICINN project TIN2011-2765-C03-02.

References

1. Lin, C.Y., Tseng, C.K., Teng, W.C., Chen Lee, W., Kuo, C.H., Gu, H.Y., Chung, K.L., Fahn, C.S.: The realization of robot theater: Humanoid robots and theatric performance. In: International Conference on Advanced Robotics (ICAR 2009), pp. 1–6 (2009)
2. Kruijff-Korbayová, I., Athanasopoulos, G., Beck, A., Cosi, P., Cuayáhuitl, H., Dekens, T., Enescu, V., Hiolle, A., Kiefer, B., Sahli, H., Schröder, M., Sommavilla, G., Tesser, F., Verhelst, W.: An event-based conversational system for the NAO robot. In: Delgado, R.L.C., Kobayashi, T. (eds.) Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop, pp. 125–132. Springer, New York (2011)
3. Cañamero, L.D.: Playing the emotion game with feelix. In: Dautenhahn, K., Bond, A., Cañamero, L., Edmonds, B. (eds.) Socially Intelligent Agents. Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 3, pp. 69–76. Springer, US (2002)
4. Shamsuddin, S., Yussof, H., Ismail, L., Hanapiah, F., Mohamed, S., Piah, H., Isma-rrubie Zahari, N.: Initial response of autistic children in human-robot interaction the-rapy with humanoid robot NAO. In: IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA 2012), pp. 188–193 (2012)

High Level Humanoid Postural Control Architecture with Human Inspiration

Santiago Martinez, Alberto Jardón, and Carlos Balaguer

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

Abstract. This paper presents the novel humanoid postural control (PC) architecture for the humanoid robot TEO. It is outlined the high level and human inspired system for improving task performance. The study of the human PC system has inspired all processes involved in the control system. The information coming from sensors is interpreted applying neurophysics concepts and, then, the resulting perceptual parameters are applied for task performance improvement. The new PC system is an anticipative module complementing an existing reactive subsystem. This design tries to replicate the operation of the human case. In this way, the reactions can be more complex and higher perturbations levels can be overcome.

1 Introduction

Lots of human-like mechanical designs have been developed during last fifty years, from first prototype Wabot-1 [1] to cutting-edge humanoid robots ASIMO [2], HUBO [3] and HRP-3 [4]. From a mechanical point of view, the development of these robots has taken the advantage of leading technologies existing in their time but the concepts used were based on traditional mechanical solutions. For instance, joint designs have been mainly created with rotary motors joined to mechanical transmissions to increase velocity and torque at their output. But mechanical limitations and the desire of high human appearance favour the searching for new solutions for the humanoid robot design. In this sense, the field of bionics seeks to design technology by mimicking the salient features of biological structures [5]. Lessons learned from bionics state that success of natural inspired designs relies on effective embodiment: on clever morphology and use of material properties [6]. Taking this in account, it is obvious that it is necessary to develop new human inspired technologies to enable this embodiment. In this way, full body humanoid robot development has been slowing down during last decade and the mechatronics research efforts have been redirected to solve more focused problems: artificial muscles, advanced materials, etc.

It has been demonstrated the feasibility of building full body humanoid robots. However, it has been recently paid attention to the second main issue involved in human being replication: the imitation of human behaviour.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_45,

This paper presents a different point of view when talking about the development of control architectures for humanoid robots. Specifically, the postural control (PC) architecture proposed is inspired by the study of the human PC system. The research carried out shows that there are two main sub-systems involved in human PC: a reactive one, and other predictive or anticipatory. The aim of the work is to replicate these control systems for the humanoid robot TEO. In this way, the work has been oriented to establish a predictive system complementing the reactive existing one.

The anticipative control system has been analysed in the human case and it has been extrapolated to the architecture for controlling the humanoid robot TEO. The operation of this control system is based on the composition of sensorial perceptions; then, the evaluation of the resulting stimulus through the use of the psychophysics theory of the surprise and, finally, the creation of events that can be used for activating determined motor reaction strategies (synergies).

The performance of the anticipative system for PC is being tested through simulations and the application of the results in the humanoid robot TEO from the RoboticsLab research group in the Systems Engineering and Automation Department from the Carlos III University of Madrid.

2 Principles of Human Posture Control

The PC correspond to a complex motor response that involves the integration of a variety of sensorial information, elaboration and execution of movement patterns [7]. The human PC system is continuously being developed from birth and it is critically influenced by sensor system maturation and the development of the Central Nervous System (CNS). During the growth process, humans learn to control posture by means of experience acquired in response to sensorial inputs. So, the human posture control system is basically composed by a sensor input system which collects information, an integration system which process this information, and an end-effector system which performs the movements to keep the right posture (Fig. 1).

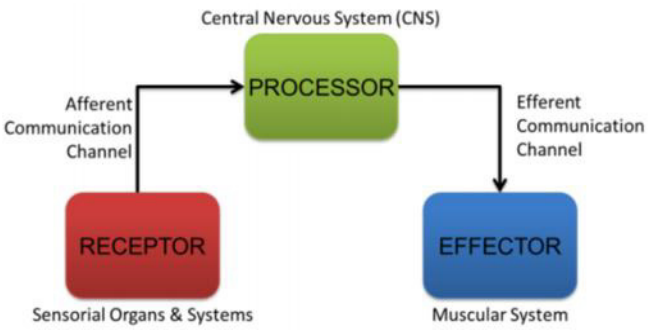


Fig. 1. Basic human PC system components

PC is performed continuously because it is the foundation of any kind of task. That is, one task can be considered as a sequence of controlled and learned postures. During the execution of each posture of the sequence, the sensorial inputs are evaluated to generate appropriate CNS stimuli that are transmitted to the neural processor centres. Depending on the result of the evaluation, different levels of PC could be fired if it would be necessary. These reactions are classified in reflex, automatic or voluntary movements depending on the response velocity required.

Table 1. Properties of the three motor systems in balance movement control

	Motor System		
	Reflex	Automatic	Voluntary
Activation	External stimulus	External stimulus	External stimulus
Role in PC	Muscle force regulation	Resist disturbances	Self-generator Purposeful movements
Latency	Fixed 20-60ms	Fixed, 130-170ms	Variable >150ms

Another factor influencing response velocity is the operation type of the human PC system. There exist two basic modes of operation of the system when a disturbance is detected: reactive and predictive. Specifically, this paper is centered on the study of the predictive PC mechanism, in which the sensorial inputs are used to predict the possible consequences from perturbations. According to the human system operation, the main function of this predictive system is preparing the effector system to apply a reaction. It is essential when higher level disturbances are detected or faster reactions are required than the feedback control loop can manage or trigger.

2.1 Basic Operational Mechanisms of the Human Postural Control System

Different theories were developed in the past to explain how the human body controls its posture. But nowadays, PC has been oriented to a systemic point of view. Today researchers recognize that PC is complex and context-dependent and that all levels of the nervous system must be examined to account for this complexity [8]. Although some controversy exists regarding the range of subsystems involved, there is general agreement that the neurological system, the musculoskeletal system, the sensory system, the environmental context, and the task demands are important contributors to PC [9].

Besides relying on their feedback systems, humans also maintain balance using anticipatory motor actions. During human movement, two control actions are performed continuously and in parallel: movement and PC. Meanwhile movement control system commands body limbs position, the PC performs actions to maintain balance taking in account the proprioceptive information. Fig. 2 shows this basic idea, but this control system is defective in the sense that it only provides information about the feedforward anticipative and feedback reactive postural adjustments produced by voluntary movements.

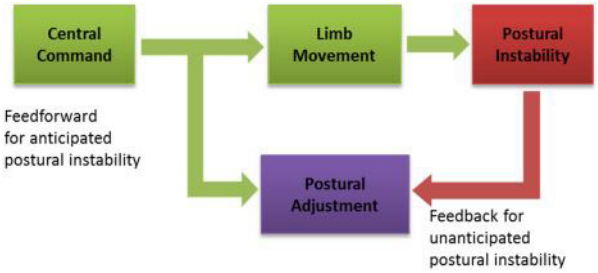


Fig. 2. Postural adjustment scheme from [10]

It is important to state that posture and movement are close related but they are essentially different. From a biomechanical point of view, the movement can be described as the combination of motor gestures. Purposeful or voluntary motor acts are performed moving one or several body segments towards a goal [11]; meanwhile other segments must be positioned in order to regain posture and equilibrium. It is ease to point out that voluntary movements are one source of postural perturbations.

Therefore, posture poses a static and dynamic dual nature. The former, static or postural fixation, is a local mechanism to maintain the body segments in stationary positions against internal (e.g. weight) or external forces (e.g. load ported) [12]. The latter, dynamic posture component is the continuous looking for keeping the desired target according to the task performed.

2.2 Sensorimotor Integration in Postural Control

To maintain PC during both static and dynamic situations, individuals rely on their sensory systems (visual, vestibular and somatosensory) to provide information such as their limbs locations and movement with respect to the surrounding environment. The CNS then interprets the sensory information and commands the musculoskeletal systems to adjust the body parts position trying to keep a desired or stable posture. However, since human beings are constantly interacting with their surroundings, one must not ignore the environment when studying PC. The influence of environmental factors such as light conditions, concurrent distracting factors, special surface characteristics, etc. are affecting the requirements to the PC. Similarly, it is easily understood that the PC demands during the task of walking and other locomotive activities are different from the demands when humans perform manipulation tasks (Fig. 3).

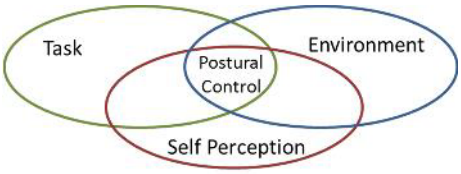


Fig. 3. PC is influenced by factors related to the individual, the task, and the environment adapted from [13]

Since PC has been defined as the control of the body's position in space, it must be performed adjusting parameters such as centre of mass, base of support, joint momentum, etc. The adjustments are counterbalancing actions of the limbs, the head and the trunk influenced by the muscular strength, previous experience, etc.

2.3 Interpretation of Sensorial Incomes for Postural Control

One key point of the human PC system is the way in which the disturbances are sensed and how this perception commands body reactions. Using psychophysical principles, it is possible to model these disturbances as unexpected events or surprise events produced depending on the task context [14].

Since surprise depends on the 'unexpectedness' of the stimulus [15], at least the three following layers and each corresponding test must be distinguished in order to provide an exhaustive model of surprisingness generated by expectation failure [16].

- Mismatch-based Surprise: based on sensory-motor expectations, it is generated by the mismatch between active knowledge and disturbance perception, exceeding some threshold value [17], [18] (function of Unexpectedness). It is based on some form of 'statistical' learning [19] and its intensity is function of the degree of certainty and the value of the goal.
- Passive Prediction-based Surprise: surprise results from a conflict or inconsistency between the updated set of knowledge and the perceived sensation [15]. Passive expectations are formed after the surprising event has occurred [20].
- Implausibility-based Surprise: This refers to those (quite numerous) situations in which the input proposition expresses information related to non-existent knowledge (function of Incredulity).

Summarizing, capture surprise events or feeling surprise seems to play several functions:

- Redirecting attention on the mismatching facts, concentrating cognitive processing resources on them.
- Activating resources for possible practical activity; physical arousal, bodily preparation for fast reaction.
- There are also long term effects (and functions) of the perceived surprise for a bad prediction (e.g. increasing controls before and during the actions).

2.4 Motor Strategies Used in Postural Control

The motor reactions during PC have been extensively studied in order to try to understand the mechanisms that humans use to overcome any kind of 'postural perturbation'. One of the main conclusions is the existence of learned patterns that are automatically triggered in response to determined stimuli. These patterns are called motor strategies or 'synergies' (e.g. [21], [22]). According to Sherrington, the control of the movement related to the synergies is composed by a reflex motor unit above the

voluntary motor unit. Such reflex movements are organised more naturally into collective functional units defined over groups of muscles and joints [23]. By the other side, Bernstein suggests that a restricted number of programs may underlie most of our behaviour.

For any given perturbation, one or more muscle synergies may be activated so that their combined influences define the resulting muscle activation pattern [24]. Extending the concept, complex synergies can be in general considered as programs for controlling some distinctive motor performance extended in space and time, built upon basic synergies of coordinated reflexes as substrate [25]. In summary, the establishment of a synergy is based on a common assumption that regularity in the behaviour of a set of elements is a sufficient sign to claim an existence of a synergy [26].

3 The Humanoid Anticipative Postural Control Architecture

The human PC system, outlined in previous sections, is the result of millions of years of evolution. Its complex operation and physiology are still being researched and they are far from being completely understood. Although technology evolution is much faster than biology evolution, the same problems must be addressed and they are continuously under development. The main studies regarding physiology and human behaviour date back the turn of 20th Century. The advances achieved in the knowledge of the human organism during the last decades have made possible a better understanding of the underlying mechanisms that produce the different human behaviours. There is a variety of human behaviours and their classification is complex. Attending to their nature, behaviours can be classified as innate or learned.

Innate or instinctive behaviours will be those, conscious or unconscious, that have a biological and genetic basis, are performed naturally, and are reinforced by practice. The human being has acquired this kind of innate behaviours thanks to thousands years of evolution, and they are “hard-wired” in the CNS. In general, instinctive behaviours are considered as “pre-programmed" responses triggered by external stimuli. They usually fit into one of the following categories [27]:

- Reflex: it is the most basic innate behaviour. Correspond to the basic reflex arc involving only a few neurons.
- Orientation behaviours: they are coordinated movements like walking, etc.
- Kinesis: it is a change on the speed of movement or a change rate of turn which are directly proportional to the stimuli intensity.
- Taxis: it is a movement directly toward (positive) or away from (negative) a stimulus.

Learned behaviours are skills acquired or modified by the experience resulting from a learning procedure. Taking this into account, it is obvious to conclude that innate and learned behaviours are close related by means of experience. The human being acquires new skills and knowledge through trial and error, observation of other

individuals or memory of past events. In general, learned behaviours will always be [27]:

- Non-heritable: behaviour acquired only through observation or experience.
- Extrinsic: it is caused by social interaction.
- Permutable: pattern or sequence may change over time.
- Adaptable: it is capable of modification to suit changing conditions.
- Progressive: subject to improvement or refinement through practice.

The better understanding of the human PC system operation has enabled the development of a large number of control schemes in cybernetic/robotic and biomechanical fields. In the former, theoretical and experimental situations as standing posture and free fall [28], walking [29], run-to-walk and vice versa [30], [31], have been studied considering static or dynamic 2D/3D problems. In the latter field, studies are focused on experimental postural analysis [32], organization of the PC [33] or biomechanical modelling to study PC [34], gait initiation [35], musculoskeletal control [36] or jumping [37].

These works are contributions for a better human motor behaviour understanding. These studies deal more generally with the selection of strategies to balance the external perturbation (force and moment) acting on the human structure. In these cases, the matter under control is the desirable posture during and after the performance of a voluntary movement. From the initial posture, the movement is the succession of instantaneous postures subjected to external perturbations. Then, the reactions against these perturbations are computed and deployed according to the response velocity required and the origin of the disturbance.

The development of humanlike machines has motivated a deeper research in human PC systems. Early developments of humanoid prototypes were built to research the first postural problem humans must face up in the first year of their life: the equilibrium maintenance. The increase in computer processing power has enabled the fast development of these prototypes and the construction of full size humanoid platforms, which are able of performing complex postural tasks.

The step up in mechatronics and computing has favoured the development of high complexity control schemes and their transformation into 'human inspired' control systems. The final goal of these control schemes is to imitate the human behaviour as much as possible.

This human inspiration has caused a change in how the researcher considers the humanoid platform. The humanoid robot was only a mechatronic platform to test tasks and control schemes. Now, new robotic platforms have been developed to study the cognitive aspect of the human nature. In these platforms, the understanding of cognition and the analysis of how humans perceive the environment, how they interact within it and how the information is processed and applied, are the key point of control. This is one of the reasons why techniques, derived from the study of human behaviour, are taking more importance in PC. Genetic algorithms, neuro-fuzzy controllers, etc. used in Artificial Intelligence are being applied in control, due to their similarity with real human processing.

During the last decade, the RoboticsLab research group has been introduced in the development of humanoid robots. The prototype RH-1 [38], [39], [40] was the first

anthropomorphic mechatronics design carried out. It was useful to understand the challenging of the humanoid robot mechatronics and control design. With the new humanoid robot prototype TEO (Task Environment Operator), Roboticslab has applied lessons learned with RH-1. New improvements in mechatronics enable the change on the control philosophy from classical control techniques towards human behaviour inspired control.

This paper deals with this change and the development of human inspired control architecture for TEO humanoid robot. The first version of the control scheme deployed for TEO humanoid robot was a classic feedback control system to regain stability. It matches with the reactive loop in the human control scheme exposed in Fig. 2. The high level architecture proposed in this work adds the open anticipative loop to complete the human inspired PC scheme.

3.1 Performing Humanoid Tasks

It has been exposed that human actions are a mixture of different kind of behaviours composed by simpler tasks. By means of the learning process, tasks become more complex and they enhance the human capabilities during growth. Thus, a task should not be considered simply as a succession of fixed motor patterns. It might be considered as a flexible, adaptable and configurable motor sequences that, as well, includes mechanisms for dealing with unexpected events.

In this way, TEO humanoid robot tasks are not mere sequences of joint angles, velocities and accelerations. There are complete sets of configurable modules established to enable the human inspired postural architecture. Fig. 4 represents the structure of TEO tasks. One task frame is composed by a main movement, which can be configured to perform different movements with the same shape. As well, the mechanisms to react against perturbations have been added by means of the integration of motor patterns or synergies. Finally, it has been included a module which combines motion sequences, depending on the control requirements.

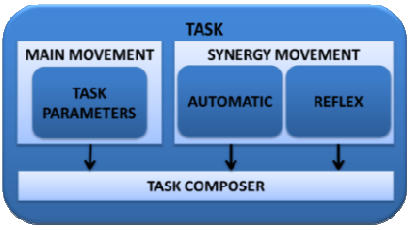


Fig. 4. Humanoid task modules

Then, complex behaviours are those groups of tasks sharing one main characteristic that defines them. Thus, the study of PC has been divided into manipulation and locomotion behaviours or groups of tasks.

3.2 Human Inspired Postural Control Architecture

The main objective of the present paper is to propose a novel human inspired and task oriented control architecture for humanoid robots. This development has been deployed for robot TEO which is the third generation of humanoid robots from RoboticsLab research group. The research in the field of humanoid robots started with the development of the platform RH-1. This mechatronic system integrated a PC to maintain equilibrium during locomotion tasks. The high level control scheme is shown in Fig. 5.

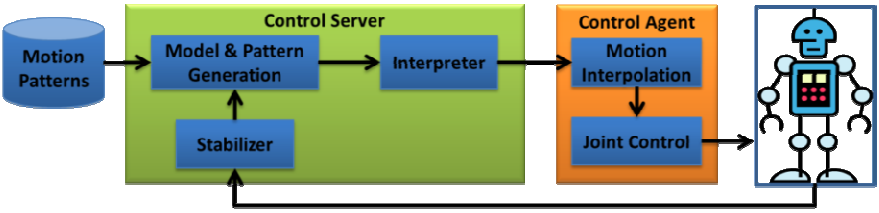


Fig. 5. RH-1 Control architecture

In this scheme, a central pattern generator computes a statically stable locomotion task. It is composed by a sequence of joint positions or postures whose stability was ensured offline. These positions are transmitted to the joint controllers and executed. An internal joint control loop minimizes the posture position error. But this loop is not enough to regain balance if higher disturbances act over the robot. In this case, a second control loop for dynamic stability or 'stabilizer' maintains postural balance during locomotion by means of ZMP and CoM allocation control. The first loop corresponds to the internal PID control of each individual joint. It is performed by the driver device that actuates every degree of freedom. Due to this nature, it is impossible to perform whole body postural corrections only with this loop. On the opposite, the 'stabilizer' considers the robot as a whole and it supervises the parameters influenced by the body dynamics. These parameters can be computed online thanks to the use of a simple inverted pendulum model that reduces significantly the processing time.

Another important aspect of this architecture is the availability of sensor data and its treatment. The only sensor devices integrated in the kinematic chain of the RH-1 robot are incremental encoders used to measure joint angles. This kind of sensors does not provide information about the dynamics of the robot. The use of the simplified model of the robot body helps to estimate the dynamics and its influence in balance. The unfeasibility of obtaining direct sensor data regarding the body dynamics, such as limb accelerations, forces exerted on the body, etc., causes an increase of computing time. Due to this, the admissible level of perturbation is lower and the time to reaction higher.

Apart from the restrictions in control, RH-1 robot presented great limitations which did not allow a correct motion performance in terms of mechanical robustness (high joint looseness), stability, and energy consumption (necessity to be connected to the

electrical net since battery could not supply the required energy for more than a while), not to mention the realization of high-level tasks such as manipulation, complex gait generation, or complex human-robot interaction [41].

TEO robot comes to substitute RH-1 humanoid robot to overcome its limitations. The design of the new platform turns the humanoid robot into a cognitive robot that enables the implementation of human inspired concepts. The novel PC architecture proposed for TEO humanoid robot is compounded by two differentiated parts: the feedback and the feedforward control loops. It is shown in Fig. 6.

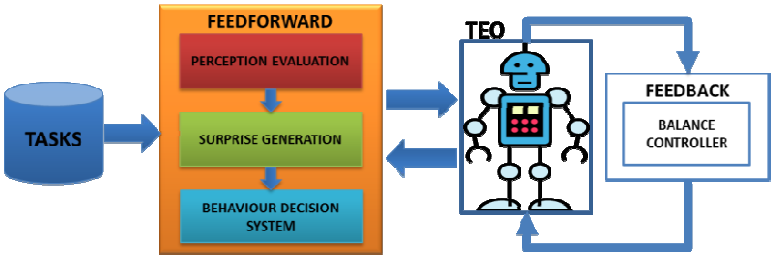


Fig. 6. TEO High level control architecture

The feedforward control loop shown in Fig. 6 has been inspired in the architectures for human PC system stated in previous sections. As commented before, human body has different levels of reaction depending of the required response velocity. Automatic and reflex acts are the fastest levels of reaction against disturbances. These reactions are activated by the inputs coming from the corresponding sensorial organs. The information from these organs is transmitted to the processing centres in which the reaction is built. Once the response is composed it is sent to the muscles to perform the programmed action.

The novel TEO postural controller has been inspired in this operation and the robot mechatronic systems have been designed to support these functions.

3.3 Perceptual Evaluation

Apart from the levels of motor reaction, the basis of the PC is the sensorimotor integration and the way of enabling it. As reviewed before, several theories have been developed to explain how the sensorial information is captured, processed and, executed through the corresponding action. Multiple systems are involved in PC in a multilevel structure. Taking this into account, it has been established a sensorial evaluation module to compute the input information and to compose the named 'perceptions'.

The exoceptive perception will capture the external perturbations (Table 2). This perception is composed by different sensor devices that accomplish the same function as one system of the human body. Then, the sensorial information detected is integrated task-dependant Fuzzy systems.

Table 2. Human vs. humanoid exoceptive perception

Exoceptive Humanoid System	Exoceptive Human System
Inertial Measurement Unit (IMU)	Vestibular System
Force / Torque Sensors (F/T)	Muscles / Skin
Vision (3D cameras)	Vision

The proprioceptive perception measures the internal body status. The sensorial data is provided by joint position and velocity sensors (Table 3).

Table 3. Human vs. humanoid proprioceptive perception

Proprioceptive Humanoid System	Proprioceptive Human System
Relative Encoder	Joint Velocity
Absolute Encoder	Joint Position

The result of the sensor data evaluation depends highly on the task being performed. It means that the resulting perception will not be the same if the task performed is, for instance, pure manipulation or pure locomotion. It is the task oriented perceptual system which filters the information and uses it in the proper way. Taking this into account, two premises can be established for TEO robot perceptual evaluation:

- Same sensorial inputs will produce different perception depending on the task performed.
- Exoceptive and proprioceptive perceptions will be composed by different sensorial sources depending on the task performed.

The first premise means, following psychophysical principles, that the processor centres filters the sensorial information to speed the result of the evaluation up and to produce an accurate response (detection, identification, discrimination and scaling). Second premise remarks the task dependant nature of the perception production as well. For instance, the use of data related with equilibrium is unnecessary in a manipulation task when the robot is seated.

Summarizing, not all information might be used in all cases and the information might not be applied in the same way in every task. Fig. 7 shows the modules in charge of perceptive evaluation in TEO control architecture. Sensations composed by the information captured by sensory devices are evaluated forming the proprioceptive and exoceptive perceptions. Both sets of perceptual information is then available in the system to be used depending of the PC necessities.

3.4 Surprise Generation

It has been described how sensory information can be processed and converted into perceptions. They are parameters that relate posture and sensorial incomes. In this stage the main problem is how to apply all this information in PC.

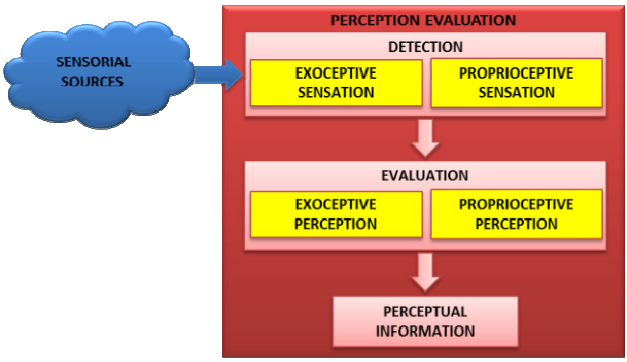


Fig. 7. Perception evaluation

The mechanism established in the feedforward loop to trigger reactions is based on the evaluation of surprising events. Perturbations can be considered as unexpected events and, the output of its processing, the activation of a surprise [15]. Then, the information from perceptions is the base for this evaluation. The system selects and combines information from the available sets, depending on the necessities of pre-established output surprises.

Nevertheless, the concept of surprise is a high level envelope that explains the process of any kind of unexpectedness. Taking this into account, surprisingness is generated by lower level mechanisms called expectations. The failure of a determined expectation elicits a surprise event. The expectation failure can be classified in:

- Active expectation failure or prediction failure. It is produced when a prediction about an outcome has not been produced by an input proposition.
- Passive expectation failure or assumption failure. It is caused when an outcome, originated by a determined input proposition, is not predicted but some assumptions about it can be established. If this output doesn't fit into these assumptions, the surprise should be elicited.
- Unanticipated incongruity. It is related to unexpected events never experimented before.

Then, reaction movements can be considered as reactions driven by surprise events. Passive and active expectations require a certain degree of previous knowledge about the matter that can cause surprise. These two forms of expectation failures have been considered to be applied for surprise generation in TEO robot. Unanticipated incongruity implies the integration of other kind of intelligent module that will be able to classify the event and to learn about it.

In this way, passive expectation failure sub-system will produce surprise events when some predefined thresholds will be exceeded by the input proposition. In the case of active expectation, predictions about critical issues related with tasks performance are continuously verified. Then, the surprise event is triggered if these predictions fail, always taking into account the task context.

3.5 Behaviour Decision System

At this stage, there has been described how perturbations are perceived by the sensorial system. After that, it has been introduced the system which transforms sensations into 'understandable' information (perceptions). At last, perceptual information is interpreted and converted into surprise events that will be used by the control system to act against perturbations.

However, the anticipated actions deployed by the decision system can be defined as reactions against future consequences evoked by sensorial stimuli. That is, given a determinate task, it is possible to know so the correct behaviour as the deviation caused by determined perturbations. It means that the perceptual knowledge will drive the decision because the consequence of the perturbation is previously known. Then, it is possible to say that the anticipative PC system reacts against the future task state.

Therefore, inside the behaviour decision module, the surprise events are processed to decide:

- If any kind of future action or reaction is needed.
- The kind of reaction that would be the most appropriate.
- How the selected reaction should be performed.

By means of evaluating the information from active and passive expectations (surprise events), it can be determined if a reaction might be selected and executed. The reaction is selected among all available motor synergies related with the corresponding task. These synergies are motion patterns that will be filled in using the results from the expectations evaluation (surprise task parameters). The outcome of this module will be a parameterized synergy that could be executed to enhance PC.

4 Conclusions

This paper presents the novel high level PC architecture for the humanoid robot TEO. Specifically, it has been exposed the guidelines for the development of the new human inspired anticipative system. It has been outlined the human PC system as design basis of a novel humanoid anticipative control system. From perturbations and sensation detection to control signal generation, the new control system established taking into account the way human's body performs these processes. This high level architecture is the framework to develop all necessary human reasoning inspired sub-systems to process task information.

Acknowledgements. The research leading to these results has received funding from the ARCADIA project DPI2010-21047-C02-01, funded by CICYT project grant on behalf of Spanish Ministry of Economy and Competitiveness and from the RoboCity2030-II-CM project (S2009/DPI-1559), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

1. Kato, I.: The WABOT-1. Bulletin of Science and Engineering Research Laboratory Waseda University special issue on WABOT, 62 (1973), http://www.humanoid.waseda.ac.jp/booklet/kato_2.html (accessed February 2, 2011)
2. Hirose, M., Haikawa, Y., Takenaka, T., Hirai, K.: Development of humanoid robot ASIMO. In: International Conference on Intelligent Robots and Systems, IROS, vol. 13, pp. 1–6. IEEE, Maui (2001)
3. Oh, J.-H., Hanson, D., Kim, W.-S., Han, I.Y., Kim, J.-Y., Park, I.-W.: Design of Android type Humanoid Robot Albert HUBO. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1428–1433. IEEE (2006), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4058572
4. Kaneko, K., Harada, K., Kanehiro, F., Miyamori, G., Akachi, K.: Humanoid robot HRP-3. In: IEEERSJ International Conference on Intelligent Robots and Systems, IROS 2008, pp. 2471–2478. IEEE, Nice (2008)
5. Vincent, J.V., Bogatyreva, O.A., Bogatyrev, N.R., Bowyer, A., Pahl, A.-K.: Biomimetics: its practice and theory. Journal of the Royal Society Interface the Royal Society 3(9), 471–482 (2006), http://rsif.royalsocietypublishing.org/content/3/9/471
6. Eaton, M.: Evolving humanoids: Using artificial evolution as an aid in the design of humanoid robots. In: Iba, H. (ed.) Frontiers in Evolutionary Robotics, pp. 127–138. InTech (2008)
7. Horak, F.B., Macpherson, J.M.: Postural orientation and equilibrium, pp. 255–292. Wiley-Blackwell (2011)
8. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science. In: Kandel, E.R., Schwartz, J.H., Jessell, T.M. (eds.) Fifth, p. 1414. McGraw-Hill (2000), http://www.amazon.com/Principles-Neural-Science-Eric-Kandel/dp/0838577016
9. Kamm, K., Thelen, E., Jensen, J.: A dynamical systems approach to motor development. Physical Therapy 70(12), 763–775 (1990)
10. Kejonen, P.: Body Movements During Postural Stabilization. Medicine 500, 693 (2002), http://herkules.oulu.fi/isbn9514267931/html/
11. Agid, Y.: From posture to initiation of movement. Revue Neurologique 146(10), 536–542 (1990), http://www.ncbi.nlm.nih.gov/pubmed/2263815
12. Martin, J.P.: The basal ganglia and posture, 1st edn., p. 152. Lippincot, Philadelphia (1967)
13. Shumway-Cook, A., Woollacott, M.: Motor control: Theory and practical applications, 2nd edn., p. 614. Lippincott Wiliams & Wilkins (2000)
14. Dey, A.: Understanding and using context. Personal and Ubiquitous Computing 5(1), 4–7 (2001)
15. Ortony, A., Partridge, D.: Surprisingness and expectation failure: what's the difference? In: McDermott, J.P. (ed.) Proceedings of the 10th International Joint Conference on Artificial Intelligence, vol. 1, pp. 106–108. Morgan Kaufmann Publishers, Milan (1987)
16. Lorini, E., Falcone, R.: Modeling expectations in cognitive agents. In: Castelfranchi, C., Balkenius, C., Butz, M., Ortony, A. (eds.) AAAI 2005 Fall Symposium From Reactive to Anticipatory Cognitive Embodied Systems, pp. 114–121. AAAI Press, Washington (2005)

17. Meyer, W.U., Reisenzein, R., Schützwohl, A.: Toward a process analysis of emotions: The case of surprise. Motivation and Emotion 21(3), 251–274 (1997), http://www.springerlink.com/content/h56w0456812680q5
18. Macedo, L., Reisenzein, R., Cardoso, A.: Modeling forms of surprise in artificial agents: empirical and theoretical study of surprise functions. In: Forbus, K., Gentner, D., Regier, T. (eds.) Proceedings of the 26th Annual Conference of the Cognitive Science Society, Chicago, pp. 873–878 (2004), http://csjarchive.cogsci.rpi.edu/Proceedings/2004/CogSci04.pdf
19. Baldi, P., Itti, L.: Of bits and wows: A Bayesian theory of surprise with applications to attention. Neural Networks 23(5), 649–666 (2010), http://www.ncbi.nlm.nih.gov/pubmed/20080025
20. Spitz, D.: A computational model of surprise 47 (2011)
21. Bernstein, N.: The co-ordination and regulation of movements. Neuropsychologia 6(1), 96 (1968)
22. Sherrington, C.: The integrative action of the nervous system. Nature 76, 122–122 (1962), http://www.nature.com/doifinder/10.1038/076122a0
23. Kelso, J.A.S., Saltzman, E.L.: Motor control: which themes do we orchestrate? Behavioral and Brain Sciences 5, 554–557 (1982)
24. Ting, L.H.: Dimensional reduction in sensorimotor systems: a framework for understanding muscle coordination of posture. In: Cisek, P., Drew, T., Kalaska, J.F. (eds.) Progress in Brain Research, vol. 165, pp. 299–321 (2007)
25. Arbib, M.A.: Brains, Machines and Mathematics, 2nd edn., p. 202. Springer, New York (1987)
26. Latash, M.L., Krishnamoorthy, V., Scholz, J.P., Zatsiorsky, V.: Postural Synergies and Their Development. Neural Plasticity 12(2-3), 119–130 (2005), http://www.ncbi.nlm.nih.gov/pubmed/16097480
27. Meyer, J.R.: Elements of Behavior (2006), http://www.cals.ncsu.edu/course/ent425/tutorial/Behavior/index.html
28. Gorce, P., Vanel, O., Ribreau, C.: Equilibrium study of "human" robot. In: 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, pp. 1309–1314. IEEE, Vancouver (1995), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=537953
29. Vukobratovic, M., Frank, A.A., Juricic, D.: On the stability of biped locomotion. IEEE Transactions on Biomedical Engineering 17(1), 25–36 (1970), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4502681
30. Hodgins, J.K.: Biped Gait Transitions. In: Proceedings 1991 IEEE International Conference on Robotics and Automation, pp. 2092–2097. IEEE (1991), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=131936
31. Hodgins, J.K.: Simulation of Human Running. In: IEEE International Conference on Robotics and Automation, vol. 2, pp. 1320–1325. IEEE, San Diego (1994), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=351304
32. Bouisset, S., Zattara, M.: A sequence of postural movements precedes voluntary movement. Neuroscience Letters 22(3), 263–270 (1981)

33. Nashner, L.M., McCollum, G.: The organization of human postural movements: A formal basis and experimental synthesis. Behavioral and Brain Sciences 8(01), 135–172 (1985), http://www.journals.cambridge.org/abstract_S0140525X00020008
34. McCollum, G., Leen, T.: Form and exploration of mechanical stability limits in erect stance. Journal of Motor Behavior 21(3), 225–244 (1989), http://cat.inist.fr/?aModele=afficheN&cpsidt=6611470
35. Brenière, Y., Dietrich, G.: Heel-off perturbation during gait initiation: biomechanical analysis using triaxial accelerometry and a force plate. Journal of Biomechanics 25(2), 121–127 (1992), http://www.ncbi.nlm.nih.gov/pubmed/1733988
36. Van Der Helm, F.C.T., Rozendaal, L.: Musculoskeletal systems with intrinsic and proprioceptive feedback. In: Winters, J.M., Crago, P. (eds.) Biomechanics and Neural Control of Posture and Movement, 1st edn., pp. 164–174. Springer, New York (2000), http://e.guigon.free.fr/rsc/incoll/vanderHelmRozendaal00.pdf
37. Levine, W.S., Zajac, F.E., Belzer, M.R., Zomlefer, M.: Ankle controls that produce a maximal vertical jump when other joints are locked. IEEE Transactions on Automatic Control 28(11), 1008–1016 (1983)
38. Arbulu, M., Kaynov, D., Cabas, L.M., Balaguer, C.: The Rh-1 full-size humanoid robot: design, walking pattern generation and control. Journal of Applied Bionics and Biomechanics 6(3), 301–344 (2009)
39. Pierro, P., Monje, C.A., Balaguer, C.: The virtual COM joints approach for whole-body RH-1 motion. In: Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication, pp. 285–290. IEEE, Toyama (2009), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5326259
40. Pierro, P., Monje, C.A., Balaguer, C.: Modelling and control of the humanoid robot RH-1 for collaborative tasks. In: Proceedings of the 8th IEEE/RAS International Conference on Humanoid Robots, pp. 125–131. IEEE, Daejon (2008), http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4755942
41. Monje, C.A., Martinez, S., Jardon, A., Pierro, P., Balaguer, C., Muñoz, D.: Full-Size humanoid robot TEO: Design attending mechanical robustness and energy consumption. In: Proceedings of the 2011 IEEE/RAS International Conference on Humanoid Robots, pp. 325–330. IEEE, Bled (2011)

Part VIII
Legal and Ethical Aspects in Robotics

Robotic Society: Privacy as a Legal and Technological Issue

Maria Rosa Llácer-Matacás

Grup de Recerca en Dret Privat, Consum i Noves Tecnologies, University of Barcelona, Faculty of Law, Avenida Diagonal 684, 08034 Barcelona, Spain
mrllacer@ub.edu

Abstract. In this paper we analyze the issues surrounding the European Proposal for General Data Protection Regulation and the adaptation of law to the technical capabilities that have an impact on privacy. We start by analyzing the new capabilities of robotic machines and the involvement of new stakeholders as engineers, designers and manufacturers. Then we present the challenges of M2M relationships and the risks of being detected through devices. Next, we discuss the need to incorporate some concepts such as 'pseudonym", anonymous processing and legal profiling. We also argue that law needs technical standards and we foresee that engineers, designers and manufacturers will not have a neutral role. Future regulation should ensure some guidelines for privacy safe robots, and guarantee privacy and freedom in a technological world.

Keywords: personal data, identification, profiling, M2M relationships, privacy impact assessment, privacy by design, privacy by default.

1 New Objects, New Features, New Stakeholders

In the information society, robots can really improve people's lives. Devices interacting between machines and people are more and more common. Some of them can work without identifying people: in principle, they are not risky from the point of view of privacy. But many other utilities provided by robotic machines rely on a device closely related to a person associated to online identifiers. Devices can be used to track or to single out persons. When behavioural profiles are related with an individual, or even when somebody fits into a profile, there is a privacy question: the service provider (data controller) has to comply with legal requirements. It is

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_46,

worth noting that the legal framework[1] protects fundamental rights: privacy and data protection[2].

Let's think for a moment about some utilities provided by robotic machines:

- Devices providing information about surrounding services and locations, based on geolocation.
- Systems used to monitor certain services in smart cities (for instance, *Bicing*, a bicycle rental service allowing the control of when and where bikes are used); or to monitor behaviour, if relevant to set the terms of a relationship (for instance, *YCar*, a car insurance policy where price depends on the driving style, detected by computerized systems).
- Equipment with sensors used for employee identification or building access (identification by smart cards, biometric data or a combination of both).
- Robots designed to make life easier for people with disabilities (they memorize customs and health needs or are connected with an external helpdesk) or, simply, to be able to check sports performance.

Clearly, robotization brings huge benefits. It is necessary to comply with law not only to avoid liability and sanctions, but also to ensure secure exploitation. Trust is a benefit both for industry and users.

There are different kinds of players involved in these new situations. Laws on data protection focus only on data subjects and controllers. The "data subject" means an identified natural person, who can be identified, directly or indirectly, by means reasonably likely to be used by the controller or by any other natural or legal person. The "controller" means the natural or legal person, public authority, agency or any other body which determine the purposes, conditions and means of the processing of personal data. But actually there are more stakeholders involved. It is true that designers and manufacturers are not responsible for the use given to the goods they produce: the data controller (provider of geolocation services, health services, etc.) remains responsible for the legal processing of personal data.

---

[1] Spanish regulation: Ley Orgánica 15/1999, de 13 diciembre, de Protección de datos de carácter personal; and Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999.

European regulation: Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data; Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 on privacy and electronic communications (Directive on privacy and electronic communications), amended by Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009.

[2] We can state that privacy is a concept ('decisional autonomy') wider than data protection (a means to ensure privacy): see Rouvroy, A. and Poullet, Y.: The right to informational self-determination and the value of self-development. Reassessing the importance of privacy for democracy. In: Reinventing Dara Protection?, Gutwirth, S., Poullet, Y., De Hert, P., De Terwange, C., Nouwt, S. (edit.), Spinger, Heidelberg (2009).

Recently, the proposal for General Data Protection Regulation[3] has taken an important step forward and introduces the concept of "data protection by design and by default" (art. 23 PRGDP). It still remains a duty of the controller to implement appropriate technical and organizational measures and procedures in such a way that the processing will meet the requirements of the law. It is clear that this goal needs the collaboration of designers and manufacturers. In this article we consider technology as an emerging key element to protect privacy. 'Privacy friendly technology' is not a new idea but, finally, a reference to privacy by design can be explicitly found in a law[4]. This means that law should seek the cooperation of the industry to meet the technical requirements and to reinvent legal normativity[5].

2 M2M Relationships: Robots Detecting People as an Object

The examples above show how the system is able to allocate consequences to people when detecting them. Sometimes terminals are absolutely related to a person (smart phones, devices controlling medication) but sometimes they are only related to an identifier without a clear relationship with somebody; however, in the latter case, a new event can relate the identifier with an individual: for example when paying for an item equipped with a tag with a credit card[6]. What's more, tags are able to connect the device holder with external data stored in the cloud. We can conclude that robots detect individuals as an object and that their information is retrieved and crossed, and always remains available.

The main feature of M2M relationships is the activation of automatic connections between the environment and devices which can finally be matched with an individual. Objects are able to interoperate and enable processing. There are two main consequences. First, the processing system is proactive and has an amazing effect: readers are true gateways to a single channel collecting, delocalizing and processing personal data; the digital person, increasingly transparent, loses control of his own

[3] Regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). Brussels, 25.1.2012, COM (2012) 11 final, 2012/0011 (COD). http://www.europarl.europa.eu/document/activities/cont/201305/20130508ATT65776/20130508ATT65776EN.pdf

[4] Art. 14.3 Directive 2002/58/EC on privacy and electronic communications (Directive on privacy and electronic communications) already contained a reference to the influence of technology: "Where required, measures may be adopted to ensure that terminal equipment is constructed in a way that is compatible with the right of users to protect and control the use of their personal data, in accordance with Directive 1999/5/EC and Council Decision 87/95/EEC of 22 December 1986 on standardization in the field of information technology and communications".

[5] Hildebrandt, M: A Vision of Ambient Law. In: Regulating Technologies, p. 185, Ed. Roger Brownsword and Karin Yeung, Oxford, Hart (2008): "Facing life in a digitalised world, in intelligent environments with hybrid multi-agent systems, with real time monitoring and real time adaptation to one's inferred preferences, legal normativity will have to be reinvented".

[6] See Art. 29 WP Opinion nº 13/2011, adopted on 16 May 2011 (WP 185) *on Geolocation services on smart mobile devices*, p. 9, http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2011/wp185_en.pdf

privacy, for the benefit of the controller. Second, this processing is not even imaginable: while some forms of electronic communication (now "conventional" such as Internet) involve invisible treatment[7], the Internet of Things involves unimaginable processing related to the most everyday acts. The system creates a continuous cycle of observation and people voluntarily participate with everyday microdecisions.

So, the mere fact of living in society provides personal information about us. Acting through devices leaves an ineffaceable and reusable track that even makes it possible to locate the holder and to send him a decision related to his profile. This processing can not only discriminate (the individual is submitted to decision-makers) but also can standardize asymmetry between individuals (ignoring the processing) and controllers (able to use information)[8]. We are dealing with a system of collective informational risk because tracking is embedded in social behaviour. There are three main threats: tracking and geographic surveillance (readers revealing the holder's identity), profiling (identity is linked to a profile embedded in a tag or maintained in the cloud) and, finally, action. As Weinberg says referent to RFID, after learning a person's identity "people or devices associated with the reader network can take actions regarding that person (…) based on their knowledge of who the subject is and what he or she is like"[9].

Looking again at the previous examples, we can analyse the kind of tracks left by users and the purposes they can be used for:

- For instance, a weather information service involves location data (processed by service provider); when matched with login data (processed by the operating system) they reveal the places I visit and can be used to send new information.
- By using smart cards to control a bike service, the provider records when and where you take or return your bike and therefore the circuits you make and their frequency. Personal habits are information that may be used against you.
- By using facial recognition patterns, it is possible to identify the same person in other environments. Moreover, physical data provides anonymous profiles to match people, for example, to deny entry into a building.
- Equipment to help people with disabilities reveals health or physical data. Let's think for a moment about asthma inhalers detecting overuse or pill boxes indicating the date of replacement: the provider knows outstandingly sensitive data which could be used by an insurance company.

When robots do not use personal data, privacy law does not interfere with industry (for instance, a device only designed to warn of danger when detecting human presence). There is a privacy issue when robots are able to identify a bystander or even a device.

---

[7] See Art. 29 WP Recommendation 1/99 *on Invisible and Automatic Processing of Personal Data on the Internet Performed by Software and Hardware*, 23.02.1999 (WP 17), http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/1999/wp17en.pdf

[8] See Solove, D. J.: The Digital Person. Technology and Privacy in the Information Age, p. 41, New York University Press, New York – London (2004): "the problem with databases and the practices currently associated with them is that they disempower people".

[9] See Weinberg, J.: Tracking RFID. 3 ISJLP, I/S A Journal of Law and Policy for the Information Society, p. 811 (2007-2008).

3 New Challenges: Pseudonyms, Anonymous Processing and Profiling

The examples above illustrate that tracking devices set up new data. It is no longer necessary to identify a person to contact a device, perhaps itself linked to a profile[10]. New technical possibilities have raised the need to adapt the legal framework. The current concept of personal data means "any information relating to an identified or identifiable natural person"; and an identifiable person is "one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity" (art. 2.a Directive 95/46/EC). But identification does not remain necessary to contact a device. This has been noticed by some documents: it is worth noting the Art. 29 Working Party Opinion 4/2007, on the concept of personal data (20-062007, WP 136)[11] and the 29 WP Opinion 01/2012 on the data protection reform proposals (23-03-2012, WP 191) encouraging to introduce explicitly the concept of pseudonymised data in the future European Regulation[12].

Recital 24 of the Proposal for General Data Protection Regulation considers that online identifiers provided by devices, applications, tools and protocols are not necessarily considered personal data in all circumstances (because the traces they leave can only be used if combined with unique identifiers and other information). However, the Draft Report on the Proposal by the Committee on Civil Liberties, Justice and Home Affairs (European Parliament, 17.12.2012) proposes many corrections and considers that the Regulation should be applicable to identifiers because it is impossible to know whether, in the future, some identifiers will make it possible to single out people. The Committee promotes 'pseudonyms' because they allow the singling out of people without identifying them. The consent to data processing in the form of pseudonyms may be given by automated means using a technical standard which allows the data subject to clearly express his or her wishes without collecting identification data[13]. This "Do Not Track" solution combines the

[10] The concept of contactability comes with the Directive 2002/58/CE on privacy and electronic communications. Traffic and location data are both required to convey or to indicate the position of a terminal. In both cases, the issue is the connection to a terminal. See Recital 14 and 15 of the Directive 2002/58/CE. We can add that the MAC address of a WiFi access point, in combination with its calculated location, shows the location of his owner see Art. 29 WP Opinion nº 13/2011, adopted on 16 May 2011 *on Geolocation services on smart mobile devices* (WP185), http://ec.europa.eu/justice/data-protection/article-9/documentation/opinion-recommendation/files/2011/wp185_en.pdf

[11] This Opinion focuses on the relationship between objects and individuals and the capability to identify individuals making "unique combinations" or analyzing retraceable pseudonymised data. http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2007/wp136_en.pdf

[12] http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2012/wp191_en.pdf

[13] See Amendments nº 85 and nº 105. Proposed art. 4. 2. a) defines 'pseudonym' as "a unique identifier which is specific to one given context and which does not permit the direct identification of a natural person, but allows the singling out of a data subject".

use of pseudonyms and technical resources to avoid tracking related to an identity[14]. But still, it will be possible to contact a device when you cannot identify the owner.

Finally, what about anonymous profiles? Robotic systems use anonymous factions or behaviours to predict concrete attitudes. Physical appearance or movements can be used as anonymous data to make discriminatory decisions on "dissident" or "right" people (for instance, refuse entry to somebody matching the profile of a "suspect"). The amended Proposal of General Regulation suggests a definition of "profiling" in a new art. 4.3.b) as "any form of automated processing of personal data intended to evaluate certain personal aspects relating to a natural person or to analyse or predict in particular that natural person's performance at work, economic situation, location, health, personal preferences, reliability or behaviour"[15].

The new Regulation reinforces the right to not be subject to profiling or measures based on profiling by means of automated processing. Such measures should only be allowed when expressly authorised by law, carried out in the course of entering or performance of a contract, or when the data subject has given his consent (proposal of amendment of art. 20). The controller must disclose the existence of profiling and measures based on it; and the profiling activities shall not include or generate any special data (inter alia, data revealing race or ethnic origin, political opinions, religion or beliefs, genetic data or data concerning health or sex life). Measures based on profiling which produce legal effects concerning the data subject (or significantly affecting him) shall not be based solely on automated processing (proposal of amendment of art. 20.2.c).

4 Legal Framework and Technical Regulation: Privacy Is Not the Problem but the Solution

The principles of European regulation on personal data are still valid, but the law must be adapted to technical capabilities. The proposal for a General Regulation incorporates the role of technology. The first General Directive 95/46/EC is dated 1995 and Directive 2002/58/EC, which includes technical references (identifiers, location data…), is only applied to public electronic communications providers. It is time to adapt the General Regulation and keep in mind technical factors. The Proposal includes technical aspects among the obligations of the data controller. This is an important development that adapts law to reality.

[14] The Draft Report refers to the German solution in *Telemedia Act* of 26 February 2007 (Federal Gazette I, p. 179), section 15.3: "For the purposes of advertising, market research or in order to design the telemedia in a needs-based manner, the service provider may produce profiles of usage based on pseudonyms to the extent that the recipient of the service does not object to this (…). These profiles of usage must not be collated with data on the bearer of the pseudonym".

[15] Recital nº 21 states that a processing activity can be considered to 'monitor' data subjects, "when individuals are tracked on the internet *or through other means* (…) with the intention to use, or potential of subsequent use of data processing techniques which consist of applying a 'profile', particularly in order to take decisions concerning her or him or for analyzing or predicting her or his personal preferences, behaviors and attitudes".

Apparently, only the service provider using robotic engines (the 'data controller') is responsible for law enforcement: designers and producers do not seem concerned[16]. Actually, the design of robotic machines creates default rules and their operation influences the processing of data in one way or another. Technical standards involve designers and manufacturers because they are responsible for incorporating privacy rules. Their role is essential to fulfill privacy standards because technology also has rule-making power[17].

Privacy should not be perceived as a problem but as a solution. Users' and controllers' trust is essential to incorporate robotic machines into social life. Increasingly, privacy is an issue of interest because people are worried about how the use of certain devices threatens personal freedom. And providers who use robots want to know if they are designed to withstand law enforcement. Technical standards are an instrument to ensure compliance with the law and give confidence to markets. Therefore designers and manufacturers will not have a neutral role[18].

This evidence has been highlighted by Art. 29 WP Opinion 02/2013, on apps on smart devices, adopted on 27 February 2013 (WP 202)[19]. The Opinion warns that "a high risk to data protection also stems from the degree of fragmentation between the many players in the app development landscape", including app developers and Operating System and device manufacturers (OS and device manufacturers). The Opinion also highlights the necessity to cooperate among them[20].

---

[16] As the Art. 29 WP Opinion 02/2013, on apps on smart devices, adopted on 27 February 2013 (WP 202) warns: "The OS and device manufacturers should also be considered as data controllers (and where relevant, as joint controllers) for any personal data which is processed for their own purposes such as the smooth running of the device, security etc. This would include user generated data (e.g. user details at registration), data automatically generated by the device (e.g. if the device has a 'phone home' functionality for its whereabouts) or personal data processed by the OS or device manufacturer resulting from the installation or use of apps".

[17] See Reidenberg, J. R.: Lex informatica: the formulation of information policy rules through technology. Tex. L. Rev. 76, p. 553 (1997–1998); Trudel, P.: Lex electronica, http://www.chairelrwilson.ca/cours/drt3808/Lexelectronicatrudel.pdf, p.10; Ulatowski, L. M.: Recent developments in RFID technology: weighing utility against potential privacy concerns. ISJLP I/S (2007–2008); Schmidt, J. M.: RFID and privacy: living in perfect harmony. Rutgers Comp. Tech. L. J. 34, p. 260 (2007–2008); A. Sanfeliu, M.R. Llácer, M. Gramunt, A. Punsola and Y. Yoshimura, Influence of the Privacy Issue in the Deployment and Design of Networking Robots in European Urban Areas. Advanced Robotics, 24 (2010).

[18] Trudel, P.: Privacy protection on the Internet: risk management and networked normativity. In: Reinventing Dara Protection?, Gutwirth, S., Poullet, Y., De Hert, P., De Terwange, C., Nouwt, S. (edit.), Springer, Heidelberg (2009).

[19] http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2013/wp202_en.pdf

[20] See Art. 29 WP Opinion 02/2013, on apps on smart devices, p. 2: "Most conclusions and recommendations in this Opinion are aimed at app developers (in that they have the greatest control over the precise manner in which the processing is undertaken or information presented within the app), but often, in order for them to achieve the highest standards of privacy and data protection, they have to collaborate with other parties in the app ecosystem. This is particularly important with regard to security, where the chain of multiple actors is only as strong as its weakest link".

5 Guidelines for Privacy Safe Robots

As we have seen, robotic machines should allow the use of pseudonyms and facilitate legal profiling. These examples show that it is important to think about design and technical duties from the perspective of law enforcement. Services providers and customers expect it to be so. It is normal to forecast safety standards to avoid legal responsibilities for damages (for example, a robot injures somebody). But to respect fundamental rights, it is important to adapt the law to technical environments and afford privacy friendly machines. Surely, this approach is more familiar to engineers than to lawyers.

The proposal for a General Data Protection Regulation points to three main duties regarding security: data protection by design and by default, data protection impact assessment (PIA), and the establishment of data protection certification mechanisms, seals and marks (arts. 23, 33 and 39 PGDPR).

The Regulation states the principle of 'data protection by design and by default', highly desired by some scholars[21] and 'Art. 29 WP'[22]. In accordance with art.23 PGDPR, the controller shall implement appropriate technical and organizational measures and procedures in such a way that the processing will meet the requirements of the Regulation. This means that any service provided by robotic machines shall be organized ensuring the recipients' privacy. Also, the controller shall implement mechanisms for ensuring that, by default, only personal data which is necessary for each specific purpose of the processing is processed, and is especially not collected or retained beyond the minimum necessary. The European Commission may lay down technical standards for these requirements. Default privacy means that the normal operation of the machines should not encourage excessive use of data according to the law. This relates to the principle of proportionality: if personal data are not essential for a performance, it is better not to process them.

The Regulation sets out the duty for the controller (or the processor acting on the controller's behalf) to carry out a privacy impact assessment of the envisaged processing operations where they present specific risks to the rights and freedoms of individuals (art. 33 PGDPR). Some operations are considered to present specific risks:

---

[21] See Cavoukian, A.: Privacy by design… take the challenge, p. 25, http: // www. privacybydesign.ca/publications.htm; see also Kenny, S. and Borking, J.: The Value of Privacy Engineering. The Journal of Information, Law and Technology (2002): they define "Privacy-Enhancing Technologies" (PETs) as a "coherent system of ICT measures that protect privacy by eliminating or reducing personal data or by preventing unnecessary and/or undesired processing (and storage) of personal data, all without losing the functionality of the system", http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2002_1/kenny/

[22] See the Communication from the Commission to the European Parliament and the Council on *Promoting Data Protection by Privacy Enhancing Technologies (PETs)*, Brussels, 2.5.2007 COM(2007) 228 final, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2007: 0228:FIN:EN:PDF the Recommendation of 12.5.2009, on *The implementation of privacy and data protection principles in applications supported by radio-frequency identification*, Brussels, 12.5.2009 http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32009 H0387:EN:HTML and the Art. 29 WP Opinion 9/2011 on the revised Industry Proposal for a Privacy and Data Protection Impact Assessment Framework for RFID Applications (WP 180) http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2011/wp180_en.pdf

we can emphasize automated processing involving a systematic and extensive evaluation of personal aspects or for analysing or predicting the economic situation, location, health, personal preferences, reliability or behaviour, which produce legal effects concerning the individual or significantly affecting him[23]. The assessment shall contain a general description of the envisaged operations, an assessment of the risks to the rights and freedoms of data subjects, the measures envisaged to address the risks, safeguards, security measures and mechanisms. The PIA is a duty before marketing a product and some companies are already offering this service in anticipation of the Regulation coming into force[24].

Finally, the Member States of the European Union shall encourage the establishment of data protection certification mechanisms and of data protection seals and marks, allowing data subjects to quickly assess the level of data protection provided by controllers and processors (art. 39 PGDPR)[25]. Also, the Commission may lay down technical standards for certification and promote mechanisms and data protection seals and marks.

Actually, these patterns are well known among designers and producers of machinery. We can mention the Directive 2006/42/EC of 17 May 2006, on machinery, and the Directive 2001/95/EC of 3 December 2001, on general product safety. The former provides the basis for the harmonisation of the essential health and safety requirements for machinery at European Union level, the latter ensures that products placed on the market are safe and do not present any risk (or only the minimum risks compatible with the product's use) considered to be unacceptable in view of maintaining a high level of protection for the health and safety of users.

In the future, privacy will also become an element of product safety and a manufacturing aspect to be certified according to UE legal framework. According to the Directive 2001/95/EC, producers shall be obliged to place only safe products on the market. A product is presumed to be safe when it conforms to safety standards. The producers shall provide consumers with the relevant information enabling them to assess the risks inherent in a product and they shall inform the authorities when they know that a product that they have placed on the market poses

---

[23] Consider also art. 33.2 b) PGDPR: “information on sex life, health, race and ethnic origin or for the provision of health care, epidemiological researches, or surveys of mental or infectious diseases, where the data are processed for taking measures or decisions regarding specific individuals on a large scale”, c): “monitoring publicly accessible areas, especially when using optic-electronic devices (video surveillance) on a large scale” and d): “personal data in large scale filing systems on children, genetic data or biometric data”.

[24] See Microsoft GS1 EPC/RFID Privacy Impact Assessment (PIA) Tool, http: //www. gs1.org/epcglobal/pia/. Referring to assessment of privacy risks under American law, see: http://www.dhs.gov/privacy-compliance. See also some International Standards on ISO web: http://www.iso.org/iso/home/search.htm?qt=PRIVACY&sort=rel&type=simple& published=on, and the guidelines regarding the security of mobile apps: ENISA "Smartphone Secure Development Guideline": http://www.enisa.europa.eu/activities/ Resilience-andCIIP/critical-applications/smartphone-security-1/smartphone-secure-development-guidelines

[25] About seals and marks as fair information practices, see King, N. J.: When mobile phones are RFID equipped- finding E.U.- U.S. Solutions to protect consumer privacy and facilitate mobile commerce. 15 Mich. Telecomm. & Tech. L. Rev., pp. 201-202 (2008-2009).

risks to the consumer. The Commission has laid down a European network of administrative cooperation (RAPEX) that enables the exchange of information and a joint surveillance and cooperation with regard to the tracing, withdrawal and recall of dangerous products. The Directive 2006/42/EC sets the EC declaration of conformity and the EC marking that ensure that machinery satisfies the relevant essential health and safety requirements relating to design and construction.

All these measures should have a translation in the field of privacy. A secure processing depends on proper design. It is important to decide where the data will be stored (in devices, service provider's systems, in the cloud) and how they are transferred, preventing a security-friendly environment and unauthorized access both in transit and when stored. In this sense, methods of user identification and authentication should be carefully designed and manufactured, through strong encryption algorithms and keys. End-users information is also important in order to allow proactivity: the user must be able to opt-in and to see what data are being processed and to selectively enable and disable permissions[26]. Devices should display icons and means to give information and have a proactive relationship with a device ('kill switch' technology to withdraw processing or to perform opt-in or opt-out options)[27].

6 Conclusions

Privacy data protection needs technical normativity. The data subject's consent to processing is not enough to guarantee privacy. The exercise of rights needs to be completed with appropriate technology standards. A European privacy mode should assess the privacy impact of robotic machines and avoid default options. In particular, it should facilitate the use of pseudonyms and the compliance of legal profiling and encryption. The collaboration of engineers and manufacturers is henceforth needed and leads to a new stage where privacy issues may be seen as a security solution rather than an obstacle preventing the development of technology. What's more, note that standardization should not be limited to mere compliance with legal duties. The standardization should look for best practices. This is particularly true where law restricts protection to identifiable users but it is still possible to contact a device according to an anonymous pattern affecting the real user although not identifiable. We can state that design also builds trust and trust is a value exceeding the mere compliance with the law.

Acknowledgements. This work is framed in the activities of Gredint 2009 SGR 944 GRC funded by Generalitat de Catalunya.

[26] Art. 29 WP Opinion 02/2013 on apps on smart devices, p. 18.

[27] Hildner, L.: Defusing the threat of RFID: protecting consumer privacy through technology-specific legislation at the state level. Harv. C.R.-C.L. L. Rev. 41, p. 165 (2006).

European Legal Issues Applied to Safe Robots

Mariló Gramunt-Fombuena

Grup de Recerca en Dret Privat, Consum i Noves Tecnologies, University of Barcelona, Faculty of Law, Avenida Diagonal 684, 08034 Barcelona, Spain
marilo.gramunt@ub.edu

Abstract. In this article we study the safety issue of deploying robots from the legal point of view. We start describing the needs for deploying robots and then the legal issues related to safe robots. Then we study the liability from the contractual and extra contractual point of views, and we explain how is the present legislation about the responsibility of the problems that can happen with robots and their deployment. Also, we discuss the company and public administration liability (quality problem) and how it is approached by the European legislation. Moreover, we discuss how tele-operation of robots by humans can affect the safety issues.

Keywords: Legal issues, autonomous robots, safe robots.

1 Introduction

Robots are becoming a reality in ( ) many different aspects of our everyday lives. In the next decades networked robots will circulate in the streets and will perform different tasks that will help to improve the quality of life of the citizens in urban areas. The robots, human beings and the environment will share data to do tasks, for example, for information assistance; for guiding or cleaning public spaces; or to do surveillance. By the way, press has recently been published that in Geneva airport, an android -Robbl- will provide information to passengers.

In order to allow robots to circulate freely, the safety issue will have to be solved from different point of views: from the technological point of view, robots must be reliable and safe to avoid hurting people; neither damaging any private or public property and other issues related to vehicle (automotive, motorcycle, etc.) safety. From the legal point of view they must follow the established rules and laws in private and public spaces.

Due that the social robot [17] is an autonomous system which has full autonomy to take its own decisions, who will be the responsible if something goes wrong? Are the society and the legal issues prepared to allow circulating robots in the street? In which way have the laws and urban rules to be changed or adapted to allow the deployment of robots from the security and safety aspect? These and other issues will be discussed in this article.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_47,

2 The State of the Art in Safety

The concept of *Safety* makes reference to the necessary technical regulations to avoid injuries to people that are in contact with the robots. It is identified with the idea that products must be usable without producing damages neither to people nor to things. It is guided through the legal dispositions that contain the safety norms and they are linked to the general duty principle of neither injuring, nor to put in danger the health and the physical integrity of people, as well as the preservation of their goods. In any case, the robot should fulfill the safety norms [6].

In the year 2000, the European Commission [7] emphasized that in order to have safe robots it is necessary to design prevention policies to the risks associated to the technological innovations. With this aim, it is necessary to identify, characterize and define what should be undertaken to protect people, the environment and other goods, following a coherence plan between the adopted measures and the state of the art. On the other hand, it is necessary to place, in perspective, the real or supposed risks, looking for the balance among risks and benefits that the science and the technology contribute to the society in general, and to each individual in particular. Let us not forget that the applications of the robotics are very general and they are applied in very diverse spheres: from the citizen security up to home assistance, as well as in applications to medical and hospital environment. Moreover, it has to be taken into account that the absolute security, "risk zero", is almost impossible to ensured in this area.

Once detected or diagnosed the risks, we should know which the liability sphere is, where the involved agents have to assume the caused damage and, mainly, which are the legal cases that can induce the liability and how to avoid them. The working papers related with the production and installation of robots in the society, insist on the ideas of the production quality, industrial security and environmental security, aspects that impact directly in the juridical imputation of liability.

We should distinguish two aspects that are related with the safety. On the one hand, we can refer to the existence of a general legal frame that imposes the execution of certain technical requirements in the machinery construction. On the other hand, it must also highlight the existence of a general legal frame related (?!) to injuries or liability damage, which will be applied to the robots, although they have not been specifically designed for them.

2.1 Applicable Norms to the Robots' Production and Commercialization

As we said, there exist(-) a series of norms that come from the use of robots in the production lines, in spite of not having been developed specifically for them. In the European Union territory, this legal frame related to technical requirements is embodied in a set of dispositions that include the guidelines that the State members should be aware of in the preparation of their norms concerned to safety in different fields:

Directive 2006/95/EC. [8] The purpose of this Directive is that the Member States shall take all appropriate measures to ensure that electrical equipment may be placed on

the market, only if they have been constructed in accordance with the technical engineering criteria in safety matters, and that does not endanger the safety of persons, domestic animals or property (art. 2). The Member States shall take all appropriate measures to ensure that, in particular, electrical equipment which complies with the safety provisions of harmonized standards shall be regarded by their competent administrative authorities as complying with the provisions of Article 2, for the purposes of placing on the market and free movement as referred to in Article 2 and 3 respectively (art. 5). The safety measures make reference to aspects such diverse as the supply of outstanding information on the product (for example, user instructions), the adoption of protection measures against the dangers coming from the own electric material (for example, to avoid that temperatures or dangerous radiations take place or to have the proper isolation system) as of external influences (for example, the electric material must prevent the danger of an overload).

Directive 2001/95/EC. [9] Their main purpose is to guarantee that the products that are introduced in the European market are safe (art. 1) from the point of view of the minimization of risks.

Directive 2006/42/EC. [10] The purpose of the present Directive is to establish a legal frame to ensure that machinery may be placed on the market and/or put into service only if it satisfies the relevant provisions of this Directive and does not endanger the health and safety of persons and, where appropriate, domestic animals or property, when properly installed and maintained and used for its intended purpose or under conditions which can reasonably be foreseen (art. 4), but only when there are not specific Community Directives that cover more specifically the hazards referred in Annex I (art. 3). To accomplish this goal, the Directive forces the producer to carry out an evaluation of the risks that allows determining the safety requirements that are applied to the machine, applying the evaluation results to the machine design and production.

2.2 Norms Applied to Injuries and Damage Liability

In the case of injuries or damage liability we must differentiate two fields: On the one hand, the damages caused by defective products and, on the other hand, the damages caused due to the robot's use. In the first supposition we have an European legal frame, but we do not have any for the second one, due that it is only applicable the legislation of each one of the Member States. However, the researchers are working determine general principles applicable to this type of liability.

Council Directive 85/374/EEC. [11] The purpose of this Directive is to approach the legislations on producer liability among the Member States, with respect to the damages caused by the defective state of its products, betting for a regime of producer strict liability derived from the inherent risks to the modern technical production (art. 1).

Tort Law. The Tort Law is structured fundamentally around the division among the subjective or fault based liability, on one side, and the objective or strict liability, on the other one. In this way, the basis of the subjective liability is the existence of a behavior contrary to a juridical duty: the harm's author is responsible for it only if he has carried out an illicit act, that is, if he has carried out an act not allowed by the Law. This fundamental distinction is actually included in all the European juridical systems [12]. In order to unify approaches, in 1992, the European Group on Tort Law was created. This Group presented in 2005 the Principles on European Tort Law, which fundamental basis is that the person to whom the damage suffered by another can be imputed legally; it is obliged to repair it. The damage can be imputed to the person if it is due to his/her culpable behavior, due to his/her abnormally dangerous activity, or if it has been produced by his assistant in the exercise of his functions. Moreover, it should converge a series of suppositions: The necessity that it exists a damage to compensate, the causality relation and the necessity that an approach of subjective imputation exists, be the blame, the risk or the duty of responding for third people's fact (art. 1:101 PETL) [13].

3 Safe and Defective Products

The applicable norms to the production and commercialization of products offer three key concepts that should be analyzed and related with the robots: the product concept, the concept of safe product and the concept of faulty product.

3.1 Is a Robot a Product?

The answer is, in any case, affirmative. The art. 2 of the Directive 85/374/CE [11] gives a very large definition of product, considering that it is *"any personal property"*, be or not incorporated to another good. In this way, a robot is a product, in the same way that it is each one of the components that integrate it (from the sensors until the electric cables).

The robot can also be included in the category of "*produc*t" to which the art. 2 a) refers, of the Directive 2001/95/CE [9], although this article contains a more restricted notion, since it includes what is the purpose of the product destination. To this precept, the robot could be considered as a product when his destination is being used by a consumer, either because it is the robot's receiver (for example, a domestic robot), or because under reasonably foregone conditions it will be used by a consumer, although it is not going to be the final receiver, in the mark of a commercial activity (for example, a robot placed in a shopping mall).

The consideration of the robots as products, with respect to the mentioned norms, implies to take in consideration the rules required to their production and commercialization, as well as the subjection to the liability that is derived in the event of non fulfillment of them.

3.2 When a Robot Can Be Considered a Safe Product?

The art. 2 b) of the Directive 2001/95 [9] defines what is a safe product on the base of diverse parameters like the normal or reasonably foreseeable conditions of use that will make (of?) this product. To study the elements that impact in the legal concept of safe product we will take the example of a domestic robot-vacuum cleaner. This robot is fabricated with components homologated in the UE and completing all the norms that are of the adoption, in the Member States, of the applicable Directive to the production of products. Moreover, in the documentation that describes the characteristics of the product, which is given to the user, there are the instructions explained to its assembly, installation and maintenance. In this documentation it is indicated that the robot's use doesn't require of any additional cleaning product and it is warned about the risk that can suppose putting in contact the robot with this type of cleaning products for the user's safety. It is also warned about the convenience of maintaining the robot far from the reach of the children. Finally, it is indicated that the robot's approximate lifespan is of 8 years, whenever the maintenance instructions and upgrade of the software have been done.

In principle, the robot will be considered a safe product, since it fulfills all the requirements that are required in art. 2 b) Directive 2001/95/CE [9].

In the definition of the safe product made above, the risk is included that the use of the product can entail. In this sense, the Directive states that the only existence of a potential risk causing damage does not exclude the consideration of safe product, whenever this risk moves in the parameters of minimization, admissibility and information. The introduction of the risk as an element of modulation on safe products, it is important in order to include it in this category and not in that of *dangerous product*. Indeed, if the product does not satisfy the characteristics reflected in the definition foreseen in the art. 2. b), it will be considered as *dangerous product*, in the way that establishes the art. 2 c) of the Directive 2001/95 [9]. Using the example of the robot-vacuum cleaner, let us imagine that the producer has not kept in mind that the robot should stop if it detects the presence of a person and for this reason, the robot injures the person; or he has not being warned about the prohibition of using cleaning products with the robot and the user uses them producing a short circuit and a fire in the robot. In both cases the robot will be included in the category of *dangerous product*.

In any case, a robot will not be considered a *dangerous product* if it has the possibility to improve the safety level or to have a low level of risk. For example, if a robot has been already considered a safe product, the improvement of the sensors for the detection of people by one producer, does not convert the robots of the other producers into dangerous because they do not have such sensors. According to art. 2 b) the robot will continue being safe whenever the risk that generates moves in the parameters of the admissibility.

3.3 When Is a Robot Considered a Defective Product?

To know when a product is or not defective is not a simple task, since the current normative links the defective character to the absence of safety that would be when it

does not provide the safety which a person is entitled to expect. So, we move in the field of the uncertain juridical concepts, since the norm does not give a concrete content about the specific reach of what is the required safety for a product. Nevertheless, there are three circumstances that will allow to value if the product is or not defective. In this way, art. 6 of the Directive 85/374/CE [11], indicates that in the determination of the defective character we will value the presentation of the product, the use that reasonably could be expected of it, and the time when the product was put on the market. Anyway, the defect that allows the qualification of a product as *defective* is not a mere material defect, but rather it must generate a safety defect of the product.

Beforehand we can affirm that any robot that cannot be considered safe according to the concept that we have given, that is to say, that should be considered dangerous, will enter in the widest category of defective product. A product can be defective for three causes: because there is a manufacturing defect, because there is a design defect or because the instructions that accompany it are defective. But are these causes sufficient?.

1. Manufacturing defect. A robot will be considered defectively fabricated when it differs from his design, even in the case that in the manufacturing process all the cautions have been taken into account. It would happen, for example, if in the robot's manufacturing process, in spite of using the materials and appropriate components and that they have followed all the rules imposed by the current legislation, the assembly process has not being well done and when robot is put in operation, the robot damages the user.

The manufacturing defect can only affect (-) one or few elements of the series, but it is not reasonable to attribute the defect to the complete series, so that the comparison between the defective product and the rest of the series can be useful to determine the existence of a manufacturing defect (this is the approach followed by the Spanish legislation in the art. 137.2 RD 1/2007 [14]).

2. Design defect. Contrary to the manufacturing defect, the design defect will affect the whole series of robots affected by the same design. This way, when it is detected that a product presents a design defect that transforms it into a non-safe or dangerous product, the product must be withdrawn from the market. Considering the robot-vacuum cleaner, if the vacuum tub end has been designed triangular to improve the effectiveness in the cleaning of the corners, but the edges have been left sharp, then the product can injure the users. We could also refer to a design defect in the case of a robot which has been designed for a height which makes it unstable and, therefore, it is susceptible of causing damages to people or to fall over them.

3. Defect in the instructions. As we have already seen, the lack of instructions or warnings converts a product to dangerous. Since we have included the category of the dangerous products in that of defective products, we remit to what we have indicated when referring of those.

As well as the concept of safe product is modulated by the notion of acceptable risk, the concept of defective product also has a shade, in this case of technological-temporary character. Art. 6 of the Directive 85/374/CE [11] considers

that the improvement of a product does not transform the previous version into a defective one. It is obvious that it is fully applicable to the world of robots.

4 Using the Robot Environment to Constraint Its Safety

The definitions of safe and defective product must be also related to the use of this product.

As we know, a robot can have a many diverse uses and for this reason, the safety norms will be also diverse in function of the expected utility. Now then, a standard group of norms relative to the safety of robots could be established in function of the environment in which there will be developed its task, since the risk of causing damages augments in the spaces open to the public and it diminishes in the spaces of restricted access.

4.1 Robots in Public and Semi-public Space

When we talk about public space we refer to the use of robots in spaces open to the public in general, whether they are indoor or outdoor robots, for example in the street, forests, shopping malls, amusement parks, museums, hospitals, etc. In all these suppositions we should keep in mind two aspects: on the one hand, the possibility that the robot's control is realized through a tele-operated or semi-tele-operated system (in this case some functions will make them in an autonomous way, for example the circulation in a street, and the other tele-operated, for example when the robot has to assist a person); on the other hand, the uncertainty of the people's behavior when they are in contact with robots, which can carry out accidental robot behaviors. This single fact entails the necessity to extreme the safety technical measures, because it is not done, the robot can be considered a defective product in the open sense. In this sense, it will be of utmost importance, for example, to establish specific circulation rules (traffic rules) to specify the maximum robot speed in a specific area, keeping in mind the sensibility in the detection of obstacles and the necessary time to stop safely the robot.

The use of robots in public or semi-public spaces increases the level of risk of accidents, which implies not only to produce safe robots but also to establish clear circulation rules in order to avoid the imputation of derivative liability, question that we will be analyzed in the following section.

4.2 Robots in Private Space

We identify the use of robots in private space when the robot's activity is developed in spaces restricted to the public in general or have the category of restricted access (private homes, industries, operating rooms). In most of the cases, the robot will have been conceived to do concrete tasks and the owner will be the one who will control the robot's activity; so, the identification and control of accident risks is easier. In these cases, with independence of the execution of the safety technical norms, the

information that producer offers to the end user, through the guiding manual, the safety warnings and the way of minimizing the potential accidents will be very relevant with respect to the liability of the producer.

5 Liability: Types and Definition

The concept of liability is linked with the existence of damage evaluable economically, that is, compensable. Nevertheless, all damage economically evaluable will not be susceptible of compensation, due that in order to be compensated it must concur all the legal circumstances to assign a responsible [16].

According to the fact that originates the damage, we will locate the liability in a contractual environment or not, which will be decisive for the application of some or other norms of liability as, for instance, Kobayashi has noted related to Japanese Law [17].

5.1 Contractual Liability

This liability is applied for robots that will be sold on the market, whether they are bought by individuals or by public organisms, and that they are applied to own use or general interest. In these cases, the basis of liability is the economic imbalance that takes place among the contractual parts. The damage that the product's buyer suffers is only economic, since he will have paid a price for a product that does not work well or that the robot does not offer the published performances. In these cases, the liability always has contractual character and it affects directly the seller (it is already the same case when he is the producer or the dealer) who, in the commercialization chain, sells the robot to an end user. As we said, if the robot does not work properly or if his benefits are not as expected, the seller should respond, not for damages caused due to a bad operation, but for breaching the sale contract (or another relationship type, like the robot's rent or loan). It would be the case, for example, of a lawn mover robot that offers the cut to three heights and only cuts one.

To analyze the forms of the seller's liability we need to distinguish two cases:

a) The consumer buyer. If the buyer is a consumer, that is, who buys the robot for a not professional purpose, his options are wider. According to the Directive 1999/44/CE [15], seller responds for lack of conformity (in Spain, the transposition norm is the *Real Decreto* 1/2007 [14]. Any advertising message or any use specially recommended, in addition to a bad operation of the robot or if it is unsuitable, allows the buyer to request, at first, its substitution or repair, and in second place, if it is impossible or fruitless, the discount of the price or the definitive resolution of the contract. Damages suffered by the buyer for not being able to use the robot will be reimbursed.

b) The professional buyer. If the buyer is a professional (for example, a company that exploits services of cleaning or surveillance or a Public Administration that places them in the street with resemblances goals), the general civil or mercantile law is applied. It is the general solution in Europe, where there is only uniformity in the norms that have

the consumer protection as purpose. In this case, the tribunals follow two directions, not always in a clear and separate way: the reparation for hidden defects (art. 1484 and following of Spanish Civil Code) and the contract breach, because it was delivered a robot that is not adjusted to that foreseen in the contract (art. 1124 of Spanish Civil Code). In the first case, the buyer can resolve the contract or request a discount of the price; in the second case it is only necessary to resolve it, but damages suffered by not being able to use the robot can be claimed (art. 1101 of Spanish Civil code). The basic difference among them is the seriousness. The seller breaches the contract if the robot's performances are far from what is conventional or habitual, that makes it not suitable to its normal use (for example, it is unable to process the number of people in a space and to carry out recount functions). There is a latent defect if it does not work correctly, for example, because of a bad assembly.

5.2 Liability for Defective Products

In the case of the liability for defective products, the starting point is located in the concept of defective product in wide sense, this is, including dangerous products. Liability will arise when a defective product causes damages to people or its goods and damages can be imputed legally to the responsible according to legal criterions. In this case the purpose is neither reestablishing a patrimonial imbalance (typical function of the liability due to hidden defects) nor determining responsibilities caused by the robot's use (non-contractual liability), but of repairing the damage that the product's user has suffered because of a product's defect.

To define the suppositions of liability we should be specific about what damages are compensable, who is the responsible and which are the causes of liability discharge.

a) Compensable damages. Art. 9 of the Directive 85/374/CEE [11] considers compensable damages caused by death or by personal injuries, damages or destruction of any item of property other than the defective product itself, whenever it is a type ordinarily intended for private use or consumption, and was used by the injured person mainly for his own private use or consumption.

Therefore, compensable damages will be those that the defective robot has caused to people or to other things different from the self robot, whenever these goods complete the use requirements and destination to that we have referred. In definitive, it will be damages that have been caused due to the defective robot in the terms that we have already pointed out. This way, the unstable domestic robot (design defect) that loses the balance and falls above a child injuring him; or robot that suffers a short circuit causing a fire in the user's home due to a manufacturing defect; or, finally, the damages caused to the user that has utilized the robot in an inadequate way due to lack of instructions.

b) Responsible. According to art. 1 of the Directive 85/374 [11], the producer shall be liable for damage caused by a defect in his product. Let us observe that in this case liability moves from seller to producer. When the responsible is the seller respect to the buyer, liability has contractual character. In this type of liability there won't necessarily be contractual obligation between the producer and the affected by the damage.

According to Directive 85/374 [11] *producer* is the manufacturer of a finished product, the producer of any raw material or the manufacturer of a component part and any person who, by putting his name, trade mark or other distinguishing feature on the product presents himself as its producer (art. 3.1). From the point of view of the robot's production, there are included in this list, the trader who manufactures the whole robot, all the suppliers of the necessary raw materials for the robot's production, any company that manufactures the robot's integral part (for example, the company that produces the robot arms or the company that programs the robot) and, finally, any person that keep the robot by himself, that is to say that in spite of not having him fabricated the robot, he appears as such on the market.

Moreover, art. 3 gives producer treatment to other persons that ***neither are*** nor have appeared as such in the product. Indeed, art. 2 e) of the Directive 2001/95 [9] adds to the list of producers, the manufacturer's representative when this is not established in the European Community, and, in his defect, to the importer. In this same way, art. 3.2 of the Directive 85/374 [11] gives producer treatment and imposes the same regime of liability to the importers and distributors of products in the European Union; and art. 3.3 gives the same treatment to the product supplier in the cases in that it cannot be identified or to be known the producer or the importer. The inclusion of these people in the category of responsible responds to the idea of endowing from a high protection grade to the aggrieved person, facilitating the exercise from the judicial actions who can demand to people domiciled in the European Union, avoiding the necessity to contest in countries where does not exist treaties of judicial cooperation and also avoiding the necessity to be held to legislations different from the effective ones in the European environment.

According to art. 2 e) of the Directive 2001/95 [9], must be included also in the list, the other professionals of the commercialization chain, in the measure that their activities can affect the product safety characteristics. In this sense, it could end up being responsible for the caused damages, the firm who has made the transportation of the robots if this transportation has not been made, for example, under the appropriate conditions of immobilization and for this reason the robot has diminished the manufacturing safety conditions; or the company that has stored the robots and has not respected the levels of temperature and the required humidity, causing the defects that finally have caused damages to people. Finally, to point out that Spanish legislation also includes the producer concept to the lenders of a service or its middlemen (art. 5 of the *Real Decreto* 1/2007 [14]).

In definitive, the juridical concept of producer does not reduce to maker's material notion, but it is expanded to all persons that is presented as such, in the presentation of the self product.

Given the wide delimitation of the producer concept offered by the Directive, it is probable that there exists more than one responsible. In these cases, art. 5 of the Directive 85/374 [11] establishes a joint liability regime, much more beneficial for the aggrieved, but more onerous for the eventual responsible. The joint regime allows the aggrieved to direct its reclamation entirely against anyone of the potentially responsible, which will fully pay the compensation in the case of considering it reasonable (articles 1137 and 1138 of the Spanish Civil Code). Once satisfied the

amount of the compensation, who has paid will be able to repeat the payment against the rest of responsibles, according to art. 5 of the Directive 85/374 [11].

c) Causes of discharge of responsibility. The liability as a consequence of damages caused by defective products is configured with an objective character, this is, detached of the necessity of proving that the defect was due to the negligence of the producer; or, with other words, that although the performance of the producer has been diligent, if the damage takes place it will be imputed the liability likewise and it will satisfy the corresponding compensation. The only prove that it is demanded to the aggrieved is that concerned to the existence of the damage and the causality relation between defect and damage (art. 4 directive 85/374 [11]).

Although in Europe concrete suppositions of objective liability already existed (Germany, France, England, Spain –in this case due to the jurisprudence criterion-), the Directive's regulation is decidedly inspired in the North American jurisprudence.

According to art. 7 of the Directive 85/374 [11], the producer will be been able to discharge of liability if he proves anyone of the following facts:

That the producer did not put the product better: on the market into circulation. It was, for example, a robot prototype dedicated to demonstrations in the facilities of the company that manufactures it, and it is object of a robbery and put on the market to be sold later on. If the robot's user suffers derivative damages of the robot's defective character, the producer will be expected to have the responsibility proving that they denounced the robbery at its moment and that the circulating setting of the product was due to circumstances completely unaware of its will.

That, having regard to the circumstances, it is probable that the defect which caused the damage did not exist at the time when the product was put on the market by him or that this defect occurred afterwards. There are facts which have a difficult prove, because they demand the accreditation that the product did not present any manufacturing defect. We could think of, for example, the case that the computer program which controls the robot and which, after some time, is damaged and causes an unexpected behavior in the robot that finishes causing damages to the end user. The producer should prove that the computer program did not present any manufacturing defect and that the defect has taken place with posteriori to the commercialization, due to circumstances completely unaware to the robot's introduction on the market.

That the producer did not manufacture the robot for sale or any form of distribution for economic purpose not manufactured or distributed by him in the course of his business. It is to prove that the producer did not manufacture it with the purpose of introducing it on the market. In fact, prove of these circumstances is very similar to that exposed in the previouse analyzed suppositions, for what we remit ourselves to that what has been mentioned.

That the defect is due to compliance of the product with mandatory regulations issued by the public authorities. The producer has been under obligation to follow some certain norms imposed by the State and the obtained result has been a defective product.

That the state of scientific and technical knowledge at the time when he put the product on the market was not such as to enable the existence of the defect to be discovered. This cause of excuse of liability resides in the impossibility of imputing it due to the excusable ignorance. For example, if in the robot's production a habitual concrete material is used in the production of other products of similar characteristics, but, after some time, it is observed that it has a harmful behavior, it will not be considered the robot's defective character due to the ignorance of the harmful effect of the material which was used.

It is necessary to keep in mind that if we identify the circulating setting of a product with their first delivery to any middleman or supplier or directly to the user or final recipient, the cause of excuse of liability will only be operative if it takes place in that moment. On the other hand, if the product was manufactured in a moment that the scientific knowledge did not allow to know the defect, but it is not put on the market until a later moment in which it is already possible to detect the defect, the producer will be considered responsible for the caused damages, since his product will be qualified as defective.

If the responsible is the manufacturer of a component that the defect is attributable to the design of the product in which the component has been fitted or to the instructions given by the manufacturer of the product. This discharge cause leaves the base that the manufacturer of a component has carried out a safe product, nevertheless that, when incorporating it to the end product, it is defective. Whether the manufacturer of the component (for example, the articulate arm of a robot) has made it following the instructions of the robot's manufacturer, or already the robot's defective character once finished is derived of the incompatibility of the piece with the robot's final design (for example, the arm is, seemingly, subject to the robot's body but with the continuous use it comes off and it causes injuries to the user), it will not be responsible the manufacturer of the component, but the robot's manufacturer).

6 Tort Law

The last type of liability has a different basis from the previous ones, since the imputation center is either the self responsible behavior that causes the damage, or in the fact of people who should respond, or for circumstances of the things or animals of which he must take the liability. In the base liability for caused damage we find the idea that the harm's author is the only responsible for it if he has carried out an illicit act, this is, if he has carried out an act not allowed by the Law. At this point, we have already pointed out that for this type of liability there is not a homogeneous regime in the States of the European Union, since it is a not an harmonized subject. Indeed, on the one hand, there are rigid systems as the German that insists on the base of a list of protected interests (§ 823 BGB) or the English that categorizes the civil illicit whose commission will have the liability (*tort of negligence*); on the other hand, there are systems like the French (art. 1.382 French Civil Code), Spanish (art. 1.902 and ss. Spanish Civil Code), Italian (art. 2.043 Italian Civil Code), Belgian (art. 1.370 Belgian Civil Code) or Portuguese (art. 483 Portuguese Civil Code) that establish a general clause of liability,

being doctrine and jurisprudence which are developing the rules and requirements that must converge to determine the suppositions that generate liability. Due to this situation, the task of determining the existence of imputable liability to the author of the damage is more difficult.

The nonexistence of a harmonized legal frame prevents us to present some guidelines of the regime of liability for damage caused in the territory of the UE. For that reason, we will expose only the regime corresponding to the general clause of liability system, more extended. To determine the existence or not of liability the concurrence of a group of elements is demanded: the existence and prove of damage, the prove of the causality relation between the behavior and the damage, and the existence of a liability imputation criterion to the harm's author.

6.1 Robots Located in Public, Semi-publics or Private Spaces Managed by the Public Administration

Let us suppose that the city council of a city decides to acquire a robot with the purpose that it carries out gardening tasks in the parks. The robot can be considered a safe product, not defective, since it has all the advantages that the scientific knowledge and technician allow, his design is adapted and it does not present manufacturing defects. The robot is designed to be tele-operated and an employee of the city council controls it. In spite of everything, one of the following circumstances could take place:

a) A person jogging in the park appears in the robot's trajectory and they collide and the corridor suffers injuries in his leg. Will there be any type of liability of the city council or should the corridor be liable for the damages?
b) While the robot is carrying out the gardening tasks in an area of non admitted access to the citizens, a person enters the area running and collides with the robot. Will there be any type of liability to the city council or should the citizen be liable for the damages?
c) The computer program installed in the robot is prepared to design alternative itineraries that facilitate the realization of the tasks. After working correctly during two years, the robot presents an unexpected behavior and instead of avoiding obstacles he rushes against them, causing injuries to some children that played in the park. Will there be any type of imputable liability to the city council or to the robot's producer?

In all the three outlined suppositions it is relatively easy to demonstrate the existence of the damage and the causal relation between the robot's intervention and the caused damage. But it is not so evident the existence of an imputability criterion to the eventual responsible.

The first question that we should clarify is who should respond for the robot's behavior. In the first and second suppositions there is no doubt that the eventual one responsible would be the city council since it has been responsible that the robot was introduced in the park and, therefore, it is erected to be responsible for the damages that it causes, in the same way as if the accident was caused because of a not well installed traffic light, some swings in bad condition, etc. In the third supposition can exist a concurrent or alternative responsible, this is, possibly when they could be responsible as (-) the city council or the producer.

The second question resides in determining if the robot's intervention in some of the accidents has been due to an imputable behavior to the city council. Let us analyze each supposition separately.

In the case **a)** it is highly probable that the tribunals consider the existence of imputable liability to the city council because the use of the parks for the development of sport activities as jogging is a habitual practice. Since the robot realizes its tasks in a public space, apart from having the appropriate technical mechanisms to prevent risks (supposedly the robot of our example counted on them) the tele-operator that controls it must be specially attentive to the environment vicissitudes, so that only if the damage is caused by the victim's the city council would be exempt from liability. Related to the Spanish supposition, the legislation articulates the Public Administration liability with a marked objective character, since the art. 141 of the Law 30/1992 [16] declares that they will be compensable only the harms coming from damages that the citizens does not have the juridical duty of supporting according to the Law. Therefore, the Public Administration responds with objective character of the caused damages, so that the only way of eliminating the liability of the Administration is to deny the existence of the causality relation or to prove that the victim was the only responsible for the damage.

The injured jogger should, in this case, demand to the city council and this, if it considers it opportune, will be able to exercise the repetition action against the tele-operator responsible of controlling the robot.

The pointed solution for the previous case practically has given us the answer of the supposed **b)**. In this case, and in accordance with the jurisprudential criterions, two circumstances could be given: or to consider that it is a supposition of the exclusive victim's fault since the damage has had its origin in its self negligent behavior when consenting to a forbidden place for people, or to consider the existence of a concurrence of causes in the production of the accident, that is to say that the damage has been partly the victim's fault and partly the lack of attention of the tele-operator. Leaving of the base that in the jurisprudence can exist diverse interpretation criterions, both solutions are possible, being, evidently, more rigorous for the city council the second of the pointed solutions, when having a more objective character that the first one. For the resolution of this supposition the jurisprudence, without a doubt, would keep in mind a warning for the development of works carried out by a robot in the enclosure of forbidden access. Anyway, still in the case of declaring the city council responsible, the quantity of the compensation would be smaller than in the case a), just due to the concurrence of the victim's fault considering the damage.

Finally we must analyze the supposed **c)**. Beforehand, we should discard the possibility that the robot was defective. The robot's producer will prove that the defect was not in the moment it was manufactured and that the scientific knowledge and technicians did not allow foresee that unwished evolution of the computer program and that this behavior would annul the functions of the robot's detention. With regard to the liability of the city council, again we face the character quasi-objective of the Public Administration liability, so that it would probably be considered responsible the city council, because it is given that it has assumed the risk of placing a robot tele-operated

in public space. For this reason, regardless of the victim's behavior, the city council must be responsible for the damages caused by the robot.

6.2 Robots Located in Public Spaces or Semi-publics Managed by a Private Company

Let us imagine the same suppositions that we have analyzed in the previous section, but now the scenario is the garden of a hotel complex managed by a private company. In this case the legal starting point is the principle of damage author's fault, although the interpretation made by the tribunals about the negligence extension is always related with the risk created by the self agent.

From our point of view, the placement of tele-operated robots in a public or semi-public space would resemble to the liability derived by the conduction of motor vehicles that is usually configured as an objective liability: proving the damage and the causation nexus, the harm's author can only be exempted if he proves that the damage went imputable only to the victim's behavior. For this reason, the solution for the cases **a)** and **b)** would be the same that we have offered in the same cases but in spaces managed by the Public Administration.

In the supposed **c)**, the hotel company could venture, with a scarce margin of success, to demonstrate that the damage was caused by a fortuitous case, that is to say, due to an unforeseeable and unavoidable event.

Whether in the case of the liability of the Public Administration as in this last one, it would be advisable to contract a liability assurance that covers the derivative risks of the use of robots in the public or semi-publics spaces.

7 Conclusions

a) The concept of Safety makes reference to the necessary technical regulations to avoid injuries to people that are in contact with the robots. It is identified with the idea that products must be usable without causing damages neither to people nor to things.
b) There are two legal aspects that are related with the safety: a general legal frame that imposes the execution of certain technical demands in the machinery construction and a general legal frame relative to injuries or damages liability, which will be applied to the robots because there is not any disposition specifically designed for the robots. These norms offer three key concepts that can apply to robots: product, safe product and defective product.
c) The definitions of a safe and defective product must be related with the use to that the product will be dedicated. Therefore, a standard group of norms related to the safety of robots could be established in function of the environment in which a robot will develop its function, since the risk of causing damages increased in the spaces open to the public in general and it diminishes in the spaces of restricted access.
d) In the case of the liability for defective products there is an European legal frame, so the European laws are quiet standardized. The starting point is located in the concept of a defective product. Liability will arise when a defective product causes damages to

people or its goods and damages can be imputed legally to the responsible according to legal criterions.
e) In the base of liability for caused damage (tort law) we find the idea that the harm's author is the only responsible for it if he has carried out an illicit act. For this type of liability (-) is not a homogeneous regime in the States of the European Union, since it is a not harmonized subject. In any case, the use of a robot in public or semi-public spaces increases the risk of damages, so liability is more objective than in private spaces.

Acknowledgements. This work is framed in the activities of Gredint 2009 SGR 944 GRC funded by Generalitat de Catalunya.

References

1. Sanfeliu, A., Punsola, A., Yoshimura, Y., Llácer, M.R., Gramunt, M.D.: Legal Challenges for networking robot deployment in European urban areas: the privacy issue. In: Workshop on Network Robots Systems; IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan (2009)
2. Sanfeliu, A., Hagita, N., Saffiotti, A.: Network Robot Systems. Robotics and Autonomous Systems 56(10), 793–797 (2008)
3. Sanfeliu, A., Andrade-Cetto, J.: Ubiquitous networking robotics in urban settings. In: Workshop on Network Robot Systems. Toward Intelligent Robotic Systems Integrated with Environments. Proceedings of 2006 IEEE/RSJ International Conference on Intelligence Robots and Systems (IROS 2006), Beijing, China (2006)
4. European Convention for protection of human rights and fundamental freedoms (Council of Europe, Rome, January 04, 1950)
5. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data
6. Ministerio de Innovación y Tecnología, Libro Blanco de la Robótica (De la investigación al desarrollo tecnológico y aplicaciones futuras). Madrid, Spain (2006)
7. European Commission staff working paper. Science, society and the citizen in Europe. SEC (2000) 1973. Brussels (2000)
8. Directive 2006/95/EC of the European Parliament and of the Council, on the harmonisation of the laws of Member States relating to electrical equipment designed for use within certain voltage limits (December12, 2006)
9. Directive 2001/95/EC of the European Parliament and of the Council, on general product safety (December 3, 2001)
10. Directive 2006/42/EC of the European Parliament and of the Council, on machinery, and amending Directive 95/16/EC (recast) (May 17, 2006)
11. Council Directive 85/374/EEC, on the approximation of the laws, regulations and administrative provisions of the Member States concerning liability for defective products (July 25, 1985)
12. Jansen, N.: Estructura de un derecho europeo de daños. Desarrollo histórico y dogmática moderna. Indret (2003), `http://www.indret.com`
13. Martín, M.: Una primera aproximación a los "Principios de Derecho europeo de la responsabilidad civil". Indret (2005), `http://www.indret.com`
14. Real Decreto 1/2007, que aprueba el Texto Refundido de la Ley General para la Defensa de los Consumidores y Usuarios

15. Directive 1999/44/EC of the European Parliament and of the Council, on certain aspects of the sale of consumer goods and associated guarantees (May 25, 1999)
16. Asaro, M.: Robots and Responsibility from a Legal Perspective. In: Proceedings of the IEEE Conference on Robotics and Automation, Workshop on Roboethics, Rome (2007)
17. Kobayashi, M.: Legal issues about Network Robot Systems. In: Workshop on Network Robots Systems; IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan (2009)
18. Ley 30/1992, de 26 de noviembre, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común
19. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots. Robotics and Autonomous Systems 42, 143–166 (2003)
20. DUSTBOT European project, Cordis, 2006

Axiological Elements to Consider Ethics Applied to Robotics

Antonio Madrid Pérez

Faculty of Law, University of Barcelona, Spain
antoniomadrid@ub.edu

Abstract. The development achieved by scientific and technical knowledge applied to robotics has long posed important ethical questions. In this paper are addressed some of these old and new questions that creates the human-robot relationship and its social dimension. This reflection is applied to the military use of drones, at a time in which the ethical and legal questions accompany some of these uses. As well as the plurality of evaluations determines the design and use of robots, the axiology is used to reflect the ethical problems of robotics.

Keywords: Ethics, robotics, responsibility, values, axiology.

1 'Roboethics': The Old and The New

The ethical reflection applied to robotics faces two main questions: an old and a new one. The first one has to do with understanding the robot as an instrument of human action: both as the result of human actions and as being used in the development of their action. Thus, ethics has as its object of reflection the design, programming, production and human use of robots. The second kind of questions, the new ones, do not arise from a strictly perception of the robot as an instrument, but it raises the relationship human-robot. From this perspective, some state-of-the-art robots begin to be thought of as autonomous agents.

The term 'roboethics' [1-2] is used in recent years in reference to the relationship between humans and robots: the elections followed in the design of robots, their use and the treatment given to artificial intelligence systems. Under the term 'roboethics' there is a group of concerns beforehand suggested and new questions that comes to light of the development and contemporary application of robotics.

What time ago was typical of science fiction today has started to become a reality that interacts socially. Now, to the extent that the development of the applied artificial intelligence enables robots to have self-awareness and ability to make decisions, opens a new field for ethical reflection: the inclusion of robots in the field of ethics as agents and not only as instruments.

Everything suggests that this field of ethical reflection will acquire greater importance in the coming years as it intensifies the social presence of the robots and they possess more and more capacity. One issue that has to solve this shift in ethical

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_48,

reflection has to do with what should be the basis in the human-robot relationship: must robotics ethics respond to the parameters of a human-based ethics or can it consider a different ethical base than the human one? The vast majority of people, whom we would ask this question, would clearly answer that we must apply an anthropocentric ethic. However, despite the apparent clarity of the answer, we believe that the question requires a more accurate answer. As an example two qualifications are suggested.

The first one is that nowadays, and as it will be seen later, is already becoming military uses of robots that for a part of the specialists and public opinion are in contradiction with the respect of the creation of important legal and ethical contents like the human rights. This military practice, made possible in part by the development and application of robotics, opens the debate about whether some contemporary Governments are violating some of the ethical and legal principles that they have formally signed at international level.

At the moment to answer the question: what should be done? Deontic ethics set performance standards. However we see some of the acts performed by humans and, what is more important, the events organized by its institutional structures collide with these standards.

The second qualification has to do with a fact that has not happened yet, but if it happens, would modify our conception of ethics focused on the human being. The goal of ethics has been expanded historically ushering other interests worthy of protection and even reviewing its foundations (the case of animalistic ethics). In the foreseeable future, we will have to consider the need to rethink the foundations of our ethical approaches at the moment that other agents like artificial or the result of the development of bio-artificial elements exist, and having self-awareness and ability to make their own decisions. In case of being able to identify these artificial or bio-artificial beings, we have to think and socially agree which are the bases of that ethics. In fact, if progress is made in this process, the same as in other historical periods which have been a major change in the political and ethical reflection will happen: these changes take place from the interaction of paradigmatic events (for example the fact of the high degree of environmental destruction or disappearance of animal species), the transformation of the dominant values, the emergence of new interests socially and politically represented, the modification of lifestyle and consumption or the social use of available technologies. That is, the development of superior capabilities in robots and their presence and social interaction will respond to an accumulative process that will extend over time.

These transverse processes raise already discussions for ethical reflection, and certainly will raise deeper debates. Questions arise enormously suggestive, still nascent: What limits should observe humans in their relationship with robots? Can they have rights and exercise them? Which rights and where they would base themselves? Could robots get to set their own idea, for example, about good life? Undoubtedly, this kind of talk has reminiscent shades of science fiction and films such as *Blade Runner* (Ridley Scott, 1982). We shouldn't forget that this film was based on a 1968 novel (Philip K. Dick, *Do Androids Dream of Electric Sheep?*). The difference is that part of what time ago was read like fiction today has become reality, a reality that is rapidly changing.

The current development of robotics raises another perspective in the ethical reflection: how the use of robots influences in interpersonal relationships. The use of robots in the production process, in personal care activities and services, in games, in educational or military uses, is a novelty in personal interaction. It is not just the question of liability for the use of robots (bugs, defective products, unwanted actions...) but how to condition the use of robots in social relations. For example, it is proposed to develop robots designed to satisfy sexual desires, provide care, to keep company or carry out military operations. To the extent that these applications are extended, it will condition the relationships between people.

Historically, applied ethics have been characterized by expanding their frame of reference to animals, the environment and future generations. Our ethical history can be seen, despite moments of historical regression, as a process of continuous expansion in which humans have reconsidered their relationships with their environment. This extension based on the reformulation, in ethical terms, about the relationship between human beings and other animate and inanimate beings has answered in part to a compelling need of the human being. In case of concern about the environment, what is at stake is the same human well-being and even their future living conditions.

This same ethic expansion has conditioned the Law development that has also seen its borders expanded. Today are usual the environmental protection laws, or the protection of animal species in danger of extinction or existence of duties to animals. There is no doubt therefore that the ethical stance of human beings is not only referred to other human beings, but also to other nonhuman beings.

Ethics applied to robotics has been used by some authors as a way of rethinking the human condition. As if it were a mirror, developing embedded applications in robots (for example, capacities of perception, learning, choice options, programming and verbal body language or facial recognition) suggests thinking on these same processes in humans. And, by comparison, inquire what place occupies these processes in the development of the human condition.

An example of changes that are taking place, and some of the debates questioned, have been revealed in the play *Three Sisters*, original by Chekhov and directed by the Japanese playwright Oriza Hirata and the engineer Hiroshi Ishiguro. Hirata had previously staged plays that pointed in the same direction: *Sayonara* and *I, Worker*. In *Three Sisters* actors and robots interact. Hirata and Ishiguro, creator of Geminoid H1-2 in the Intelligent Robotics Laboratory at Osaka University, have collaborated to develop this project which explores the relationship between human and robot. The same Ishiguro explained time ago that he developed the Geminoid (defined as an android of an existing person controlled by a computer) to understand what is a human being [3].

2 A Question of No Return: The Extension of Liability

The issue of ethics in relation to robotics has long been facing a question: we will call it the question of the Rubicon. The question, with no return from my point of view, is:

Where do we place now the border between humanity and artificial beings with artificial intelligence? It is true that this question contains an uncertainty that could only be answered in terms of its current development, by researchers with fully aware of applied artificial intelligence and, in the future, will be confirmed by the practical robotics applications. However, the question will not be resolved in the field of what should be, but on the ground of being (of what ends up being).

Probably, we may be involved in a dynamic process that leads to the emergence of new forms of existence, or, if we want to be more cautious, it can lead to new forms of existence. When considering this possibility we must clarify the role that ethics can play in relation to that. In this way we must address to the scientific and technical processes, the political-strategic guidelines and the ongoing production and trade procedures: the development of robotics and the multiplication of its applications is unstoppable. It is wrong to think that ethical reflection may be able to stop this development. The reality is that the development of robotics is a phenomenon that conditions the ethical reflection, but I doubt that ethical reflection could mark the development of artificial intelligence. I don't think that ethical reflection can solve the question of whether robots should be able to develop self-awareness, emotions and preferences. This reality, in case it really happens, will force to an ethical reflection. It is even possible it may be imposed to the law; in such a way that once its regulatory activity arises, there is already specific technological reality and emerging issues affecting people. This has happened, for example, with the development of Internet and the tremendous activity that it enables, the biomedical research or, previously, the regulation of polluting activities.

If we accept that science is mainly responsible for what it is, not what it should be [4], ethical reflection applied to robotics will have to start from processes that are already underway and answer to opportunities and interests that are made in other areas of decision and under other objectives than the strictly scientific-technical. Therefore, ethical reflection can even propose criteria and limits to the development of robots, but it cannot be expected to go beyond this. Especially if we consider that ethical reflection is done on a scientific activity that takes place in a context in which, as in other areas of scientific activity, dominates the uncertainty and the probability (as well as the possibility of giving unintended results) and not certainty.

However, it is increasingly less accepted the value-neutrality of science so that they have increasingly incorporated elements of ethical and political reflection to science and its application. Science and technology can no longer be seen as indifferent to good and evil, because they are made of systems of intentional actions. These systems include agents who deliberately seek certain goals based on certain interests. Doing that, they act and apply beliefs, knowledge, values and norms susceptible to moral evaluation [5]. It is therefore possible to ethically evaluate intentions, the interests at stake, the purposes, the results obtained and the methods used.

In an age where ethics seems that replaces political discussion (it would seem that there is more concern for ethics than politics), we must recover the role of political discussion on the issue at hand. It is recognized that, in our societies, political discussion is influenced by ethical perspectives, but once the importance of the

development of robotics is given, the public debate on ethical and political has to be promoted.

The ethical label, which in recent decades has been so widespread in our society (just remember the multiplication of institutes, ethic experts and observatories in charge of ethical studies or development of the ethic business) should not replace the political debate, because it is in the political arena in which they must make decisions that bind the contemporary democratic societies.

In 1979, Hans Jonas published one of the most influential texts in ethical reflection over the development and use of technology: *The Imperative of Responsibility. In Search of an Ethics for the Technological Age* [6]. In this text, Jonas spoke about changes that the world had experienced and how these transformations had changed elements of the common life of human beings. He was worried about the ethical dimension of human performance, which was also a political action to him. The development and social application of technology had set a new kind of power: the capacity to cause effects on the world from scientific and technological joint agents combined with business, financial and state. This author suggested thinking about responsibility from power: the more power you own, the more responsibility you have. And this responsibility needed to be specified in duties [6].

The Declaration on science and the use of scientific knowledge [7] driven in the UNESCO framework, established some principles of action that could be transferred to the political field: 1. ethical standards that have to govern over the scientific activity have to be based on a broad public debate; 2. the free flow of information on all his possible uses and consequences of new discoveries and technologies, so that ethical issues can be discussed appropriately; 3. to promote the dialogue between the scientific community and society. This dialogue should also involve the governments to which the development of artificial intelligence has a strategic value, as well as companies and consortia that feature the development of robotics and its application.

3 The Plurality of Ethical Status

What constitutes the question of the Rubicon is the possible existence of robots whose characteristics we had previously considered too much important to give them a different ethical status. But which ethical status and why we should attribute ethical status to the robots?

It is assumed that on ethical and legal terms, robots are not people [8]. From this perspective it is argued that ethical equality is between persons, so no-persons are in a different situation. The inclusion in the human ethics community for infants and children, the sick, and the mentally or physically disabled happens because they belong to the human species, not for being available or having certain capabilities to develop an specific behavior. But once an ethical community establishes the criteria to identify those who are part of that community, the most interesting thing is to observe in which position are their different members within that community. In other words, which rights and obligations are established for them? If we think that all the recognized members as belonging to an ethical community have the same rights and

obligations, soon we will realize that the reality contradicts that assumption: the combination of factors and the economic, political and cultural interests produce some cases in which is not possible to identify the not-ethical status of the person with his legal status. For example, it is common that people who migrate from one country to another belong to the same ethical community, at least from the perspective of universal ethics. However, in any contemporary State are immigrants recognized with the same rights as nationals. Therefore, the ethical community will not always match the political and legal community.

In the case of primates, although they are not persons, it is understood that in some ethical conceptions, there is a closeness between these (first cousins of Homo sapiens) and humans. Primates are considered to have capabilities that resemble human. This reasoning model could at least be extended to robots and (this is an important issue) those who may have a closer interaction with humans. This interaction is established at different levels. For example, in the case of the play Three Sisters, the performance of an android generates in the viewer the activation of empathic incitements derived from the human figure that would not so easily be generated if the same robot would have given the appearance of a lamppost. At the level of the human-computer relationship, the designs HCI (Human-Computer Interaction) try to improve this interaction [9-10].

When Richard Dawkins [11] explains the appearance of the human being, he asks a question impossible to answer: Who was the first human? The answer is that the humanization process was a gradual process. At some point, probably less than a million years but more than 100,000 years ago, our ancestors were sufficiently different from us, so one current person could not have offspring with them. Something different is how we qualify the *Homo erectus*: as proto-humans, as almost-humans, as ancestors or as distant relatives? This is a matter of how we use the words and the meaning we give to them.

In the case of robots, the development of artificial intelligence could come up in the future the same issue of impossible answer: we speak of a gradual process, a cumulative and multifactorial one. In the development of this process will have great importance the application of adaptive models that provide autonomy to robots. If this pro-future process happens, it will be in connection with the development of symbiotic mechanisms between biological and artificial. Biorobotics is one of the combinations in which one works and whose development can be more innovative. If this development occurs (and to the extent it happens and how it happens), it will intensify a question that remains opened: the symbolic construction of the human and the implications of this construction in the human condition.

Humans tend to intuitively establish naturalized ethical judgments: what we call ethics has played an important adaptive function in our evolution as a species. If we see a person burning a doll, it's possible that this event may trigger a sense of doom, including physical alteration, not for the loss of the materials with which the doll was manufactured, or the pollution that could generate, but fundamentally by intuitive assimilation between the doll and the human being. The doll resembles a human being, so mentally embodied responses are activated and generate a feeling of discomfort.

With robots something similar could happen in those cases in which the robot is given a human form [12]. And probably it would not happen either with the robot that, without having being configured or designed in a way to remember develop skills or characteristics of human beings, has the same capabilities [9]. In this issue, human mechanisms of instinctive recognition of other members of the same species will play an important role. At least until other mechanisms of intellectual, ethical and legal recognition doesn't appear.

4 Robots as Artificial Moral Agents

The Edinburgh School [13] proposed to contemplate science as a social process, which development is affected by a variety of factors: social context and political, economic or ideological factors. These ethical questions and their responses interact in this complex and transverse context. Peter M. Asaro [14] has adopted this view to talk about robots as almost-moral agents. This author, instead of talking about robots without moral, prefers to use expressions that allow him to approach to this new reality. He uses expressions such as: 'moral agency-robots with moral significance', 'robots with moral intelligence' and 'robots with dynamic moral intelligence'.

Wendell Wallach and Colin Allen [15] talk in recognition of this reality about 'Artificial Moral Agents' (AMAs) [16]. The greater autonomy of robots, the more important is the clarification of what the contents of the artificial morality should be. The use of robots, as well as the development of other technological tools that increase the power of human action, and therefore the possible damage as a result of it, increase the responsibility of persons (natural and legal) and not the reduction of their responsibility. Accordingly, we must try to avoid demotivation in the design and implementation of technologies. Similarly we must achieve the realization of these responsibilities. Something different is that it could prove the existence of an intention to discourage the design and implementation of technologies or that trying to carry out these responsibilities would be even more difficult.

Asaro [14] recognizes that these systems cannot yet be classified as 'fully autonomous moral agents'. However, this possibility is implicitly accepted and eagerly awaited, for example, by the heads of major military forces of the world: "The expectation is that robots on the battlefield will form the bulk of detachments, such as infantry units that would be comprised of 150 human soldiers working alongside 2,000 robots" [17]. At the moment, some of the most advanced projects in the military use of robots is the DARPA (The Defense Advanced Research Projects Agency): (http://www.darpa.mil), and in particular, the ARM program (The Autonomous Robotic Manipulation):

- (http://www.darpa.mil/Our_Work/DSO/Programs/Autonomous_Robotic_Manipulation_%28ARM%29.aspx).

In the field of artificial moral have been developed some tests in order to assess the morality of robots. This means, to evaluate the programming and performance of the robots. Wareham [18], for instance, discusses the advantages and disadvantages of

three types of test used for this assessment by comparing the morality of a person and the robot artificial moral: the moral Turing test, the humanity test and the psychological test. Meanwhile, Sullins [19] observes that the artificial intelligence unit can be considered as a moral agent if it meets three features: autonomy, intentionality and responsibility for other moral agents.

What is relevant is to establish the parameters to guide the action and how it acts instead of the humanity or artificiality of the actor. If we study the artificial morality from this perspective, two paradoxes could happen. The first one would be that the artificial morality of robots could become more reliable than the human one. And the second one would be that robots could remember and evaluate humans in relation to the fulfilment of their moral obligations. Undoubtedly, these issues are disturbing.

This approach to the question of morality in robots has the advantage of dynamic perspectives. Moral development is seen as a gradual process, adaptive and socially interconnected. In fact, this has happened in the historical evolution of man as a moral being. This perspective helps to explain how morality in relation to robots is configured in a complex way not only from the development of the programming, but also, for example, from consumption habits and claiming needs by individuals (for example, care or military activities). The arrangement of robots and their development and type of development does not happen in a vacuum, but responds to a process aimed and linked to the uses of these robots. The same happens with the artificial moral development linked to that context.

5 The Military Use of Drones (LARs): An Axiological Question Applied to Robotics

Christof Heyns [20], United Nations special rapporteur on extrajudicial executions, has warned about the use of a new generation of weapon systems: "Lethal Autonomous Robots" (LARs). It was known that some armies used drones remotely piloted for attacks in Yemen, Somalia and on the border between Afghanistan and Pakistan. It was also known that other governments were developing and beginning to use this technology. In addition to the information disseminated by different media, the UN had already documented in 2010 this use of drones.

Philip Alston [21] presented a report to the UN General Assembly in which the practice that some states were developing under the term "targeted killings" was explained. To commit these murders, the different states used drones model "Predator" or "Reaper". These machines were initially designed for military espionage in order to use them in various territories considered as dangerous. Different countries used and use this technology: United States, Israel, Russia, Turkey, China, India, Iran, Great Britain and France among others. Over time, the military uses aimed at getting information, allowed the development of drones with destructive capacity that could be guided in order to kill people or disable targets. As a result, since 2004, 3,587 people have been killed by drones. 884 of these persons were civilians (The Guardian, El País, 2013) [22].

The novelty on which Christof Heyns warned in his document is the following: the new models of drones have the ability to make decisions about their pre-programmed targets. Apparently, it is not necessary the machine being controlled by remote control to achieve the extrajudicial killing of the person, but the same machine has margins of decision. This decision-making comes from the predefined installation on the machine, but also from the self-learning processes that develop these robots. In fact, today is a reality that robots controlled by remote control can kill people and destroy goals. It has been done and it is being done.

This practice raises an important ethical debate: on the ethics of these uses and also about the ethical content of the programming of these robots. For example, a question of ethical content (besides having an important political and legal dimension) is: in case of military intervention of robots, should we contemplate the surrender of the person who is going to be attacked? Is this possibility not considered once its execution has been decided? It is possible that the practical answer to this question goes through a technical question: How can a robot distinguish from an opponent and an opponent who wants to surrender? In which time has the decision be made? In case of no attack, what does the robot with the person who has surrendered? Becomes the opponent a prisoner? How? These questions need to be answered, but depending on the answers (including technology) a mode of military interventions will be being establishing and, consequently, the degree of respect and/or violation of the international law and, therefore, its ethical content. For these reasons, Christof Heyns has called for a moratorium on the use of drones because of the international implications involved, especially when it is used in operations of attack and not strictly defensive purposes.

In case of this destructive use of drones, the ethical, political and legal debate appear linked one another. One of the arguments used to defend this use of drones is avoiding deaths on the army that use this technology, and allows faster actions to be less dependent on assistance from people who have to control their operation. We must recognize, therefore, that there are arguments in favor of the military uses of available technology [17]. Other arguments are: increased national and international security, the development of technologies that can be used for welfare purposes or the accuracy of these military interventions that avoids an increase of military presence on the territory. These arguments can be refuted and considered obviously unacceptable, but not because of it are extinguished values to which they are associated.

The confrontation between these arguments and their opponents is actually a confrontation between different premises, between opposing interests and between disparate objectives. This argument actually advised us to place ourselves, not so much in the realm of absolute judgments, but to considered judgments that are poised in values, interests, practices and objectives. In this ethical openness, the axiology is particularly fruitful. Axiology studies the linkages between values (for example, security vs. human dignity) and also considers the structures that shape these values and the study of the contexts from which these values arises and are applied. For these reasons, axiology helps to analyze complex contexts as are those in which it develops and the ones in which robotics will be developed over the coming decades.

Nevertheless from this opening of ethical evaluation that allows axiology, I understand the ethical notion of 'limit' needs to be preserved to guide the scientific and social application of their results. If in contemporary societies there is a plurality of values (core-natural, epistemic, technical, military, political, economic, social, legal, ecological, aesthetic, religious or moral) the discussion in axiological terms that we could ask is what values predominate at a time and concrete society in the development of research and application of robotics. Echeverría argues that the dominance of military values in a scientific community in a society is one of the biggest axiological risks of the scientific practice: precisely because science and technology increase the capacity of human action. And one of these capabilities is the destructive action. "*Lo que es un bien científico desde una perspectiva militar, porque supone un nuevo instrumento que supera la capacidad de destrucción de los anteriores, puede ser un mal desde la perspectiva de otros subsistemas de valores, y en concreto desde el punto de vista de los valores básicos*" [23].

6 Concluding Remarks

The development of robotics, and their increasing application, involves the expansion of areas of ethical reflection: robots as object of ethical reflection, but also the ethical reflection on robots as agents.

The development of robots emerges historically as a cumulative process, gradual and multifactorial. For these reasons, it is reasonable to think that we are engaged in a complex process that can lead to the emergence of new forms of existence. It is normal to us not being able to have the necessary historical perspective that explains this process and its consequences. This story will possibly not be written by us, but by future generations.

What we have today are rudimentary and partial applications which pose ethical questions. Ethical reflection provides criteria to guide us when we face these problems. Specifically, axiology allows us to be better oriented in the labyrinths in which we are immersed. To the extent that we are witnessing the confrontation between values, we need to analyze them, see how they are related with the contexts in which they arise, and think about in which contexts these values are going to be applied. The weighting between the values at stake offers practical flexibility that is shown as necessary. However, this flexibility must not lead to an extreme relativism that justifies any act, any design or any use.

Ethics, like axiology as part of ethics, provides criteria, but it should not replace the responsibility of political decision. A large proportion of opened issues that have been raised in this article are subject to political decision. The military use of drones, for example, can be analyzed from an ethical perspective, but it cannot be forgotten that their use is based on a government decision. We therefore need to develop an ethics applied to robotics, but an ethics that dialogue with politics. A political ethics to serve and help people to think about robotics: their development and their uses.

References

1. Veruggio, G., Operto, F.: Roboethics: a Bottom-up Interdisciplinary Discourse in the Field of Applied Ethics in Robotics. International Review of Information Ethics 6, 3–8 (2006)
2. Lin, P., Abney, K., Bekey, G.A.: Robot Ethics. The Social and Ethical Implications of Robotic. MIT Press, Cambridge (2012)
3. Ishiguro, H., Nishio, S.: Building artificial humans to understand humans. Journal of Artificial Organs 10(3), 133–142 (2007)
4. Barnes, B.: About Science. Basil Blackwell, Oxford (1985)
5. Olivé, L.: Ética aplicada a las ciencias naturales y la tecnología. In: Ibarra, A., Olivé, L. (eds.) Cuestiones éticas en Ciencia y Tecnología en el Siglo XXI, pp. 181–223. Biblioteca Nueva, Madrid (2003)
6. Jonas, H.: The Imperative of Responsibility. In: Search of an Ethics for the Technological Age, University of Chicago Press (1984); (original edition: Das Prinzip Verantwortung. Versuch einer Ethik für die technologische Zivilisation, Suhrkamp, Frankfurt am Main, 1979)
7. Budapest, Statement (1999), http://www.unesco.org/science/wcs/eng/declaration_e.htm
8. Nagenborg, M., Capurro, R., Weber, J., Pingel, C.: Ethical regulations on robotics in Europe. AI & Society 22, 349–366 (2008)
9. Terrence, F., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots. Robotics and Autonomous Systems 42(3-4), 143–166 (2003)
10. Weng, Y., Chen, C., Sun, C.: Toward the Human–Robot Co-Existence Society: On Safety Intelligence for Next Generation Robots. International Journal of Social Robotics, 2009 1, 267–282 (2009)
11. Dawkins, R.: The magic of reality. How we know what's really true. Bantam Press, London (2011)
12. Nishio, S., Taura, K., Sumioka, H., Ishiguro, H.: Teleoperated android robot as emotion regulation media. International Journal of Social Robotics 5 (2013)
13. Bloor, D., Henry, J.: Scientific Knowledge: A Sociological Analysis, Atholone, Londres (1996)
14. Asaro, P.M.: What Should We Want From a Robot Ethic? International Review of Information Ethics 6, 9–16 (2006)
15. Wallach, W., Allen, C.: Moral Machines. Teaching Robots Right from Wrong. Oxford University Press, Oxford (2009)
16. Allen, C., Wallach, W.: Moral Machines: Contradiction in Terms or Abdication of Human Responsibility? In: Lin, P., Abney, K, eds. (2012)
17. Thurnher, J.S.: No one at the controls. Legal implications of fully autonomous targeting. Joint Force Quarterly 67, 77–84 (2012)
18. Wareham, C.: On the Moral equality of Artificial Agents. International Journal of Technoethics 2(1), 35–42 (2011)
19. Sullins, J.P.: When Is a Robot a Moral Agent? International Review of Information Ethics 6, 23–30 (2006)
20. Heyns, C.: A call for a moratorium on the development and use of lethal autonomous robots (2013), http://www.ohchr.org/EN/NewsEvents/Pages/Acallforamoratoriumonthedevelopmentrobots.aspx

21. Alston, P.: Report of the Special Rapporteur on extrajudicial, summary or arbitrary executions. Human Rights Council to General Assembly, A/HRC/14/24/Add.6 (2010)
22. The Guardian (May 29, 2013) http://www.guardian.co.uk/science/2013/may/29/killer-robots-ban-un-warning
23. Echeverría, J.: Ciencia del bien y el mal, Herder, Barcelona. El País (2007), http://internacional.elpais.com/internacional/2013/05/30/actualidad/1369939836_365608.html (May 30, 2013)

Part IX
Robot Vision

LineSLAM: Visual Real Time Localization Using Lines and UKF

Eduardo Perdices, Luis M. López, and José M. Cañas

Universidad Rey Juan Carlos, Fuenlabrada, Spain
{eperdices,jmplaza}@gsyc.es, luismiguel.lopez@urjc.es
http://jderobot.org

Abstract. In visual simultaneous location and mapping (SLAM) with a single camera, the use of 3D points as a basic feature has been shown sufficient to reliably estimate the camera position and orientation. Nevertheless, the resultant maps are not clear enough for certain applications, even for a large amount of point features. We propose a novel SLAM technique that uses lines as basic features, and the unscented Kalman filter (UKF) as a tracking algorithm. This paper discusses the mathematical foundations as well as the practical implementation of this technique, along with the results of preliminary experiments.

Keywords: Cameras, simultaneous localization and mapping, SLAM, visual localization, line detection, Plücker coordinates, unscented Kalman filter.

1 Introduction

The automatic extraction of relevant information from the image flow of a camera is one of the main challenges of computer vision. One of the main successful techniques in this field are the visual simultaneous localization and mapping (SLAM) algorithms. They provide 3D location of the camera in real time taking as input only the image flow. An important precedent of SLAM is structure from motion (SFM), which has been extensively studied with impressive results over the past three decades [1]. However, the developed approaches usually didn't work in real time since its applications didn't require it. New applications such as augmented reality (AR) have motivated new approaches that focus in real-time algorithms.

Many real-time AR applications with mobile cameras have been used in videogames (e.g. Invizimals^{TM}), in industrial environments (e.g. Seabery's Augmented Training technology for welding) and in medical scenarios. Wearable vision, the new Google glasses and robotics are other scenarios where these algorithms can be applied.

Monocular Simultaneous Localization and Mapping (MonoSLAM) [2] was the first approach achieving good performance working in real time. This approach was based on sequential Bayesian estimation with an extended Kalman filter

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_49,

(EKF). Other approaches aimed at improving the MonoSLAM technique providing a new parameterization to work with features in the infinite [3], adding new techniques to tolerate spurious features [5] or using particle filtering methods to scale better with the number of features [4]. Another issue to take into account before using previous approaches is linearization; even though EKF is able to manage non linear systems, it can be imprecise or unstable when applied on a highly non-linear model. The UKF [6] is an alternative that can cope with these non-linearities; however, it has more computational complexity that the EKF. Novel works [7] state that in this case a square-root UKF can be used, which maintains the same order of magnitude (O^2) as an EKF.

Afterwards, a new kind of algorithm appeared to solve SfM problem in real time. This new approach, called parallel tracking and mapping [8], introduced a new point of view using optimization methods and decoupling mapping and tracking into two different threads, since most applications need to work in real time only for tracking and not for mapping. In [10] both methods MonoSLAM and PTAM were compared and they came to the conclusion that EKF based approaches work better for small feature sets, whereas PTAM based approaches are more suitable elsewhere. Recently, PTAM approach has been improved with new techniques to make it more robust [9].

Most of the state of art explained before is based on point features. Each landmark is modeled as a point in a 3D space; 2D interesting points are detected in each observation (e.g. with a corner detector) and their surrounding small image patches are stored. Recent research suggests that the use of more complex geometrical features such as straight lines can improve the performance of the MonoSLAM algorithms. See [11] and [12] for EKF implementations to use straight lines and points at the same time; and [13] for an analysis of different parameterizations to work with point and line features.

So far, there are no visual SLAM implementations that use lines as a basic feature and make use of the UKF for tracking at the same time. This paper presents a novel algorithm, named LineSLAM, that covers this gap. Hopefully lines would improve accuracy, robustness to blur and fast movements, and help in the construction of more abstract and reusable maps. In addition, the proposed line representation has been carefully designed to facilitate an efficient and powerful management. The use of lines may be the first step towards more complex objects that may improve the localization quality.

Several algorithms to detect straight lines in an image are available. A basic technique [16] is to apply an edge filter on an image and then use a Hough transform to detect lines, but it consumes too much time to work in real time. More efficient and precise implementations are available, [17,18]; the detector chosen for the present work [18] is explained in Section 3.3.

The rest of the paper is organized as follows: Section 2 of the paper describes the proposed line representation and some related operations. In Section 3, the UKF-, line-based SLAM algorithm is detailed, along with the image pressing required. Some preliminary experiments performed with the software implementation of our

visual SLAM are shown in Section 4. Conclusions and future lines in section 5 wrap up the paper.

2 Line Representation

We aim to take advantage of environments that contain straight lines by parameterizing them in both the 3D world and the 2D image plane. This section describes the main mathematical tools (structures and operations) that will be used to model the environment of the camera. In structured real environments, especially indoors, straight lines are ubiquitous as they are formed by the intersection of 2 planes (such as walls) and they are present between corners of artificial objects such as pieces of furniture.

Some of the advantages of using lines instead of (or additionally to) points are: i) lines having homogeneous visual features along themselves are easy to detect and parameterize; ii) the schemes can be adapted to deal with incomplete observations (segments) resulting from occlusions and image-plane cutting; iii) the combination of corners (points) and edges (lines) can help detect and identify objects in the scene; iv) scale- and perspective- invariance characteristics of edge features [14] can be exploited to make the algorithm more robust.

Points in our model are represented using the homogeneous coordinates: the 2D point $[u, v]^T$ corresponds to the equivalence class $\lambda[u, v, 1]^T \ \forall\lambda \neq 0$. A 2D point corresponds to a subspace of a 3D space. Similarly, the 3D point $[x, y, z]^T$ corresponds to the class $\lambda[x, y, z, 1]^T \ \forall\lambda \neq 0$ (subspace of a 4D space).

Representing 2D lines by their slope and origin ordinate renders vertical lines impossible to represent, since their slope equals infinity. To avoid such an ill-posing, a line in the image plane is associated with the equation $ax + by + c = 0$, different choices of $[a, b, c]$ giving rise to different lines. Thus, a line may naturally be represented by the vector subspace $k[a, b, c]^T \ \forall k \neq 0$.

The homogeneous representation of lines and points allows us to write some properties and line-point operations as simple vector operations.

This paragraph addresses planes and lines in 3D (see [1] for a more detailed explanation). Expressing the plane as an equation similarly to the 2D line equation, the vector equation $\boldsymbol{\pi}^T\mathbf{x} = 0$ expresses that the point $\mathbf{x}$ is on the plane defined by the 4-component vector $\boldsymbol{\pi}$. Having this in mind, note that lines in 3D cannot be represented by one scalar equation: at least two equations are required. Moreover, the 3D line can be intuitively represented as the join of 2 separate points or, equivalently, the intersection of 2 non-parallel planes. As a consequence, a line can be represented by a 4×4 skew-symmetric homogeneous matrix. In particular, the line joining the two points $\mathbf{A}, \mathbf{B}$ is represented by the Plücker matrix $\mathbf{L} = \mathbf{A}\mathbf{B}^T - \mathbf{B}\mathbf{A}^T$. The matrix $\mathbf{L}$ is independent of the points $\mathbf{A}, \mathbf{B}$ used to define it and has the required 4 degrees of freedom for a line in 3D. Point and plane transformations defined by matrices can easily be adapted to define a line transformation. A line in Plücker format can be more compactly represented as a 6-entry vector corresponding to the 6 nonzero elements of $\mathbf{L}$, namely $\mathcal{L} = [L_{12}, L_{13}, L_{14}, L_{23}, L_{42}, L_{34}]$. This form is more convenient for the

Kalman filter formulation as using it is equivalent to updating only the relevant components of the original 4×4 Plücker matrix.

2.1 Point, Plane and Line Operations

Some basic operations concerning points, planes and lines are presented next which are relevant to the line-based SLAM (LineSLAM) formulation.

The most important operation associated with the camera model is the observation function. This function is characterized by the 3×4 projective[1] matrix $\mathbf{E}$ that can be computed as $\mathbf{E} = \mathbf{KRT}$, where $\mathbf{K}, \mathbf{R}, \mathbf{T}$ are the projection, rotation and translation matrix. While $\mathbf{R}$ and $\mathbf{T}$ are 4×4 matrices that convert absolute (world) coordinates into camera frame coordinates, $\mathbf{K}$ is the 3×4 projection matrix that converts camera frame coordinates into image plane (2D) coordinates. More precisely, while $\mathbf{R}$ and $\mathbf{T}$ are time variant and depend on the camera state (position and orientation) vector; $\mathbf{K}$ is constant and it depends on the camera intrinsic parameters.

Once we have defined the observation projective matrix $\mathbf{E}$, we describe the basic transformations associated with it.

- Point projection. If we have a point $\mathbf{p}$ in absolute coordinates, its projection on the image plane is $\mathbf{p'} = \mathbf{Ep}$
- Line projection. If we have a line $\boldsymbol{L}$ in 3D absolute coordinates and in Plücker matrix format, its projection on the image plane is $\boldsymbol{L'} = \mathbf{E}\boldsymbol{L}\mathbf{E}^{\mathrm{T}}$. To convert the resulting 3×3 matrix into a homogeneous line vector, we take $\boldsymbol{l} = [L'_{23}, L'_{13}, L'_{21}]^{\mathrm{T}}$.

$$\begin{aligned} \boldsymbol{L'} &= \mathbf{E}\boldsymbol{L}\mathbf{E}^{\mathrm{T}} \\ \boldsymbol{l} &= \begin{bmatrix} L'_{23} & L'_{13} & L'_{21} \end{bmatrix}^{T} \end{aligned} \tag{1}$$

- Finding the φ-plane corresponding to an observed line. When the true position of a line is not available, but a projection of it (from a known camera state) is available, its coordinates cannot be estimated just from one view (observation), but it is possible to find the plane that contains all the lines that could give rise to that view. Since that plane also contains the *focal* point of the camera in the position of the current observation, we call that plane the φ-*plane*. Given the projection matrix $\mathbf{E}$ and the 2D line (the view) $\boldsymbol{l}$, the φ-plane is computed as $\boldsymbol{\varphi} = \mathbf{E}^{\mathrm{T}}\ \boldsymbol{l}$. The proof is straightforward and is omitted for space limitations. This representation is useful because observations of the same line from different viewpoints give rise to a pencil of φ-planes that intersect on the same line. As a consequence, a set of φ-planes that match the same line and their viewpoints have enough parallax can be used to estimate the line vector. This will be further explained in Sect. 2.2.

[1] Note that the projective matrix that maps from 3D to 2D is 3×4 because of the homogeneous coordinates. The same applies to the dimension of $\mathbf{K}, \mathbf{R}$ and $\mathbf{T}$.

2.2 Line Initialization

This section is devoted to the process of initializing newly observed features (lines), which is key to the creation and maintenance of a map from the processed visual information.

Mathematically, this can be defined as robustly estimating an unknown 3D line which is the "rolling axis" of a pencil of planes, from noisy observations (projections) of the planes belonging to such pencil. This is useful because, despite all φ-planes associated with observations of a single line intersect in that line, the observations are usually corrupted with noise and that noise makes the planes not intersect on the same line. Moreover, errors in the visual line matching could give rise to outliers in the set of φ-planes. A measure of the reliability of the estimated line is needed to know when the available φ-plane observations are "different enough" to reliably compute a 3D line.

To estimate the line most likely to give rise to a set of φ-planes, we rely on the singular value decomposition (SVD) method. To do so, we construct an $n \times 4$ matrix $\mathbf{M}$, such that each line of the matrix is a φ-plane in the set, and we factorize it as $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$. Intuitively, the 4×4 matrix $\mathbf{U}$ contains 4 "representative" φ-planes, and the diagonal, 4×4 matrix $\mathbf{\Sigma}$ contains the singular values associated to these representative planes. The estimated line is the intersection of the φ-planes associated with the 2 largest singular values.

In a scenario where the set of φ-planes is not corrupted with noise and there are at least 2 different φ-planes, $\mathbf{M}$ has rank 2, and therefore, the resulting $\mathbf{\Sigma}$ has only 2 nonzero values. In the case of noisy observations, if the different points of view (location of the camera associated with an observation) of the line have enough parallax, then $\mathbf{M}$ will have 2 large SVs (singular values) and 2 SVs close to 0. If there is only 1 large singular value, that means that the observations have not enough parallax and the line cannot be reliably estimated. If there are 3 or 4 large singular values, that means that the amount of noise is too high. In practice we say that the matrix has 2 large singular values when the 2nd SV divided by the 3rd SV is greater than a pre specified threshold value. This threshold has been set to 20 in experimental tests with reasonably good results. This provides a means to decide when to initialize a line, i.e. incorporate it to the set of known lines that are used to refine the estimated camera location.

3 LineSLAM

LineSLAM is a SLAM algorithm that, starting from N_i known (landmark) 3D lines, keeps these lines under track from the image flow of a single camera and creates a map with N 3D lines. It also incorporates the detected lines in the scene to the set of landmark lines. This algorithm is inspired by MonoSLAM approaches, with the novelty of using lines as features instead of isolated points.

Most MonoSLAM approaches deal with their non-linear dynamic models by using an EKF, which basically linearizes the non-linear functions replacing these functions with their Jacobians where needed. Instead of using an EKF, LineSLAM has been implemented with a UKF. The associated dynamic model's

functions need not be linearized; instead, a deterministic sampling technique known as unscented transform is used to generate a reduced set of samples around the mean of statistic distributions.

Figure 1 shows the main steps of the algorithm. At each iteration, a new video frame is obtained and lines are detected from it [cf. Sect. 3.3]. Afterwards, the UKF prediction step is performed, what involves predicting the probability distribution of the current camera state and landmark line vectors. The predicted line observations (resulting from projecting the known landmark lines into the predicted camera image plane) are matched with the 2D lines detected (this steps will be further explained in Sects. 3.2 and 3.4). The last step consists in creating new "candidate" lines from unmatched observations and initialize them as valid 3D lines whenever they fulfill the conditions explained in Sect. 2.2. The newly initialized 3D lines are added to the initial map with the aim of improving the accuracy of the camera localization.

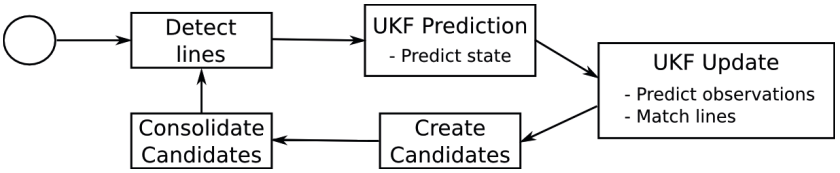


Fig. 1. Main LineSLAM algorithm design

3.1 State and Observation Models

The dynamic model's estimated state vector $\hat{x}$ contains the camera state $\hat{x_c}$ and N line states $\hat{l_i}$ corresponding to the known landmarks. Generally not all known landmarks will be in the camera field of view, so that the state vector size will change dynamically depending on the number of observed lines.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_c \\ \hat{l_1} \\ \hat{l_2} \\ \dots \\ \hat{l_N} \end{bmatrix} \; ; \quad \hat{\mathbf{x}}_c = \begin{bmatrix} r^W \\ v^W \\ q^{WC} \\ \omega^C \end{bmatrix} \; ; \quad \hat{\mathcal{L}}_i = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{bmatrix}$$

The camera state vector $\hat{\mathbf{x}}_c$ is composed of its 3D position r^W, its orientation quaternion q^{WC} and its linear and angular velocities v^W and ω^C. Each line's state is in turn a 6-dimension Plücker vector [cf. Sec. 2]. Coordinates are defined in Fig. 2.

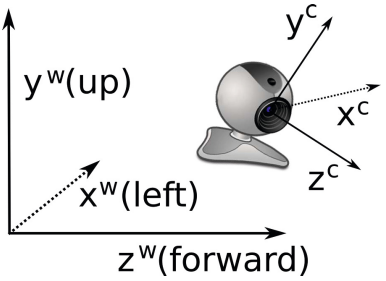


Fig. 2. Camera and world coordinates

The state-transition (movement) model F_t in a time step Δt is constant for each line. The camera transition model F_{t_c} is defined by the following equation:

$$F_{t_c}(\mathbf{x}_c) = \begin{bmatrix} r^W_{|t-1} + v^W_{|t-1}\Delta t \\ v^W_{|t-1} \\ q(\omega^C_{|t-1}\Delta t) \times q^{WC}_{|t-1} \\ \omega^C_{|t-1} \end{bmatrix}$$

The observation vector H_t is formed by concatenating all visible (2D) lines at time t. The observation model associated to this is as follows: for each line $\mathcal{L}_i$ defined by its 6-dimension Plücker vector, a 3D vector H_{t_i} is computed converting the Plücker vector into a $4x4$ matrix:

$$\mathcal{L}_i = \begin{bmatrix} L_{12} & L_{13} & L_{14} & L_{23} & L_{42} & L_{34} \end{bmatrix}^T \Rightarrow \begin{bmatrix} 0 & L_{12} & L_{13} & L_{14} \\ -L_{12} & 0 & L_{23} & -L_{42} \\ -L_{13} & -L_{23} & 0 & L_{34} \\ -L_{14} & L_{42} & -L_{34} & 0 \end{bmatrix} = \mathbf{L}$$

and then projecting the line according to (1).

3.2 UKF Prediction and Update

As it was stated before, the UKF uses a deterministic sampling technique to handle non-linear dynamic equations. For a state x with N_p parameters, whose state prediction is $\hat{x}$ and its covariance P, the UKF needs to calculate $2N_p + 1$ states χ_i (called "sigma" vectors) and its corresponding weights $W_i^{(m)}$ and $W_i^{(c)}$ using the following formulas:

$$\begin{cases} \chi_i = \hat{x} \text{ if } i = 0 \\ \chi_i = \hat{x} + (\sqrt{(N_p + \lambda)P})_i \text{ if } i = 1, ..., N_p \\ \chi_i = \hat{x} - (\sqrt{(N_p + \lambda)P})_i \text{ if } i = N_p + 1, ..., 2N_p \end{cases}$$
$$\begin{cases} W_i^{(m)} = \frac{\lambda}{N_p + \lambda} \text{ if } i = 0 \\ W_i^{(m)} = \frac{1}{2(N_p + \lambda)} \text{ if } i = 1, ..., 2N_p \end{cases}$$
$$\begin{cases} W_i^{(c)} = \frac{\lambda}{N_p + \lambda} + (1 - \alpha^2 + \beta) \text{ if } i = 0 \\ W_i^{(c)} = \frac{1}{2(N_p + \lambda)} \text{ if } i = 1, ..., 2N_p \end{cases}$$

where $\lambda = \alpha^2(N_p + \kappa) - N_p$ is a scale factor, α represents how each sigma will be spread around $\hat{x}$, and κ and β are scale factors that can be tuned depending on the a priori knowledge about the distribution of x.

Once the vectors χ_i and its corresponding weights $W_i^{(m)}$ and $W_i^{(c)}$ have been computed, the Kalman prediction step is as follows:

$$\hat{x}_{t|t-1} = \sum_{i=0}^{2N_p} W_i^{(m)} F(\chi_i)$$
$$P_{t|t-1} = \sum_{i=0}^{2N_p} (W_i^{(c)} (F(\chi_i) - \hat{x}_{t|t-1})(F(\chi_i) - \hat{x}_{t|t-1})^T) + Q_t$$

where $\hat{x}$ is the state vector, P the state covariance matrix and Q the covariance error matrix. The Kalman update step is in turn computed through these equations:

$$Y_t = \sum_{i=0}^{2N_p} W_i^{(m)} H(\chi_i)$$
$$S_t = \sum_{i=0}^{2N_p} (W_i^{(c)} (H(\chi_i) - Y_t)(H(\chi_i) - Y_t)^T) + R_t$$
$$C_t = \sum_{i=0}^{2N_p} (W_i^{(c)} (\chi_i - \hat{x}_{t|t-1})(H(\chi_i) - Y_t)^T)$$
$$\mathcal{K}_t = C_t S_t^{-1}$$
$$\hat{x}_t = \hat{x}_{t|t-1} + \mathcal{K}_t (z_t - Y_t)$$
$$P_t = P_{t|t-1} - \mathcal{K}_t S \mathcal{K}_t^T$$

where Y is the prediction mean vector, S the prediction covariance matrix, C the prediction state cross-covariance matrix, $\mathcal{K}$ the Kalman gain matrix, z the real observation vector and R the observation error matrix.

3.3 Line Detection

Each video frame is analyzed to detect segments of straight lines. We have adapted the approach described by Solis et al. [18], which uses a compilation of different image processing schemes involving normalization, Gaussian smoothing, thresholding, and Laplace edge detection to extract edge contours from input images; then, these contours are labeled according to its orientation and segments are recognized joining consecutive contours with similar orientations. Figures 3 and 4 shows the algorithm results in indoor and outdoor image samples.

Fig. 3. Line detection in an indoor scenario, 36 lines detected in 4.5 ms

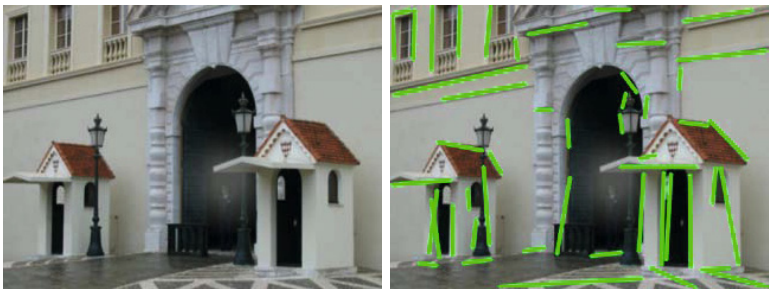


Fig. 4. Line detection in an outdoor scenario, 49 lines detected in 4.9 ms

This solution is far more accurate than other line detection algorithms; it also exhibit robustness and obtains good results in a few milliseconds, what is essential to maintain our algorithm running in real time. The execution time of the algorithms vary from 3 to 6 ms with a 320×240 image resolution depending on the image characteristics.

3.4 Line Matching

While carrying out the UKF update step, the lines stores in MonoSLAM must be matched with the segments detected in the image. To do so, each 2D segment is converted into the general equation form of the line solving the determinant with its extremes ($\mathbf{p}_s$ and $\mathbf{p}_e$):

$$\mathbf{v}_i = (x_i, y_i, z_i) = \begin{vmatrix} i & j & 1.0 \\ p_{s_x} & p_{s_y} & 1.0 \\ p_{e_x} & p_{e_y} & 1.0 \end{vmatrix}$$

These 2D lines are compared with the known 3D lines projecting each 3D line into the image plane to obtain a 2D vector $v_l = (x_l, y_l, z_l)$ [cf. Sec. 2.1]. To match each detected line with the LineSLAM 3D lines we have implemented a method to compare two 2D lines. The first step is projecting the 2D vector into a subset of $\mathbb{R}^3$ with:

$$\mathbf{w} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \Rightarrow \begin{bmatrix} x/\sqrt{x^2+y^2} \\ y/\sqrt{x^2+y^2} \\ z/\sqrt{x^2+y^2} \end{bmatrix}$$

This subspace makes easier to compare distances and angles among 2D lines. Thus, comparison between two 2D lines converted to this subspace (w_i and w_l) is made by the equation:

$$d = \begin{bmatrix} a_i - a_l \\ b_i - b_l \\ c_i - c_l \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} a_i - a_l \\ b_i - b_l \\ c_i - c_l \end{bmatrix}$$

where α balances the importance between line-to-line distance and orientation. Even with this equations, consecutive segments with the same orientation are difficult to match because both segments might have the same general equation form of the line. To rule out these cases we also store representative image patches of the 3D line the first time it is observed and calculate the correlation between theses patches and current segments patches with a zero-mean sum of squared differences algorithm (ZMSSD), a technique that is fast to compute, and invariant to brightness.

To validate a match between lines both d and paths correlation must be below certain thresholds. These thresholds are set depending on image resolution, image quality and current environment.

4 Experiments

We have tested our LineSLAM implementations first in simulated environments with Matlab and afterwards with real observations from one single camera.

4.1 Simulated Data

The main objective of the simulations is to validate the numerical procedures described in the body of this article. Moreover, they are intended to give insights on the impact of different parameters such as noise, number of detected lines, and the distance of the lines to the camera (relative to their length).

The results shown in this section are the results of a Monte Carlo estimation of the mean square error of the camera position and orientation, as a function of different parameters. The methodology of the experiments is as follows: for each tuple of parameters, several sets of random 3D lines are generated. The 3D lines are contained in a unitary sphere and the simulated camera traces a complete circle around the center of that sphere. The position and orientation of the camera is estimated from the noisy observations of those lines. The mean square error is averaged over all the independent runs.

The first set of simulations shows the effect of the noise in the camera localization error for different numbers of detected lines. The radius of the camera

circular trajectory is fixed at 28 cm. The error curves show that the performance of the filter is limited for low levels of noise. Note that the slope of the logarithmic error curve decreases as the noise tends to zero, meaning that it is not possible to arbitrarily improve the quality of the estimation just by reducing the noise in the observations (that would need an unaffordable increase in the camera resolution). The results also suggest that, in order to improve the reliability of the estimated camera's state, the effect of reducing the noise by a factor of 10 is similar to that of increasing the detected lines by a factor of 2. Another conclusion is that a reasonable performance can be obtained by using as less as 4 (non-coplanar) lines.

The second set of simulations shows the effect of the average distance from the camera to the detected lines. As expected, the error increases monotonically with the distance to the observed pattern.

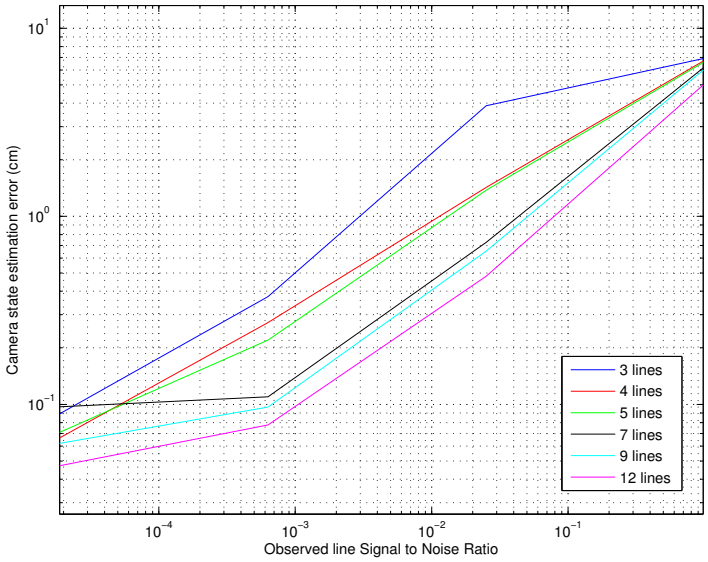


Fig. 5. Error in camera's state estimation vs. observation noise for different number of detected lines

4.2 Experiments with Real Data

We have implemented our LineSLAM approach in C++ to perform experiments with real images in real time. For these real experiments we have used a Logitech QuickCam Pro 9000 camera whose calibration matrix $\mathbf{K}$ is:

$$\mathbf{K} = \begin{bmatrix} 277.2 & 0 & 162.0 & 0 \\ 0 & 274.9 & 121.0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

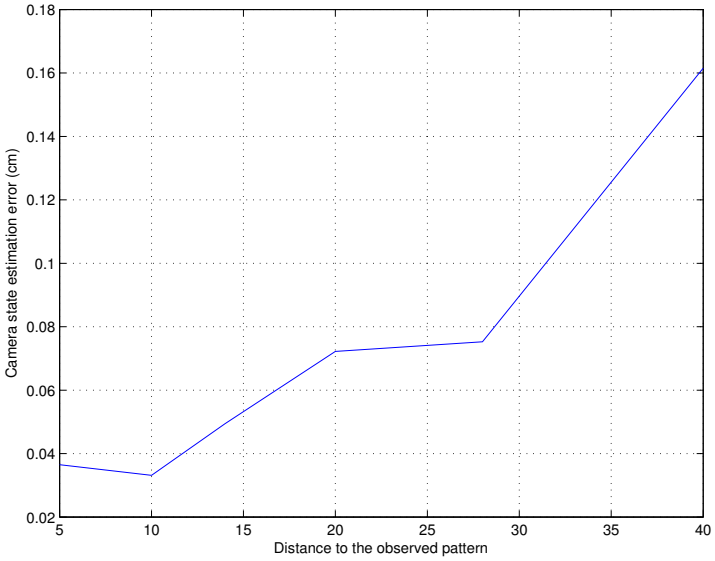


Fig. 6. Error in camera's state estimation vs. average distance to the observed lines

The design of the algorithm is the same than in Matlab but we face new problems such as less accuracy in line detection, what predictably will take to a worse state estimation.

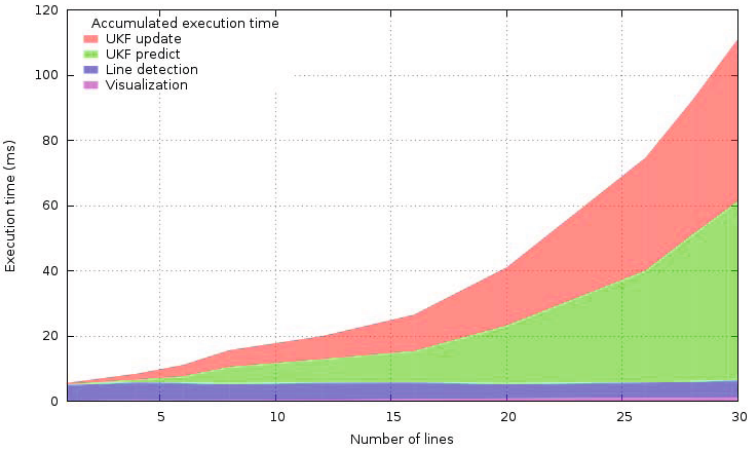


Fig. 7. Execution time vs. number of lines. It includes line detection (3-6ms), visualization (1ms) and UKF prediction and update.

Figure 7 shows how execution time increases with the number of lines N. Since we have used an UKF, whose complexity order is $O(n^3)$, the execution time increases quickly with N, being able to manage only until 18 lines in real time (30FPS). Execution time would improve if we used an EKF, which has a complexity order of $O(n^2)$.

Next experiment (Figs. 8 and 9) demonstrates the creation of new 3D lines with the method explained in Sect. 2.2. The experiment begins with 4 known 3D lines (labelled from 1 to 4), which are matched to their corresponding 2D

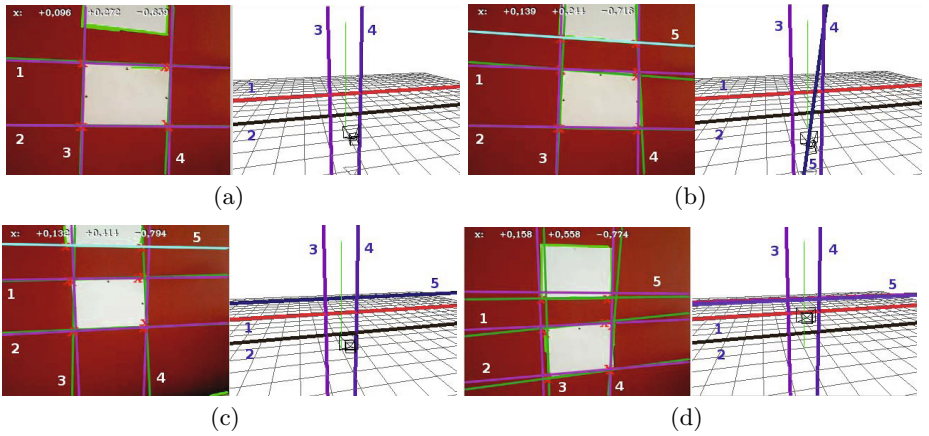


Fig. 8. Left images are real observations which contains detected segments (green), projections of consolidated lines (purple) and candidate lines (cyan). Right images show a 3D simulated environment with current camera position and lines stored.

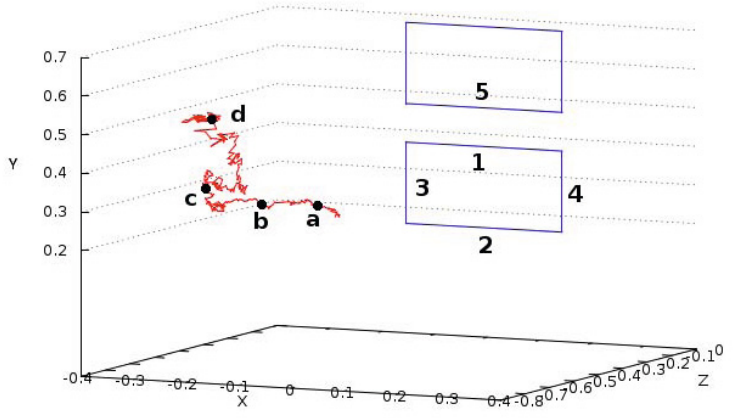


Fig. 9. Camera calculated trajectory (red) and real objects (blue). Labels 1 to 5 correspond to the lines stored in the map whereas labels a,b,c and d are the camera positions which correspond to the observations from Fig. 8.

detected lines and allow the UKF to estimate a reliable camera's state. Then (Fig. 8(b)), a new candidate line in 2D is selected (labelled as 5), but its 3D line is not properly calculated because the camera has not move enough and its parallax is too small to calculate a proper 3D line. After some iterations (Fig. 8(c)) the parallax is big enough and an accurate 3D line is calculated, and therefore the candidate is consolidated as a valid 3D line (Fig. 8(d)). From then on, this consolidated 3D line serves as a reliable and stable 3D line and helps the UKF to estimate its state vector.

5 Conclusions

This article has presented a novel approach to the monocular vision SLAM based on the detection of straight lines and the recursive estimation of the camera state by means of the UKF. The mathematical details of the line modeling, the signal processing (filtering) techniques as well as the image analysis tools related to the detection of lines in the image frame, have been presented, and a novel algorithm implementing all these aspects, LineSLAM, has been developed and experimentally validated.

After testing several different methods, the Solis 2D line [18] algorithm has been chosen as the best available technique for straight line detection. Regarding the filtering technique, the UKF is preferred to the EKF because with UKF does not require to compute the Jacobian matrix for the update step. The problems typically associated with linearization-based approximation are thus avoided.

The representation of lines in 2D and in 3D has been carefully chosen, because the quality of the estimated camera state depends on how they are represented. The Plücker matrix format has been chosen because the functions associated with this representation are expected to work well in conjunction with the family of EKF and UKF filters. Additionally, we have presented a robust algorithm to create new 3D lines from several 2D observations taken from known camera locations. This algorithm is based on SVD and it is the mathematical basis to incorporate 3D lines in the map. This is key to extend the built map to unknown areas, as well as addressing the map maintenance task. A map built in terms of 3D straight lines provides more abstracted information and can be more useful than a map based only on point features.

The performance of the developed algorithms has been assessed by two sets of preliminary experiments. The first one with synthetic data shows the effects of noise and camera-pattern distance in the camera location accuracy. The second one validates the system real-time functioning with a real camera and shows an example of addition of new lines in the map.

The obtained initial results encourage us to extend this work in several directions: i) making the algorithm able to deal with line segments of bounded length, instead of infinite lines; ii) using EKF and square root UKF for filtering instead of UKF to speed up the algorithm, and iii) making a thorough comparison between LineSLAM and point-based visual SLAM in terms of speed, robustness and potential for the closed-loop problem [15]. Two long-term future lines of

work are: i) exploring the use of a wider class of objects (e.g. more abstracted shapes such as cubes or cylinders, widespread objects such as pens and books, or scenario-specific objects such as surgical instruments) as features for visual SLAM, and ii) the use of lines in bundle adjustment approaches like PTAM.

Acknowledgements. This research has been partially sponsored by the Community of Madrid through the RoboCity2030-II project (S2009/DPI-1559).

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004) ISBN: 0521540518
2. Davison, A.J.: Real-time simultaneous localization and mapping with a single camera. In: Proc. International Conference on Computer Vision (2003)
3. Montiel, J., Civera, J., Davison, A.: Unied inverse depth parametrization for monocular slam. In: Proceedings of Robotics: Science and Systems, Philadelphia, USA (August 2006)
4. Eade, E., Drummond, T.: Scalable monocular SLAM. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2006)
5. Civera, J., Grasa, O.G., Davison, A.J., Montiel, J.M.M.: 1-Point RANSAC for EKF Filtering: Application to Real-Time Structure from Motion and Visual Odometry. Journal of Field Robotics (2010)
6. Wan, E.A., Van Der Merwe, R.: The unscented Kalman filter for nonlinear estimation. In: The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000, pp. 153–158. IEEE (2000)
7. Holmes, S., Klein, G., Murray, D.W.: A square root unscented Kalman filter for visual monoSLAM. In: Proc. of the IEEE International Conference on Robotics and Automation (2008)
8. Klein, G., Murray, D.: Parallel Tracking and Mapping for Small AR Workspaces. In: Proc. International Symposium on Mixed and Augmented Reality, ISMAR 2007, Nara (2007)
9. Strasdat, H., Davison, A.J., Montiel, J.M.M., Konolige, K.: Double Window Optimisation for Constant Time Visual SLAM (PDF format). In: ICCV (2011)
10. Strasdat, H., Montiel, J.M.M., Davison, A.J.: Visual SLAM: Why Filter? Image and Vision Computing (2012)
11. Smith, P., Reid, I., Davison, A.J.: Real-Time Monocular SLAM with Straight Lines (PDF format). In: BMVC (2006)
12. Jeong, W.Y., Lee, K.M.: Visual slam with line and corner features. In: Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf., pp. 2570–2575 (2006)
13. Sol, J., Vidal-Calleja, T., Civera, J., Montiel, J.M.M.: Impact of Landmark Parametrization on Monocular EKF-SLAM with Points and Lines. International Journal of Computer Vision 97(3), 339–368 (2012)
14. Mikolajczyk, K., Zisserman, A., Schmid, C.: Shape recognition with edge-based features. In: British Machine Vision Conference, BMVC 2003 (2003)
15. Williams, B., Klein, G., Reid, I.: Automatic Relocalization and Loop Closing for Real-Time Monocular SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(9), 1699–1712 (2011)

16. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM 15(1), 11–15 (1972)
17. Kŏsecká, J., Zhang, W.: Video compass. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 476–490. Springer, Heidelberg (2002)
18. Solis, A., Nayak, A., Stojmenovic, M., Zaguia, N.: Robust Line Extraction Based on Repeated Segment Directions on Image Contours. In: Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, Canada, July 8-10 (2009)

Event-Based Visual Servoing with Features' Prediction

G.J. Garcia, J. Pomares, F. Torres, and P. Gil

Physics, Systems Engineering and Signal Theory Department, University of Alicante, San Vicente del Raspeig, Spain
{gjgg,jpomares,Fernando.Torres,Pablo.Gil}@ua.es

Abstract. Event-based visual servoing is a recently presented approach that performs the positioning of a robot using visual information only when it is required. From the basis of the classical image-based visual servoing control law, the scheme proposed in this paper can reduce the processing time at each loop iteration in some specific conditions. The proposed control method enters in action when an event deactivates the classical image-based controller (i.e. when there is no image available to perform the tracking of the visual features). A virtual camera is then moved through a straight line path towards the desired position. The virtual path used to guide the robot improves the behavior of the previous event-based visual servoing proposal.

Keywords: Visual Servoing, Event-based Control, Event-trigger, Visual Robot Control, Path planning.

1 Introduction

Currently, image-based control is a well-known approach to guide a robot using visual information [3]. These conventional approaches employ frame-based image acquisition and processing technologies in order to continuously obtain the image, extract features and apply the visual servoing controller. In general, the use of this kind of technologies is not computationally efficient because they do not take advantage of the dynamic characteristics of visual scenes. The constant image information processing does not stop even when nothing relevant occurs. In opposite with the previous approach, the event-based visual systems allows to increase the system performance [4]. In this paper, an event-based visual servoing approach is proposed which permits to reduce the image data stream using the event-based control theory [1]. In this case, an event is considered as something that occurs which requires some response. Therefore, it is only necessary to obtain and process image information when an event is generated. Event-based control has been applied to many fields, as in [12] where this strategy is used to control the level of a water tank. A method approached to the evaluation of optical flow using an asynchronous event-based acquisition is developed in [2]. From a pair of event-based cameras, in [11] is described an event-based stereo matching algorithm exploiting the asynchronous visual events. Recently, these cameras have been used in microrobotic applications

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*, Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_50,

[9]. In these works Ni et al. introduce an event-based iterative closest point algorithm to track a microgripper's position at a high rate frequency. These dynamic vision sensors (eDVS) have also been used to track angular 2D coordinates frame to guide and operate in real-time autonomous mobile robots [8].

As it is previously indicated, the visual controller only is applied when an event is detected. Therefore, a behavior must be indicated when an event is not detected in order to continuously guide the robot. To do this, an approach based on virtual visual servoing [10] is proposed. Using virtual visual servoing systems the camera parameters can be estimated iteratively. This is done so that the extracted visual features correspond to the same features computed by the projection of the 3D model according to the current camera parameters. This approach is employed to determine the 3D position of the camera when an event is generated. Therefore, when an event is not obtained the last 3D camera position is known. This information is used to propose a new path planning algorithm which allows the robot guidance in the image virtual space when no real image information is obtained.

The main contribution of the paper is the use of an event-based approach to guide a robot using image information. Furthermore, a path planning algorithm is integrated in order to determine the desired behaviour in the virtual image space when no real image information is obtained. This virtual information is employed in order to guide the robot when events are not generated.

The paper is structured as follows: first, the basics of image-based visual servoing are detailed; the event-based controller using visual servoing is described in Section 3; Section 4 presents an improvement of this event-based visual servoing consisting on features' prediction, Section 5 presents different experiments to validate the proposal; and finally, in Section 6 the main conclusions are discussed.

2 Image-Based Visual Servoing

Image-based visual servoing is based in minimizing the error between current and desired features on the image plane. The image acquired by the camera is the only information needed to obtain that error. The image function describing the task can be represented by $\mathbf{e}_t = \mathbf{s} - \mathbf{s}^*$, where $\mathbf{s}$ is an M x 1 vector containing M visual features corresponding to the current state, while $\mathbf{s}^*$ denotes the visual features values in the desired state.

In order to relate the variations in the image to the variations in the camera, the interaction matrix, $\mathbf{L}_s$, is employed, $\dot{\mathbf{s}} = \mathbf{L}_s \dot{\mathbf{r}}$ [6], where $\dot{\mathbf{r}}$ indicates the camera velocity.

The control law of classical image-based visual servoing is obtained by imposing an exponential decrease of $\mathbf{e}_t$ ($\dot{\mathbf{e}}_t = -\lambda \mathbf{e}_t$):

$$\mathbf{v}_c = -\lambda \hat{\mathbf{L}}_s^{+} \left(\mathbf{s} - \mathbf{s}^* \right) \tag{1}$$

where $\widehat{\mathbf{L}_s^{+}}$ is the pseudoinverse of an approximation of the interaction matrix and λ is the proportional control gain.

Image-based visual servoing can be schematized as in Fig. 1. This scheme shows how the image acquired by the camera located at the robot end-effector closes the control loop. This image provides the new position of the visual features, which are then compared to the reference, consisting in the visual features position at the desired configuration. The error computed in this way is then used in the control law shown in (1). The output of the controller is a robot end-effector velocity in the Cartesian 3D space. This velocity is performed by the robot's internal controller, which computes the joint configuration to assure the desired end-effector velocity.

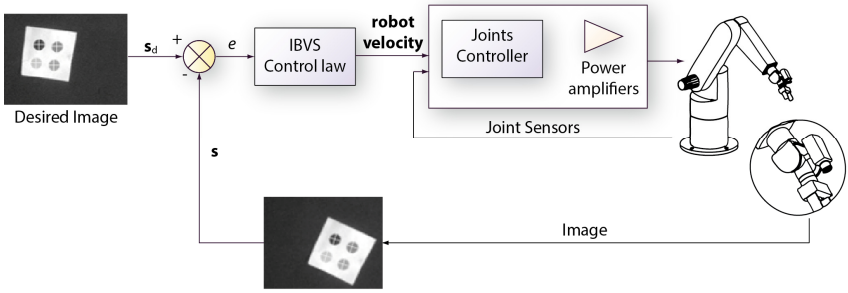


Fig. 1. Classical image-based visual servoing scheme

3 Event-Based Visual Servoing

There are several reasons to modify this scheme in order to improve the image-based visual servoing performance using the event-based control theory. As stated before, an event-based strategy can be employed in order to reduce the transfer of data between camera and controller. Thus, there will be an image transmission only when something relevant occurs (e.g. visual features are near the desired position or in a region where it can be loosen going out the image plane). In addition, large images from modern high-res and high-speed cameras increase considerably the image processing time necessary to segment the objects and obtain the position of the visual features. Finally, events can prevent a visual feature from going out of the field of vision, which may spoil the positioning task.

The event-based visual servoing proposed in [5] is shown in Fig. 2. The main modifications over the scheme of a classical image-based visual servoing are related to the event generator and the event detector blocks.

Event generator is a module that produces an event when any of the visual features enter into a specific region of the image. An event-based camera has been emulated in the researches described in this paper. Thus, image is acquired from a standard camera, but after processing the image in a computer (the cpu is not embedded into the sensor), the Event generator module produces an event every time a visual feature touches any of the predefined image regions. This emulation is required in order to validate the proposed scheme. The image regions are defined so that the visual features cannot leave the field of view. Fig. 3 shows the different regions in the image space and the lines that set the event trigger.

Once the event has been triggered, the Event detector module that can be seen in Fig. 2 must determine the way in which the system will compute the robot's velocity. Fig. 3 depicts the decision scheme of Event detector. If any of the pixel coordinates of the visual features is in the green region marked as ON in Figure 3, the event-based controller activates the classical image-based visual sevoing to compute the robot end-effector velocity. This velocity is then stored in order to be used if the Event detector determines that all the visual features in the red region are marked as OFF in Figure 3. Image-based visual servoing must be activated during the first iteration of the positioning task wherever the visual features are in the image (i.e., even though an event is triggered in this first iteration).

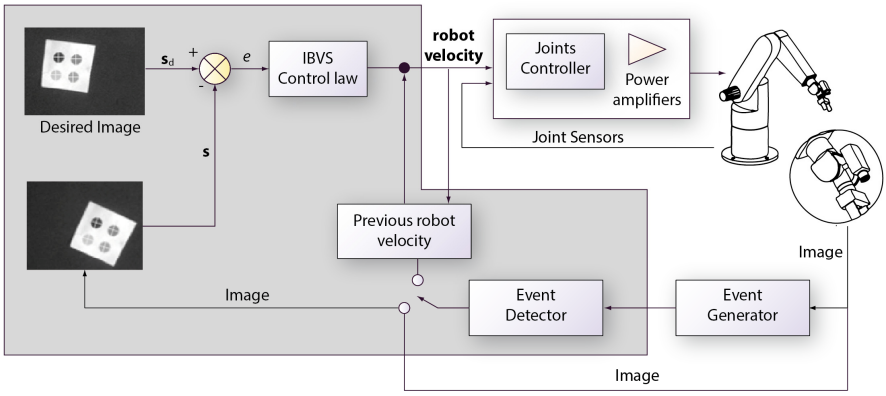


Fig. 2. Event-based visual servoing scheme without features' prediction

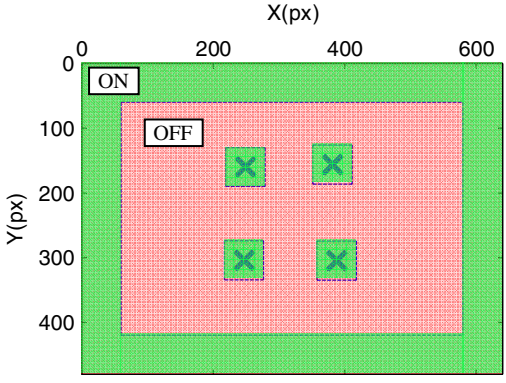


Fig. 3. Event generator regions and Event detector conditions into image space

4 Features' Prediction with Virtual Visual Servoing

The scheme shown in Fig. 2 uses a very simple method to continue guiding the robot when the features enter into the OFF zone depicted in Fig. 3. This scheme proposes to deactivate the image-based visual servoing controller (as in this zone there is no image available) and maintain the last velocity computed in the previous loop iteration. In order to improve the behavior of the controller at this point, in this Section a new path planning algorithm is proposed. This approach is based on virtual visual servoing.

4.1 Virtual Visual Servoing

Virtual visual servoing is based on the classical image-based visual servoing scheme described in Section 2. In virtual visual servoing, a virtual camera is positioned in the scene. Virtual visual servoing moves iteratively this virtual camera in order to obtain the location of the virtual camera from which the object is observed at the same position as in a reference image acquired by the real camera. In order to define the virtual visual servoing systems, the observed features in the reference image are denoted by $\mathbf{p}_d$, and $\mathbf{p}$ are the current positions of the image features projected using the camera intrinsic parameters, ξ, and the current extrinsic parameters, ${}^c\mathbf{M}_o$ (pose of the object frame with respect to the camera frame).

In order to determine the camera extrinsic parameters, it is necessary to minimize iteratively the error, $\mathbf{e}$, between the observed data, $\mathbf{p}_d$, and the position of the same features $\mathbf{p}$ computed using the following equation:

$$\mathbf{p}=pr_{\xi}\left({}^{c}\mathbf{M}_{o}\,{}^{o}\mathbf{P}\right) \tag{2}$$

where ${}^o\mathbf{P}$ are the 3D coordinates (in the object coordinate frame) of the points extracted by the camera and pr_ξ denotes the perspective projection model according to the intrinsic parameters. From the previous defined error, $\mathbf{e}$, it is possible to obtain (the camera intrinsic parameters do not vary. They can be obtained using an off-line calibration [14]):

$$\dot{\mathbf{e}}=\dot{\mathbf{p}}-\dot{\mathbf{p}}_{d}=\frac{\partial\mathbf{p}}{\partial\mathbf{r}}\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{t}} \tag{3}$$

where $\mathbf{r}$ is the camera pose. Equation (3) can be rewritten as:

$$\dot{\mathbf{e}}=\mathbf{L}_{p}\mathbf{v} \tag{4}$$

where $\mathbf{v}$ is the instantaneous virtual camera velocity and $\mathbf{L}_p$ the interaction matrix related to $\mathbf{p}$ [7].

In order to make $\mathbf{e}$ decrease exponentially to 0 ($\dot{\mathbf{e}}=-\lambda_1\mathbf{e}$) the following control law is obtained:

$$\mathbf{v} = -\lambda_l \mathbf{L}_p^{+} \mathbf{e} \tag{5}$$

When $\mathbf{e} = 0$, the extrinsic camera parameters are obtained.

4.2 Features' Prediction

After performing a virtual visual servoing task, the pose of the camera in the Cartesian 3D space is obtained. Through this camera pose and the desired one, a straight line path for the camera can be planned. The tracking of this path assures that the robot arrives at the final desired position. In order to track the path, a simple time-dependent method is proposed:

$$\mathbf{v}_c = -\lambda \hat{\mathbf{L}}_s^{+} \left(\mathbf{s}(t) - \mathbf{s}^{*} \right) \tag{6}$$

The path of the camera is computed in the Cartesian 3D space because computing it in the image space is not possible (a path planned for a particular feature may not be coherent with the path obtained for another). The problem addressed in this section is to determine the 3D location ${}^{c}\mathbf{M}_{ok}$, which represents the extrinsic camera parameters at iteration k of the straight path planned. Fig. 4 summarizes the algorithm proposed to obtain this 3D location.

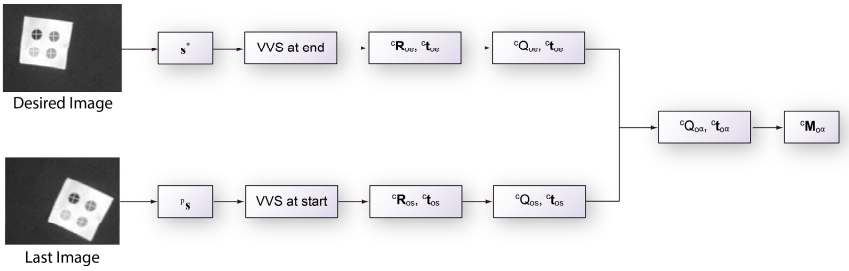


Fig. 4. Scheme of the algorithm to predict the camera pose in the blind zone

The camera extrinsic parameters ${}^{c}\mathbf{M}_{os}$ and ${}^{c}\mathbf{M}_{oe}$ are obtained from the set of features ${}^{p}\mathbf{s}$ and $\mathbf{s}^{*}$ respectively by using virtual visual servoing (see section 4.1). ${}^{p}\mathbf{s}$ is the set of visual features at the last image obtained from the camera. This last image is obtained before the camera launches the event where the features enter the OFF zone depicted in Fig. 3.

The translations ${}^{c}\mathbf{t}_{os}$ and ${}^{c}\mathbf{t}_{oe}$ can be easily extracted from ${}^{c}\mathbf{M}_{os}$ and ${}^{c}\mathbf{M}_{oe}$. A linear interpolation between both translations can be obtained using the following equation:

$${}^{c}\mathbf{t}_{o\alpha} = {}^{c}\mathbf{t}_{oe} + \alpha \cdot \left({}^{c}\mathbf{t}_{os} - {}^{c}\mathbf{t}_{oe} \right) \tag{7}$$

with $0 \le \alpha \le 1$. Furthermore, the rotations ${}^{c}\mathbf{R}_{os}$ and ${}^{c}\mathbf{R}_{oe}$ can also be extracted from ${}^{c}\mathbf{M}_{os}$ and ${}^{c}\mathbf{M}_{oe}$. In order to develop the linear interpolation for the orientation, it is necessary to represent the previous rotations by the quaternions ${}^{c}\mathbf{Q}_{os}$ and ${}^{c}\mathbf{Q}_{oe}$.

Spherical linear interpolation [13] is used to interpolate the orientation, ${}^{c}\mathbf{Q}_{o\alpha}$, between each pair of quaternions:

$$ {}^{c}\mathbf{Q}_{o\alpha} = \frac{{}^{c}\mathbf{Q}_{oe}.\sin\left((1-\alpha)\theta\right) + {}^{c}\mathbf{Q}_{os}.\sin\left(\alpha\theta\right)}{\sin\theta} \tag{8} $$

where θ is the angle between the orientation of the camera in the end position and the orientation in the start pose of the camera obtained by virtual visual servoing through ${}^{p}\mathbf{s}$. After the computation of ${}^{c}\mathbf{Q}_{o\alpha}$, the rotation matrix ${}^{c}\mathbf{R}_{o\alpha}$ is obtained by transforming the quaternion.

When the linear interpolation ${}^{c}\mathbf{M}_{o}(\alpha) = [{}^{c}\mathbf{R}_{o\alpha}\ {}^{c}\mathbf{t}_{o\alpha}]$ is generated, the location ${}^{c}\mathbf{M}_{ok}$ is obtained by iterating over this linear interpolation. In order to achieve a natural behavior of the visual controller this location is achieved as a function of time using an exponential distribution:

$$ \alpha = 1 - e^{-t/\beta} \tag{9} $$

where β is a constant that can be used to increase the slope of the exponential distribution. Exponential distribution represents better the natural performance of a classic image-based visual servoing system than a linear distribution. After computing ${}^{c}\mathbf{M}_{ok}$ from the current value of t, the visual features predicted can be computed from:

$$ \mathbf{s}(t) = \mathrm{s}(\alpha) = pr_{\xi}\left({}^{c}\mathbf{M}_{o}(\alpha)\,{}^{o}\mathbf{P}\right) \tag{10} $$

where ${}^{o}\mathbf{P}$ is the set of object points considered and pr_{ξ} denotes the perspective projection model according to the intrinsic parameters, ξ.

The algorithm proposed in this Section is schematized in Fig. 5. The Event detector described in Section 3 activates the prediction module when there is no image available. The virtual information is then used to improve the controller performance.

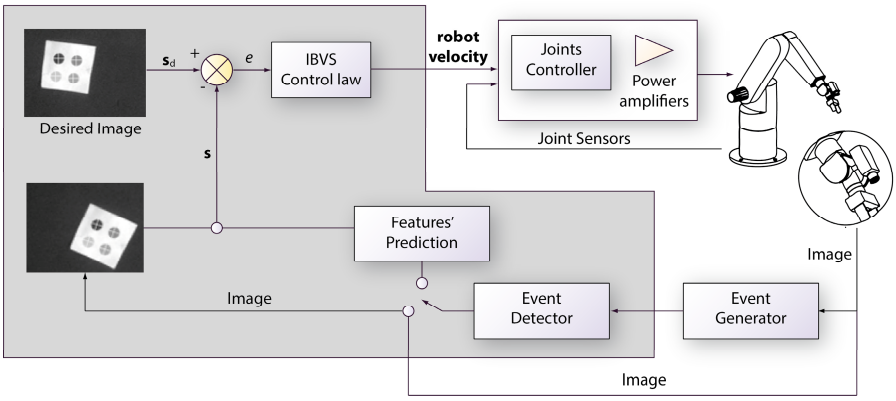


Fig. 5. Event-based visual servoing scheme with features' prediction

5 Results

Two different experiments have been developed using both the event-based visual servoing described in Fig.2 and the new proposal depicted in Fig 3. The testing platform consists in a webcam placed at the end-effector of a Mitsubishi PA-10. The PA-10 is a robot manipulator of 7 degrees of freedom. The webcam employed can acquire images at an image resolution of 640x480 px.

Image processing issues are not the goal of this work, so, in order to guide the robot in these experiments, four visual features, **s,** (centroid points from very easily segmentable circular marks) have been used. The webcam does not allow performing a quick image acquisition. However, image acquisition is not a crucial parameter in the proposed controller validation.

Regions depicted in Fig. 3, can be adjusted to any experiment. For both the two experiments shown in this Section, the external ON region has been defined of 60 px width. In addition, the four internal ON regions are obtained from the image position of the points at the desired robot's pose. The square internal region is centered at the desired visual features location and are 60 px width.

5.1 Experiment without Features' Prediction

The first experiment starts with one of the four points in the external ON region. The set of desired visual features is $\mathbf{s}_d$=[249, 156, 382, 156, 388, 305, 247, 304]. With a gain of λ=0.1, Fig. 6 shows the evolution of the robot's end-effector during the positioning task. The blue cross sets the desired position. The event-based visual servoing proposed in [5] is a valid scheme to position the robot. However, this experiment demonstrates that deactivating visual controller when visual features enter the OFF zone cannot assure direct convergence of the positioning. In this experiment the Event generator triggers six events. The evolution of the end-effector presents three different changes of direction. These changes match the main event triggered during the positioning task.

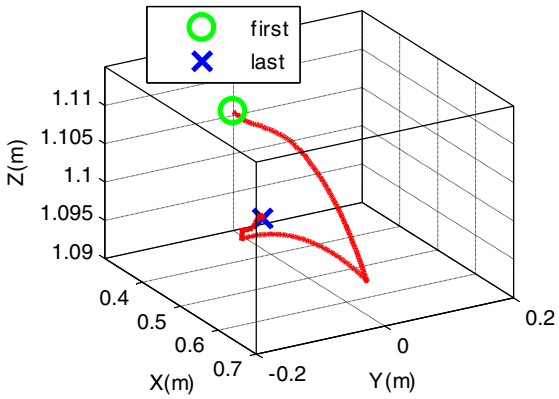


Fig. 6. Experiment without features' prediction. 3D trajectory of the robot end-effector.

Robot evolution in the Cartesian 3D space is not the best measure factor for the proposed controller. Image-based visual servoing systems' input is the image. Therefore, the best measure factor for these kinds of controllers is the evolution of the features in the image. Fig. 7 shows this evolution for this first experiment. The change in the evolution of the visual features corresponds to an event. The test begins with a visual feature in the external ON area. An event is launched when this feature leaves this external region and enters the OFF region (E1). The Event detector deactivates the visual servoing controller and the last velocity is maintained until the next event occurs. Event 2 (E2) is triggered when the bottom right visual features enters the 60 px security bottom area. The Event detector activates the visual servoing controller and a new velocity is sent to the robot. This velocity minimizes the image error, pushing the visual feature away from the bottom edge towards its desired position. The event 3 (E3) is triggered once this point enters again into the OFF region. Although the last velocity computed by the visual servoing controller pushed the visual features towards the desired position, the lack of new images prevents the correction of the velocity to achieve the desired position. Thus, event 4 (E4) is triggered when the top right visual feature enters into the top ON region. A new velocity is computed and the event 5 (E5) deactivates the visual controller. This last velocity guides the features towards the internal square ON areas. The last event (E6) is triggered when one of the visual features enters into its square internal ON area, and after this, the Event detector activates the visual servoing to guide the robot towards the desired position.

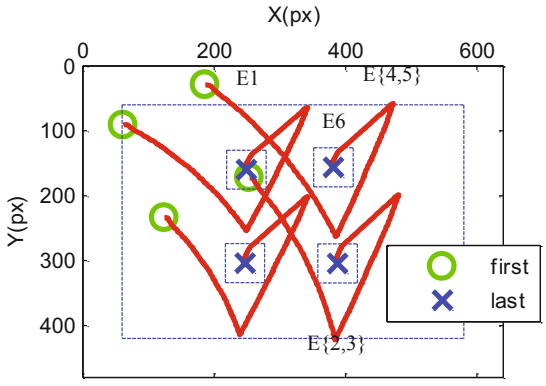


Fig. 7. Experiment without features' prediction. Evolution of the features in the image.

The six events triggered during the experiment deactivate the visual servoing controller three times. During the deactivation (the visual features are all inside the OFF region), the velocity is constant. Figure 8 illustrates the lineal and angular velocity of the end-effector during the task. Figure 9 shows a zoom over time regarding the lineal velocity depicted in Figure 8.a. Over Figure 9, different instants have been shaded in green or red.

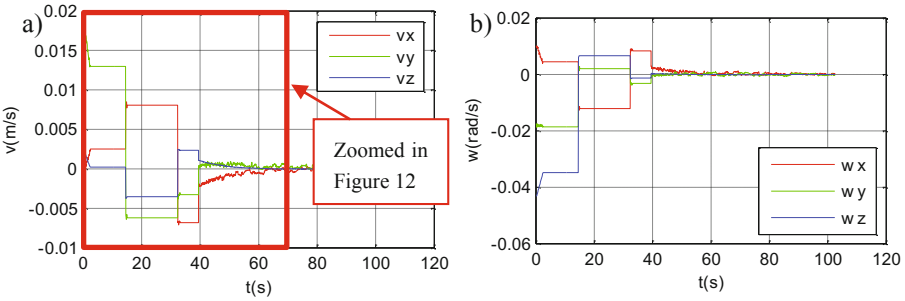


Fig. 8. Experiment without features' prediction. Velocity of the end-effector: a) lineal velocity, b) rotational velocity.

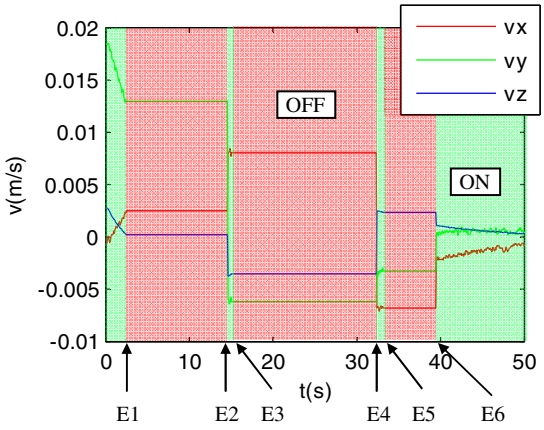


Fig. 9. Experiment without features' prediction. Zoom in lineal velocity of the end-effector. Event detector decisions.

Furthermore, the six events marked in Figure 7 over the evolution of the features in the image have been also marked in Figure 9. A green shaded area represents the lineal velocity sent to the robot when the Event detector activates the visual servoing controller. The velocity in this case presents an exponential decrease towards zero. This is the typical performance of the output of classic image-based visual servoing. Red shaded zones denote the velocity sent to the robot when the Event detector deactivates the visual servoing. When this occurs, the event-based visual servoing sends to the robot the last velocity calculated by the visual servoing. This is the reason why in the red shaded zones there is constant velocity.

5.2 Experiment with Features' Prediction

This test uses the proposed algorithm presented in Section 4 to adjust the velocity of the robot when the visual features enter the OFF zone and, thus, there is no image to compute the current features. The previous experiment has demonstrated the necessity

of computing the velocity in this blind zone. The set of desired visual features is $\mathbf{s}_d$=[249, 162, 381, 156, 389, 303, 247, 310]. With a gain of λ=0.1, Fig. 10 shows the evolution of the robot's end-effector during the positioning task. The camera tracks a straight trajectory when there is no real image to compute real visual features. The experiment shows that the positioning is more robust than the obtained in the previous experiment as the robot converges towards the 60 px width zone of the desired features.

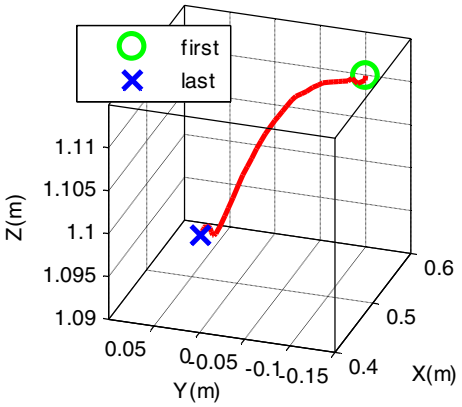


Fig. 10. Experiment with features' prediction. 3D trajectory of the robot end-effector.

Fig. 11 shows the evolution of the features in the image for this experiment. There is not a clear change in the evolution of the visual features. Only two events are triggered in this experiment. The initial position of the robot presents one of the visual features in the external ON area. Thus, the Event detector described in Section 3 activates the visual servoing controller in order to obtain a velocity that approaches the robot to the desired position. In addition, this velocity pushes this visual feature away from the image edges, avoiding thus losing it. The first event (E1) is triggered when the last visual feature leaves the external ON region. The Event detector activates the visual features' prediction, obtaining a straight path from the position of the last image acquired in the previous iteration and the desired camera position. Using this path, future iterations in the blind zone compute different camera poses obtained following the scheme proposed in Section 4. Once the camera pose is obtained the set of visual features is computed by using the intrinsic camera parameters. Finally, with this set of visual features the velocity sent to the robot is computed using the same image-based visual servoing used in the ON zone. This velocity is supplied to the robot. The second event (E2) is launched when one of the features enters its internal square ON area. This square region is centered at the desired feature position. The Event detector activates then the visual servoing controller and a new velocity obtained newly from a real image is sent to the robot. The new velocity sent to the robot attracts it towards the desired position.

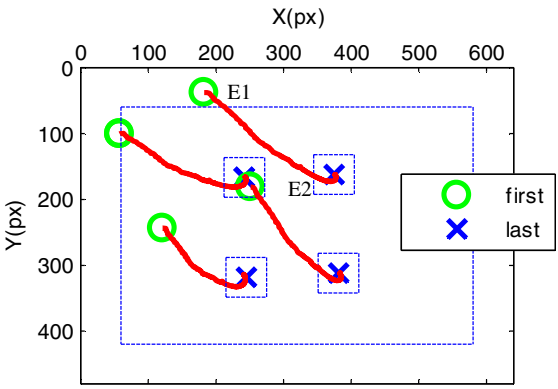


Fig. 11. Experiment with features' prediction. Evolution of the features in the image.

The behavior of the method proposed in this paper to obtain virtual features to guide the robot in the blind zone is very similar to a classical image-based visual servoing scheme. The main advantage of using the proposed scheme is that there is no image to process when visual features enter the blind zone. The time saved during this phase can be spent in any other task.

6 Conclusions

Event-based visual servoing is a control scheme to guide a robot from any initial pose to the desired one using images only when it is necessary. It not only reduces the transfer load between the camera and the processing unit but also reduces the processing time during the visual servoing task. Moreover, this controller prevents visual features from leaving the field of view, which would cause the failure of the task.

The system proposed in this paper improves the robustness of the event-based visual servoing. The Event detector of the previous proposal can only switch between visual servoing and the last visual servoing calculated velocity. In order to avoid situations like that of the first experiment, where visual features bounce more than one time over the external security area, an image position estimator based on virtual visual servoing has been used. Thus, the time saved during the deactivation of the visual servoing can be spent on estimating the set of visual features **s**. This estimation pushes the visual features towards the desired position as if the controller had a real image.

The works presented in this paper opened a new researching topic. After validating the proposed controller using a standard webcam, the next step is the use of an event-based camera in order to quantify the improvement in terms of data transfer load between camera and computer.

The controller proposed allows avoiding outliers using a simple strategy. Events can be used in the future to perform any change in the parameters of the visual servoing controller. An event can order the system to increase the image resolution to

perform a more precise feature tracking, or to decrease the image resolution to save the data transfer bandwidth. An event can also trigger a visual feature change, using points, lines or ellipses depending on the different events.

Acknowledgements. The research leading to these results has received funding from the Spanish Ministry of Education and Science and European FEDER funds, the Valencia Regional Government and the Research and Innovation Vice-president Office of the University of Alicante, through the research projects DPI2012-32390, GV2012/102 and PROMETEO/2013/085, GRE12-17, respectively.

References

1. Aström, K.J.: Event Based Control. In: Analysis and Design of Nonlinear Control Systems. Springer, Heidelberg (2008)
2. Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C., Srinivasan, M.: Asynchronous frameless event-based optical flow. Neural Networks 27, 32–37 (2012)
3. Chaumette, F., Hutchinson, S.A.: Visual servo control. I. Basic approaches. IEEE Robotics & Automation Magazine 13(4), 82–90 (2006)
4. Delbrück, T., Linares-Barranco, B., Culurciello, E., Posch, C.: Activity-driven, event-based vision sensors. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2426–2429 (2010)
5. García, G.J., Pomares, J., Torres, F., Gil, P.: Event-based visual servoing. In: International Conference on Informatics in Control, ICINCO (2013)
6. Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation 12(5), 651–670 (1996)
7. Marchand, E., Chaumette, F.: A new formulation for non-linear camera calibration using VVS. Publication Interne 1366, IRISA Rennes, France (2001)
8. Müller, G.R., Conradt, J.: Self-Calibrating Marker Tracking in 3D with Event-Based Vision Sensors. In: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. (eds.) ICANN 2012, Part I. LNCS, vol. 7552, pp. 313–321. Springer, Heidelberg (2012)
9. Ni, Z., Bolopion, A., Agnus, J., Benosman, R., Regnier, S.: Asynchronous Event-Based Visual Shape Tracking for Stable Haptic Feedback in Microrobotics. IEEE Transactions on Robotics 28(5), 1081–1089 (2012)
10. Pomares, J., Chaumette, F., Torres, F.: Adaptative Visual Servoing by Simultaneous Camera Calibration. In: ICRA 2007, pp. 2811–2816 (2007)
11. Rogister, P., Benosman, R., Ieng, S.-H., Lichtsteiner, P., Delbrück, T.: Asynchronous Event-Based Binocular Stereo Matching. IEEE Transactions on Neural Networks and Learning Systems 23(2), 347–353 (2012)
12. Sánchez, J., Guarnes, M.A., Dormido, S.: On the application of different event-based sampling strategies to the control of a zsimple industrial process. Sensors 9, 6795–6818 (2009)
13. Shoemake, K.: Animating Rotation with Quaternion Curves. Computer Graphics 19(3), 245–254 (1985)
14. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: Proceedings of the International Conference on Computer Vision, vol. 1, pp. 666–673 (1999)

Visual Hybrid SLAM: An Appearance-Based Approach to Loop Closure

Lorenzo Fernández, Luis Payá, Oscar Reinoso, Arturo Gil, and David Valiente

Departamento de Ingeniería de Sistemas Industriales, Miguel Hernández University, Avda. de la Universidad s/n, 03202, Elche (Alicante), Spain
{l.fernandez,lpaya}@umh.es
http://arvc.umh.es

Abstract. This paper proposes an appearance-based method to detect loop closure in visual SLAM (Simultaneous Localization and Mapping). To solve this problem, we make use of omnidirectional images and the internal odometry captured by a robot in a real indoor environment. We build an appearance-based model and, subsequently, two maps of the environment are constructed, one metric and other topological with relationships between them. These relationships are updated in each step of our hybrid approach. The topological map is a graph built from the appearance information in the scenes. A new node is added when the new visual information is different enough from the previous information. At the same time, we check a possible topological loop closure with previous nodes. On the other hand we estimate the metric position of the new pose using a Monte-Carlo approach with the aim of building a metric map. The experimental results demonstrate the reasonable performance of our method.

Keywords: Appearance-base descriptor, Omnidirectional Images, Monte-Carlo SLAM, Hybrid Metric-Topological mapping, Loop closure detection.

1 Introduction

Simultaneous localization and mapping (SLAM) has received significant attention in the past decades as it constitutes a fundamental step when designing an autonomous mobile robot. The SLAM technique considers the situation in which an autonomous vehicle builds up a map within an unknown environment, while simultaneously using this map to estimate its location. We can face the SLAM problem from three different approaches: the Topological SLAM, which represents the position of the robot through a list of states, and the map is composed of a list of locations and the connectivity relationships between them [6]; the Metric SLAM, which includes in the map a set of robot poses with some metric relationships between them and the position of the robot in the

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_51,

map can be estimated with geometrical accuracy at each step [4]; and the Hybrid Metric/Topological SLAM, that combines the computational efficiency of a topological map with the precision and detail of metric implementation [12]. Nowadays, it is usual that SLAM techniques make use of computer vision to build the map and localize the robot. These processes can be approached from two points of view: feature-based approach, where distinctive landmarks (SURF, SIFT) are recognized in the scenes and tracked [13] and appearance-based approaches, where a database containing global image descriptors (PCA, Fourier components, image frequency filtering) is built [8,2]. We present a unified framework to carry out a hybrid metric-topological SLAM. We organize the SLAM problem into a discrete estimation problem (topological SLAM) and a continuous problem (metric SLAM).

An efficient loop closure detection is fundamental in SLAM to build a correct map. It is necessary that the robot decides whether the current location has been visited in a previous step. Two fundamental features to solve this problem efficiently are choosing an adequate method to describe the scenes and optimizing the data association problem [5,15]. The scene representation is one of the most significant problem. In this paper, we use an appearance-based descriptor to describe the scenes captured by the robot at each new pose. Therefore, we carry out the loop closure mechanism when the appearance of the current scene is similar enough to a previously captured scene.

The rest of the paper is organized as follows: Section 2 introduces the method employed to construct the topological graph. In Section 3 a description of the algorithm implemented to built the hybrid map is given. This is then followed by the experimental results in Section 4. Finally, the conclusions are presented in Section 5.

2 Constructing a Topological Graph

2.1 Appearance Descriptor

Since we have decided to work with the global appearance of the images, we need to build a specific descriptor that represents the global information that each image has. The descriptor should retain as much information as possible with a relatively small size, it must be computed quickly and it must be robust against changes in the environmental lighting conditions. In this work we make use of and compare two different image descriptors that are suitable to get a good description of the appearance of the image: Fourier Signature [7] and Histogram of Oriented Gradient (HOG) [1].

2.2 Topological SLAM

In our hybrid topological-metrical mapping approach, the topological map describes the environment using a graph, where each node represents an area of the environment with similar visual appearance (containing one or more images),

and the edges indicate the connectivity relationships between the nodes [9]. To compute the most representative image of each node, we compute the similarity between every two images in the node $S(i,j)$ through the following equations system:

$$S(i,j) = \frac{1}{D(i,j)}$$
$$D(i,j) = \sqrt{\sum_m \sum_n (FS_i(m,n) - FS_j(m,n))^2} \quad (1)$$

Where FS_i is the descriptor of the panoramic image I_i, and m and n are the number of components the descriptor has in rows and columns, respectively. Then, the closer the images are, the smaller the distance D is and the greater the similarity S is. Once we have computed the similarity between every two images in the node, we compute the most representative image using the following equations:

$$R = arg \max_{i \in P}(\min_{j \in P, i \neq j}(S(i,j)))$$
$$N_R = \max_{i \in P}(\min_{j \in P, i \neq j}(S(i,j))) \quad (2)$$

Where P represents the set of nodes in the graph. R is the image that best represents the node and N_R is its minimum similarity factor.

Whenever the robot captures a new image I, it checks whether this image is part of the region that defines the current node using the representative of this node, R. If the similarity S is over a certain threshold (Th), the new image is added to the current node. If the similarity does not exceed the threshold, it is compared with the node representative of all the neighbor nodes to the current one, and the node with the higher similarity is chosen (loop closure). We use a new threshold (Th_{nei}) lower than the previous one in order to give priority to the neighbor nodes. In the case that the similarity does not exceed the neighbor limit, the image is compared to the rest of the node representatives and if it exceeds the minimum threshold (Th) it is added to the corresponding node (loop closure). Finally, in the case that no match is found, a new node is added to the graph, and subsequently an edge between the new node and the previous one is added too. On the other hand, every time we add an image to an existing node, if $Th \leq S(i,j) \leq N_R$, the node representative is re-calculated.

3 Metric SLAM

In the metric solution of SLAM, we estimate the position and orientation of the robot with respect to a reference system with geometrical accuracy. In the proposal that we present in this work we develop a visual SLAM process based in a prior work [14], in which, we developed a visual odometry to estimate the

trajectory followed by a robot by means of a camera sensor solely. In the present work, we use the visual information in combination with the internal odometry of the robot to compute an hybrid metric-topological map of the environment. In addition, we use an approach based on fastSLAM ([8]), but with some important changes. We combine a Monte-Carlo localization (MCL) algorithm with a landmark estimation process, so that, the robot decides when its current location is included as a metric map landmark.

3.1 Monte Carlo Algorithm

We have built a formulation of mobile robot localization as a Monte-Carlo estimation that presents some differences with the traditional formulation. The aim is the estimation of the robot's pose $x_t = (x, y, \theta)$ at time t using a set of measurements $z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ from the environment and the movements $u_{1:t} = \{u_1, u_2, \ldots, u_t\}$ of the robot [3]. In a Monte-Carlo Localization problem [11], the probability density function $p(x_t|z_{1:t}, u_{1:t})$ is represented by a set of M random samples $\chi_t = x_t^i, i = 1 \ldots M$ extracted from it, named particles. Each particle can be understood as a hypothesis of the true state of the robot $x_t^i = (x^i, y^i, \theta^i)$. The set of particles defines a discrete probability function that approximates the continuous belief and the weight of each particle determines the importance of the particle.

The original MCL algorithm consists of three main phases: *Prediction phase*, in which a set of particles $\overline{\chi_t}$ is generated based on the set of particles χ_{t-1} and a control signal u_t; *Update phase*, in which the observation z_t is used to compute the weight w_t^i of each particle in the set $\overline{\chi_t}$ and *Resampling phase* that is used to compute the resulting set χ_t. Finally, the set χ_t represents the distribution $p(x_t|z_{1:t}, u_{1:t})$. In our case (SLAM), as we begin the experiment without a map of the environment, the initial set of particles is represented by a set of samples drawn from a narrow Gaussian centered at the initial point $(x = 0, y = 0)$. When the experiment starts, the first map landmark (l_0) corresponds with the first pose of the robot (X_0) and when the robot moves and captures a new image, the set of particles of the initial pose also moves adding some error to the movement of each particle, taking into account the movements $u_{1:t} = \{u_1, u_2, \ldots, u_t\}$ of the robot.

Moreover, the topological algorithm computes the representative pose of the node every time a new image is added, and it will appear on the metric map as the set of particles that represents the metric localization of the node. Although we compute each set of particles that represents each movement of the robot, we only add to the metric map the set of particles that represents the node. The main difference with respect to the traditional MCL algorithm is that we do not carry out an update and resampling process in each movement of the robot but when the topological algorithm detects a loop closure. On the other hand we perform a prediction phase at each movement of the robot.

We activate the update and resampling process when a metric loop closure is detected. The weight of each particle $w_t^i = p(z_t|x_t^i)$ that represents the current pose of the robot (p_t) is computed using as input both, the metric information

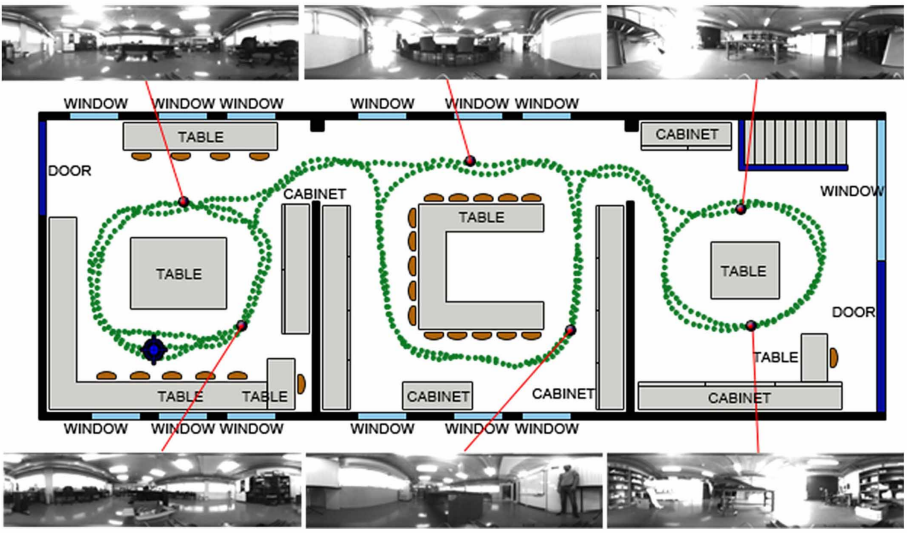


Fig. 1. Bird's eye view of the environment where the experiments have been carried out. This is a realistic and challenging environment with visual aliasing and some people walking into the laboratory. We can see the path followed by the robot to get the necessary data to test the performance of our algorithm (green dots), some examples of the images captured along this path and the initial and final position (big blue dot). Several loop closures are presented.

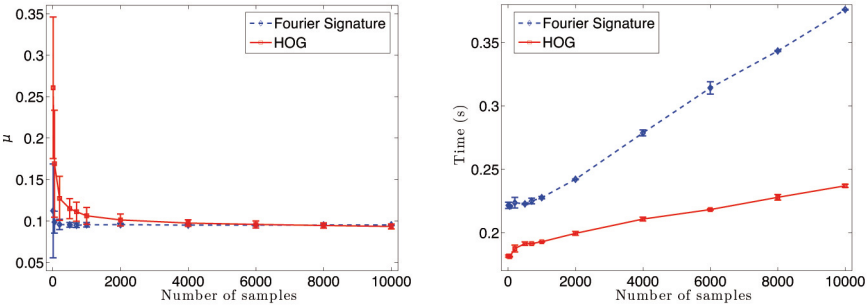
(metric distance) and the visual information (appearance distance) through the following equation:

$$\omega_t^i = \exp\{-v_i \Sigma_l^{-1} v_i^T\} \exp\{-h_j \Sigma_d^{-1} h_j^T\} \tag{3}$$

Where v_i is the difference between the position of the landmark l_j and the position (x_i, y_i) of the particle i ($v_i = (l_x^j, l_y^j) - (x^i, y^i)$). Σ_l is a diagonal matrix $\Sigma_l = diag(\sigma_l^2, \sigma_l^2)$ where the variance σ_l^2 has been choosen in order to minimize the error in the localization of the robot. $h = |d_j - d_t|$ is the difference between the appearance descriptor associated to the current observed image and the descriptor associated to the landmark l_j. The resulting set χ_t is computed by resampling with replacement from the set $\overline{\chi_t}$, where the probability of resampling each particle is proportional to its importance weight w_t^i.

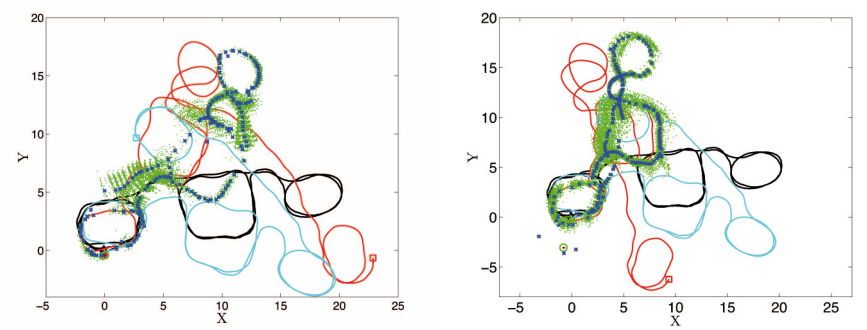
4 Experimental Results

With the purpose of evaluating our hybrid metric-topological SLAM proposal, we performed a realistic experiment with a mobile robot in an laboratory-like environment. First, we have captured the data we need and then we have carried out several sets of experiments varying the main parameters of our algorithm (Figure 1). To evaluate our method we need a mechanism to compute how similar



(a) (b)

Fig. 2. (a) Shape difference μ versus number of samples using the Fourier Signature (blue line) and HOG (red line). (b) Processing step time to build the map versus number of samples using the Fourier Signature (blue line) and HOG (red line).



(a) (b)

Fig. 3. Example of a simulation experiment using a number of particles equal to 200 and (a)the Fourier Signature descriptor and (b) the HOG descriptor. The particles that represent the position of the landmarks are plotted as green dots and the position of these landmark is represented as a blue cross. The particles that represent the current position of the robot are plotted as red points and their position is represented as a green circle. The internal odometry is plotted as a blue line where the blue squares represent the current position of the robot according to the internal odometry. The visual odometry is plotted as a red line where the red square represent the position of the robot according to the visual odometry. Finally, the ground truth is represented with a black line where the black square represent the position of the robot according to the ground truth.

is the layout of the resulting map comparing to the real layout of the captures. As we have begun the experiment without knowledge about the shape of the map, it can occur that the resulting map has a similar shape comparing to the original grid but with a rotation. We have decided to use the Procrustes analysis [10]. Thanks to this analysis we get a measure of how accurate is the layout of

the landmark after the SLAM process, comparing to the real layout. As a result of this process we obtain a parameter $\mu \in [0, 1]$, where μ is a measure of the shape correspondence between the sets of points A and B, so that the lower is μ, the more similar are A and B. In this work we name μ as *shape difference*.

An exhaustive experimentation has been carried out, using as input the data previously acquired in the real scenario. We have tested our approach using the computed visual odometry and the two appearance-based descriptors. We use the visual odometry as the input of the prediction phase in MCL. In the simulations we have tested the influence of the number of particles in the final result of the map. We have used the shape difference μ to evaluate the accuracy of the resulting map (metric). In Figure 2 we can see the μ factor and the step time needed by the algorithm t, versus the number of particles in the Monte Carlo algorithm for both descriptors. It can be seen that in both cases as we increase the number of particles, the shape factor decreases but it reaches a minimum value from which the decrease is hardly noticeable. In the case of Fourier Signature we reach the minimum value of μ equal to $0,096$ using 2000 particles. In the case of HOG we reach the minimum value when we use 4000 particles, μ equal to 0.097. Regarding the time, we can see that the time required for each step of the process increases when the number of particles does. In the case of Fourier Signature, μ takes a lower value than in *HOG* (except for a very high particle number, when μ is slightly higher in Fourier Signature). On the other hand, the time is lower in the case of the HOG although both descriptors allow us to work in real time while maintaining good accuracy using a number of particle about 200. We can see an example of a simulation experiment using a number of particles equal to 200 and both descriptors in Figure 3. We can see that the map obtained by means of our algorithm is more accurate than the map obtained through the internal odometry in both cases. It can be seen that in our map, and when we use the Fourier Signature, the initial and final position of the robot coincides as occurs in the actual map. In the case of HOG, although this position does not coincide, it is closer than in the case of the odometry.

5 Conclusions and Future Work

In this paper, we have presented our own method of visual SLAM. This SLAM approach uses the global appearance of panoramic images in combination with the internal odometry of the robot. A hybrid metric-topological algorithm has been developed to built a metric map, to maintain a topological graph and to detect loop closures. The experiments have been carried out with a set of omnidirectional images captured by an omnidirectional system mounted on the mobile robot under an indoor environment.

In the experiments, we test our hybrid approach when we use the Fourier Signature and HOG to describe the scenes globally. When we tune correctly the parameters of the algorithm, we get a correct map in both cases. We show that we need a minimum number of particles to get an acceptable accuracy in both cases. We present detailed results of an experiments for both methods

when we use a number of particle equal to 200 in both cases. Both experiments results demonstrate the accuracy of our algorithm. The accuracy of the Fourier Signature outperforms the results of HOG but HOG needs a lower computation time. Since both methods allow us to work in real time we reach the conclusion that the Fourier Signature is the best of both methods for our algorithm.

A future work includes the development of a new method to extract more information of the sampling process when we detect a loop closure to get a better accuracy. We will also try to test our algorithm in a large and outdoors environment.

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

References

1. Amorós, F., Payá, L., Reinoso, Ó., Fernández, L., Marín, J.M.: Visual map building and localization with an appearance-based approach - comparisons of techniques to extract information of panoramic images. In: Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2010, pp. 423–426 (2010)
2. Amorós, F., Payá, L., Reinoso, Ó., Jiménez, L.M.: Comparison of global-appearance techniques applied to visual map building and localization. In: Proceedings of the International Conference on Computer Vision Theory and Applications, VISAPP 2012, pp. 395–398 (2012)
3. Fox, D., Burgard, W., Thrun, S.: Markov localization for mobile robots in dynamic environments. Journal of Artificial Intelligence Research (JAIR) 11, 391–427 (1999)
4. Gil, A., Reinoso, O., Ballesta, M., Juliá, M.: Multi-robot visual SLAM using a rao-blackwellized particle filter. Robotics and Autonomous Systems, enviado (2008)
5. Lui, W.L.D., Jarvis, R.: A pure vision-based topological slam system. Int. J. Rob. Res. 31(4), 403–428 (2012)
6. Lui, W., Jarvis, R.: A pure vision-based approach to topological slam. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3784–3791 (2010)
7. Paya, L., Fernandez, L., Reinoso, O., Gil, A., Ubeda, D.: Appearance-based dense maps creation. comparison of compression techniques with panoramic images. In: Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics, pp. 238–246. INSTICC, Milan (2009)
8. Paya, L., Fernandez, L., Gil, A., Reinoso, O.: Map building and monte carlo localization using global appearance of omnidirectional images. Sensors 10(12), 11468–11497 (2010)
9. Romero, A., Cazorla, M.: Topological slam using omnidirectional images: Merging feature detectors and graph-matching. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010, Part I. LNCS, vol. 6474, pp. 464–475. Springer, Heidelberg (2010)
10. Seber, G.: Multivariate observations. Wiley Interscience (1984)

11. Thrun, S., Fox, D., Burgard, W., Dellaert, F.: Robust monte carlo localization for mobile robots. Artificial Intelligence 128(1-2), 99–141 (2000)
12. Tully, S., Kantor, G., Choset, H.: A unified bayesian framework for global localization and slam in hybrid metric/topological maps. Int. J. Rob. Res. 31(3), 271–288 (2012)
13. Valgren, C., Lilienthal, A.: SIFT, SURF and seasons: Long-term outdoor localization ugin local features. In: Proc. of the 3rd European Conference on Mobile Robots (ECMR), Freiburg, Germany (2007)
14. Valiente, D., Fernandez, L., Aparicio, A.G., Castello, L.P., Garcia, O.R.: Visual odometry through appearance and feature-based method with omnidirectional images. Trans. Rob. 26(6), 1051–1064 (2010)
15. Zhiwei, L., Xiang, G., Yanyan, C., Songhao, Z.: A novel loop closure detection method in monocular slam. Intell. Serv. Robot. 6(2), 79–87 (2013)

Multi-robot Formations: One Homography to Rule Them All

Gonzalo López-Nicolás, Miguel Aranda, and Carlos Sagüés

Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain
{gonlopez,marandac,csagues}@unizar.es

Abstract. The problems of convergence to a desired configuration for a set of robots and leader-following in formation are considered in a framework where the robots have nonholonomic constraints, move in a plane and are observed by a calibrated flying camera, which provides the only sensory information used for the control. We propose a homography-based visual control method only requiring a priori single image of the desired configuration to perform the task. The proposed method consists of an image-based control scheme using the homography induced by the multi-robot system. Therefore, an interesting property is that the whole information regarding the multi-robot system is encapsulated in one single homography. The results show that the system is able to track the leader with the robots in formation despite the leader and camera motion are unknown.

1 Introduction

It is well known that some complex tasks cannot be adequately carried out by a single robot or its performance can be greatly improve by using multiple robots. From the variety of problems related with the topic of multi-robot systems, we focus in this paper in the goal of driving a set of robots to a desired configuration while following a leader, which is also part of the formation. A number of research works in this field focus on the problem of reaching and maintaining a robot team in a particular configuration [8] [13] [6].

Vision sensors have been extensively used for robot localization, navigation and control. Visual control is a wide field of research that has attracted the attention of many researchers [4]. In multi-robot systems, it is common to have a setup where each robot is equipped with a local perception system, and they share their information to accomplish the global task. This is the case, for example, of the localization method for multiple mobile robots presented in [5]. Another related work is [13], where groups of mobile robots are controlled to visually maintain formations, including the situation where communication between the robots is not available. The vision-based formation control with feedback-linearization proposed in [8] tackles the issue of switching between decentralized and centralized cooperative control.

Centralized multi-robot control approaches provide several advantages: they allow simple and cheap robots, and release their local resources by transferring expensive computations to an external computer. A centralized architecture is considered for the leader-follower control proposed in [6], where the perception system consists of a fixed camera on the ceiling. In general, vision-based tasks become more robust when multiple

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_52,

view geometry constraints are imposed [11]. Particularly, the homography is a well-known geometric model across two views induced by a plane of the scene, and it has been used extensively in visual control [3], [7], [10].

In the framework considered here, the multiple robots are assumed to move in a planar surface and constrained to nonholonomic motion. The goal of the control scheme proposed is to drive the multiple robots to a desired configuration defined by an image previously taken of that configuration. One of the robots is designated as leader, and performs an arbitrary motion without following the commands of the control law used to maintain the formation. Therefore, the proposed control scheme allows the robots to track the leader in formation. The visual information is acquired by a flying camera looking downward that undergoes an arbitrary planar motion, in such a way that its translation is parallel to the robots motion plane and the rotation is parallel to the plane normal.

We propose a homography-based control approach that takes advantage of the planar motion constraint to parameterize the homography. The image features we employ to compute the homography are the projections of the multiple robots on the image plane. Then, the computed homography gives information about the configuration of the set of robots. In particular, from the homography we can determine if the configuration of the robots is rigid, i.e. they maintain the desired configuration defined by the target image, or nonrigid, meaning that the robots are in a different configuration. We propose an image-based control law where a desired homography is defined as a reference for the control in order to drive the robots to the desired configuration while tracking the leader robot. We can highlight several advantages of our approach. First, any arbitrary target configuration can be defined simply with one image, avoiding the need for information such as 3D measurements or relative positions between the robots. Another advantage is that in our method the camera can move, which allows it to carry out different or additional tasks independently of the control. Moreover, the control performance is not affected by the camera or leader motion, which can therefore be arbitrary.

This paper extends the work presented in [12] and [9]. The idea of homography-based control of a set of robots by using a flying camera was introduced in [12], where the camera performs an arbitrary motion that was constrained to be in a plane parallel to the floor. This constraint was avoided in [9], where the method was generalized and supported with real experiments. Additionally, we tackle the problem of collision avoidance when controlling multiple mobile robots [1], which has to deal with both inter-agent and external collisions. In that work, we use a potential-like collision avoidance method based on gyroscopic forces. We also proposed a purely image-based control strategy that drives the robots while minimizing the sum of the squared distances the robots have to travel [2]. That method features multiple cameras, each of them observing a subset of the robot team, and overcomes the field-of-view limitations of single-camera methods increasing their scalability by exploiting the advantages of both, centralized and distributed multi-robot control strategies. In this paper, we extend and test the proposal for a control law which carries out the multi-robot configuration and leader-following control task. The method is able to lead the robots to the desired configuration while tracking one robot named as leader. The tracking task is hardened because the leader

motion is not known a priori, but the control system handles it properly showing good performance.

The paper is organized as follows. Section 2 presents the parametrization of the homography and the definition of the desired homography for reaching the multi-robot goal configuration. The control law for the multi-robot system is presented in section 3. In Section 4 the performance of the proposal is illustrated in simulation. Finally, the conclusions of the paper are given in Section 5.

2 Homography-Based Scheme

The setup consists of a set of robots undergoing planar motion in the $x-y$ plane. The motion of the camera occurs in a plane parallel to the floor and the rotation of the camera is also parallel to the floor normal ($\mathbf{n}$). The motion of the camera is arbitrary and unknown while holding the previous constraints. One of the robots acts as leader, so that although it still belongs to the formation it also performs an arbitrary and unknown motion. In the following, we parameterize the homography in this framework and propose a procedure to define the desired homography that corresponds to the desired configuration of the multi-robot system.

Two perspective images can be geometrically linked through a plane by a homography $\mathbf{H} \in \mathbb{R}^{3\times 3}$. This projective transformation $\mathbf{H}$ relates points of the plane projected in both images. Pairs of corresponding points $(\mathbf{p}, \mathbf{p}')$ are then related up to scale by $\mathbf{p}' = \mathbf{H}\,\mathbf{p}$. The calibrated homography can be related to camera motion and plane parameters as follows

$$\mathbf{H} = \mathbf{R} + \mathbf{T}\,\mathbf{n}^T/d\,, \tag{1}$$

where $\mathbf{R}$ and $\mathbf{T}$ are the relative rotation and translation of the camera, $\mathbf{n}$ is the unit normal of the plane with respect to the reference frame and d is the distance along $\mathbf{n}$ between the plane and the reference position. In the framework considered, the position of the camera $(x, y, z)^T$ is constrained to the plane $x-y$ (i.e. $z=0$) and rotation ϕ about the z-axis. This constraint yields

$$\mathbf{R} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\,,\quad \mathbf{T} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}\,, \tag{2}$$

with $\mathbf{T} = -\mathbf{R}(x, y, 0)^T$. In our framework, the mobile robots move in a planar surface that generates the homography. Besides, the camera undergoes planar motion: the translation is parallel to the plane and the rotation is parallel to the plane normal, i.e. the z-axis, and $\mathbf{n} = (0, 0, -1)^T$. Notice that the distance d is the height of the camera with respect the motion plane of the robots. Therefore, the homography matrix is

$$\mathbf{H}_{rigid} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \sim \begin{bmatrix} \cos\phi & \sin\phi & -t_x/d \\ -\sin\phi & \cos\phi & -t_y/d \\ 0 & 0 & 1 \end{bmatrix}\,. \tag{3}$$

In our framework, the homography across two views can be computed from a minimal set of two point correspondences solving a linear system. In particular, the points

considered consist of the projection of the robots on the image plane, and each point correspondence gives two independent equations. Given that $\mathbf{H}_{rigid}$ is defined by seven unknown entries, and using the homography constraints $h_{11} = h_{22}$ and $h_{21} = -h_{12}$, a set of two point correspondences allows to determine the homography up to a scale factor by solving a linear system. Given that h_{33} is never zero because of the particular form (3), the scale of the homography can always be normalized and fixed by this entry.

Similarly to the work [2], we wish to compute a 2D Euclidean transformation, but we will derive it from a homography parametrization simpler than (3). This requires a prior translation of the positions in the two sets of points. Here, this translation can be such that the points are expressed with respect to the respective global centroids. These centroids, $\mathbf{c}$ for the sets of points $\mathbf{p}$ of the current formation and $\mathbf{c}'$ for the sets of points of the desired formation $\mathbf{p}'$, are also expressed in the reference frame of robot and we assign

$$\mathbf{p}' := \mathbf{p}' - \mathbf{c}' \ , \quad \mathbf{p} := \mathbf{p} - \mathbf{c} \tag{4}$$

The new coordinates transformed with the centroid are used hereafter (although in the following some comparison is provided with or without this transformation). Then, the two sets of transformed positions $\mathbf{p}'$ and $\mathbf{p}$ are used to compute a homography constrained to have the following form:

$$\mathbf{H}_{nonrigid} = \begin{bmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ 0 & 0 & h_{33} \end{bmatrix} \sim \begin{bmatrix} s\cos\phi & s\sin\phi & 0 \\ -s\sin\phi & s\cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} . \tag{5}$$

Unlike (3), the homography (5) is not Euclidean. However, as shown in [2], it can be computed linearly using SVD and allows to obtain the least-squares Euclidean homography that we are looking for, $\mathbf{H}^d$ as follows:

$$\mathbf{H}^d = \mathbf{H}_{nonrigid}\ diag(1/s,\ 1/s,\ 0) \ . \tag{6}$$

Each pair of robots induce a homography across two images, the current image and the image of the desired configuration. If all of these homographies are equal, the relative motion of the robots is rigid. Otherwise, if any of the homographies is different to the others, the relative motion of the set of robots is not rigid and they are not in the desired configuration. A desired homography computed using all robots needs to be defined in order to lead the robots to the desired configuration. The $\mathbf{H}_{nonrigid}$ relates each point $\mathbf{p}$ of the current image with the corresponding point $\mathbf{p}'$ in the desired formation image with $\mathbf{p}' = \mathbf{H}_{nonrigid}\,\mathbf{p}$. The desired homography $\mathbf{H}^d$ is used now to define the goal location of the robots (except the leader) in the image as

$$\mathbf{p}^d = (\mathbf{H}^d)^{-1}\,\mathbf{p}' + \mathbf{c} \ . \tag{7}$$

The effect of using or not the centroid (4) in the computation of the desired homography is shown in Fig. 3. In this example, nine robots are depicted in an arbitrary configuration. The desired configuration is an octagonal shape with the robots in the contour and the leader in the center. The target location of the octagon is drawn with thin line. Thick lines join each robot with its corresponding goal. The plots of the center

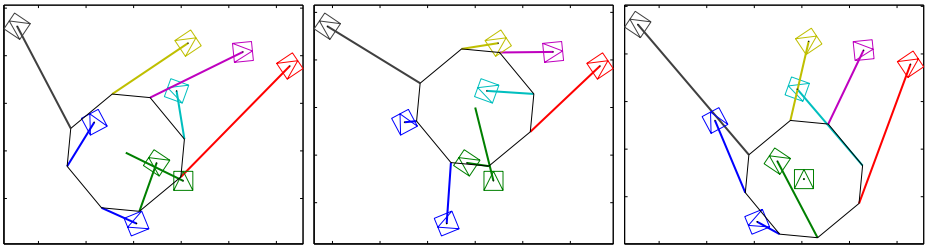


Fig. 1. Comparison of the desired image location for the robots provided by the goal homography in different cases for some instant time. The control law is performed without considering the centroid of the robot distribution (left), whereas it is considered in the next case (center). The last case translates the centroid of the desired robot locations to the leader (right).

and left shows the difference of using or not the centroid in the point target location, respectively. This procedure minimizes the sum of squared distances between the current and desired positions of the robots to reach the desired configuration. It can be seen that considering the centroid the trajectories required to the robots are shorter. Furthermore, one of the robots acts as leader and will not follow the control scheme commands to reach the desired configuration. Thus, instead of using the centroid of the set of robots in (4), we propose to center the coordinates in the image coordinates of leader. In this way, the control law is not requiring any action from the leader to reach the formation. This is illustrated in the right plot of Fig. 3. This procedure improves the control performance as shown in the experimental section.

3 Visual Control Law

From the desired homography computed as explained in the previous section, we propose a control scheme to drive the robots to the desired configuration defined by an image of that configuration. Basically, in each iteration of the control loop, the flying camera takes a current image of the robots, the desired homography $\mathbf{H}^d$ is obtained and used in the control law to compute the robot velocities necessary to reach the desired configuration of the multi-robot system. These velocities are sent to the robots except the leader, which performs its arbitrary motion.

Different coordinate systems defined in the 3D space are depicted in Fig. 2 (left). The state of each robot is given by $(x, y, \phi)^T$, where ϕ is the orientation of the robot expressed as the angle between the robot body y-axis and the world y-axis. Each robot has two velocity inputs, the linear velocity v and angular velocity ω, with v in the direction of the robot y-axis, and ω around the robot z-axis. The kinematics of each robot can be then expressed in general in polar or Cartesian coordinates as

$$\begin{cases} \dot{\rho} = v\,\cos\alpha \\ \dot{\alpha} = \omega - \frac{v}{\rho}\sin\alpha \\ \dot{\phi} = \omega \end{cases}, \quad \text{and} \quad \begin{cases} \dot{x} = -v\,\sin\phi \\ \dot{y} = v\cos\phi \\ \dot{\phi} = \omega \end{cases}, \tag{8}$$

respectively, being $x = -\rho\sin\psi \quad \text{and} \quad y = \rho\cos\psi$. The alignment error α is defined as the angle between the robot body y-axis and the distance vector ρ with $\alpha = \phi - \psi$.

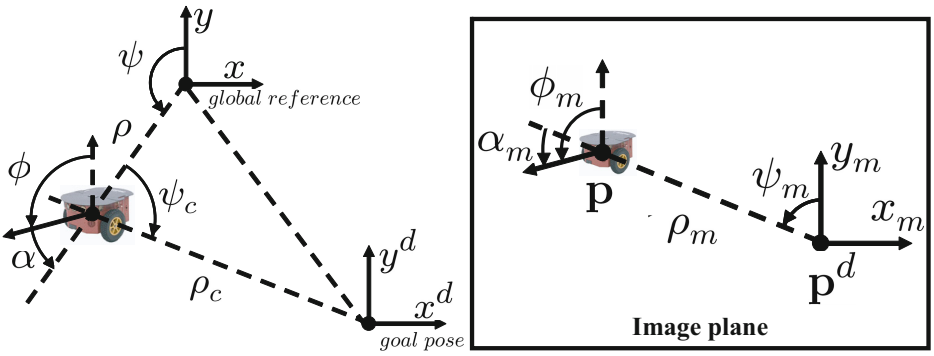


Fig. 2. (Left) Coordinate systems from a top view of the 3D scene. The robot position is given by $(x, y, \phi)^T$ or $(\rho, \alpha, \phi)^T$ in the global reference. (Right) Coordinate systems on the image plane for each robot. Subindex m denotes that the variable is defined on the image plane (the same variable without subindex m refers to the 3D space). Point $\mathbf{p}$ is the image projection of a robot and $\mathbf{p}^d$ its location to reach the desired configuration of the multi-robot system.

We now introduce several variables, depicted in Fig. 2 (right), to define the state of each robot on the image plane with (ρ_m, ψ_m, ϕ_m). The origin of the coordinate system for each robot $\mathbf{p}$ on the image plane is placed in the desired location $\mathbf{p}^d$, i.e. the robots are in the desired configuration when all of them are in the origin of their respective references ($\mathbf{p}^d$). The alignment error on the image α_m is also defined as $\alpha_m = \phi_m - \psi_m$.

Finally, we define the control law in order to bring the robot team to the desired configuration as follows:

$$\begin{cases} v = -k_v\, \rho_m \\ \omega = -k_\omega\, (\alpha_m - \pi) \end{cases}, \tag{9}$$

where $k_v > 0$ and $k_\omega > 0$ are control gains. This control law drives the robots to their image target positions so that the team reaches the desired configuration while tracking the leader. The image projection of the distance to the desired position ρ_m and the alignment error α_m are measured directly in the image plane.

4 Simulation Experiments

The approach presented in this work has been tested is simulation providing good results. In this section, we describe an example in order to illustrate the performance of the control scheme. The setup consists of a set of robots moving on the floor. In particular, nine robots are considered with one of them as the leader. The desired configuration is an octagonal shape with one of the robots, the leader, in the center. This is shown, together with the arbitrary initial configuration of the robots, in the first row of Fig. 3. The camera looking downwards is flying over the scene taking images of the set of robots and, using the image of the desired configuration, the controller computes the desired homography that defines the goal location of the robots in the image plane.

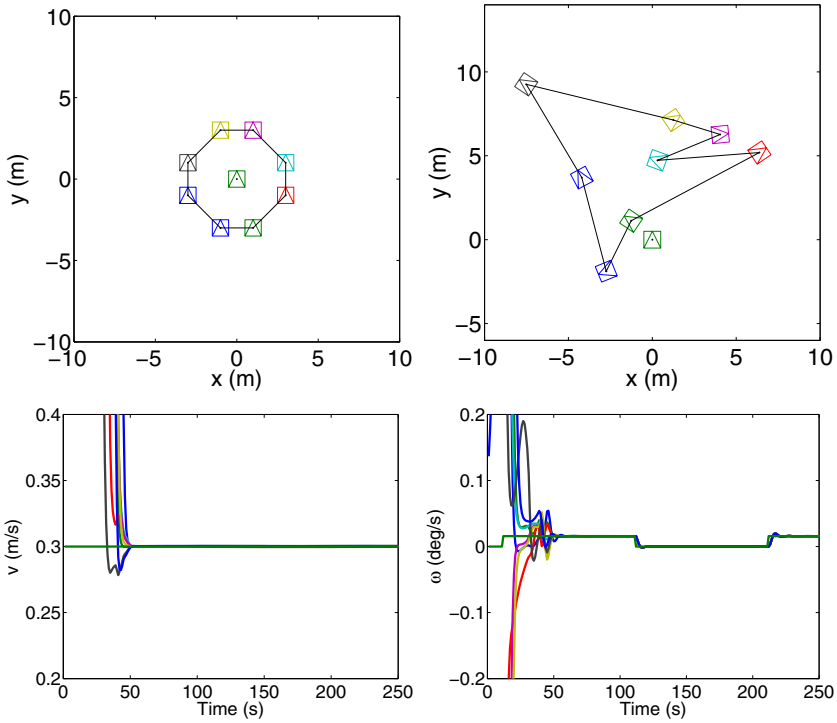


Fig. 3. The desired configuration of the robots and their initial positions for the simulation presented are depicted, respectively, in the first row. The robots are distributed in the desired formation forming an octagonal shape with the leader robot in the centre. The linear and angular velocities given by the controller to follow the leader in formation are depicted in the second row.

Note that the flying camera moves arbitrarily parallel to the floor plane and its motion is unknown for the control system. The velocity commands are sent to the robots, except the leader, to reach the desired formation. The leader does not accept commands and moves arbitrarily being its motion unknown for the control system.

The resultant velocities provided by the control law are shown in the second row of Fig. 3. The leader moves following a rectangular shape with rounded corners. The linear velocity of the leader is $v = 0.3$ m/s and the angular velocity is $\omega = \{0, 0.0157\}$ rad/s. It can be seen in the plots that the robots' velocities converge to the leader's velocities adequately in order to follow it. At the beginning, the velocities are higher as the initial configuration is very different to the desired one. Once the robots are in the formation, the control law tracks the leader velocities while keeping the robots in formation at the same time. A top view of the resultant motion of the set of robots and the flying camera is shown in Fig. 4. In this example, the flying camera follows a circular motion also compounded by sinusoids. The motion of the camera is depicted with black line and its initial location is $(-40, 0)\, m$ with height $50\, m$. The initial location of the robots is around the coordinates $(0, 0)$ and the path followed by the robots during the simulation

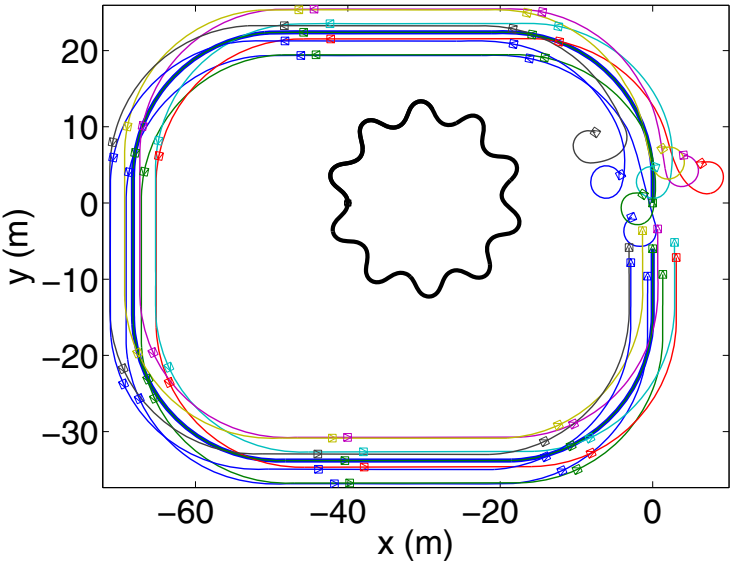


Fig. 4. Top view of the workspace showing the camera and robots motion in a simulation with the flying camera undergoing a circular motion compounded with sinusoids. The trajectory of the camera is depicted with a black line, starting at coordinates $(-40, 0)$. The initial positions of the robots are around $(0, 0)$ and the leader of the formation follows a rectangular shape motion anticlockwise. The path followed by the robots to reach the desired configuration and follow the leader is shown with coloured lines. The robots are depicted in some intermediate instants during the simulation.

is shown with coloured lines. The robots are also depicted in some intermediate instants to show that the correct formation is reached and maintained while tracking the leader.

The evolution of the robots in the image plane is depicted in Fig. 5. In this figure, the initial and final location of the robots is depicted with small coloured circles. Notice that the robots' trajectories in the image plane can be quite complex depending on the coupled motion between the robots and the arbitrary motion of the camera and the leader. Nevertheless, the proposed control law handles it correctly and in a simple way.

In order to evaluate the accuracy of the approach presented, we show in Fig. 6 the evolution of the error in the formation. The distance between each pair of robots in the the actual formation is measured and compared with the corresponding distances in the desired formation. The mean of the absolute difference values are then given as a measurement of the total error in the formation. Two different cases are considered in the error computation. In the first case, an additional simulation has been carried out where the control law is performed without considering the centroid of the set of robots in the computations (first row). In the second case, the centroid in considered and related with the leader robot of the formation (second row), this case corresponds with the simulation presented in this section. Additionally, the left plots in Fig. 6 corresponds to the error distances between robots without considering the leader, whereas the right plots consider the errors distances in where the leader is involved. It can be

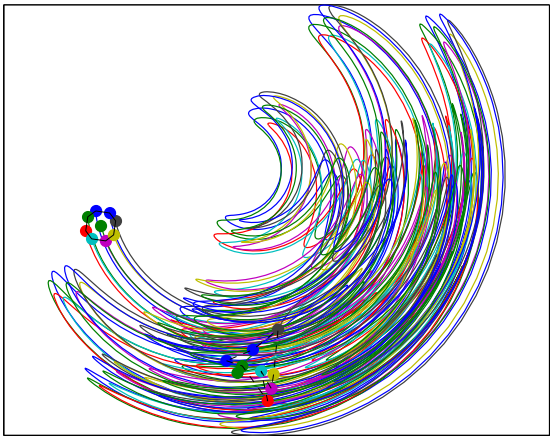


Fig. 5. Trace of the robots in the image plane. The initial and last configuration of the robot formation is depicted with coloured circles. It can be seen that the traces of the robots in the image plane are quite complex because of the coupled motion between the robots and the arbitrary motion of the camera and the leader. Thanks to the homography-based approach, the control law handles correctly the problem independently of the complex trajectories in the image plane.

seen that the controlled robots reach the desired formation without error, independently of considering the centroid or not in the control law computations. The error distance in the formation regarding the leader is different to zero in any case and higher when not considering the centroid of the robots in the control law. Not considering the centroid means that the control law assumes that the leader cooperates in reaching the formation, but actually, the leader follows an arbitrary motion. Then, using the centroid centered in the leader, means that the control law realized that no cooperation comes from the leader and the robots compensate the leader motion to maintain the formation. Thus, reducing the formation error. Note that the motion of the leader is unknown and therefore, the control law cannot guarantee zero error while the leader is moving. The final error can be reduced by increasing the control law gains, but too high gains could reduce the system performance and lead to instability.

Finally, the evolution of the homography entries is depicted in Fig. 7. In particular, entries h_{11}, h_{12}, h_{13}, and h_{23} are depicted given that $h_{22} = h_{11}$ and $h_{21} = -h_{12}$. Again, we separately present the homography entries of the set of robots without considering the leader (left column) and the entries of homographies where the leader is involved (right column). The robots reach the desired formation when the desired homography is finally reached. The results in terms of homography entries is coherent with the evolution of the error formation previously shown in Fig. 6. The homographies generated by the robots except the leader converge to the desired homography, i.e. they reach the desired formation, whereas the homographies that include the leader fluctuate around the desired homography with a small error, denoting that the robots follows the leader in formation with some tracking error. Notice that the final homography does not converge to a constant value given that it evolves because of the arbitrary motions of the camera and the leader of the formation.

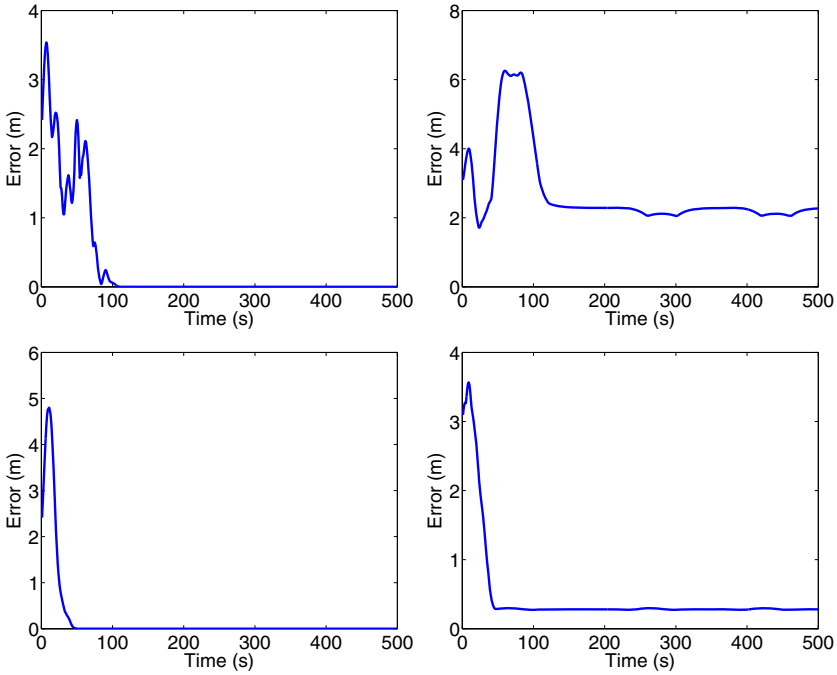


Fig. 6. Evolution of the error between the current formation and the desired formation. The distances between each robot in the current and desired formation are measured and the mean of the absolute error values is depicted. Different cases are tested: In the first row, we do not consider the centroid of the formation in the control law, contrary to the second row, where it is considered. In both cases, Left plot shows the error of all the robots in the formation without including the leader errors, whereas right plots show the errors of the leader with respect the rest of the robots in the formation.

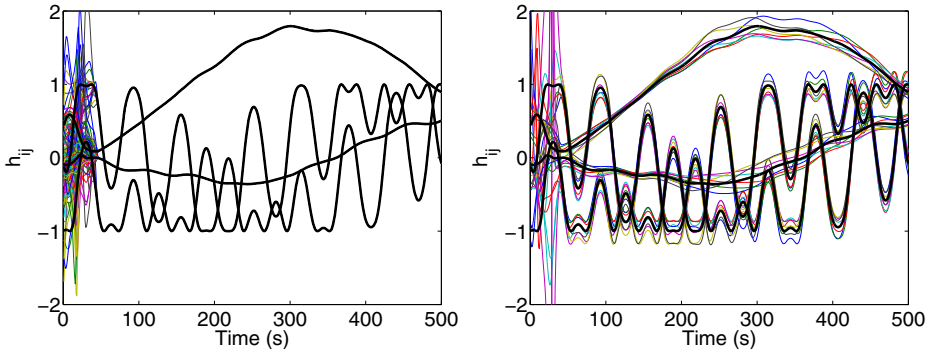


Fig. 7. Evolution of matrix entries of the desired homography (thick lines) and actual homographies (thin lines) between the robots. Entries h_{11}, h_{12}, h_{13}, and h_{23} are depicted. Left column shows the actual homographies except those where the leader robot is involved. Right column shows all the homographies in which the leader is involved. All the robots reach the desired homography with zero error except the leader, which is always moving away from the desired configuration.

5 Conclusion

The proposed control scheme is able to lead a group of robots to a desired configuration and follow the robot leader while maintaining the formation. The key point is that the control law is based on the computation of one single desired homography that allows to define the goal location of the robots in the image plane. The control scheme shows good performance, specially taking into account that neither the camera motion nor the leader trajectory are known or estimated. The final error distances in the formation during the navigation are zero between the robots but not with respect the leader, which presents some final error. This error regarding the tracking of the leader depends on its dynamics and can be reduced by tuning the control gains. However, this error cannot be reduced to zero unless the leader stops or, as classical approaches consider, the motion of the leader is known a priori or estimated online.

Acknowledgment. This work was supported by Ministerio de Economía y Competitividad/European Union (project DPI2012-32100), Ministerio de Educación under FPU grant AP2009-3430, and DGA-FSE (group T04).

References

1. Aranda, M., López-Nicolás, G., Mezouar, Y., Sagüés, C.: One homography to control multiple robots. In: Workshop on Visual Control of Mobile Robots (ViCoMoR) in conjunction with the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 7–12 (2011)
2. Aranda, M., Mezouar, Y., López-Nicolás, G., Sagüés, C.: Partially distributed multirobot control with multiple cameras. In: American Control Conference, pp. 6323–6329 (2013)
3. Blanc, G., Mezouar, Y., Martinet, P.: Indoor navigation of a wheeled mobile robot along visual routes. In: IEEE International Conference on Robotics and Automation, pp. 3365–3370 (April 2005)
4. Chaumette, F., Hutchinson, S.: Visual servo control, part I: Basic approaches. IEEE Robotics and Automation Magazine 13(4), 82–90 (2006)
5. Chen, H., Sun, D., Yang, J.: Global localization of multirobot formations using ceiling vision SLAM strategy. Mechatronics 19(5), 617–628 (2009)
6. Chen, J., Sun, D., Yang, J., Chen, H.: Leader-Follower Formation Control of Multiple Nonholonomic Mobile Robots Incorporating a Receding-horizon Scheme. The International Journal of Robotics Research 29(6), 727–747 (2010)
7. Courbon, J., Mezouar, Y., Martinet, P.: Indoor navigation of a non-holonomic mobile robot using a visual memory. Autonomous Robots 25(3), 253–266 (2008)
8. Das, A.K., Fierro, R., Kumar, V., Ostrowski, J.P., Spletzer, J., Taylor, C.J.: A vision-based formation control framework. IEEE Transactions on Robotics and Automation 18, 813–825 (2002)
9. López-Nicolás, G., Aranda, M., Mezouar, Y., Sagüés, C.: Visual control for multirobot organized rendezvous. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42(4), 1155–1168 (2012)
10. López-Nicolás, G., Gans, N.R., Bhattacharya, S., Guerrero, J.J., Sagüés, C., Hutchinson, S.: Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints. IEEE Transactions on Systems, Man, and Cybernetics, Part B 40(4), 1115–1127 (2010)

11. López-Nicolás, G., Guerrero, J.J., Sagüés, C.: Visual control of vehicles using two-view geometry. Mechatronics 20(2), 315–325 (2010)
12. López-Nicolás, G., Mezouar, Y., Sagüés, C.: Homography-based multi-robot control with a flying camera. In: IEEE International Conference on Robotics and Automation, pp. 4492–4497 (2011)
13. Vidal, R., Shakernia, O., Sastry, S.: Following the flock: Distributed formation control with omnidirectional vision-based motion segmentation and visual servoing. Robotics and Automation Magazine 11(4), 14–20 (2004)

Robust Scale-Invariant Object Tracking

Ahmed Salaheldin, Sara Maher Elkerdawi, and Mohamed ElHelw

Center for Informatics Science, Nile University, Giza, Egypt
ahmed.hussein@nileu.edu.eg

Abstract. Tracking by detection methods are becoming increasingly popular in recent years. They use samples classified in previous frames to detect object in a new frame. These methods have shown successful results. However, due to the self updating nature of this approach, tracking by detection methods usually suffer from object drift. Inaccurately detected samples are added to the training set which degrades the performance. Another problem is that the object may change in shape and size which increases the potential for inaccurate detection and subsequently the chance of losing the object. We propose a robust object tracking algorithm that adapts to changes in the size of the object. The algorithm is divided into two steps. The first step uses random projections compressed using a sparse matrix to describe positive and negative samples generated around the object. These samples are used to train a classifier to detect the position of the object in the next frame. The second step searches for the best size to fit the object around the position chosen in the first step. Experimental results show that our method provides robust tracking and help alleviate the drift problems.

1 Introduction

Object tracking is an important problem in many applications such as robotics, surveillance, augmentation reality and traffic control. Many methods are used to address object tracking in videos such as template based tracking, shape based methods, mean shift methods, flow based tracking, etc. Recently tracking by detection is gaining popularity. These methods use the current position of the target to generate positive samples with a small degree of variation in position. Negative samples are generated from the background surrounding the object. These samples are then used to solve a classification problem in the next frame. A classifier trained with the generated samples is used to label patches in the scene as foreground or background. The new position of the object is chosen from the patches labeled as foreground. This process is repeated for each frame. Tracking by detection has demonstrated success in many applications [1-6].

There might be a conflict between the underlying paragraph style and some individual character and paragraph formatting.

However, since this approach is based on self learning (i.e. uses its own predictions for training), tracking by detection is prone to drift. False positives used to update the classifier cause the detection to degrade. Moreover, even if the samples are correctly

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_53,

classified, the sample may not fit the object tightly. In this case, part of the tracking bounding box is occupied by background. Over a few iterations of self updating, the tracker is likely to drift towards the background and loose the object. This problem is aggravated if the size of the object varies significantly due to its movement in relation to the camera. If the object is scaled down, a greater are of the bounding box will contain background. This increases the chance of the tracker drifting to the background and loosing the object. On the other hand if the size of the object increases, the tracker would only fit part of the object. In this case the tracker cannot utilize all the details of the object which decreases its chances of accurate detection. If the increase in size is substantial, parts of the object surrounding the bounding box may be counted as negative samples, further compromising the performance of the classifier.

The problem of scale variations in tracking has been tackled in the literature, mainly using mean shift methods [7-9]. Yilmaz [9] used a mean shift method with an asymmetric kernel to estimate the scale and orientation of the object. The scale and orientation are added as two more features to the x and y position of the object. The mean shift for all dimensions is evaluated simultaneously. This approach produced promising results. However, due to the constancy of the kernel shape, this tracker works best if the shape of the object is unaltered. In most real applications the shape of the object often changes due to deformation, out of plane rotation or camera movement; which would affect the performance of the tracker.

Babenko et al [10] used a tracking by detection method. When generating positive and negative samples for training, the samples are of the same size as the tracking bounding box. However, the patches to be labeled by the classifier may be larger or smaller than the bounding box. This approach has its advantages and disadvantages. On one hand, patches of varying sizes may be better aligned with object, thus improving classification and providing tighter tracking. On the other hand adding a scale parameter to the patches increases the search space, so there is more room for error, unless the number of patches to be detected is increased which is time consuming. So adding the scale parameter improves tracking in some cases and degrades it in others [10].

In this paper we propose a robust two-step approach to tracking by detection. We use a discriminative model to separate the object from the background using compressed features. Random projections have been used recently in tracking by detection [11]. Studies in compressed sensing have shown that a small number of randomly drawn linear measurements can keep most of the information in visual images [12], [13]. For the first step we use information about the object to generate positive and negative samples and random projections are used to project them into a low dimension feature space. A classifier trained with these samples is used to label patches of the same dimensions as the training samples. The best candidate of the positively labeled is then selected as the current position of the object. In the second step patches of different size are generated close to the position chosen in the first step. The classifier is used to separate the background from the fore ground and the positive sample most similar to the original target is chosen as the new object. This

prevents the tracker from drifting due to self updating. We demonstrate that our method provides robust tracking for objects with scale variations and decreases the chance of drifting.

This paper is organized as follows: In the next section we describe the random projections used to generate features for the training and testing samples. Section 3 describes our proposed scale-invariant approach. We present our experiments and results in section 4 and conclude the paper in section 5.

2 Random Projections

Random projections are a method of projecting a high dimensional space to a lower dimensional space while preserving most of the information in the original signal. Bingham et al [14] show that using random projections for dimensionality reduction is favorable to principal component analysis (PCA) in image and text applications. The theory relies on the Johnson-Lindenstrauss lemma[15] that states that with high probability the distance between a pair of samples in a high dimensional space is preserved if they are projected onto a randomly selected subspace with sufficient dimensions[11]. A random matrix R satisfying the Johnson-Lindenstrauss lemma can project compressive signals such as images onto a lower dimensional space without losing most of the information [16]. This allows for the use of robust features in detection without the computational cost that comes with high dimensional feature spaces.

A random matrix R of size m × n projects data from an m dimensional space to an n dimensional space where n is much smaller than m. R is usually a Random Gaussian matrix with zero mean and unit variance. However, to further reduce the computational cost a sparser random matrix is employed where a member of R rij is given by the equation:

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 \; with \; probability \; \dfrac{1}{2s} \\ 0 \; with \; probability \; 1 - \dfrac{1}{s} \\ -1 \; with \; probability \; \dfrac{1}{2s} \end{cases} \tag{1}$$

This matrix satisfies the Johnson-Lindenstrauss lemma for s= 2 or 3[15]. For s=2 half the matrix is zeros and therefore half the computations for the projections may be ignored. For s=3 two thirds of the matrix is zeros. This not only saves in computational load but also in memory as only the non zero elements of the matrix R are stored.

3 Scale-Invariant Tracking

In this section we present our proposed tracking method, employing two steps of detection to perform object tracking.

3.1 First Step

In the first step we employ the algorithm for tracking by detection using random projections in [11]. Given the position and size of the object in the first frame we generate positive samples near the object and negative samples surrounding it. All the samples have the same dimensions as the given object. Haar like feature are extracted from the samples by applying a set of convolution masks with varying scales. The samples are now represented with a very high dimensional feature space which makes the computational load very heavy. The Haar like features are then compressively sensed with a very sparse measurement matrix R to randomly project them onto a lower dimensional subspace. We only need to construct the measurement matrix R once. We then use the same matrix for the rest of the tracking process.

The positive and negative samples –now projected to a lower dimensional feature space- are used to train a base classifier. We use to types of classifier for detection: 1- the ratio classifier used in [11] which models the samples with a naïve Bayesian classifier. When new samples are added, the mean and variance of the underlying Gaussian distribution are updated. 2- A K nearest neighbor (KNN) classifier which classifier samples based on their proximity to the positive and negative training samples. When new samples are added, older samples are averaged into fewer samples to save memory and computations.

The base classifier is then used to classify patches generated randomly from the image around the last detected position of the object. The patches are projected into the feature space using the same measurement matrix R. The classifier then chooses from the positively labeled samples the best candidate for the position of the object. In case of the ratio classifier, it is the sample with the highest posterior probability. In case of the KNN it is the sample closest to the positive training samples.

3.2 Second Step

In this step we estimate the scale variation of the object and provide the new bounding box for the object. New patches are generated around the position provided from the first step. This time the patches may be up to 10% smaller or larger than the training samples. The new samples are projected using the same random matrix and labeled using the base classifier. The positive sample most similar to the original target given in the first frame is chosen as the new position and dimensions of the object. Step one is started using the new bounding box and the process is repeated. Since the ratio classifier can't be trained with a single sample to choose the point most similar to the original object, we use Euclidean distance as a measure of similarity.

By dividing the detection process into two steps, we treat the problem as two separate tasks: 1- finding the position of the object. 2- Find the size of the object.

This way we avoid the problem of having a large search space, where we needlessly search through different size in wrong positions. Since the radius in which we generate the patches around the object in the second step is very small, the generated patches are very dense and so we have a large chance of accurately fitting the object while only slightly increasing the computational load. The original object is used as a reference in the second step instead of the self labeled samples to prevent the problems of loose fitting or jittering that arise from self updating. It serves as a correction step to prevent drifting. There is no risk however from the problems associated with not adaptive algorithms, since an adaptive algorithm is used in the first step, and the second step is restricted to a small radius around the tracked position.

4 Experiments and Results

In this section we describe our experiments and present our results. We compare our scale-invariant approach to the fixed scale approach used in [11]. The parameters for our setup is as follows: The radius for drawing positive samples is 4, the inner radius for generating negative samples is 120 and the outer radius is 150. In step 1, the search radius to detect the object is set to 70 and 1000 samples are generated. In step 2 the search radius is set to 3 and 500 samples are generated. The dimensionality of the projected space is 50 and the learning rate is set to 0.9. The experiments are repeated using the ratio classifier and KNN as base classifiers. For fair comparison, the same random matrix is used to compare the two step and the single step approaches.

We conduct our experiments on a video sequence of a toy cow. The sequence contains high scale variations and rapid change. The object is rotated off plane and otherwise. It also leaves the scene and is totally lost before returning to view.

To evaluate the performance of the tracker and compare it to fixed scale tracking in a quantitative way we use 2 quality measures. Firstly we use a score representing the average Euclidean distance between the center of the tracking bounding box and the ground truth (in pixels). This score is only concerned with the position of the center of the bounding box and doesn't take scale into consideration. Secondly we use a score based on the PASCAL challenge used in [17] and [18] which calculates the overlap between the detected bounding box and the ground truth. The overlap is calculated as:

$$\text{score} = \frac{\text{area}(\text{ROI}_\text{D} \cap \text{ROI}_\text{GT})}{\text{area}(\text{ROI}_\text{D} \cup \text{ROI}_\text{GT})} \tag{2}$$

where $\text{ROI}_\text{D}$is the detected bounding box and $\text{ROI}_\text{GT}$is the ground truth bounding box. We use this score to calculate the percentage of correctly tracked frames in the sequence. A frame is considered positive if the overlap score exceeds 0.5. The run time for the fixed scale tracker using ratio classifier and KNN is 13.8 and 5.9 fps respectively, while the run time for the scale adaptive tracker is 11.9 and 5.6 fps respectively.

Figure 1 shows the results based on the distance score. The graph shows the error in distance between the centroids along the frames of the sequence. It shows that the average distance is reduced by more than 50% when employing our method. This is true for both KNN and ratio classifiers. It is worth noting that this measure of evaluation doesn't take the size of the bounding box into consideration; so the reduction in error is due to increased accuracy in tracking the position of the object.

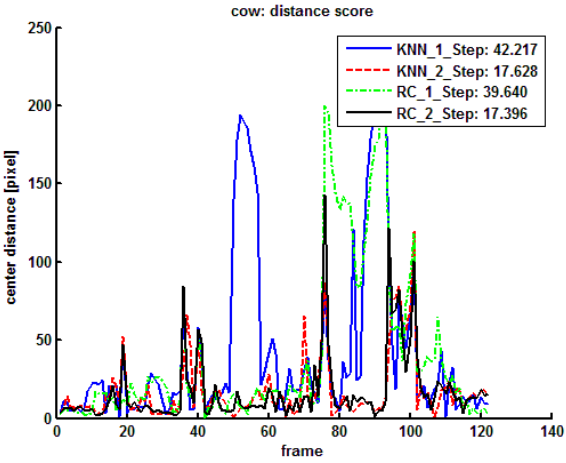


Fig. 1. Tracking results using distance score

Figure 2 shows the results of the Pascal score throughout the sequence. It also shows the percentage of frames in which the object was detected correctly (with more the 50% overlap between detection and ground truth). From the graph we can see where the tracker drifts and loses the object. We can also see a point where the object exits the scene completely before returning to it and is momentarily lost by all

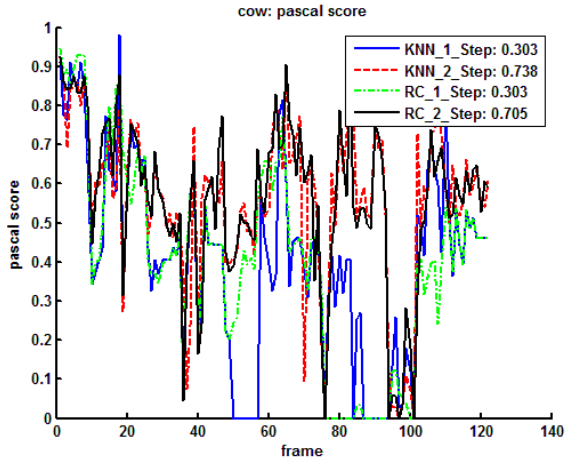


Fig. 2. Tracking results using Pascal score

trackers. Using KNN as a classifier, the accuracy improved from 30.3% to 73.8% when using our method. Using ratio classifier, the accuracy improved from 30.3% to 70.5% when using our method. The results from both figures clearly show the improved robustness and accuracy of our tracker.

Figure 3 shows samples of the results for both methods at different frames. The figure shows results using both the ratio classifier (left) and KNN (right) as base classifiers. Both classifiers perform similarly and we can draw similar observations. The scale-invariant tracking accurately updates the size as well as the position of the object and provides tight tracking for the object throughout the video sequence. We can see several examples of object drift fixed by using a second step. a) Shows a bounding box too big for the object and so is drawn to the background. b) Is an example where the size of the bounding box is correct but the tracking drifts due to the sudden rotation and translation of the object. e) Shows the case where the bounding box is too small for the object and can only capture some of its features. f) Is a case where the object is partially culled. Notice that our proposed tracker scales down the bounding box to fit the visible part of the object more tightly and avoid capturing a lot of the background.

Figure 4 shows the object returning to view after being totally culled. Notice that tracking without a corrective step detects the object incorrectly when using the ratio classifier, while two step tracking reacquires the object upon its return. The target is not reacquired simply due to the employment of the original target in the second step; since the radius for finding the object is limited, it is obvious that the object was detected in the first step. Instead, this improvement is attributed to the accurate detection performed by the scale invariant tracker throughout the sequence; which decreases the chances of mislabeled samples degrading the performance of the classifier.

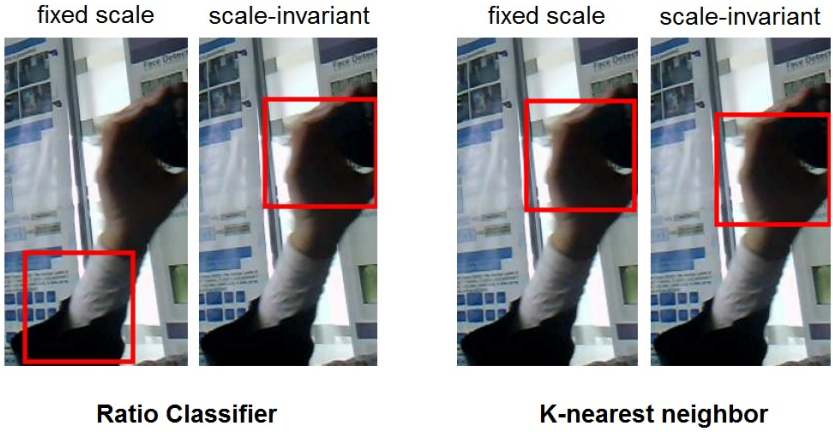


Fig. 3. Tracking results using the proposed tracking method vs. fixed scale tracking. The object reappears after exiting the scene completely.

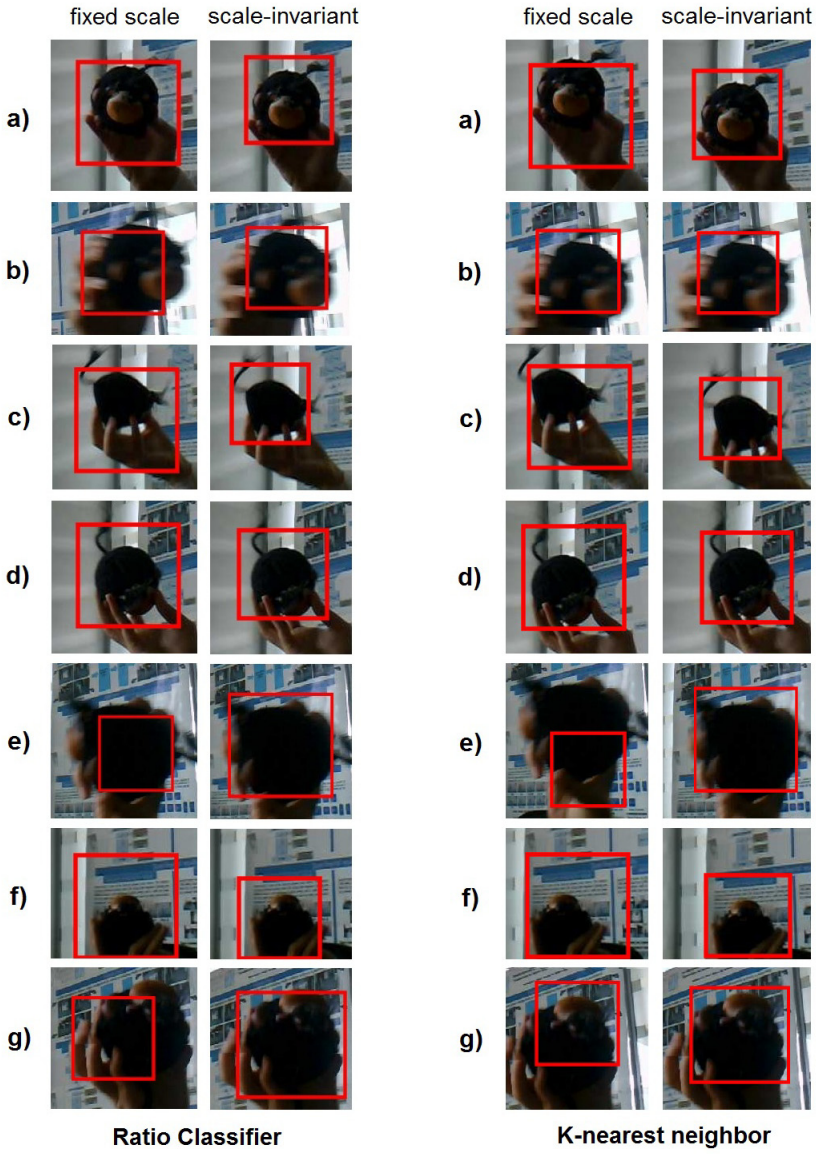


Fig. 4. Tracking results using the proposed tracking method vs. fixed scale tracking

5 Conclusion and Future Work

In this paper we propose a robust real-time, scale-invariant tracking method to deal with object scaling and drifting. We employed random projections to describe image samples effectively in a low dimensional space. In the first step of our method the

position of the object is found by searching through fixed sized samples in a large area. In the second step we search for the object through samples of varying size in a small area. Our results show that our method provides tight tracking throughout the sequence and decreases drifting and inaccurate detection compared to the single step approach. In the future we aim to repeat the experiment on more datasets to establish the benefit of our method. Moreover, since a small number of projected features is used (n=50), the results are dependent on the generated random matrix. Using a different random matrix may significantly change the results. We aim to investigate techniques to alleviate the sensitivity of the detection to the random projections.

References

1. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1515–1522. IEEE (September 2009)
2. Andriluka, M., Roth, S., Schiele, B.: Monocular 3d pose estimation and tracking by detection. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 623–630. IEEE (June 2010)
3. Stalder, S., Grabner, H., Van Gool, L.: Cascaded confidence filtering for improved tracking-by-detection. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 369–382. Springer, Heidelberg (2010)
4. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, June 23-28 (2008), doi:10.1109/CVPR.2008.4587583
5. Grabner, H., Grabner, M., Bischof, H.: Real-Time Tracking via On-line Boosting. BMVC 1(5), 6 (2006)
6. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I–511. IEEE (2001)
7. Zivkovic, Z., Krose, B.: An EM-like algorithm for color-histogram-based object tracking. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 1, pp. I–798. IEEE (June 2004)
8. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5), 564–577 (2003)
9. Yilmaz, A.: Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–6. IEEE (June 2007)
10. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(8), 1619–1632 (2011)
11. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
12. Donoho, D.: Compressed sensing. IEEE Trans. Inform. Theory 52, 1289–1306 (2006)
13. Candes, E., Tao, T.: Near optimal signal recovery from random projections anduniversal encoding strategies. IEEE Trans. Inform. Theory 52, 5406–5425 (2006)

14. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–250. ACM (August 2001)
15. Achlioptas, D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins. J. Comput. Syst. Sci. 66, 671–687 (2003)
16. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. Constr. Approx. 28, 253–263 (2008)
17. Grabner, H., Grabner, M., Bischof, H.: Real-Time Tracking via On-line Boosting. BMVC 1(5), 6 (2006)
18. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 723–730. IEEE (June 2010)

A Comparison between Active and Passive 3D Vision Sensors: BumblebeeXB3 and Microsoft Kinect

Diana Beltran and Luis Basañez

Technical University of Catalonia, Barcelona, Spain
{diana.beltran,luis.basanez}@upc.edu

Abstract. This paper shows a performance comparison of two sensors capable of obtaining depth information using two different methods, i.e. stereo information and infrared based depth measurement. The sensors are a Bumblebee XB3 and a Microsoft Kinect, and they provide in-depth information with some advantages and disadvantages that will be presented and evaluated in this paper. The analysis compares the devices single characteristics and tests their performance.

Keywords: BumblebeeXB3, Kinect, Point Cloud, 3D measurements.

1 Introduction

At the present time there exist different techniques to acquire depth information from a scene. These techniques are generally grouped in two main categories [1]: active and passive. The active group refers to the techniques that use a controlled source of structured energy emission, such as a scanning laser source or a projected pattern of light, and a detector like a camera. A common active vision device is a laser range scanner, where an active source moves around an object in order to scan the entire object surface. These sensors are dense in 3D measurements, but most of them are limited to static environments [2]. Few years ago a new class of active depth sensing systems based on the time-of-flight (TOF) principle, has emerged [3], [4]. The operational principle of these sensors is similar to other laser scanners but their advantage is that they can capture the whole scene at the same time, enabling their use in dynamic scenes applications [5]. The disadvantages of the TOF sensors are the big price and the low resolution. A recent development in active range sensing technology is the Microsoft Kinect sensor [6] that has a good working range, low price, a reasonable resolution and a low computational cost.

On the other hand, the passive techniques do not use a specific structured source of energy in order to form an image and hence, the light source may not be directly used in the range calculation. The basic principle used in recovering 3D information is the triangulation principle. In active vision techniques, a triangle is created between the light, the object and the sensor. In passive stereo vision techniques, the triangle may be created between the object and two sensors. Many kinds of sensors have been

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 252,
DOI: 10.1007/978-3-319-03413-3_54,

designed to acquire depth information using specific techniques making them suitable for a specific application.

The objective of this paper is to analyze a passive and an active sensor in order to show their performance and to emphasize their differences. Some previous comparison of active and passive sensor for acquiring 3D information have been made; for instance in [7] a Kinect and a bumblebe2 are used under different light condition with the objective of appreciating the performance of each sensor under light variation; other interesting comparison can be found in [8] where the accuracy of a PMD and a stereo vision, both systems under optimal condition, is compared.

2 BumblebeeXB3 Stereo Camera

The BumblebeeXB3 is a stereo camera with three sensors that can be used with two different baselines; 24 cm and 12 cm. The 24 cm baseline allows obtaining 3D points with more accuracy in longer ranges making it useful for outdoor applications, while the narrow baseline is more suitable for indoor areas improving close range matching and minimum-range-limitation [9]. Table 1 shows the main specifications of the BumblebeeXB3 camera [10], and Fig. 1 shows an image of the BumblebeeXB3 camera.

Table 1. Main specifications of the BumblebeeXB3 camera and the Microsoft Kinect sensor

BumblebeeXB3 Camera Specifications	
Imaging Sensor	Three Sony ICX445 a/3¨ progressive scan CCD´s
	1280x960 max pixels, 3.75µm square pixel
Baseline	12cm and 24cm
Lens Focal Length	2.5mm with 100° HFVO or 3.8mm with 70°HFVO or 6mm with 50° HFVO
A/D Converter	Analog Device 12-bit analog to digital converter
Video Data Output	8 and 16-bit digital data
Frame Rates	15, 7.5, 3.75, 1.875 FPS

Microsoft Kinect Specifications	
Imaging Sensor	IR Projector. RGB Camera IR Camera
Resolution, Depth Stream	QVGA (320x240)
Resolution, Color Stream	VGA (640x480)
Frame Rate	30 FPS
Mechanical Tilt Rate	±28°
Field Of View	Horizontal: 57° Vertical: 43°
Working Range	1.2 to 3.5 meters

Fig. 1. BumblebeeXB3 camera

2.1 Operation Mode

The process followed by BumblebeeXB3 camera to obtain depth information is similar to the used by a normal stereo vision system. In general, the purpose of stereo vision is to perform range measurements based on the left and right images obtained from stereoscopic cameras. Basically, an algorithm is implemented to establish the correspondence between image features in different views of the scene and then calculate the relative displacement between features coordinates in each image [11].

A general description of the process to obtain 3D points is described next.

Camera Calibration. The first step is to calibrate the camera; the BumblebeeXB3 is accurately precalibrated by Point Grey Research for lens distortions and camera misalignment [12]. The camera calibration process is a necessary step to know the parameters that define the camera model in order to obtain scene measurements from images. The accuracy of the calibration will determine the precision of the measurements obtained from the images.

Matching. To obtain a 3D data set from a scene, it is necessary to solve a crucial problem: find corresponding points in each image; i.e., points in image A, $P(x, y)$, and image B, $P'(x', y')$, corresponding to the same point of the scene. This is a difficult process because it is possible to have areas of the scene where there is no solution (occluded areas) or where exist multiple solutions (areas without enough texture).

Reconstruction. The reconstruction is the method by which the spatial layout of a scene can be recovered from two views. Therefore, once the corresponding points are found (P, P'), the final step is to find the depth of the points $Z = bf/d$, where b is the baseline, f is the focal length and d is the disparity $d = x' - x$.

3 Microsoft Kinect

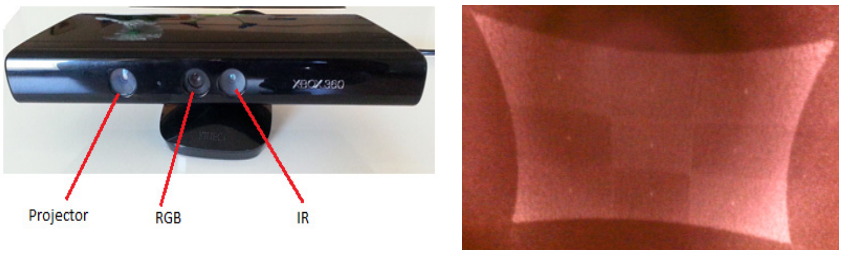
The Kinect sensor was originally designed as a game interface, but the robotics community has seen in it an interesting 3D sensor for robotics application. This sensor is being used by the researchers because of its high quality and low price. Some applications where the Kinect sensor is being used are: 3D reconstruction, face detection, slam, object detection, and augmented reality, among others.

Table 1 shows the main specifications of the Microsoft Kinect. Fig. 2a shows the sensor´s placement on the device. Each lens is associated with a camera or a projector [13]. The RGB sensor has a resolution of 640x480 pixels and a frame rate of 30 fps and the infrared (IR) camera has a resolution of 320x240 pixels and a frame rate of 30 fps.

3.1 Operation Mode

The Kinect depth is calculated by triangulation against a known pattern from the projector, (Fig 2b). The pattern is memorized at a known depth. For a new image, at

each pixel in the IR image, a small correlation window is used to compare the local pattern at that pixel with the memorized pattern. The best match gives an offset from the known depth, in terms of pixels: this is called disparity. The Kinect device performs a further interpolation of the best match to get sub-pixel accuracy of 1/8 pixel. Given the known depth of the memorized plane and the disparity, an estimated depth for each pixel can be calculated by triangulation [13].



(a) (b)

Fig. 2. (a) Kinect sensors placement, (b) Pattern projected from the Kinect

3.2 Disparity to Depth Relationship

In a normal stereo system, the cameras are usually calibrated so that the rectified images are parallel and have corresponding horizontal lines. At zero disparity, the rays from each camera are parallel, and the depth is infinite. Larger values for the disparity mean shorter distances [13].

In the Kinect case, it returns a raw disparity that is not normalized in this way, that is, a zero Kinect disparity does not correspond to infinite distance. The Kinect disparity is related to a normalized disparity by the relation:

$$d = (d_{off} - k_d)/8$$

Where d is a normalized disparity, k_d is the Kinect disparity, and d_{off} is an offset value particular to a given Kinect device. The factor 1/8 appears because the values of k_d are in 1/8 pixel units [13].

4 Experiments

In order to show the differences between the stereo camera BumblebeeXB3 and Microsoft Kinect sensor, a real experiment has been done for comparison. (The experiment has been performed during the afternoon in order to have enough external light, because the BumblebeeXB3 does not work without enough light).

For a specific scene, the aim is to obtain the disparity image and the points cloud given by the BumblebeeXB3 and the Kinect. In order to compare the same area of a scene the cameras has been placed according to the setup of Fig. 3.

Fig. 3. Kinect and BumblebeeXB3 setup

4.1 BumblebeeXB3 Images Acquisition

The Robot Operating System (ROS) has been used to perform the experiments, [14]. ROS has a package for the Bumblebee2, the previous version of the BumblebeeXB3, so it has been necessary to make some changes in this code to adapt it to the BumblebeeXB3. The adapted package allows to obtain the raw images from each of the sensors of the camera (left, center and right) and, adding the parameters obtained from the stereo_calibration ROS package, it is possible to obtain calibrated images from each stereo pair (left-center, right-center, left-right). To get the disparity image and the 3D point cloud it was necessary to use the stereo_image_proc ROS package. In Fig. 4, the left and right images acquired from the BumblebeeXB3 using its maximum resolution are showed.

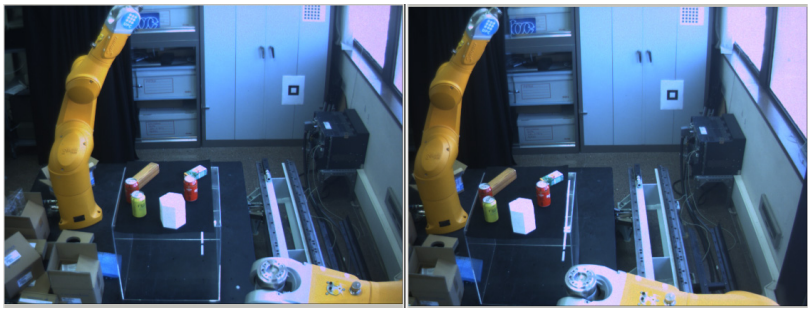


Fig. 4. Left and right images from BumblebeeXB3

To obtain the disparity image, *stereo_image_proc* uses the class *stereobm* of *opencv*. Fig. 8a shows the disparity image obtained using the left and right images of the BumblebeeXB3 (Fig. 4). The *stereobm* class computes stereo correspondence using the block matching algorithm. The class uses some parameters that define the size of disparity range for an optimal search and determine the size of the averaging window used to match pixel blocks. These parameters and the prefiltering and postfiltering parameters could be changed by the user through the *dynamic_reconfigure* ROS packages with the window showed in Fig. 5.

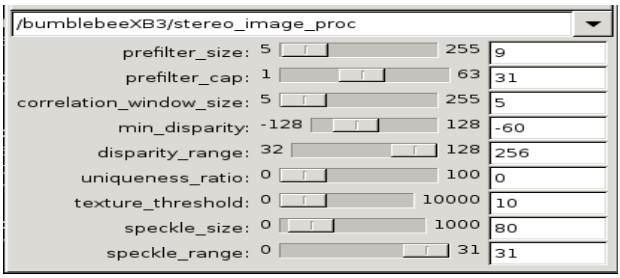


Fig. 5. Dynamic reconfigure window

4.2 Kinect Images Acquisition

For the Kinect sensor, the *Openni_camera* ROS package has been used. This package allows obtaining the raw IR image, the raw RGB image, the disparity image and the 3D point cloud. Using the *image_calibration* ROS packages it is possible to get the calibrated IR image and the calibrated RGB images (Fig. 8) and, in consequence, obtain a better disparity image (Fig. 7b) and a 3D point cloud (figure 11b).

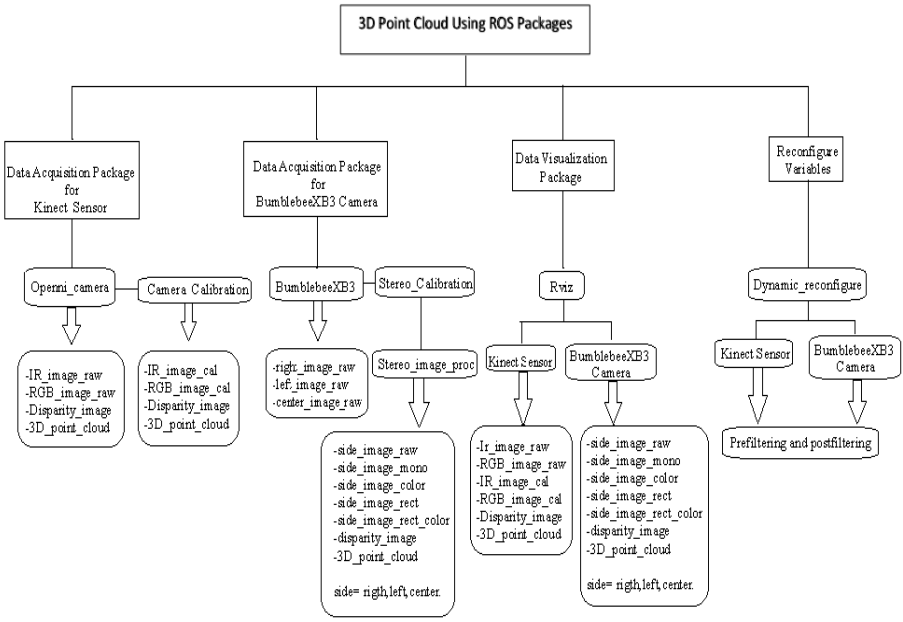
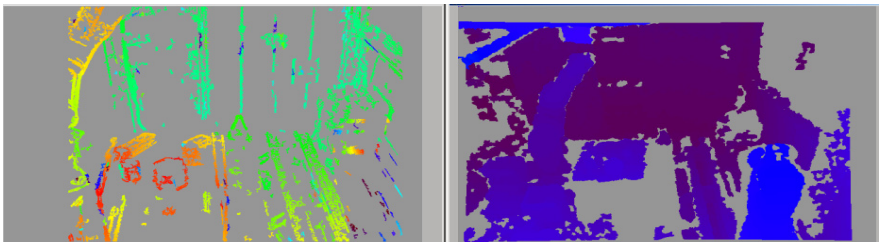


Fig. 6. Software used to obtain 3D points from BumblebeeXB3 and Kinect

5 Experimental Analysis

In order to visualize the images captured by the BumblebeeXB3 and the Kinect sensor, the *image_view* and *rviz* packages has been used. Fig. 6 shows an overview of the software used to obtain the 3D points from the two sensors.

Analyzing the disparity images from the Kinect and the BumblebeeXB3 sensors (Fig.7), the differences between them are evident. By one side the BumblebeeXB3 generates a disparity map (the colors of the disparity map indicate the distance of the objects from the camera) without information in the textureless areas, but describing the edges of the transparent objects. By the other side, the Kinect sensor generates a disparity maps with rich information in textureless areas but has problems with reflective objects.



(a) (b)

Fig. 7. (a) Disparity map from BumblebeeXB3, (b) Disparity map from Kinect

Fig. 11 presents the point cloud images obtained from the Kinect and the BumblebeeXB3 cameras. Since the disparity is proportional to the depth ($Z = bf/d$), it is coherent that in the point cloud there are not points in the areas of each image where the corresponding sensor has no information (explained above). In order to have some real distance measurements, several points from the scene have been chosen, and their real distance from each camera has been obtained with the digital laser distance meter Bosch DLE50 professional that has a range from 0.05m to 50m and a precision of ±1.5mm.

(a) (b)

Fig. 8. (a) IR image from Kinect, (b) RGB image from Kinect

Table 2. Depth measurements

Sample	Real World (cm)	Kinect (cm)	Error(cm)	BumblebeeXB3	Error (cm)
1	145,8	146,1	0,300	NaN	NaN
2	164,0	164,9	0,900	1,659	1,900
3	169,0	170,6	1,600	1,710	2,000
4	175,0	NaN	NaN	1,728	2,250
5	184,6	183,3	1,300	1,867	2,100
6	193,4	196,9	3,500	1,956	2,200
7	206,9	207,6	0,700	2,058	1,100
8	207,9	208,8	0,900	2,092	1,300
9	233,7	235,9	2,200	2,314	2,300
10	241,4	249,6	8,200	2,431	1,700
11	247,1	251,4	4,300	2,484	1,300
12	252,8	229,7	23,100	NaN	NaN
13	260,1	266,9	6,800	2,626	2,500
14	299,1	301,9	2,800	2,957	3,400
15	480,8	490,7	9,900	4,781	2,700
16	490,4	483,8	6,600	4,749	15,500

Table 2 allows a comparative analysis of the accuracy of both cameras. In some areas the BumblebeeXB3 works better than the Kinect camera and in other areas the Kinect works better than the BumblebeeXB3; it is due to the different specifications of each camera, the characteristics of the scene and the intrinsic parameters like radial distortion; in points that are around the optical center the accuracy is better, but as one moves away from the center the accuracy decreases. In point 1, the BumblebeeXB3 has no information because this point is out of the working space of the camera (using the 24cm baseline), while point 4 lies in an area where the Kinect has lost information because of an external interference. In point 16 the BumblebeeXB3 measurements has a big error because this point is in an area of low texture and at big distance away from the camera. By other hand, point 12 has a big error because this area had high external illumination and the Kinect is affected for it. In the case of the BumblebeeXB3 this point 12 is out of its field of view. In Fig. 11 it is possible to appreciate the 3D point cloud used to make the measurements of both cameras.

Fig. 9 and Fig. 10 show a dispersion of the points in the plane (x,y) (each point has the sample´s number and its corresponding error). It can observe how near or far are the points from the optical center of each 3D camera. Points near to the optical center of the camera should have less error, because of less radial distortion, but it is not the only factor that affect the quality of the measurements of a point The influence of

external light, the camera calibration and the distance from the sensor are also important factors to take into account; that is why some points, despite being close of the optical center, have higher error than others.

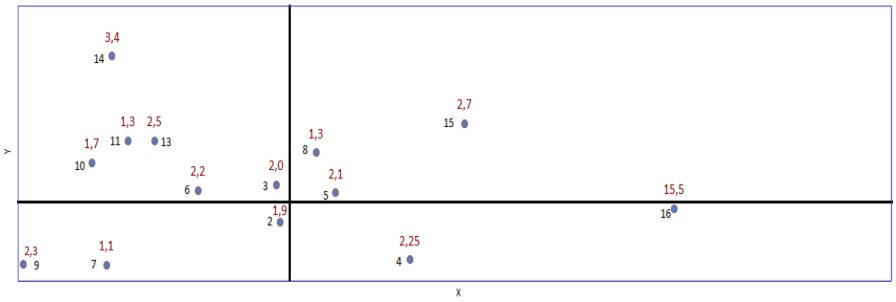


Fig. 9. Spatial distribution of the points in Bumblebee's reference frame

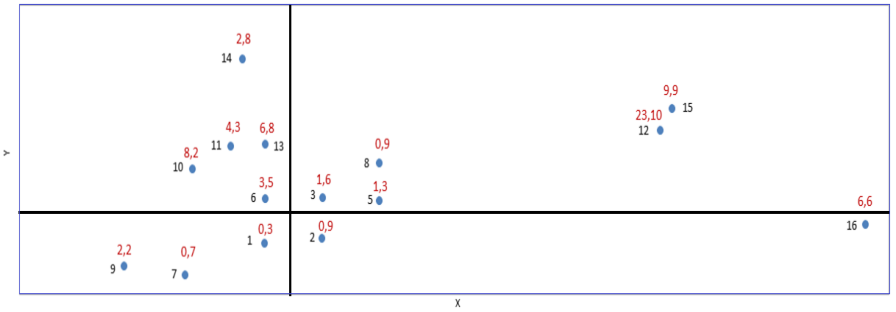
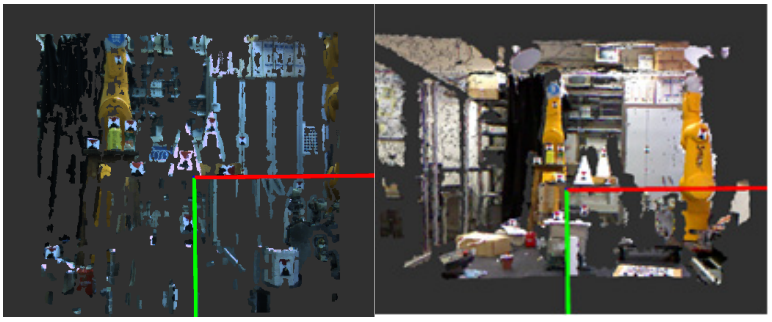


Fig. 10. Spatial distribution of the points in Kinect's reference frame



(a) (b)

Fig. 11. (a) Point cloud from BumblebeeXB3, (b) Point cloud from Kinect

6 Conclusions

The BumblebeeXB3 is a camera with high accuracy in areas with enough texture, has a high resolution and a big field of work, making it appropriate for indoor and outdoor scenes. Its disadvantages appear in textureless, low illuminated areas. By the other side, the Kinect sensor has high accuracy in textureless areas, is faster and works well in its work space (1,2-3.5 meters), its resolution is enough to obtain good 3D measurements and it is a sensor with low computer cost (because, most of the process is performed in the device chip). But it is not appropriate for outdoor applications because its sensibility to the interferences by external light, and it loses information in reflective objects. With this analyze we can conclude that these two sensor complement each other making think the fusion of their information as a future work.

References

1. Srrenivasa, K., Mada, M., Smith, L., Smith, S.: An Overvire of Passive and Active Vision Techniques for Hand -Held 3D Data Acquisition. In: Proceedings of SPIE Opto-Ireland: Optical Metrology, Imaging, and Machine Vision, vol. 4877 (2003)
2. Salvi, J., Pages, J., Batlle, J.: Pattern Codification Strategies in Structured Light Systems. Pattern Recognition 37(4), 827–849 (2004)
3. 3DV systems, z-cam, http://inition.co.uk/3D-Technologies/3dv-systems-zcam
4. MESA Imaging, http://www.mesa-imaging.ch/
5. Pmd technologies, http://www.pmdtec.com/
6. Microsoft Kinect, http://www.xbox.com/en-us/kinect/
7. Beder, C., Bartczak, B., Koch, R.: A Comparison of PMD-Cameras and Stereo-Vision for the Task of Surface Reconstruction using Patchlets. In: Proceedings of the Second International ISPRS Workshop BenCOS (2007)
8. Tarfin, S., Bugra, O., Bugra, A., Konukseven, E.: Comparison of Kinect and Bumblebee2 in Indoor Environments. Midle East Technical University (2011)
9. Point Grey, http://www.avsupply.com/Point-Grey/bumblebee-xb3.php
10. Point Grey datasheet, http://ww2.ptgrey.com/stereo-vision/bumblebee-xb3
11. Nguyen, S., Nguyen, T.H., Nguyen, H.T.: Semi-autonomous Wheelchair System Using Stereoscopic Cameras. In: 31st Annual International Conference of the IEEE EMBS, Minneapolis, Minesota, USA (September 2009)
12. Rossu, L., Molinier, T., Akhloufi, M., Tison, Y.: Measurement of Laboratory fire Spread Experiments by Stereovision. In: Image Processing Theory, Tools and Applications. IEEE (2010)
13. Kinect_calibration, http://www.ros.org/wiki/kinect_calibration/technical
14. ROS Packages, http://www.ros.org/browse/list.php

Author Index