

Manuel F. Silva · José Luís Lima ·  
Luís Paulo Reis · Alberto Sanfeliu ·  
Danilo Tardioli *Editors*

# Robot 2019: Fourth Iberian Robotics Conference

Advances in Robotics, Volume 2

# **Advances in Intelligent Systems and Computing**

**Volume 1093**

## **Series Editor**

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,  
Warsaw, Poland

## **Advisory Editors**

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India

Rafael Bello Perez, Faculty of Mathematics, Physics and Computing,  
Universidad Central de Las Villas, Santa Clara, Cuba

Emilio S. Corchado, University of Salamanca, Salamanca, Spain

Hani Hagras, School of Computer Science and Electronic Engineering,  
University of Essex, Colchester, UK

László T. Kóczy, Department of Automation, Széchenyi István University,  
Gyor, Hungary

Vladik Kreinovich, Department of Computer Science, University of Texas  
at El Paso, El Paso, TX, USA

Chin-Teng Lin, Department of Electrical Engineering, National Chiao  
Tung University, Hsinchu, Taiwan

Jie Lu, Faculty of Engineering and Information Technology,  
University of Technology Sydney, Sydney, NSW, Australia

Patricia Melin, Graduate Program of Computer Science, Tijuana Institute  
of Technology, Tijuana, Mexico

Nadia Nedjah, Department of Electronics Engineering, University of Rio de Janeiro,  
Rio de Janeiro, Brazil

Ngoc Thanh Nguyen, Faculty of Computer Science and Management,  
Wrocław University of Technology, Wrocław, Poland

Jun Wang, Department of Mechanical and Automation Engineering,  
The Chinese University of Hong Kong, Shatin, Hong Kong

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 such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within “Advances in Intelligent Systems and Computing” are primarily 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.

**\*\* Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink \*\***

More information about this series at <http://www.springer.com/series/11156>

Manuel F. Silva · José Luís Lima ·  
Luís Paulo Reis · Alberto Sanfeliu ·  
Danilo Tardioli  
Editors

# Robot 2019: Fourth Iberian Robotics Conference

Advances in Robotics, Volume 2



Springer

*Editors*

Manuel F. Silva  
School of Engineering  
Polytechnic Institute of Porto  
Porto, Portugal

José Luís Lima  
Department of Electrical Engineering  
Polytechnic Institute of Bragança  
Bragança, Portugal

Luís Paulo Reis   
Faculty of Engineering  
University of Porto  
Porto, Portugal

Alberto Sanfeliu  
UPC  
Universitat Politècnica de Catalunya  
Barcelona, Spain

Danilo Tardioli  
Centro Universitario de la Defensa (CUD)  
Zaragoza, Spain

ISSN 2194-5357

ISSN 2194-5365 (electronic)

Advances in Intelligent Systems and Computing

ISBN 978-3-030-36149-5

ISBN 978-3-030-36150-1 (eBook)

<https://doi.org/10.1007/978-3-030-36150-1>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

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

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

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

This book contains a selection of papers accepted for presentation and discussion at Robot 2019—Fourth Iberian Robotics Conference—held in Porto, Portugal, November 20–22, 2019. Robot 2019 is part of a series of conferences that are a joint organization of SPR—Sociedade Portuguesa de Robótica/Portuguese Society for Robotics and SEIDROB—Sociedad Española para la Investigación y Desarrollo en Robótica/Spanish Society for Research and Development in Robotics. The conference organization had also the collaboration of several universities and research institutes, including School of Engineering of the Polytechnic Institute of Porto, Polytechnic Institute of Bragança, University of Porto, University Politécnica de Cataluña, University of Zaragoza/I3A, INESC TEC, Centro Universitario de la Defensa, CeDRI and LIACC.

Robot 2019 builds upon several previous successful events, including three biannual workshops (Zaragoza—2007, Barcelona—2009 and Sevilla—2011) and the three previous editions of the Iberian Robotics Conference held in Madrid in 2013, Lisbon in 2015 and Seville in 2017. The conference is focused on presenting the research and development of new applications, on the field of robotics, in the Iberian Peninsula, although open to research and delegates from other countries.

Robot 2019 featured five plenary talks on state-of-the-art subjects on robotics by Mirko Kovac, Director of the Aerial Robotics Laboratory, Reader in Aero-structures at Imperial College London and Royal Society Wolfson Fellow, UK, on “Soft Aerial Robotics for Digital Infrastructure Systems;” Gianni A. Di Caro, Associate Teaching Professor at the Department of Computer Science of the Carnegie Mellon University, Qatar, on “Robot Swarms and the Human-in-the-Loop;” Luis Merino, Associate Professor of Systems Engineering and Automation and Co-Principal Investigator of the Service Robotics Laboratory at the Universidad Pablo de Olavide, Spain, on “Human-Aware Decision Making and Navigation for Service Robots;” Nuno Lau, Assistant Professor at Aveiro University, Portugal, on “Optimization and Learning in Robotics;” and Elon Rimon, Professor in the Department of Mechanical Engineering at the Technion—Israel Institute of Technology, Israel, on “Perspectives on Minimalistic Robot Hand Design and a New Class of Caging-to-Grasping Algorithms.”

Robot 2019 featured 16 special sessions, plus a main/general robotics track. The special sessions were about Aerial Robotics for Inspection and Maintenance; Agricultural Robotics and Field Automation; Autonomous Driving and Driver Assistance Systems; Autonomous Sailboats and Support Technologies; Collaborative Robots for Industry Applications; Core Concepts for an Ontology for Autonomous Robotics; Genealogy and Engineering Practice; Educational Robotics; Field Robotics In Challenging Environments; Future Industrial Robotics; Intelligent Perception and Manipulation; Machine Learning in Robotics; Mobile Robots for Industrial Environments; Radar-Based Applications for Robotics; Rehabilitation and Assistive Robotics; Simulation in Robotics; and the Workshop on Physical Agents.

In total, after a careful review process with at least three independent reviews for each paper, but in some cases 5 or 6 reviews, a total of 112 high-quality papers were selected for publication, with a total number of 468 authors, from 24 countries, including Aland Islands, Australia, Belgium, Brazil, Canada, Colombia, Croatia, Czechia, Ecuador, Estonia, France, Germany, Italy, Japan, Netherlands, Pakistan, Portugal, Puerto Rico, Spain, Sweden, United Arab Emirates, UK, USA and Venezuela.

We would like to thank all special sessions' organizers for their hard work on promoting their special session, inviting the Program Committee, organizing the special session review process and helping to promote Robot 2019 Conference. This acknowledgment goes especially to Adrià Colomé, Alberto Olivares Alarcos, Alejandro Mosteo, Angel Sappa, Armando Sousa, Artur Pereira, Arturo de la Escalera, Begoña Arrue, Benedita Malheiro, Brígida Mónica Faria, Bruno Ferreira, Cristina Manuela Peixoto Santos, Danilo Tardioli, Eurico Pedrosa, Filipe Neves dos Santos, Francisco Bellas, Francisco Curado, Francisco Rovira Más, Germano Veiga, Guillem Alenyà, Guillermo Heredia, Ismael García Varea, Jan Rosell, João Quintas, Jon Agirre Ibarbia, Jorge Cabrera Gámez, José Lima, Juan C Moreno, Julita Bermejo-Alonso, Luis Merino, Luis Piardi, Luis Riazuelo, Manuel Silva, Miguel Ángel Cazorla Quevedo, Miguel Oliveira, Nuno Cruz, Nuno Lau, Pablo Bustos García de Castro, Paulo Goncalves, Pedro Guedes, Pedro Neto, Raul Morais, Ricardo Sanz, Roemi Fernandez, Vicente Matellán Olivera and Vitor Santos.

We would also like to take the opportunity to thank the rest of the organization members (André Dias, Benedita Malheiro, Luís Lima, Nuno Dias, Paulo Ferreira, Pedro Costa, Pedro Guedes and Teresa Costa) for their hard and valuable work on the local arrangements, publicity, publication and financial issues. We also express our gratitude to the members of all the Program Committees and additional reviewers, as they were crucial for ensuring the high scientific quality of the event and to all the authors and delegates that with their research work and participation

made this event a huge success. To the end of this preface, special thanks to our editors, Springer, that was in charge of this conference proceedings edition and, in particular, to Dr. Thomas Ditzinger.

October 2019

Manuel F. Silva  
José Luís Lima  
Luís Paulo Reis  
Alberto Sanfeliu  
Danilo Tardioli

# Organization

## Program Committee

Jon Agirre Ibarbia	Tecnalia Research and Innovation, Spain
António Pedro Aguiar	University of Porto, Portugal
Teodoro Aguilera	University of Extremadura, Spain
Eugenio Aguirre	University of Granada, Spain
Daniel Albuquerque	Polytechnic Institute of Viseu, Portugal
David Alejo	University Pablo de Olavide, Spain
Gillem Alenyà	Universitat Politècnica de Catalunya, Spain
Jorge Almeida	University of Aveiro, Portugal
Luis Almeida	University of Porto, Portugal
Fernando Alvarez	University of Extremadura, Spain
Angelos Amanatiadis	Democritus University of Thrace, Greece
Josep Amat	Universitat Politècnica de Catalunya, Spain
Rui Araújo	University of Coimbra, Portugal
Manuel Armada	Consejo Superior de Investigaciones Científicas, Spain
Begoña C. Arrue	University of Seville, Spain
Helio Azevedo	University of São Paulo, Brazil
José Azevedo	University of Aveiro, Portugal
Mihail Babciński	University of Coimbra, Portugal
Pilar Bachiller	Universidad de Extremadura, Spain
Stephen Balakirsky	Georgia Institute of Technology, USA
Antonio Bandera	University of Málaga, Spain
Juan Pedro Bandera Rubio	Universidad de Málaga, Spain
Antonio Barrientos	Universidad Politécnica de Madrid, Spain
Richard Bearee	Arts et Métiers ParisTech, France
José Antonio Becerra Permu	University of A Coruña, Spain
Daniel Beffler	University of Bremen, Germany
Francisco Bellas	University of A Coruña, Spain
Luis M. Bergasa	Universidad de Alcalá, Spain

Julita Bermejo-Alonso	Universidad Isabel I, Spain
Marko Bertogna	University of Modena, Italy
Mehul Bhatt	Örebro University, Sweden
Estela Bicho	University of Minho, Portugal
Ole Blaurock	Fachhochschule Lübeck, Germany
Thadeu Brito	Research Center in Digitalization and Intelligent Robotics, Portugal
Adrian Burlacu	The Gheorghe Asachi Technical University of Iasi, Romania
Pablo Bustos García de Castro	Universidad de Extremadura, Spain
Fernando Caballero Benítez	University of Seville, Spain
Jorge Cabrera	Universidad de Las Palmas de Gran Canaria, Spain
João Calado	Instituto Superior de Engenharia de Lisboa, Portugal
Rafael Caldeirinha	Polytechnic Institute of Leiria, Instituto de Telecomunicações, Portugal
Joel Luis Carbonera	Federal University of Rio Grande do Sul, Brazil
Angela Cardoso	INEGI, Portugal
Carlos Carreto	Polytechnic Institute of Guarda, Portugal
Alicia Casals	Institute for Bioengineering of Catalonia, Spain
Jose A. Castellanos	University of Zaragoza, Spain
Miguel Ángel Cazorla Quevedo	Universidad de Alicante, Spain
Jose Maria Cañas Plaza	Universidad Rey Juan Carlos, Spain
Marco Ceccarelli	University of Rome Tor Vergata, Italy
Abdelghani Chibani	Université Paris-Est Créteil, France
Benoit Clement	ENSTA Bretagne, France
Adrià Colomé	Universitat Politècnica de Catalunya, Spain
Luís Correia	University of Lisboa, Portugal
Paulo Costa	University of Porto, Portugal
Pedro Costa	University of Porto, Portugal
Hugo Costelha	Polytechnic Institute of Leiria, Portugal
Micael Couceiro	Ingeniarius, Ltd., Portugal
Vincent Creuze	University of Montpellier, France
Nuno Cruz	University of Porto, Portugal
Francisco Curado	University of Aveiro, Portugal
Arturo de la Escalera	Universidad Carlos III de Madrid, Spain
Félix de La Paz López	Universidad Nacional de Educación a Distancia, Spain
Antonio J. Del-Ama	National Hospital for Spinal Cord Injury, Spain
Mohammed Diab	Universitat Politècnica de Catalunya, Spain
Paulo Dias	University of Aveiro, Portugal

Antonio C. Domínguez-Brito	Universidad de Las Palmas de Gran Canaria, Spain
Fadi Dornaika	University of the Basque Country, Spain
Richard Duro	University of A Coruña, Spain
Luis Emmi	Centre National de la Recherche Scientifique, France
João Alberto Fabro	Federal Technological University of Paraná, Brazil
Andres Faina	IT University of Copenhagen, Denmark
Brígida Mónica Faria	Polytechnic Institute of Porto, Portugal
Paulo Farias	Federal University of Bahia, Brazil
Vicente Feliu	Universidad de Castilla-La Mancha, Spain
Fernando Fernandez	Universidad Carlos III de Madrid, Spain
Roemi Fernandez	Center for Automation and Robotics CSIC-UPM, Spain
Camino Fernández Llamas	Universidad de León, Spain
Manuel Ferre	CAR UPM-CSIC, Spain
Bruno Ferreira	INESC TEC, Portugal
Hugo Ferreira	INESC TEC, Portugal
Paulo Ferreira	Polytechnic Institute of Porto, Portugal
Pedro Fonseca	University of Aveiro, Portugal
Oscar Fontenla-Romero	University of A Coruña, Spain
Anna Friebel	Åland University of Applied Sciences, Finland
Oren Gal	Technion, Israel
Nicolas Garcia-Aracil	Universidad Miguel Hernandez, Spain
Ismael Garcia-Varea	Universidad de Castilla-La Mancha, Spain
Fernando García	University Carlos III, Spain
Paulo Goncalves	Polytechnic Institute of Castelo Branco, Portugal
Pablo Gonzalez-De-Santos	Centre for Automation and Robotics CSIC-UPM, Spain
Pedro Guedes	Polytechnic Institute of Porto, Portugal
Tamás Haidegger	Óbuda University, Hungary
Florian Haug	Åland University of Applied Sciences, Finland
Guillermo Heredia	University of Seville, Spain
Carlos Hernández Corbato	Delft University of Technology, The Netherlands
Roberto Iglesias Rodríguez	University of Santiago de Compostela, Spain
Eduardo Iáñez	Miguel Hernández University of Elche, Spain
Nicolas Jouandéau	Paris8 University, France
Nuno Lau	University of Aveiro, Portugal
Fabrice Le Bars	ENSTA Bretagne, France
Agapito Ledezma	Universidad Carlos III de Madrid, Spain
Wilfried Lepuschitz	Practical Robotics Institute Austria, Austria
Frederic Lerasle	LAAS-CNRS, France
Howard Li	University of New Brunswick, Canada

José Luís Lima	Polytechnic Institute of Bragança, Portugal
Pedro U. Lima	University of Lisboa, Portugal
Alejandro Linares-Barranco	University of Seville, Spain
Vitor Lobo	Escola Naval–CINAV, Portugal
Ana Lopes	University of Coimbra, Portugal
Joaquín López Fernández	University of Vigo, Spain
Jose Manuel Lopez-Gude	Basque Country University, Spain
Pedro Machado	Nottingham Trent University, UK
Erik Maeble	University of Luebeck, Germany
Benedita Malheiros	Polytechnic Institute of Porto, Portugal
André Luís Marcato	Federal University of Juiz de Fora, Brazil
Yarik Marchukov	Universidad de Zaragoza, Spain
Yaroslav Marchukov	Universidad de Zaragoza, Spain
Lino Marques	University of Coimbra, Portugal
Humberto Martinez Barbera	Universidad de Murcia, Spain
Francisco Martín Rico	Universidad Rey Juan Carlos, Spain
Vicente Matellán Olivera	Universidad de León, Spain
Nuno Mendes	University of Coimbra, Portugal
Munir Merdan	Practical Robotics Institute Austria, Austria
Luis Merino	Pablo de Olavide University, Spain
Luis Montano	University of Zaragoza, Spain
Paulo Monteiro	University of Aveiro, Portugal
Hector Montes Franceschi	Universidad Tecnologica de Panama, Panama
Eduardo Montijano	Universidad de Zaragoza, Spain
Raul Morais	University of Trás-os-Montes e Alto Douro, Portugal
Antonio Morales	Universitat Jaume I, Spain
Antonio P. Moreira	University of Porto, Portugal
Juan C. Moreno	Cajal Institute, Spain
Luis Moreno	Universidad Carlos III de Madrid, Spain
Alejandro R. Mosteo	Centro Universitario de la Defensa de Zaragoza, Spain
Paulo Moura Oliveira	University of Trás-os-Montes e Alto Douro, Portugal
Ana C. Murillo	University of Zaragoza, Spain
Giovanni Muscato	University of Catania, Italy
Alberto Nakano	Federal Technological University of Paraná, Brazil
Hirenkumar Chandrakant Nakawala	University of Verona, Italy
Pedro Navarro	University Carlos III, Madrid, Spain
Pedro Neto	University of Coimbra, Portugal
António J. R. Neves	University of Aveiro, Portugal

Urbano Nunes	University of Coimbra, Portugal
Pedro Núñez	Universidad de Extremadura, Spain
Manuel Ocaña Miguel	Universidad de Alcalá de Henares, Spain
Alberto Olivares Alarcos	Universitat Politècnica de Catalunya, Spain
Alexandra Oliveira	Polytechnic Institute of Porto, Portugal
Andre Oliveira	Federal Technological University of Paraná, Brazil
Josenalde Oliveira	Federal University of Rio Grande do Norte, Brazil
Miguel Oliveira	University of Aveiro, Portugal
Joanna Isabelle Olszewska	University of West Scotland, UK
Jun Ota	The University of Tokyo, Japan
Dimitrios Paraforos	University of Hohenheim, Germany
Nieves Pavon-Pulido	Universidad Politécnica de Cartagena, Spain
Eurico Pedrosa	University of Aveiro, Portugal
Peter Peer	University of Ljubljana, Slovenia
Ana Pereira	Polytechnic Institute of Bragança, Portugal
Artur Pereira	University of Aveiro, Portugal
Marcelo Petry	INESC TEC, INESC P&D Brasil, Federal University of Santa Catarina, Brazil
Luis Piardi	Research Center in Digitalization and Intelligent Robotics, Portugal
Tatiana M. Pinho	INESC TEC, Portugal
Andry Pinto	INESC TEC, Portugal
Vítor Hugo Pinto	University of Porto, Portugal
Frédéric Plumet	Institut des Systèmes Intelligents et de Robotique, France
Mihai Pomarlan	Universität Bremen, Germany
Cristiano Premebida	Loughborough University, UK
Edson Prestes	Federal University of Rio Grande do Sul, Brazil
Domènec Puig Valls	Universitat Rovira i Virgili, Spain
Noé Pérez-Higueras	Pablo de Olavide University, Spain
Jonas Queiroz	University of Porto, Portugal
João Quintas	Instituto Pedro Nunes, Portugal
Sandro Rama Fiorini	IBM, USA
Francisco Ramos	University of Castilla-La Mancha, Spain
Pedro Jesús Reche López	Universidad de Jaén, Spain
Carlos V. Regueiro	Universidade da Coruña, Spain
Giulio Reina	University of Salento, Italy
Oscar Reinoso	Universidad Miguel Hernandez, Spain
Luis Paulo Reis	University of Porto, Portugal
Luis Riazuelo	Universidad de Zaragoza, Spain
Lluís Ribas-Xirgo	Universitat Autònoma de Barcelona, Spain
Isabel Ribeiro	University of Lisboa, Portugal
Eugénio A. M. Rocha	University of Aveiro, Portugal

Luis Rocha	INESC TEC, Portugal
Rui P. Rocha	University of Coimbra, Portugal
Nelson Rodrigues	University of Porto, Portugal
Francisco J. Rodríguez Lera	University of Luxembourg, Luxembourg
Ronnier Rohrich	Federal Technological University of Paraná, Brazil
Roseli Romero	University of São Paulo, Brazil
Cristina Romero González	Universidad de Castilla-La Mancha, Spain
Kostia Roncin	ENSTA Bretagne, France
Jan Rosell	Universitat Politècnica de Catalunya, Spain
Rosaldo Rossetti	University of Porto, Portugal
Francisco Rovira Más	Universitat Politècnica de València, Spain
José Rufino	Polytechnic Institute of Bragança, Portugal
Mohammad Safeea	University of Coimbra, Portugal
Veera Ragavan Sampath Kumar	Monash University, Malaysia
Cristina Manuela Peixoto Santos	University of Minho, Portugal
Filipe Neves Santos	INESC TEC, Portugal
Jose Santos	University of A Coruña, Spain
Vitor Santos	University of Aveiro, Portugal
Ricardo Sanz	Universidad Politécnica de Madrid, Spain
Rafael Sanz Dominguez	Universidad de Vigo, Spain
Angel Sappa	CVC Barcelona, Spain
Colin Sauze	Aberystwyth University, UK
Sophia Schillai	University of Southampton, UK
Alexander Schlaefer	Hamburg University of Technology, Germany
Craig Schlenoff	NIST, USA
Michael Schukat	NUI Galway, Ireland
André Scolari	Federal University of Bahia, Brazil
Joao Sequeira	University of Lisboa, Portugal
João Silva	Altran Portugal, Portugal
Manuel F. Silva	Polytechnic Institute of Porto, Portugal
Daniel Skocaj	University of Ljubljana, Slovenia
Filomena Soares	University of Minho, Portugal
Armando Sousa	University of Porto, Portugal
Bo Tan	Tampere University, Finland
Danilo Tardioli	University of Zaragoza, Spain
Ana Maria Tome	University of Aveiro, Portugal
Carme Torras	Institut de Robòtica i Informàtica Industrial CSIC-UPC, Spain
Elisa Tosello	University of Padua, Italy
Jesus Urena	University of Alcalá, Spain
Alberto Vale	University of Lisboa, Portugal

Antonio Valente	University of Trás-os-Montes e Alto Douro, Portugal
Enrique Valero	Heriot Watt University, UK
Germano Veiga	INESC TEC, Portugal
Jose Vieira	University of Aveiro, Portugal
Antidio Viguria	Center for Advanced Aerospace Technologies, Spain
Blas M. Vinagre	University of Extremadura, Spain
David Vázquez Bermúdez	Computer Vision Center, Spain
Matias Waller	Åland University of Applied Sciences, Finland
Angel Garcia-Olaya	Universidad Carlos III de Madrid, Spain
Arturo Torres-González	Universidad de Sevilla, Spain
Augusto Gómez Egúiluz	Universidad de Sevilla, Spain
Bruno J. N. Guerreiro	University of Lisboa, Portugal
Fernando Gomez-Bravo	Universidad de Huelva, Spain
Francisco Javier Perez-Grau	CATEC, Spain
Ja Acosta	Universidad de Sevilla, Spain
Jesus Capitan	Universidad de Sevilla, Spain
Jose Joaquin Acevedo	Universidad de Sevilla, Spain
Julián Estévez	Universidad del País Vasco, Spain
Julio L. Paneque	Universidad de Sevilla, Spain
Marco Wehrmeister	Federal Technological University of Paraná, Brazil
P. J. Sanchez-Cuevas	Universidad de Sevilla, Spain
Pablo Ramon Soria	Universidad de Sevilla, Spain
Saeed Rafee Nekoo	Universidad de Sevilla, Spain
Shayok Mukhopadhyay	American University of Sharjah, Dubai
Valter Costa	INEGI, Portugal

## Additional Reviewers

Alireza Asvadi	Université de Bretagne Occidentale, France
Ana Lopes	Polytechnic Institute of Tomar, Portugal
Anis Koubaa	Prince Sultan University, Saudi Arabia
Carlos Gómez-Huélamo	University of Alcalá, Spain
Daniele Di Vito	University of Cassino and Southern Lazio, Italy
Edmanuel Cruz	University of Alicante, Spain
Félix Escalona	University of Alicante, Spain
Francisco Gomez-Donoso	University of Alicante, Spain
Francisco J Rodríguez Lera	University of León, Spain
Gledson Melotti	Federal Institute of Espírito Santo, Brazil
Juan Monroy	University of A Coruña, Spain
Kai Li	Polytechnic Institute of Porto, Portugal
Luís Ramos Pinto	University of Porto, Portugal

Luka Čehovin Zajc	University of Ljubljana, Slovenia
Miguel Armando Riem De Oliveira	University of Aveiro, Portugal
Mihail Babciński	University of Coimbra, Portugal
Nara Doria	Federal Institute of Sergipe, Brazil
Pedro Miraldo	University of Lisboa, Portugal
Pedro Santos	University of Porto, Portugal
Ramon Fernandes	Federal University of Bahia, Brazil
Ricardo Sutana de Mello	Brazil
Richard Duro	University of A Coruña, Spain
Tiago Trindade Ribeiro	Federal University of Bahia, Brazil
Vitor Hugo Pinto	University of Porto, Portugal

# Contents

## General Robotics

<b>Generalized Camera Array Model for Standard Plenoptic Cameras . . . . .</b>	3
Nuno Barroso Monteiro and José António Gaspar	
<b>Robot Navigation to Approach People Using <math>G^2</math>-Spline Path</b>	
<b>Planning and Extended Social Force Model . . . . .</b>	15
Marta Galvan, Ely Repiso, and Alberto Sanfeliu	
<b>Evaluation of SLAM Algorithms for Highly Dynamic Environments . . . . .</b>	28
Oliver Roesler and Vignesh Padubidri Ravindranath	
<b>An Aerial Robot Path Follower Based on the ‘Carrot Chasing’ Algorithm . . . . .</b>	37
Hector Perez-Leon, Jose Joaquin Acevedo, Jose A. Millan-Romera, Alejandro Castillejo-Calle, Ivan Maza, and Anibal Ollero	
<b>ROSS-LAN: RObotic Sensing Simulation Scheme for Bioinspired Robotic Bird LANding . . . . .</b>	48
Juan Pablo Rodríguez-Gómez, Augusto Gómez Eguíluz, José Ramiro Martínez-de Dios, and Aníbal Ollero	
<b>Low Cost Binaural System Based on the Echolocation . . . . .</b>	60
Thiago Fernandes Moya Moreira, José Lima, Paulo Costa, and Márcio Cunha	
<b>Fully Coupled Six-DoF Nonlinear Suboptimal Control of a Quadrotor: Application to Variable-Pitch Rotor Design . . . . .</b>	72
Saeed Rafee Nekoo, José Ángel Acosta, and Aníbal Ollero	
<b>Human-Robot Collaborative Navigation Search Using Social Reward Sources . . . . .</b>	84
Marc Dalmasso, Anaís Garrell, Pablo Jiménez, and Alberto Sanfeliu	

<b>Mind Perception of a Sociable Humanoid Robot: A Comparison Between Elderly and Young Adults .....</b>	96
Maryam Alimardani and Sonia Qurashi	
<b>Velocity-Based Heuristic Evaluation for Path Planning and Vehicle Routing for Victim Assistance in Disaster Scenarios .....</b>	109
Manuel Toscano-Moreno, Anthony Mandow, María Alcázar Martínez, and Alfonso García-Cerezo	
<b>Emergency Landing Spot Detection for Unmanned Aerial Vehicle .....</b>	122
Gabriel Loureiro, Luís Soares, André Dias, and Alfredo Martins	
<b>Intelligent Perception and Manipulation</b>	
<b>ROS Framework for Perception and Dual-Arm Manipulation in Unstructured Environments .....</b>	137
Delia Sepúlveda, Roemi Fernández, Eduardo Navas, Pablo González-de-Santos, and Manuel Armada	
<b>Modular Dual-Arm Robot for Precision Harvesting .....</b>	148
Eduardo Navas, Roemi Fernández, Delia Sepúlveda, Manuel Armada, and Pablo Gonzalez-de-Santos	
<b>Integrating Multiple Sources of Knowledge for the Intelligent Detection of Anomalous Sensory Data in a Mobile Robot .....</b>	159
Manuel Castellano-Quero, Juan-Antonio Fernández-Madrigal, and Alfonso J. García-Cerezo	
<b>A Manipulation Control Strategy for Granular Materials Based on a Gaussian Mixture Model .....</b>	171
Carlos M. Mateo, Juan Antonio Corrales, and Youcef Mezouar	
<b>Visual and Tactile Fusion for Estimating the Pose of a Grasped Object .....</b>	184
David Álvarez, Máximo A. Roa, and Luis Moreno	
<b>Object Classification for Robotic Platforms .....</b>	199
Samuel Brandenburg, Pedro Machado, Pranjali Shinde, João Filipe Ferreira, and T. M. McGinnity	
<b>Machine Learning in Robotics</b>	
<b>Effects of a Social Force Model Reward in Robot Navigation Based on Deep Reinforcement Learning .....</b>	213
Óscar Gil and Alberto Sanfeliu	
<b>A Cross-Situational Learning Based Framework for Grounding of Synonyms in Human-Robot Interactions .....</b>	225
Oliver Roesler	

<b>A Novel Model for Emotion Detection from Facial Muscles Activity . . . . .</b>	237
Elahe Bagheri, Azam Bagheri, Pablo G. Esteban, and Bram Vanderborght	
<b>Producing Parameterized Value Functions Through Modulation for Cognitive Developmental Robots . . . . .</b>	250
Alejandro Romero, Francisco Bellas, Jose A. Becerra, and Richard J. Duro	
<b>Benchmarking Deep and Non-deep Reinforcement Learning Algorithms for Discrete Environments . . . . .</b>	263
Fernando F. Duarte, Nuno Lau, Artur Pereira, and Luís P. Reis	
<b>Mobile Robots for Industrial Environments</b>	
<b>Versatile and Massive Experimentation of Robot Swarms in Industrial Scenarios . . . . .</b>	279
Vivian Cremer Kalempa, Marco Antonio Simoes Teixeira, and André Schneider de Oliveira	
<b>Edge-Enabled Autonomous Navigation and Computer Vision as a Service: A Study on Mobile Robot's Onboard Energy Consumption and Computing Requirements . . . . .</b>	291
Jens Lambrecht and Eugen Funk	
<b>A Bio-Inspired Approach for Robot Swarm in Smart Factories . . . . .</b>	303
Ronnier Frates Rohrich, Marco Antonio Simoes Teixeira, Luis Piardi, and André Schneider de Oliveira	
<b>Positioning System for UAV Precision Tasks Near Walls in GPS Denied and Metallic Environments . . . . .</b>	315
Felix Orjales, Javier Losada-Pita, Alejandro Paz-Lopez, and Alvaro Deibe	
<b>CRAS (Climbing Robot for Autonomous InSpection): The Challenges of a High-Temperature Tank . . . . .</b>	327
Nicolas Dalmedico, Higor Barbosa Santos, Juliano Scholz Slongo, Marco Antônio Simões Teixeira, Piatan Sfair Palar, Vinicius Vargas Terres, André Schneider de Oliveira, Lúcia Valéria Ramos de Arruda, Flávio Neves Júnior, and Julio Endress Ramos	
<b>Indoor Environment Monitoring in Search of Gas Leakage by Mobile Robot . . . . .</b>	339
João Braun, Luis Piardi, Thadeu Brito, José Lima, Ana Pereira, Paulo Costa, and Alberto Nakano	
<b>Performance of New Global Appearance Description Methods in Localization of Mobile Robots . . . . .</b>	351
Vicente Román, Luis Payá, María Flores, Sergio Cebollada, and Óscar Reinoso	

**Radar-Based Applications for Robotics**

- Asynchronous mmWave Radar Interference for Indoor Intrusion Detection . . . . .** 367

Edgar S. Gonçalves, Francisco C. Teixeira, Daniel F. Albuquerque,  
and Eurico F. Pedrosa

- Machine Learning Methods for Radar-Based People Detection and Tracking by Mobile Robots . . . . .** 379

José Castanheira, Francisco Curado, Eurico Pedrosa, Edgar Gonçalves,  
and Ana Tomé

- Tracking Drones with Drones Using Millimeter Wave Radar . . . . .** 392

Sedat Dogru, Rui Baptista, and Lino Marques

**Rehabilitation and Assistive Robotics**

- Classification of Daily Activities Using an Intelligent Tip for Crutches . . . . .** 405

Asier Brull, Aitor Gorrotxategi, Asier Zubizarreta, Itziar Cabanes,  
and Ana Rodriguez-Larrad

- ExoFlex: An Upper-Limb Cable-Driven Exosuit . . . . .** 417

David Pont, Aldo Francisco Contreras, José Luis Samper,  
Francisco Javier Sáez, Manuel Ferre, Miguel Ángel Sánchez,  
Ricardo Ruiz, and Ángel García

- Using a Collaborative Robot to the Upper Limb Rehabilitation . . . . .** 429

Lucas de Azevedo Fernandes, José Luis Lima, Paulo Leitão,  
and Alberto Yoshiro Nakano

- Grasping Angle Estimation of Human Forearm with Underactuated Grippers Using Proprioceptive Feedback . . . . .** 441

Francisco Pastor, Juan M. Gandarias, Alfonso J. García-Cerezo,  
Antonio J. Muñoz-Ramírez, and Jesús M. Gómez-de-Gabriel

- Integration of a Surgical Robotic Co-worker in an Endoscopic Neurosurgical Assistance Platform . . . . .** 453

Álvaro Muñoz García, Irene Rivas, Javier P. Turiel, Víctor Muñoz,  
Juan Carlos Fraile Marinero, Eusebio de la Fuente,  
and Jose María Sabater

- Smart Companion Pillow – An EPS@ISEP 2019 Project . . . . .** 465

Alexandre Soares dos Reis, Elien Gielen, Ko Wopereis, Marcel Pasternak,  
Vaido Sooäär, Tobias Schneider, Abel J. Duarte, Benedita Malheiro,  
Jorge Justo, Cristina Ribeiro, Manuel F. Silva, Paulo Ferreira,  
and Pedro Guedes

<b>Advantages of the Incorporation of an Active Upper-Limb Exoskeleton in Industrial Tasks . . . . .</b>	<b>477</b>
Andrea Blanco, José M. Catalán, Jorge A. Díez, José V. García, Luis D. Lledó, Emilio Lobato, and Nicolás M. García-Aracil	
<b>Assistance Strategies for Robotized Laparoscopy . . . . .</b>	<b>485</b>
Alicia Casals, Albert Hernansanz, Narcís Sayols, and Josep Amat	
<b>Suitable Task Allocation in Intelligent Systems for Assistive Environments . . . . .</b>	<b>497</b>
Manuel Vinagre, Joan Aranda, and Alicia Casals	
 <b>Simulation in Robotics</b>	
<b>Coverage Path Planning Optimization for Slopes and Dams Inspection . . . . .</b>	<b>513</b>
Iago Z. Biundini, Aurelio G. Melo, Milena F. Pinto, Guilherme M. Marins, Andre L. M. Marcato, and Leonardo M. Honorio	
<b>Evaluating an AEF Swimming Microrobot Using a Hardware-in-the-loop Testbed . . . . .</b>	<b>524</b>
José Emilio Traver, Inés Tejado, Cristina Nuevo-Gallardo, Javier Prieto-Arranz, Miguel A. López, and Blas M. Vinagre	
<b>Learning Low-Level Behaviors and High-Level Strategies in Humanoid Soccer . . . . .</b>	<b>537</b>
David Simões, Pedro Amaro, Tiago Silva, Nuno Lau, and Luís Paulo Reis	
 <b>Workshop on Physical Agents</b>	
<b>3D Hand Joints Position Estimation with Graph Convolutional Networks: A GraphHands Baseline . . . . .</b>	<b>551</b>
John-Alejandro Castro-Vargas, Alberto Garcia-Garcia, Sergiu Oprea, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez	
<b>Map Slammer: Densifying Scattered KSLAM 3D Maps with Estimated Depth . . . . .</b>	<b>563</b>
Jose Miguel Torres-Camara, Felix Escalona, Francisco Gomez-Donoso, and Miguel Cazorla	
<b>Integration of the Alexa Assistant as a Voice Interface for Robotics Platforms . . . . .</b>	<b>575</b>
Alejandro Hidalgo-Paniagua, Andrés Millan-Alcaide, Juan P. Bandera, and Antonio Bandera	

<b>Adapting ROS Logs to Facilitate Transparency and Accountability in Service Robotics . . . . .</b>	587
Francisco Javier Rodríguez-Lera, Ángel Manuel Guerrero-Higuera, Francisco Martín-Rico, Jonathan Gines, Juan Felipe García Sierra, and Vicente Matellán-Olivera	
<b>A Story-Telling Social Robot with Emotion Recognition Capabilities for the Intellectually Challenged . . . . .</b>	599
David Azuar, Guillermo Gallud, Felix Escalona, Francisco Gomez-Donoso, and Miguel Cazorla	
<b>A Novel Inter-device Calibration for Wi-Fi-aided Indoor Localization Systems . . . . .</b>	610
Miguel Martínez del Horno, Cristina Romero-González, Luis Orozco-Barbosa, and Ismael García-Varea	
<b>A Review of Segmentation Methods for 3D Semantic Mapping . . . . .</b>	620
Cristina Romero-González, Jesus Martínez-Gómez, and Ismael García-Varea	
<b>Learning from the Individuals and the Crowd in Robotics and Mobile Devices . . . . .</b>	632
Fernando E. Casado, Dylan Lema, Roberto Iglesias, Carlos V. Regueiro, and Senén Barro	
<b>Socially-Accepted Path Planning for Robot Navigation Based on Social Interaction Spaces . . . . .</b>	644
Araceli Vega, Ramón Cintas, Luis J. Manso, Pablo Bustos, and Pedro Núñez	
<b>A Modelling and Formalisation Tool for Use Case Design in Social Autonomous Robotics . . . . .</b>	656
Alba Gragera, Alba María García, and Fernando Fernández	
<b>People Detection and Tracking Using an On-Board Drone Camera . . . . .</b>	668
Cristian Cifuentes-García, Daniel González-Medina, and Ismael García-Varea	
<b>Design of a Robotic as a Service Platform to Perform Rehabilitation Therapies . . . . .</b>	681
Adrián Gallego, José Carlos Pulido, José Carlos González, and Fernando Fernández	
<b>Author Index . . . . .</b>	693

# **General Robotics**



# Generalized Camera Array Model for Standard Plenoptic Cameras

Nuno Barroso Monteiro<sup>1,2</sup>✉ and José António Gaspar<sup>1</sup>

<sup>1</sup> Institute for Systems and Robotics, University of Lisbon, Lisbon, Portugal  
{nmonteiro,jag}@isr.tecnico.ulisboa.pt

<sup>2</sup> Institute for Systems and Robotics, University of Coimbra, Coimbra, Portugal

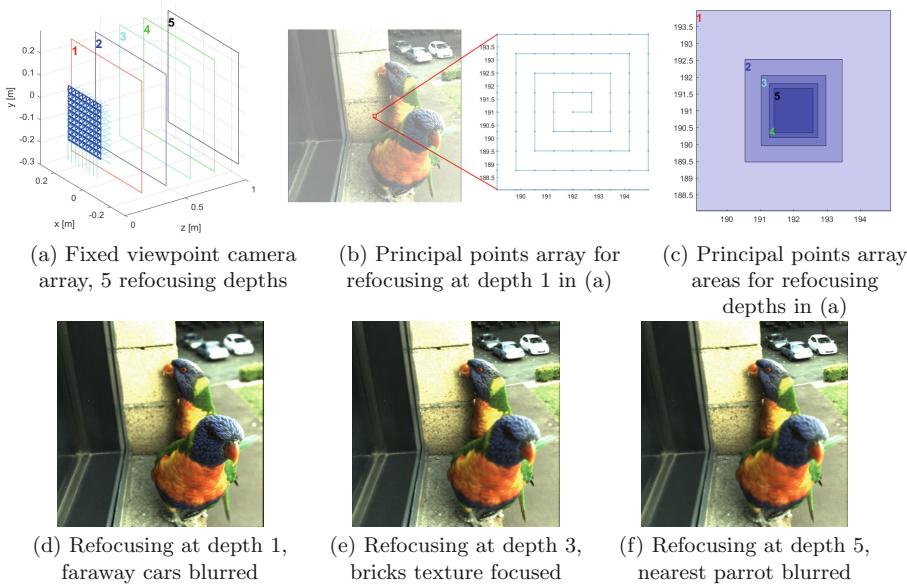
**Abstract.** Plenoptic cameras discriminate the contribution of each ray emanating from a particular point by placing a microlens array between the main lens and the image sensor. The collection of rays captured by these cameras can represent the physical microlens camera array or can be rearranged to represent a virtual camera array with a very narrow baseline (viewpoint camera array). In this work, we extend the common camera arrays considered for standard plenoptic cameras (SPCs) and define the geometry associated with the different virtual camera arrays that can be obtained by shearing the lightfield (LF). This geometry is validated using a publicly available calibration dataset and calibration toolbox. The results show that the geometry proposed is capable of describing the multiple viewpoint and microlens camera arrays.

**Keywords:** Standard plenoptic camera · Viewpoint camera array · Microlens camera array · Shearing

## 1 Introduction

Plenoptic cameras discriminate the contribution of each light ray that emanate from a given point due to the positioning of a microlens array between the main lens and the image sensor. This allows to project a point in the scene onto several positions of the sensor (Fig. 2a). The collection of rays acquired by these cameras is called a lightfield [7, 11].

In this work, we will focus on the standard plenoptic camera (SPC) [15] whose geometry generates unfocused microlens images (MIs) (Fig. 2c). SPCs define several types of camera arrays by reorganizing the pixels captured by the camera on the 2D raw image (Fig. 2b) [15]. The raw image displays the pixels collected by each microlens in the microlens array (Fig. 2c) and represents the images captured by the physical microlens array placed in front of the sensor. There is another arrangement of pixels that is commonly used in SPCs, the viewpoint images (VIs). These images are obtained by selecting the same pixel position relatively to the microlens center for each microlens [15]. This rearrangement defines a virtual camera array with co-planar projection centers and with a very narrow baseline [1].



**Fig. 1.** Viewpoint camera arrays obtained considering shearing for refocusing at depths  $z = 0.2, 0.4, \dots, 1.0$  m (a). The spacing among projection centers has been scaled 100 times to be perceptible on the 3D plot. The distribution of the principal points for the viewpoint camera arrays at different refocusing depths are depicted in (b) and (c). The corresponding refocused images are depicted in (d), (e) and (f).

A SPC allows to define additional cameras that collect rays that intersect at an arbitrary point in the scene [13] either by applying a shearing operation or creating surface camera images (SCams). Although these strategies are commonly used for disparity estimation [5, 16], the geometry associated with the corresponding cameras has not been defined. In this work, we derive the mappings between a SPC [6] and the multiple viewpoint and microlens camera arrays that can be obtained from this camera.

In terms of structure, we present in Sect. 2 a review of the camera array mappings for SPCs. In Sect. 3, we introduce the SPC model and the viewpoint and microlens camera arrays mappings considered in the literature from this model. The generalized mappings proposed for the viewpoint and microlens camera arrays are presented in Sect. 4. In Sect. 5, these mappings are validated experimentally and the major conclusions are presented in Sect. 6.

**Notation:** Italic letters correspond to scalars, lower case bold letters correspond to vectors, and upper case bold letters correspond to matrices. Vectors represented in homogeneous coordinates are denoted by  $(\tilde{\cdot})$ .

## 2 Related Work

Dansereau *et al.* [6] proposed a mapping between the LF in the image space defined in pixels and microlenses indices and the LF in the object space defined by a position and a direction in metric units (Fig. 2a). Nonetheless, there is not provided a connection between this mapping and the projection matrix for either the microlens or viewpoint cameras. The definition of the projection matrices for the microlens and the viewpoint cameras appeared in the work of Bok *et al.* [4]. The geometry of the camera arrays is described using the parameters of the optical setup and the knowledge of the corresponding microlenses centers in the raw image but no relationship with the originally proposed model for SPCs [6] is provided. Additionally, the geometry proposed for the viewpoint cameras does not explain the zero disparity for points in the world focal plane of the main lens.

Marto *et al.* [12] represented a camera array composed of identical co-planar cameras using a mapping similar to the one proposed by Dansereau *et al.* [6]. However, the mapping proposed does not explain the zero disparity. The mapping between the SPC model [6] and the viewpoint camera array that is consistent with the zero disparity for points in the world focal plane of the main lens is described in [1]. This model for the viewpoint cameras consider co-planar cameras with a shifted principal point among the different viewpoint cameras. The corresponding mapping between the SPC model [6] and the microlens camera array is described in [2].

In this work, we extend the characterization of the microlens and viewpoint camera arrays found in the literature [1, 2, 4] and define the geometry of the several cameras that can be defined by collecting the rays captured by a SPC that intersect in an arbitrary point in the object space.

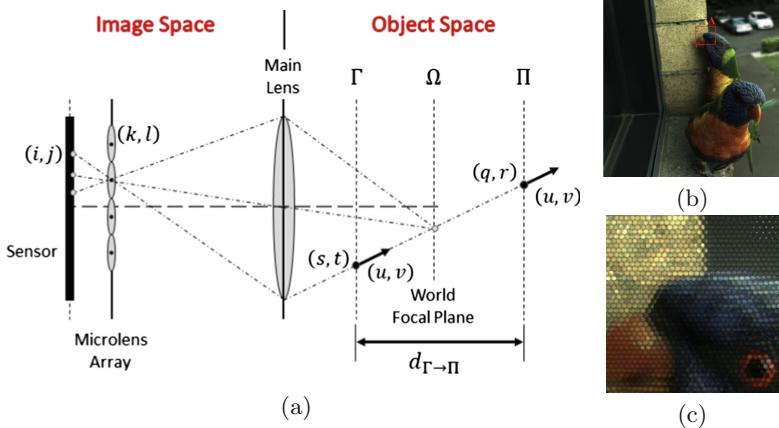
## 3 Standard Plenoptic Camera

A SPC can be represented by a  $5 \times 5$  matrix  $\mathbf{H}$  [6] which maps rays  $\tilde{\Phi} = [i, j, k, l, 1]^T$  in the image space to rays  $\tilde{\Psi} = [s, t, u, v, 1]^T$  in the object (metric) space:

$$\tilde{\Psi} = \mathbf{H} \tilde{\Phi} \quad (1)$$

where rays  $\Phi$  are parameterized using pixels  $(i, j)$  and microlenses  $(k, l)$  indices and rays  $\Psi$  are parameterized using a position  $(s, t)$  and a direction  $(u, v)$  defined on a plane  $\Gamma$  in metric units [14] (Fig. 2a). The mapping  $\mathbf{H}$  [6] has 12 non-zero entries

$$\mathbf{H} = \begin{bmatrix} h_{si} & 0 & h_{sk} & 0 & h_s \\ 0 & h_{tj} & 0 & h_{tl} & h_t \\ h_{ui} & 0 & h_{uk} & 0 & h_u \\ 0 & h_{vj} & 0 & h_{vl} & h_v \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \quad (2)$$



**Fig. 2.** Geometry of a SPC whose main lens focal plane corresponds to plane  $\Omega$  (a). The LF in the image space is parameterized using pixels  $(i, j)$  and microlenses  $(k, l)$  indices while the LF in the object space is parameterized using a point  $(s, t)$  and a direction  $(u, v)$  defined on the parameterization plane  $\Gamma$ . (b) shows a raw image acquired with a SPC with the mains lens world focal plane placed near the wall and (c) exhibits the details of the microlenses in red box A.

In the following, we denote  $\mathbf{H}$  as the lightfield intrinsics matrix (LFIM)<sup>1</sup>.

One ray  $\Phi = [s, t, u, v]^T$  can be represented as one parametric 3D line [8], namely  $[x, y, z]^T = [s, t, 0]^T + \lambda[u, v, 1]^T$  for  $\lambda \in \mathbb{R}$ . Therefore, the LFIM matrix (2) allows to define the relationship between an arbitrary point  $[x, y, z]^T$  in the object space and the ray  $\Phi$  in the image space [14] as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{H}_{ij}^{st} \begin{bmatrix} i \\ j \end{bmatrix} + \mathbf{H}_{kl}^{st} \begin{bmatrix} k \\ l \end{bmatrix} + \mathbf{h}_{st} + z \left( \mathbf{H}_{ij}^{uv} \begin{bmatrix} i \\ j \end{bmatrix} + \mathbf{H}_{kl}^{uv} \begin{bmatrix} k \\ l \end{bmatrix} + \mathbf{h}_{uv} \right) \quad (3)$$

where the LFIM is partitioned in four  $2 \times 2$  sub-matrices and two  $2 \times 1$  vectors  $\mathbf{h}_{st} = [h_s, h_t]^T$  and  $\mathbf{h}_{uv} = [h_u, h_v]^T$ . The sub-matrices follow the notation  $\mathbf{H}_{(\cdot)}$  where the subscript selects the columns and the superscript selects the lines, *i.e.* for example,  $\mathbf{H}_{ij}^{st}$  selects the first two columns, denoted by  $ij$ , and the first two lines, denoted by  $st$ .

**Viewpoint Camera Array.** The SPC can be represented by a camera array of viewpoints [1]. Let us represent the viewpoint camera array by a parametric projection matrix  $\mathbf{P}^{ij}$  varying with the coordinates  $(i, j)$

$$\mathbf{P}^{ij} = \mathbf{K}^{ij} [\mathbf{I}_{3 \times 3} \mathbf{t}^{ij}] {}^c \mathbf{T}_w \quad (4)$$

<sup>1</sup> We note that LFIM is a simplified term, as  $\mathbf{H}$  effectively contains intrinsic parameters information, however, it also contains baseline information, as detailed in Sect. 3. Conventional extrinsic parameters, as found in pinhole camera models, defining a world coordinate system, are in fact not contained in  $\mathbf{H}$ .

where  $\mathbf{K}^{ij}$  denotes the intrinsic matrix,  $\mathbf{I}_{3 \times 3}$  is a  $3 \times 3$  identity matrix,  $\mathbf{t}^{ij}$  is the projection center and  ${}^c\mathbf{T}_w = \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{t}_w \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$  defines the rigid body transformation between the world and camera coordinate systems with rotation  ${}^c\mathbf{R}_w \in SO(3)$  and translation  ${}^c\mathbf{t}_w \in \mathbb{R}^3$ , and  $\mathbf{0}_{1 \times 3}$  is the  $1 \times 3$  null matrix. Note that the intrinsic matrix and the projection center are different for each viewpoint camera  $(i, j)$ . More in detail, the intrinsic camera model takes into account that the principal point is different for each viewpoint while the scale factor remains the same. The intrinsic matrix and projection center are

$$\mathbf{K}^{ij} = \begin{bmatrix} \frac{1}{h_{uk}} & 0 & -\frac{h_u}{h_{uk}} - i \frac{h_{ui}}{h_{uk}} \\ 0 & \frac{1}{h_{vl}} & -\frac{h_v}{h_{vl}} - j \frac{h_{vj}}{h_{vl}} \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{t}^{ij} = - \begin{bmatrix} h_s - \frac{h_{sk}}{h_{uk}} h_u + i \left( h_{si} - \frac{h_{sk}}{h_{uk}} h_{ui} \right) \\ h_t - \frac{h_{tl}}{h_{vl}} h_v + j \left( h_{tj} - \frac{h_{tl}}{h_{vl}} h_{vj} \right) \\ -\frac{h_{sk}}{h_{uk}} \end{bmatrix}. \quad (5)$$

**Microlens Camera Array.** The SPCs can also be represented by a microlens camera array [2]. Let us represent the microlens camera array by a parametric projection matrix  $\mathbf{P}^{kl}$  varying with the coordinates  $(k, l)$

$$\mathbf{P}^{kl} = \mathbf{K}^{kl} \left[ \mathbf{I}_{3 \times 3} \ \mathbf{t}^{kl} \right] {}^c\mathbf{T}_w \quad (6)$$

where  $\mathbf{K}^{kl}$  denotes the intrinsic matrix and  $\mathbf{t}^{kl}$  is the projection center. As for the viewpoint camera array, the intrinsic matrix and the projection center are different for each microlens camera  $(k, l)$ . Namely, the intrinsic camera model takes into account that the principal point is different for each microlens while the scale factor remains the same. The intrinsic matrix and projection center are

$$\mathbf{K}^{kl} = \begin{bmatrix} \frac{1}{h_{ui}} & 0 & -\frac{h_u}{h_{ui}} - k \frac{h_{uk}}{h_{ui}} \\ 0 & \frac{1}{h_{vj}} & -\frac{h_v}{h_{vj}} - l \frac{h_{vl}}{h_{vj}} \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{t}^{kl} = - \begin{bmatrix} h_s - \frac{h_{si}}{h_{ui}} h_u + k \left( h_{sk} - \frac{h_{si}}{h_{ui}} h_{uk} \right) \\ h_t - \frac{h_{tj}}{h_{vj}} h_v + l \left( h_{tl} - \frac{h_{tj}}{h_{vj}} h_{vl} \right) \\ -\frac{h_{si}}{h_{ui}} \end{bmatrix}. \quad (7)$$

## 4 Generalized Camera Arrays

In Sect. 3, one defined a parametric projection matrix to define either a viewpoint or a microlens array. In this section, one shows that a plenoptic camera can define multiple camera arrays by collecting rays with different combinations of pixel and microlens coordinates.

Consider the LF in the object space  $L_\Gamma(s, t, u, v)$  acquired by a plenoptic camera with the plane  $\Omega$  in focus (Fig. 2a). The LF captured by the plenoptic camera can be defined on another plane by shifting the parameterization plane along the normal to the plane  $\Gamma$ . Assuming that the plane  $\Pi$  is at a distance  $d_{\Gamma \rightarrow \Pi}$  from the plane  $\Gamma$ , one can re-parameterize the LF captured by the

plenoptic camera relatively to the plane  $\Pi$  [3],  $L_\Pi(q, r, u, v)$ , by  $\tilde{\Psi}_\Pi = \mathbf{D} \tilde{\Psi}$  where

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & d_{\Gamma \rightarrow \Pi} & 0 & 0 \\ 0 & 1 & 0 & d_{\Gamma \rightarrow \Pi} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

and  $\tilde{\Psi}_\Pi = [q, r, u, v, 1]^T$  correspond to rays parameterized by a point  $(q, r)$  and a direction  $(u, v)$  on plane  $\Pi$ . Mapping the LF in the object space  $L_\Gamma(s, t, u, v)$  to the LF in the image space  $L(i, j, k, l)$  by the intrinsic matrix  $\mathbf{H}$  (2), one obtains

$$\tilde{\Psi}_\Pi = \mathbf{D} \mathbf{H} \tilde{\Phi} . \quad (9)$$

The new intrinsic matrix  $\mathbf{H}_\Pi = \mathbf{D}\mathbf{H}$  allows to relate the LF in the object space  $L_\Pi(q, r, u, v)$  and the LF in the image space  $L(i, j, k, l)$ . The re-parameterization (9) allows to define a constraint to identify the rays that intersect at an arbitrary point of the plane  $\Pi$  [13]. Let  $\Phi_a$  and  $\Phi_b$  be two rays in the image space with the same coordinates  $(q, r)$  on plane  $\Pi$ , by taking their difference one defines a constraint on the LF coordinates in the image space as

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{H}_{ij}^{qr} \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix} + \mathbf{H}_{kl}^{qr} \begin{bmatrix} \Delta k \\ \Delta l \end{bmatrix} \quad (10)$$

where  $\Delta(\cdot) = (\cdot)_b - (\cdot)_a$ , and  $\mathbf{H}_{(\cdot)}^{qr}$  corresponds to  $2 \times 2$  sub-matrices of  $\mathbf{H}_\Pi$  obtained from selecting the entries of the first two rows, denoted by  $qr$ , and selecting either the entries of the  $1^{st}$  and  $2^{nd}$  columns, denoted by  $ij$ , or the  $3^{rd}$  and  $4^{th}$  columns, denoted by  $kl$ . Using the constraint (10) and considering  $(i_r, j_r)$  as reference coordinates to enforce the constraint, one defines a sampling on the viewpoint coordinates  $(i, j)$  as

$$k_S = k + \beta_{ik} (i - i_r) \wedge l_S = l + \beta_{jl} (j - j_r) \quad (11)$$

where the parameters  $\beta_{ik} = \frac{h_{si} + d_{\Gamma \rightarrow \Pi} h_{ui}}{h_{sk} + d_{\Gamma \rightarrow \Pi} h_{uk}}$  and  $\beta_{jl} = \frac{h_{tj} + d_{\Gamma \rightarrow \Pi} h_{vj}}{h_{tl} + d_{\Gamma \rightarrow \Pi} h_{vl}}$  correspond to the disparities considered on the VIs for a point at depth  $d_{\Gamma \rightarrow \Pi}$ .

The sampling (11) corresponds to the sampling performed during the shearing operation defined by Tao *et al.* [16]. The shearing can be interpreted as a redefinition of the epipolar plane images (EPIS) [15, 16] of the acquired LF  $L(i, j, k, l)$  according to a given slope that corresponds to disparity on the VIs. Assuming that  $\beta_{ik} = \beta_{jl} = \beta$  and denoting the rays in the sheared LF as  $\Phi_S = [i, j, k_S, l_S]^T$ , the relationship between the rays of the acquired and the sheared LF (11) can be redefined as

$$\tilde{\Phi}_S = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \beta & 0 & 1 & 0 & -\beta i_r \\ 0 & \beta & 0 & 1 & -\beta j_r \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{U}} \tilde{\Phi}. \quad (12)$$

The matrix  $\mathbf{U}$  allows to define a virtual microlens camera  $(k_S, l_S)$  that collects rays from the acquired LF that intersect at a common point in plane  $\Pi$ . Let us analyze the influence of shearing on the camera arrays presented in Sect. 3.

**Viewpoint Camera Array.** Following the strategy defined in [1] to compute the caustic surface, one can conclude that the caustic profile for the non-sheared and sheared viewpoint camera  $(i, j)$  is the same. Namely, the constraint to ensure a unique projection center does not change as well as the location of the projection center relatively to the non-sheared viewpoint camera. The projection matrix  $\mathbf{P}_S^{ij}$  for the sheared viewpoint camera is obtained considering the back-projection equation (3) redefined with the LFIM  $\mathbf{H}_S = \mathbf{H} \mathbf{U}$  and solving relatively to  $(k, l)$ . This gives a projection matrix defined by

$$\mathbf{P}_S^{ij} = \mathbf{K}_S^{ij} [\mathbf{I}_{3 \times 3} \mathbf{t}^{ij}]^c \mathbf{T}_w \quad (13)$$

where  $\mathbf{K}_S^{ij}$  denotes the intrinsic matrix for the sheared viewpoint camera. More in detail, the camera model for the sheared viewpoint camera only differs on the principal point relatively to the non-sheared viewpoint camera counterpart (4) (Fig. 1b–c), which is consistent with the strategy to translate the VIs to perform shearing of the LF [10, 16]. The intrinsic matrix is given by

$$\mathbf{K}_S^{ij} = \begin{bmatrix} \frac{1}{h_{uk}} & 0 & -\frac{h_u}{h_{uk}} - i \frac{h_{ui}}{h_{uk}} - \beta(i - i_r) \\ 0 & \frac{1}{h_{vl}} & -\frac{h_v}{h_{vl}} - j \frac{h_{vj}}{h_{vl}} - \beta(j - j_r) \\ 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

Setting  $\beta = 0$ , the intrinsic matrix  $\mathbf{K}_S^{ij}$  (14) reduces to the intrinsic matrix  $\mathbf{K}^{ij}$  (5).

**Microlens Camera Array.** Following the strategy defined in [2] to compute the caustic surface, one can conclude that the caustic profile for the sheared microlens camera  $(k, l)$  is different from the non-sheared counterpart. Namely, the constraint to ensure a unique projection center is given by

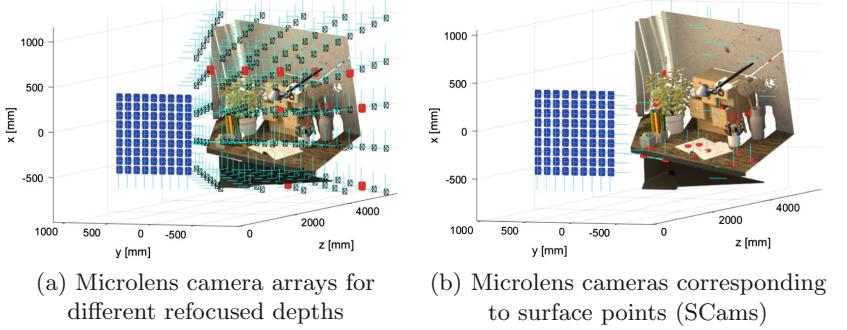
$$\frac{h_{si} + \beta h_{sk}}{h_{ui} + \beta h_{uk}} = \frac{h_{tj} + \beta h_{tl}}{h_{vj} + \beta h_{vl}}, \quad (15)$$

and the projection center is defined on a plane at a depth  $z_\beta = -\frac{h_{si} + \beta h_{sk}}{h_{ui} + \beta h_{uk}}$  (Fig. 3) by

$$\mathbf{t}_S^{kl} = - \begin{bmatrix} h_s + z_\beta h_u + (h_{sk} + z_\beta h_{uk})(k - \beta i_r) \\ h_t + z_\beta h_v + (h_{tl} + z_\beta h_{vl})(l - \beta j_r) \\ z_\beta \end{bmatrix}. \quad (16)$$

The projection matrix  $\mathbf{P}_S^{kl}$  for the sheared microlens camera is obtained considering the back-projection equation (3) redefined with the LFIM  $\mathbf{H}_S$  and solving relatively to  $(i, j)$ . This gives a projection matrix defined by

$$\mathbf{P}_S^{kl} = \mathbf{K}_S^{kl} [\mathbf{I}_{3 \times 3} \mathbf{t}_S^{kl}]^c \mathbf{T}_w \quad (17)$$



**Fig. 3.** Microlens camera arrays obtained considering shearing for refocusing at different depths for the synthetic Table dataset [9]. Shearing allows to obtain microlens cameras with projection centers at different depths (a). These cameras obtain relevant information for depth estimation [5] when the projection center corresponds to a surface point, *i.e.* a SCam is defined (b). The viewpoint camera array is represented in blue with the spacing among projection centers scaled by 4 times.

where  $\mathbf{K}_S^{kl}$  denotes the intrinsic matrix for the sheared microlens camera and is given by

$$\mathbf{K}_S^{kl} = \begin{bmatrix} \frac{1}{h_{ui} + \beta h_{uk}} & 0 & -\frac{h_u - \beta h_{uk} i_r}{h_{ui} + \beta h_{uk}} - k \frac{h_{uk}}{h_{ui} + \beta h_{uk}} \\ 0 & \frac{1}{h_{vj} + \beta h_{vl}} & -\frac{h_v - \beta h_{vl} j_r}{h_{vj} + \beta h_{vl}} - l \frac{h_{vl}}{h_{vj} + \beta h_{vl}} \\ 0 & 0 & 1 \end{bmatrix}. \quad (18)$$

Setting  $\beta = 0$ , the intrinsic matrix  $\mathbf{K}_S^{kl}$  (18) and the projection center  $\mathbf{t}_S^{kl}$  (16) reduce to the intrinsic matrix  $\mathbf{K}^{kl}$  and projection center  $\mathbf{t}^{kl}$  defined in (7). Additionally, if we replace  $\beta = -\frac{h_{si} + d_{\Gamma \rightarrow \Pi} h_{ui}}{h_{sk} + d_{\Gamma \rightarrow \Pi} h_{uk}}$  in  $z_\beta$ , one can see that the depth of the projection center corresponds to the plane  $\Pi$  at  $d_{\Gamma \rightarrow \Pi}$ .

**Generalized EPI Geometry.** Considering Eq. (3) and the sheared viewpoint cameras (13), one can obtain the EPI geometry that relates the depth of a point with the disparity on the VIs  $\left[ \frac{\Delta k}{\Delta i}, \frac{\Delta l}{\Delta j} \right]^T$  for the sheared LF

$$\frac{\Delta k}{\Delta i} = -\frac{h_{si} + zh_{ui}}{h_{sk} + zh_{uk}} - \beta \quad \text{and} \quad \frac{\Delta l}{\Delta j} = -\frac{h_{tj} + zh_{vj}}{h_{vl} + zh_{vl}} - \beta. \quad (19)$$

The EPI geometry shows that the zero disparity plane, also known as the optical focal plane [15] of the SPC main lens is affected by the shearing operation. This is in accordance with the creation of a virtual focal plane during the refocus operation that implicitly requires a shearing of the LF [15] (Fig. 1d–f). These equations reduce to the ones presented in [12] for  $\beta = 0$ .

## 5 Experimental Results

In this section, the mappings proposed in Sect. 4 are validated experimentally using the publicly available calibration dataset [6] (Dataset A) acquired with

a 1<sup>st</sup> generation Lytro camera. Namely, the viewpoint and microlens cameras obtained after calibration of the sheared versions of the calibration dataset LFs are compared with the cameras obtained using the mappings proposed in Sect. 4 with the LFIM obtained from the calibration of the non-sheared calibration dataset.

Let us start by calibrating the non-sheared calibration dataset using the calibration procedure [6]. The estimated LFIM  $\mathbf{H}$  (2) and the corresponding viewpoint (4) and microlens (6) cameras are given in Tables 1 and 2, respectively, where  $k_{nm}^{(\cdot)}$  denotes the entry  $(n, m)$  of the intrinsic matrix and  $t_n^{(\cdot)}$  denotes the entry  $n$  of the projection center associated with the viewpoint  $(i, j)$  or microlens  $(k, l)$ . Using the values in Table 1 and the mappings (13) and (17), one obtains the characterization of the camera arrays for different values of disparity  $\beta$ .

**Table 1.** LFIM obtained after calibration of Dataset A [6] with  $h_{sk} = h_{tl} = 0$ .

$h_{si}$	$h_s$	$h_{tj}$	$h_t$	$h_{ui}$	$h_{uk}$	$h_u$	$h_{vj}$	$h_{vl}$	$h_v$
0.0003	-0.0013	0.0003	-0.0013	-0.0011	0.0019	-0.3508	-0.0011	0.0019	-0.3515

**Table 2.** Intrinsic matrices and projection centers for viewpoint and microlens cameras. These values are obtained after applying the mappings (5) and (7) with  $\Delta i = \Delta j = 1$  and  $\Delta k = \Delta l = 1$ , respectively.

$k_{11}^{ij}$	$k_{22}^{ij}$	$k_{13}^{ij}$	$k_{23}^{ij}$	$t_1^{ij}$	$t_2^{ij}$	$t_3^{ij}$	$k_{11}^{kl}$	$k_{22}^{kl}$	$k_{13}^{kl}$	$k_{23}^{kl}$	$t_1^{kl}$	$t_2^{kl}$	$t_3^{kl}$
538.6	534.9	189.6	188.6	0.001	0.001	0	881.0	892.0	-307.5	-311.9	-0.081	-0.081	-0.227

The characterization of the viewpoint and microlens cameras obtained using the mappings proposed in Sect. 4 is compared with the characterization obtained by applying (4) and (6) to the LFIM obtained from the calibration of the sheared versions of the calibration dataset LFs. The sheared LFs are obtained considering different disparities  $\beta$  for the re-parameterization of the EPIS (shearing). The disparities considered range from 0.1 to 2.0 pixels. Figure 4 depicts the entries of the viewpoint intrinsic matrix and projection center with the disparity  $\beta$  used for shearing considering a unitary displacement from the reference viewpoint  $(i_r, j_r)$ , *i.e.*  $\Delta i = \Delta j = 1$ . Similarly, Fig. 5 depicts the entries of the microlens intrinsic matrix and projection center considering  $\Delta k = \Delta l = 1$ . Tables 3 and 4 represent the mean and Standard Deviation (STD) of the errors  $\epsilon_{(\cdot)} = |(\cdot)^M - (\cdot)^E| / |(\cdot)^M|$ , in percentage, for each entry of the intrinsic matrix and projection center for the viewpoint and microlens camera, respectively. In the error  $\epsilon_{(\cdot)}$ ,  $(\cdot)^M$  corresponds to the entries obtained from the mappings (13) and (17), and  $(\cdot)^E$  corresponds to the entries obtained from the mappings (4) and (6).

The viewpoint mapping (13) models the changes with the disparity  $\beta$  very accurately (Fig. 4). In Table 3, one can see that the mean error is below 0.2% which shows that the estimate values are in accordance with the mapping (13).

The difference on the estimated values appears to be the result of the interpolation and discretization that occurs in the shearing operation. This also affects the position of the detected corners that are used in the calibration.

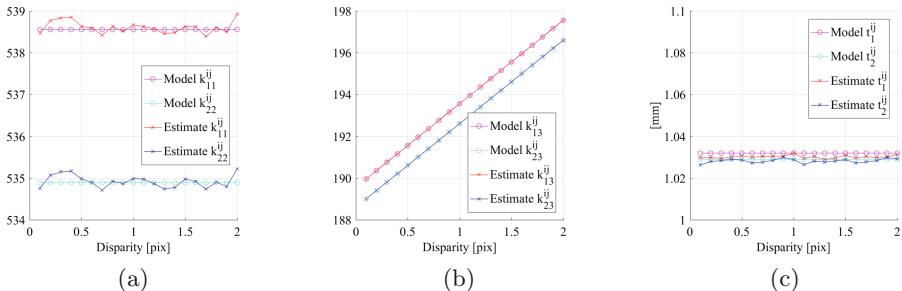
The microlens mapping (17) also models the changes with the disparity  $\beta$  very accurately except for  $\beta = 0.6$  (Fig. 5). This disparity value is close to the singularity that occurs for  $\beta = -h_{ui}/h_{uk} = 0.611$  which causes some numerical instability in the mapping. Indeed, in Table 4, one can see that the mean error considering all disparity values is below 4.5%. Nonetheless, removing the disparity  $\beta = 0.6$ , one obtains a mean error below 0.5% which shows that the estimate values are in accordance with the mapping (17). Notice that the viewpoint mapping obtains a lower error than the microlens mapping. This can be justified by the strategy of the calibration procedure [6] that calibrates a SPC using detected corners on VIs.

**Table 3.** Mean and STD error, in percentage, for each entry of the viewpoint intrinsic matrix and projection center.

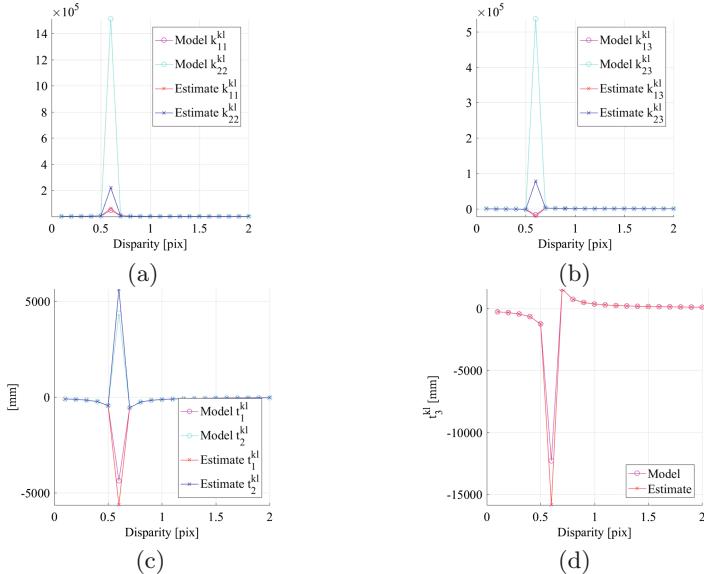
$k_{11}^{ij}$	$k_{22}^{ij}$	$k_{13}^{ij}$	$k_{23}^{ij}$	$t_1^{ij}$	$t_2^{ij}$
$0.022 \pm 0.018$	$0.022 \pm 0.017$	$0.004 \pm 0.003$	$0.002 \pm 0.002$	$0.174 \pm 0.067$	$0.100 \pm 0.075$

**Table 4.** Mean and STD error, in percentage, for each entry of the microlens intrinsic matrix and projection center. First line considers all disparity values while the second line excludes the disparity  $\beta = 0.6$ .

$k_{11}^{kl}$	$k_{22}^{kl}$	$k_{13}^{kl}$	$k_{23}^{kl}$	$t_1^{kl}$	$t_2^{kl}$	$t_3^{kl}$
$1.85 \pm 6.45$	$4.45 \pm 19.07$	$1.84 \pm 6.45$	$4.44 \pm 19.07$	$1.89 \pm 6.38$	$1.89 \pm 6.38$	$1.90 \pm 6.38$
$0.41 \pm 0.44$	$0.19 \pm 0.18$	$0.40 \pm 0.44$	$0.18 \pm 0.18$	$0.47 \pm 0.42$	$0.47 \pm 0.42$	$0.48 \pm 0.43$



**Fig. 4.** Variation of viewpoint camera intrinsic matrix and projection center with disparity  $\beta$  for shearing. These entries are estimated considering that  $\Delta i = \Delta j = 1$ . The scale factors of the intrinsic matrix  $k_{11}^{ij}$  and  $k_{22}^{ij}$  are represented in (a). The principal point  $[k_{13}^{ij}, k_{23}^{ij}]^T$  is depicted in (b). In (c), the  $x$ - and  $y$ - components of the projection are presented. The  $z$ -component of the projection center is not represented since it is always zero regardless of the disparity  $\beta$  considered for shearing.



**Fig. 5.** Variation of microlens camera intrinsic matrix and projection center with disparity  $\beta$  for shearing. These entries are estimated considering that  $\Delta k = \Delta l = 1$ . The scale factors of the intrinsic matrix  $k_{11}^{kl}$  and  $k_{22}^{kl}$  are represented in (a). The principal point  $[k_{13}^{kl}, k_{23}^{kl}]^T$  is depicted in (b). The  $x$ - and  $y$ - components of the projection are presented in (c) while the  $z$ -component is presented in (d).

## 6 Conclusions

In this work, one defined the geometry of the multiple camera arrays that can be obtained from the rays captured by a SPC. This geometry extends the characterization of the microlens and viewpoint camera arrays found in the literature [1, 2, 4] and that are associated with the images that can be obtained directly from the LF acquired by a SPC. The mappings proposed for the different microlens and viewpoint cameras obtained after shearing were validated using a publicly available dataset and calibration toolbox [6]. The results show that the mappings proposed are in accordance with the calibration estimates obtained.

In terms of future work, we want to characterize the camera arrays originated considering a sampling on the microlens coordinates  $(k, l)$  instead of the sampling on the viewpoint coordinates  $(i, j)$  considered in this work.

**Funding.** This work was supported by the Portuguese Foundation for Science and Technology projects [UID/EEA/50009/2019] and [PD/BD/105778/2014], the RBCog-Lab [PINFRA/22084/2016] and FIREFRONT [PCIF/SSI/0096/2017].

## References

1. Anonymous: Standard plenoptic cameras mapping to camera arrays and calibration based on DLT. TCSV 2018 Submission ID TCSV 02137-2018 (2018)
2. Anonymous: Geometric calibration of multi-focus plenoptic cameras. ICCV 2019 Submission ID 2256 (2019)
3. Birkbauer, C., Bimber, O.: Panorama light-field imaging. *Comput. Graph. Forum* **33**(2), 43–52 (2014)
4. Bok, Y., Jeon, H.G., Kweon, I.S.: Geometric calibration of micro-lens-based light field cameras using line features. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(2), 287–300 (2017)
5. Chen, C., Lin, H., Yu, Z., Bing Kang, S., Yu, J.: Light field stereo matching using bilateral statistics of surface cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1518–1525 (2014)
6. Dansereau, D.G., Pizarro, O., Williams, S.B.: Decoding, calibration and rectification for lenslet-based plenoptic cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1027–1034 (2013)
7. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), vol. 96, pp. 43–54. ACM (1996)
8. Grossberg, M.D., Nayar, S.K.: The raxel imaging model and ray-based calibration. *Int. J. Comput. Vision* **61**(2), 119–137 (2005)
9. Honauer, K., Johannsen, O., Kondermann, D., Goldluecke, B.: A dataset and evaluation methodology for depth estimation on 4D light fields. In: Asian Conference on Computer Vision, pp. 19–34. Springer (2016)
10. Jeon, H.G., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y.W., So Kweon, I.: Accurate depth map estimation from a lenslet light field camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1547–1555 (2015)
11. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), vol. 96, pp. 31–42. ACM (1996)
12. Marto, S.G., Monteiro, N.B., Barreto, J.P., Gaspar, J.A.: Structure from plenoptic imaging. In: Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), pp. 338–343. IEEE (2017)
13. Monteiro, N.B., Barreto, J.P., Gaspar, J.: Surface cameras from shearing for disparity estimation on a lightfield. In: RECPAD - Portuguese Conference on Pattern Recognition (2018)
14. Monteiro, N.B., Marto, S., Barreto, J.P., Gaspar, J.: Depth range accuracy for plenoptic cameras. *Comput. Vis. Image Underst.* **168**, 104–117 (2018)
15. Ng, R.: Digital light field photography. Ph.D. thesis, Stanford University (2006)
16. Tao, M.W., Hadap, S., Malik, J., Ramamoorthi, R.: Depth from combining defocus and correspondence using light-field cameras. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 673–680 (2013)



# Robot Navigation to Approach People Using $G^2$ -Spline Path Planning and Extended Social Force Model

Marta Galvan, Ely Repiso<sup>(✉)</sup>, and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain  
[{mgalvan,erepiso,sanfeliu}@iri.upc.edu](mailto:{mgalvan,erepiso,sanfeliu}@iri.upc.edu)

**Abstract.** When a robot has to interact with a person in a dynamic environment, it has to navigate to reach a close distance and to be in front of the person. This navigation has to be smooth and take care of the person’s movements, the static obstacles and the motion of other people. In this paper, we present a new method to approach a person, that combines  $G^2$ -Splines ( $G^2$ S) paths with the Extended Social Force Model (ESFM) to allow the robot to move in dynamic environments avoiding static obstacles and other people. Moreover, we use the Bayesian human motion intentionally prediction (BMP) in combination with the Social Force Model (SFM) to be able to approach a moving person and also to avoid moving people in the environment. The method computes several paths using the  $G^2$ S and taking into account the person’s position and orientation. Then, the method selects the best path using several costs that consider distance, orientation, and interaction forces with static obstacles and moving people. Finally, the robot is controlled with the ESFM to follow the best path. The method was validated by a set of simulations and also by real-life experiments with a humanoid robot in a dynamic environment.

**Keywords:** Human-robot approaching · Robot navigation · Human-robot interaction · Human-robot collaboration

## 1 Introduction

Our society is evolving to include intelligent robots in daily live, which have to interact and collaborate with humans. These robots have to develop several skill and behaviors, among them social or collaborative navigation [8, 14], learning

---

Work supported by the Spanish Ministry of Science and Innovation under EU project AI4EU (H2020-ICT-2018-2-825619), the national R&D project ColRobTransp (DPI2016-78957-RAEI/FEDER EU), by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656). Ely Repiso is also supported by Spanish Ministry of Science and Innovation under a FPI-grant, BES-2014-067713. M. Galvan and E. Repiso—contributed equally.



**Fig. 1.** The robot uses the implemented method to approach a static and moving person, while avoiding several static obstacles of the environment.

how to approach people and develop original ways of reaching them [4, 18], or understanding and predicting human intentions [6, 12, 22].

Social navigation and approach a person are common tasks for people. We tend to approach other people when: we need help to arrive at a specific place, we want to buy something in a shop, we meet someone somewhere or we know someone in the street, and so on.

For humans moving around people and obstacles while approaching one person socially and predictably is very natural, but for robots is a challenging behavior, that implies develop different skills to be able to do different things at the same time. In particular, robots have to navigate autonomously towards humans, predict people movement, recognize the person to be approached and know its position and orientation, deal with uncertainties like momentary occlusions of the approached person by other people, approach in a human-like way and initiate interactions following social rules and patterns. Furthermore, the difficulty increases if the person to be approached is moving. Figure 1 shows two real experiments, firstly the robot approaches a static person (the two fist images at left) and secondly approaches a dynamic person (the two last images at right), while it has to avoid several static obstacles of the environment.

Some researches like Carton et al. [3, 4] try to use appropriate human-like features, such as smooth trajectory shapes, specified approach speed, appropriate human-robot distance, etc. With these characteristics the approaching of the robot appears significantly more natural, enhancing non-verbal interaction initiation with humans, mutual collision avoidance and reduction of interference. Also, Takayama et al. [20] do a user study to know the influence of several factors in the preferences of the personal spaces in the approaching behavior for Japanese people. Furthermore, Joosee et al. [13], study the appropriateness of the robot's approach behavior in different cultures, where they found that Chinese participants prefer closer approaches compared to participants from the U.S. or Argentina. In our work, we also try to use most of these human-like features.

Other methods to approach some people are based on learning algorithms like in [1], where they use learning methods to find the personal comfort field, without invading the personal space. That work is based on an on-line learning algorithm where the robot learns user's specific personal comfort space from user's reactions, by exploring regions nearby the user to search for a more

comfortable approaching trajectory. Another method is based on the collection of real data to obtain human-like approaching behaviors like in [2], where they collect navigation trajectories in which a person approaches to another human, to create a model that can be used by a robot's path planner to get more socially acceptable paths.

In our institute, we started to develop methods to approach a person in [16], where the robot tries to approach a person while accompanying another person. The model includes a framework to calculate a moving goal taken into account the movement of the approached person, the movement of the group and the best path to go throw the obstacles of the environment. After that in [17] we enhanced the previous approach by computing the best encounter point using a gradient descent method, taking into account all people's predictions. In the encounter point, the robot performs a triangle formation to achieve an engagement with both people.

Now in this paper, we go a step further from the previous approaching methods realizing a path planning algorithm that allows us to have a more human-like robot navigation to approach a person. First, we obtain some smoothed paths using  $G^2$ -Splines computation; then, we formulate a new cost function, to select the best path to go to the goal while avoiding collisions with the entities in the environment, which uses the social forces presented in the ESFM [8]. Furthermore, we use the ESFM to control the robot to deal with real-time navigation in dynamic environments. The computation is done online and in real-time, to keep the environment constantly updated and to have always a feasible path for the robot.

In the remainder of the paper, we start by introducing the implemented approach, that combines the  $G^2$ -Splines ( $G^2$ S) with the Extended Social Force Model (ESFM) in Sect. 2. Then, we show the developed metrics of performance to evaluate the task in Sect. 3. In Sect. 4 we include the simulation results. The real-life experiments with our robot are shown in Sect. 5. Finally, Sect. 6 presents the conclusions.

## 2 Extended Social Force $G^2$ -Spline Navigation Method to Approach a Person

In this section, we present a robot navigation method to approach a person that combines the  $G^2$ -Splines [15] ( $G^2$ S) path planning with the Extended Social Force Model (ESFM) [8]. The path planning algorithm for the robot behavior is summarized in Algorithm 1. In Fig. 1 you can see two different instants of time of two approaching situations and in Fig. 2, you can see a simulation example of the path planning generation. In this article, we have used a similar notation of [8].

The basic idea is to start approaching a person when the robot is close to he/she (around 15 m), using a combination of  $G^2$ S path planning and ESFM. The

**Algorithm 1.** AKP approaching( $s_{ini}, \Delta t$ )

---

```

1: Initialize  $\mathcal{T}(\mathcal{V}, \mathcal{E}) \leftarrow \{\emptyset\}$ 
2:  $\mathcal{V} \leftarrow s_{ini}$ 
3:  $\{D_{pj,n}^{goal}\} = \text{SCENE\_PREDICTION}(s_{ini})$ 
4:  $\{D_{r,n}^{goal}\} = \text{FINAL\_DESTINATIONS}(s_{ini}, N)$ 
5: for  $n = 1$  to  $N$  do
6:    $\{x_1, \dots, x_k\} = \text{SPLINES\_GENERATION}(s_{ini}, D_{r,n}^{goal})$        $\triangleright x_1 = s_{ini}$  and  $x_k = D_{r,n}^{goal}$ 
7:   for  $i = 1$  to  $K$  do
8:      $u_{ri} = \text{CALCULATE\_EDGE}(x_i, x_{i+1}, i, dist_{col})$ 
9:   end for
10:   $J_n = \text{PATH\_COST\_COMPUTATION}(\{x_1, \dots, x_k\}, \{u_{r1}, \dots, u_{rk}\})$ 
11: end for
12:  $[D_{r,best}^{goal}, J_{best}] = \text{BEST\_PATH\_SELECTION}(\{J_1, \dots, J_N\})$ 
13:  $u_{rf} = \text{CALCULATE\_EDGE}(s_{ini}, step_{goal}, index, dist_{col})$ 
14:  $s_{new} = \text{ROBOT\_PROPAGATION}(s_{ini}, u_{rf})$ 
15:  $s_{new} = \text{ORIENTATION\_ADJUSTING}(s_{new})$ 
16:  $\mathcal{V} \leftarrow \{s_{new}, J_{best}\}$ 
17:  $\mathcal{E} \leftarrow \{u_{rf}\}$ 
18: return BRANCH( $\mathcal{T}$ )

```

---

method assumes that the robot has to move from its location up to a surrounding circle around the person that has 1.5 m of radius, and we suggest a 5 good final goal locations over this circle to approach that person, covering and angle view of 80° around the point to face directly the person (their computation is better explained, next in the algorithm). We use only 5 final destinations around the person to allow the robot to do not approach directly facing the person if it is necessary, but neither allow the robot to approach the person very laterally using a difference of orientations between them smaller than 45°. You can see how these 5 final locations are distributed around the person in the upper images of Fig. 2. The distance between the person and the robot was based on a previous work of our institute [9]. At each cycle time, the robot computes  $G^2$ -Splines paths to go from its current position to each one of the 5 possible final destinations. For each one of the paths, the algorithm uses the ESFM to evaluate the paths (and selects the best one) and to control the robot approaching behavior, while avoiding static and moving obstacles (moving people). As the environment is dynamic (has other people around), the robot has to generate new paths in each iteration (every 0.2 s) in order to adapt to it.

In the following, we will present the entire procedure to obtain the approaching behavior of the robot. At each iteration a robot plan is computed, and the linear and angular velocities of the robot state, included in  $s_{new}$ , are executed by the robot controller to move it. The input of the algorithm is the  $s_{ini} \in \mathcal{S}$ , where  $\mathcal{S} = S_r \times \cup S_{pi}$  is the state that contains the information of the robot state plus all people's states considered on the scene, included the approached person (see [8]). This state includes position, velocity and orientation for the robot and people. Moreover, the cycle time is set to  $\Delta t = 0.2$  s (It is the maximum time

allowed by the controller to work in real time). Furthermore, all the laser scans are processed to obtain the static and dynamic obstacles of the environment.

In the function SCENE\_PREDICTION( $s_{ini}$ ), the robot infers all the final destinations for all the people of the environment,  $D_{p_j,n}^{goal}$ , by using the people's direction of movement, and using these destinations and SFM [11] to predict the people paths inside a window of time of 5 s, using the BMP [6].

Then, the algorithm computes  $N = 5$  candidate paths using the  $G^2$ -Splines function [15], in SPLINES\_GENERATION(). Moreover, the needed information to generate a path is only related to the initial and final states. For details we refer to [15]. Then, these paths start in the robot position and finish around the position of the approached person. To find the final destinations of all the paths we use the function in Algorithm 1-line 4. This function computes 5 final destinations at 1.5 m around the position of the approached person, taking into account the person orientation and computing the possible final states starting from the one facing directly the person in front of it and using an increment of  $\pm 20^\circ$  or  $\pm 40^\circ$  from the starting goal, where the robot face directly the person in front of it, to both sides (left and right) to compute the other four approaching positions. We use this type of geometric splines, because they present very nice properties: the quintic  $G^2$ -Splines offers flexibility and, since they are geometric polynomials of 5th order with second order geometric continuity ( $G^2$ ), the curvature  $\kappa$  is continuous. Furthermore, to obtain the robot iterative and dynamic behavior we need to split these paths of the  $G^2$ -Splines in  $K$  steps with length of 0.2 m. This length is the maximum distance that can cover the robot during one iteration time of the algorithm, due to the maximum robot's velocity that is 1 m/s and the maximum iteration time of the robot's controller that is 0.2 s (this not mean that the robot always cover this path distance, it depends of the resultant force of this step of the path, this distance is only the maximum one to allow the robot to navigate at its maximum speed if does not have any obstacles around). Then, we obtain the robot steps  $\{x_1, \dots, x_k\}$  inside the path.

After that for each path we need to know the forces in each step of the path, because the robot navigates using the ESFM. These forces follow the Helbing definition [11] and are computed in function CALCULATE\_EDGE(), where  $dist_{col}$  defines the radius of the circular area were the interactions with obstacles and other people are considered. The ESFM to control the robot uses as local goals the steps of the path computed by the  $G^2$ -Splines and includes repulsive forces between the robot and static and dynamic obstacles, like people. According to the model, both humans and robots are free particles in a 2D space, following the laws of Newtonian mechanics, and the *resulting force*  $\mathbf{F}_e$  that governs the trajectory of the movement of each entity ( $e = \{r, p\}$ , where the entity  $e$  can be robot ( $r$ ) or person ( $p$ )) is described in Eq. 1. This force is used to control the robot in a dynamical environment where people or other robots are moving around, and also to predict the people movement.

$$\mathbf{F}_e = \alpha \mathbf{f}_{e,d}^{goal}(D_{e,n}^{goal}) + \gamma \left( \sum_{j \in P} \mathbf{f}_{e,j}^{int} + \sum_{b \in R} \mathbf{f}_{e,b}^{int} \right) + \delta \sum_{o \in O} \mathbf{f}_{e,o}^{int}, \quad (1)$$

where,  $P$ ,  $O$  and  $R$  are the sets of people, obstacles and robots of the environment, respectively. The resultant force,  $\mathbf{F}_e$ , is composed by the attractive force until the destination and the repulsive forces respect other people, robots or obstacles. Furthermore, the  $\{\alpha, \gamma, \delta\}$  parameters were used in [7], to implement three different robot behaviors (aware, balanced and unaware) and here we select different robot behaviors respect to the final goal, and the people and obstacles interactions, that face better our approaching case. We set these parameters equal to  $\{\alpha = 1.2, \gamma = 1.4, \delta = 1.0\}$ , to allow a balanced robot behavior respect obstacles, an aware behavior respect people and an intermediate behavior between unaware and balanced respect to the final goal. The aware robot's behavior respect people allows the robot to hinder less the people's path.

The first force of Eq. 1 is the *attraction force* to reach the goal Eq. 2. This force assumes that the entity tries to adapt its velocity within a relaxation time  $k^{-1}$  to arrive to the destination and is given by:

$$\mathbf{f}_{e,d}^{goal}(D_{e,n}^{goal}) = k(\mathbf{v}_e^0(D_{e,n}^{goal}) - \mathbf{v}_e) \quad (2)$$

where  $k = 1/\tau$ , and  $\tau$  is the time for a human to take a step, 0.5 s approximately.  $\mathbf{v}_e$  is the current velocity of the entity,  $D_{e,n}^{goal}$  is any of the possible final destinations for the robot or people and  $\mathbf{v}_e^0(D_{e,n}^{goal})$  is the desired velocity to reach the goal. Moreover, each repulsive interaction forces are modeled using the Helbing [11] social force model, as:

$$\mathbf{f}_{e,z}^{int} = A_{ez} e^{(d_{ez} - d_{e,z})/B_{ez}} \hat{\mathbf{d}}_{e,z} w(\varphi_{e,z}, \lambda_{ez}) \quad (3)$$

where  $z \in P \cup O \cup R$  is either a person or a static obstacle or a robot, of the environment.  $A_{ez}$  and  $B_{ez}$  denote respectively the strength and range of the repulsive interaction force, between  $e$  and  $z$ .  $d_{e,z}$  is the distance between the centers of the two entities, and  $\mathbf{d}_{ez} \equiv \mathbf{r}_e + \mathbf{r}_z$ , is a parameter that depends only from the interaction between each type of entities. Moreover, given the limited field of view, influences might not be isotropic, requiring then a scaling anisotropic factor,  $w(\varphi_{e,z}, \lambda_{ez})$ , further details in [5]. Finally, all the parameters  $\{k, A_{ez}, B_{ez}, \lambda_{ez}, d_{ez}\}$  are defined depending on the interaction type, where we use the parameters learned in [21] for the interaction force between robot-obstacle, and we use the parameters learned in [5] for the interaction force between human-robot.

Now, we need to calculate the final cost of each path, Algorithm 1-line 10, to be able to finally select the best path,  $J_{best}$ , until the best final destination,  $D_{r,best}^{goal}$ , using the cost, Algorithm 1-line 12. The best path has the minimum cost and the cost for each path requires three steps calculation. First, each individual cost function of  $\mathbf{J}(S, U) = [J_d(S), J_{or}(S), J_p(U), J_o(U)]$  is computed. There are multiple objectives to be minimized in dynamic planning, and we use different and independent criteria, where each single cost is related to the considered criteria: distance of all the steps of the path (path length), change of the robot orientation during all the steps of the path (path curvature) and repulsive forces respect to people and obstacles during all the steps of the path according to the

**Algorithm 2.** Robot propagation

---

```

1: function ROBOT_PROPAGATION( $s_{ini}, u_{rf}, \Delta t$ )
2:    $s_{ini} = [x_{t_i}, y_{t_i}, \theta_{t_i}, v_{t_i}, \omega_{t_i}, t_i]^T$ 
3:    $a_{v_{t_i}} = f_x \cos(\theta_{t_i}) + f_y \sin(\theta_{t_i})$ 
4:    $a_{\omega_{t_i}} = -f_x \sin(\theta_{t_i}) + f_y \cos(\theta_{t_i})$ 
5:    $v_{t_{i+1}} = v_{t_i} + a_{v_{t_i}} \cdot \Delta t$ 
6:    $\omega_{t_{i+1}} = \omega_{t_i} + a_{\omega_{t_i}} \cdot \Delta t$ 
7:    $x_{t_{i+1}} = x_{t_i} + v_{t_i} \cos(\theta_{t_i}) \cdot \Delta t + a_{v_{t_i}} \cos(\theta_{t_i}) \cdot \Delta t^2 / 2$ 
8:    $y_{t_{i+1}} = y_{t_i} + v_{t_i} \sin(\theta_{t_i}) \cdot \Delta t + a_{v_{t_i}} \sin(\theta_{t_i}) \cdot \Delta t^2 / 2$ 
9:    $\theta_{t_{i+1}} = \theta_{t_i} + \omega_{t_i} \cdot \Delta t + a_{\omega_{t_i}} \cdot \Delta t^2 / 2$ 
10:  return  $s_{new} = [x_{t_n}, y_{t_n}, \theta_{t_n}, v_{t_n}, \omega_{t_n}, t_n]^T$ 
11: end function

```

---

ESFM (interactions between people and obstacles). Each different cost computed for all the steps along the path is defined by:

$$\begin{aligned}
J_d(S) &= \sum_{t_{ini}}^{t_{end}} \|\mathbf{x}_r(t+1) - \mathbf{x}_r(t)\|^2, & J_{or}(S) &= \sum_{t_{ini}}^{t_{end}} \|\theta_r(t+1) - \theta_r(t)\|^2, \\
J_p(U) &= \sum_{t_{ini}}^{t_{end}} \sum_{i=1}^P \|u_{p_i}(t)\|^2, & J_o(U) &= \sum_{t_{ini}}^{t_{end}} \sum_{i=1}^O \|u_{o_i}(t)\|^2
\end{aligned}$$

where,  $\mathbf{x}_r = (x_r, y_r)$  is the robot position,  $\theta_r$  is the robot orientation  $u_{p_i}(t)$  and  $u_{o_i}(t)$  are the repulsive forces respect to people and obstacles. Second, in order to avoid the scaling effect of a weighted-sum method, each cost function is normalized to  $(-1, 1)$ , with Eq. 4-left. Third, a projection via weighted sum is obtained with Eq. 4-right to convert the multi-cost function in a single cost for each path.

$$\bar{J}_i(X) = \operatorname{erf}\left(\frac{x - \mu_x}{\sigma_x}\right), \quad J(S, U) = w_i \sum_i \bar{J}_i(S, U) \quad (4)$$

Finally, the algorithm selects the best path that has the minimum cost, which represents minimum distance and minimum orientation changes, as well as avoiding interactions between people and obstacles. Then, we use the function `CALCULATE_EDGE()` to obtain the resultant force Eq. 1,  $u_{rf} = (a_x, a_y) = (F_{xr}, F_{yr})/m_r$ , to move the robot following the best path, where  $m_r = 1$ . Now, the entity means robot. Next, with function `ROBOT_PROPAGATION()` of Algorithm 2, we convert the forces into linear and angular accelerations to propagate the robot position and to obtain the angular and linear velocities that needs our robot controller to move the robot. It is to say that we obtain  $s_{new}$  for the robot. Finally, the function `ORIENTATION_ADJUSTING()` is used to adjust the robot orientation until obtain a small difference, less than  $20^\circ$ , between its orientation and the desired orientation of the best path, only if the robot reaches the final goal with a big difference of orientation.

### 3 Metrics of Performance for Positioning the Robot Respect to the Approached Person

In this section, the performance metrics used to evaluate the robot behavior are described. These metrics are based on previous studies on humans [19] and the proxemic rules, proposed by Hall [10]. Furthermore, the limits of the interaction distances used were based on a previous work of our institute [9], and a similar version of these performances is included in [17].

The final position of the robot respect to the approached person was evaluated using three types of performance metrics. One related with proxemics, based on several areas of performance, to obtain if the robot arrives to the best approached area to face the person. Other two metrics that serve to differentiate if the robot arrives inside a desired margin of distances with respect to the approached person and if also the robot is oriented to face the person.

The first performance takes into account the spatial relationship in 2D between the robot and the approached person and it is defined by three areas: (i) Human's personal space  $\mathcal{C}$ , is the area where the robot can not be in order to avoid invading any human's personal space and it takes into account the space delimited by the radius of the person, that includes some free space, correspondent to  $R_s$ ; (ii) Social distance area  $\mathcal{A}$ , is the area where the robot should be to be socially accepted; and (iii) Human's best approaching area  $\mathcal{B}$ , is the area where the robot must be placed so that the person perceives a socially accepted approach behavior. This area is a dynamic area inside the social distance that depends on the final path destinations and the best path selection from the 5 possible planned paths of the robot, Algorithm 1-line 4 and 12. The description of these areas was included in [17]. Here only change in the  $\mathcal{A}$  the  $P_c$  correspondent to the companion person to  $P_a$  for the approaching person, and also the physical position of the area  $\mathcal{B}$  changes because here we have only the approached person and this area is defined by the final goal of the best path. Furthermore, the formulation of how the performance was extracted from these areas and the description of the robot area is detailed in [17], next we only explain the general idea to understand it. The Area metric has the maximum performance of 1 when the robot is in the area described by  $\mathcal{B}$ , since it is the best position to approach the human. Additionally, if the robot is in the area  $\mathcal{A}$ , but not in area  $\mathcal{B}$ , is a partial success, since the robot is inside the social distance of the human, but not in the best approaching position. Then the performance has a value of 0.5. Finally, if the robot is further than 3 m from the human's position, then we consider that there is not approaching interaction between robot and person, and therefore its performance is 0. Also, if the robot invades any human's personal space is penalized with 0 performance.

Regarding the distance and angle performances, we consider that the robot achieves a good distance performance if it keeps its central position inside the interval of distances [1.25 – 2] m, respect to the position of the approached person. This margin is around the ideal value of 1.5 m. Then, between 2 m until 3 m the performance decreases from 1 until 0, and also between 0.75 until 1.25 the performance increases from 0 to 1. The reader is referred to [17] for further

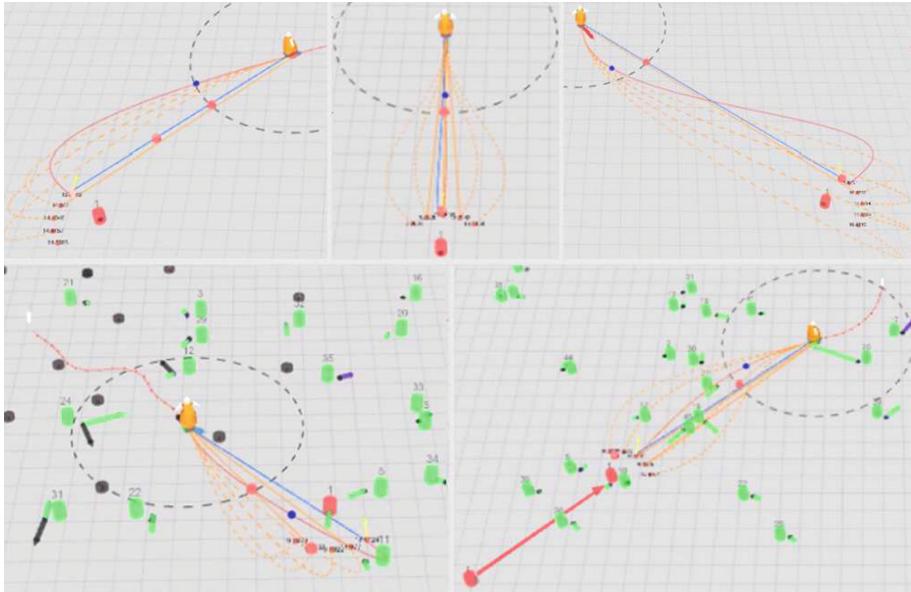
explanation and see the equation of the metric of performance in distance. In terms of angle, the best angle performance is, at most, a difference of  $20^\circ$  from the ideal orientation to face the approached person (performance value of 1). Then, if the difference of the angle increases, we penalized it until we obtain a 0 value of performance, when the robot has an error of  $90^\circ$  with respect to the ideal value of orientation. The equation of the angle performance metric is shown in Eq. 5, where  $P_{\theta_{diff}}$  means angle performance and is the difference of orientations between the robot and the approached person,  $\theta_{diff} = (180^\circ - \theta_r) - \theta_t$ , and  $\theta_t$  means the orientation of the approached person and  $\theta_r$  is the real orientation of the robot. For a better understanding of the performance metrics, the reader can find an image that graphically shows the performances in this link: <http://www.iri.upc.edu/people/erepiso/ROBOT2019.html>

$$P_{\theta_{diff}} = \begin{cases} 1 & \text{if } 0^\circ \leq \theta_{diff} \leq 20^\circ \\ -\frac{1}{70}(\theta_{diff}) + \frac{9}{7} & \text{if } 20^\circ < \theta_{diff} \leq 90^\circ \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

## 4 Simulation Experiments

The actual section describes our simulation environment and all the possible situations that we used to test and validate our implemented approach. To be able to test these situations we used a complex simulation environment, used in all our previous works [16, 17] and also a Gazebo simulator. All is coded using C++ and ROS. This environment includes random people moving through different destinations, represented by green cylinders, and several obstacles, represented by dark grey cylinders, that we can put in any place. The people movement follows the Social Force Model [11] and their desired velocities were randomly selected inside the interval of [0–1] m/s. Furthermore, our simulated robot follows the laws of a non-holonomic vehicle and uses the presented algorithm to approach a person. The robot has a maximum velocity of 1 m/s. Between the robot and the target person we have used a final social distance of 1.5 m as minimum distance to approximate. We have selected this distance to prevent the robot to go too close to the person and we are based in one previous work [9].

To test and validate a large field of situations we perform more than 7200 simulations. These simulations include different approaching situations, where the robot has to approach one person (static or moving towards any possible destination) while avoiding several static and dynamic obstacles. A representative number of the simulation cases is shown in Fig. 2, where the simulation environment contains the robot model, the approached person in red. The possible paths for the robot are drawn in orange and in red the best path. The time window to compute the paths and take into account the people and obstacle interactions, which is a black dashed circle around the robot. The resultant force for the robot is represented with a red arrow over the robot, the blue arrow is the force until the goal, the green arrows are repulsive forces respect to people,



**Fig. 2. Synthetic experiments:** simulation environment to test and evaluate the new method to approach a person. *Up*: three images that show different approaching angles respect to a static person. *Down-left*: shows the robot approaching to one person, using the fourth approximation angle, while avoids static and dynamic obstacles. *Down-right*: shows the robot approaching a moving person while avoids dynamic obstacles.

the black arrows are repulsive forces respect to static obstacles and the purple arrows are repulsive forces for people with respect to the robot.

The first group of simulations were carried out in an empty environment, where we tested the approaching behavior of the robot with a static person that has 4 different approaching orientations ( $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ ). We used only 4 cases because with these 4 selected orientations we covered a huge range of approaching behaviors of the robot. The second group of simulations were also in the empty space, but now the approached person was moving using three different approaching directions (right, center and left), that also represent well most of the possible real interactions with a moving approaching person. The third ones included the before approaching behaviors of the person, but with other people moving around. In the fourth ones the robot had to avoid several static obstacles while approaching to the person with the same behaviors described before (static and dynamic). The fifth group of simulations included people and obstacles at the same time.

The performance of the robot was evaluated in all of these simulations using the metrics described in Sect. 3. Where the performance is inside the interval of [0–1] and the best value of performance corresponds to 1. Table 1 shows all the performances for all the cases.

**Table 1.** Performance results for the approaching task of all the simulations. The performance value equal to 1 is considered the best value and the values between brackets are the standard errors of each mean value.

Approaching performance simulations	mean ( $P_{2R_i}$ )	mean ( $P_{\theta_{diff}}$ )	mean ( $P(r, p_a)$ )
Person stop, without obst	0.98 ( $\pm 0.04$ )	1 ( $\pm 0$ )	0.98 ( $\pm 0.07$ )
Person stop, with dynamic obst	0.97 ( $\pm 0.07$ )	0.66 ( $\pm 0.19$ )	0.99 ( $\pm 0.05$ )
Person stop, with static obst	0.96 ( $\pm 0.06$ )	0.85 ( $\pm 0.13$ )	0.97 ( $\pm 0.08$ )
Person moving, without obst	0.96 ( $\pm 0.05$ )	1 ( $\pm 0$ )	0.99 ( $\pm 0.02$ )
Person moving, with dynamic obst	0.97 ( $\pm 0.1$ )	0.73 ( $\pm 0.03$ )	0.99 ( $\pm 0.05$ )
Person moving, with static obst	0.84 ( $\pm 0.13$ )	0.92 ( $\pm 0.16$ )	0.94 ( $\pm 0.13$ )
Person stop, static & dynamic obst	0.86 ( $\pm 0.12$ )	0.55 ( $\pm 0.21$ )	0.96 ( $\pm 0.095$ )
Person moving, static & dynamic obst	0.84 ( $\pm 0.13$ )	0.79 ( $\pm 0.15$ )	0.96 ( $\pm 0.08$ )

**Table 2.** Performance results for the approaching task of all the real-life experiments. The performance value equal to 1 is considered the best value and the values between brackets are the standard errors of each mean value.

Approaching performance real-life	mean ( $P_{2R_i}$ )	mean ( $P_{\theta_{diff}}$ )	mean ( $P(r, p_a)$ )
Person stop, without obst	0.86 ( $\pm 0.14$ )	0.77 ( $\pm 0.2$ )	0.98 ( $\pm 0.02$ )
Person stop, with dynamic obst	0.8 ( $\pm 0.13$ )	0.82 ( $\pm 0.23$ )	0.97 ( $\pm 0.04$ )
Person stop, with static obst	0.96 ( $\pm 0.07$ )	0.62 ( $\pm 0.25$ )	0.97 ( $\pm 0.04$ )
Person moving, without obst	0.87 ( $\pm 0.14$ )	0.8 ( $\pm 0.23$ )	0.93 ( $\pm 0.11$ )
Person moving, with dynamic obst	0.86 ( $\pm 0.19$ )	0.67 ( $\pm 0.37$ )	0.98 ( $\pm 0.03$ )
Person moving, with static obst	0.93 ( $\pm 0.12$ )	0.64 ( $\pm 0.3$ )	0.92 ( $\pm 0.14$ )

## 5 Real-Life Experiments

The implemented method was tested also in a real-life environment. The experiments were developed in the FME (Facultat de Matemàtiques i Estadística) lab, an outdoor urban environment located at the South Campus of the Universitat Politècnica de Catalunya (UPC). There, we have a controlled environment where we can test the algorithm without any obstacles and with an approached person situated at different points of the environment and with different orientations (static or moving), with static or dynamic obstacles and the same configurations for the approached person as the case without obstacles. The mean and standard deviation of the approaching performance of the 82 real-life experiments are shown in Table 2 and Fig. 1 shows several interesting moments of these experiments. In addition, the reader can find some videos of the real experiments in this link: <http://www.iri.upc.edu/people/erepiso/ROBOT2019.html>.

## 6 Conclusions

We have presented a method that combines the  $G^2$ -Splines with the Extended Social Force Model to allow the robot to approach a person as much natural

human-like as possible. The major contribution of this work is how we combine both methods,  $G^2S$  and ESFM, to obtain a more smooth and natural approaching behavior for the robot, while the robot navigates well inside dynamic environments and at the same time facilitate the walking behavior of other people. Furthermore, the robot's final path will be free of obstacles by using the ESFM applied to the  $G^2S$  path generation. The computation is done in real time. The new method has been tested and validated over simulations and good results have been obtained. Furthermore, we validated the algorithm in real-life experiments on the FME, where the robot achieved a good approaching behavior.

## References

1. Ahn, H., Oh, Y., Choi, S., Tomlin, C.J., Oh, S.: Online learning to approach a person with no regret. *IEEE Robot. Autom. Lett.* **3**(1), 52–59 (2018)
2. Avrunin, E., Simmons, R.: Using human approach paths to improve social navigation. In: Human-Robot Interaction (HRI), ACM/IEEE International Conference on Human-Robot Interaction, March 2013
3. Carton, D., Olszowy, W., Wollherr, D., Buss, M.: Socio-contextual constraints for human approach with a mobile robot. *Int. J. Soc. Robot.* **9**, 309–327 (2017)
4. Carton, D., Turnwald, A., Wollherr, D., Buss, M.: Proactively approaching pedestrians with an autonomous mobile robot in urban environments. In: Experimental Robotics, pp. 199–214. Springer (2013)
5. Ferrer, G., Garrell, A., Sanfeliu, A.: Robot companion: a social-force based approach with human awareness-navigation in crowded environments. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1688–1694. IEEE (2013)
6. Ferrer, G., Sanfeliu, A.: Bayesian human motion intentionality prediction in urban environments. *Pattern Recognit. Lett.* **44**, 134–140 (2014)
7. Ferrer, G., Sanfeliu, A.: Multi-objective cost-to-go functions on robot navigation in dynamic environments. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3824–3829. IEEE (2015)
8. Ferrer, G., Sanfeliu, A.: Anticipative kinodynamic planning: multi-objective robot navigation in urban and dynamic environments. *Auton. Robot.* **43**(6), 1473–1488 (2019)
9. Garrell, A., Sanfeliu, A.: Cooperative social robots to accompany groups of people. *Int. J. Robot. Res.* **31**(13), 1675–1701 (2012)
10. Hall, E.T.: The Hidden Dimension. Anchor Books, New York (1969). 20, 71
11. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. *Phys. Rev. E* **51**(5), 4282 (1995)
12. Ikeda, T., Chigodo, Y., Rea, D., Zanlungo, F., Shiomi, M., Kanda, T.: Modeling and prediction of pedestrian behavior based on the sub-goal concept. *Robot.: Sci. Syst.* **VIII**, 137 (2013)
13. Joosse, M., Poppe, R., Lohse, M., Evers, V.: Cultural differences in how an engagement-seeking robot should approach a group of people. In: CABS 2014, 20–22 August 2014, Kyoto, Japan (2014)
14. Khambaita, H., Alami, R.: Viewing robot navigation in human environment as a cooperative activity. arXiv preprint [arXiv:1708.01267](https://arxiv.org/abs/1708.01267) (2017)

15. LoBianco, C.G., Piazzì, A.: Optimal trajectory planning with quintic G<sup>2</sup>-splines. In: Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, Michigan, USA, 3–5 October 2000, pp. 620–625 (2000)
16. Repiso, E., Garrell, A., Sanfeliu, A.: On-line adaptive side-by-side human robot companion to approach a moving person to interact. In: Iberian Robotics Conference, pp. 113–125. Springer (2017)
17. Repiso, E., Garrell, A., Sanfeliu, A.: Adaptive side-by-side social robot navigation to approach and interact with people. *Int. J. Soc. Robot.* 1–22 (2019)
18. Satake, S., Kanda, T., Glas, D.F., Imai, M., Ishiguro, H., Hagita, N.: How to approach humans?-strategies for social robots to initiate interaction. In: 2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 109–116. IEEE (2009)
19. Stubbs, K., Hinds, P.J., Wettergreen, D.: Autonomy and common ground in human-robot interaction: a field study. *IEEE Intell. Syst.* **22**(2), 42–50 (2007)
20. Takayama, L., Pantofaru, C.: Influences on proxemic behaviors in human-robot interaction. In: IEEE/RSJ international conference on Intelligent robots and systems, IROS 2009, pp. 5495–5502. IEEE (2009)
21. Zanlungo, F., Ikeda, T., Kanda, T.: Social force model with explicit collision prediction. *EPL (Europhys. Lett.)* **93**(6), 68005 (2011)
22. Zanlungo, F., Ikeda, T., Kanda, T.: Potential for the dynamics of pedestrians in a socially interacting group. *Phys. Rev. E* **89**(1), 012811 (2014)



# Evaluation of SLAM Algorithms for Highly Dynamic Environments

Oliver Roesler<sup>(✉)</sup> and Vignesh Padubidri Ravindranath

Rösler Software-Technik Entwicklungs - und Vertriebsgesellschaft mbH,  
28816 Stuhr, Germany  
[oliver@roesler.co.uk](mailto:oliver@roesler.co.uk), [v.ravindranath@minidat.de](mailto:v.ravindranath@minidat.de)

**Abstract.** Simultaneous Localization And Mapping (SLAM) has received considerably attention in the mobile robotics community for more than 25 years. Most SLAM algorithms have been developed for and successfully tested in static environments. Previous studies that investigated the use of SLAM algorithms in dynamic environments only considered partially dynamic environment in which only a few objects are non-static. In this paper, we evaluate several popular SLAM algorithms for use in highly dynamic environments in which all objects are only temporarily static, i.e. all objects will be moved within a short time frame. To this end, we built a static test environment and defined two different scenarios based on a warehouse environment to simulate highly dynamic environments. Four different 2D SLAM algorithms that are available in Robotic Operating System (ROS) are employed and evaluated through visual inspection of produced maps and the difference between the object positions in obtained maps and their real positions in the environment. Based on our conducted evaluation Hector Mapping achieves the best performance in both scenarios.

**Keywords:** SLAM · ROS · Highly dynamic environments · Benchmarking · TurtleBot3 Burger

## 1 Introduction

Simultaneous Localization and Mapping (SLAM), which can also be referred to as Concurrent Mapping and Localization, is one of the major computational problems in the field of robotic mapping and navigation [12]. SLAM was first proposed by Thrun et al. [15] as the task of producing a map of an unknown environment, while simultaneously calculating the robot's position in the environment. The precise information of the position and orientation plays a pivotal role in the map building process because the generated map can in return be used to improve localization. SLAM is used in many areas, such as surveillance with unmanned aerial vehicles [4], self driving cars [2, 15] and autonomous underwater vehicles [6].

Most previous studies investigated SLAM in static environments, i.e. they assumed that objects are static and do not change their positions [3]. However, this assumption is often not true because most environments contain moving agents or objects that can be moved by agents. This dynamicity can lead to faults during object detection, errors in data association and inaccurate state estimations causing corruptions in the produced map of the environment so that it becomes useless for further applications. Thus, during the last years several SLAM algorithms have been explicitly developed for dynamic environments [1, 13, 14, 17, 18, 21]. The used environments consisted of many static objects, which should be represented as landmarks in the produced map and one or more moving objects, which should not be represented in the map. An example for the described environments is an office environment, which contains many static objects, e.g. desks and flower pots, and one or more humans as non-static objects. However, there exist environments that can be seen as even more dynamic because all objects in the environment are only temporarily static so that the basic layout of the environment changes continuously. An example is a warehouse without internal walls and shelves in which different objects of varying size can be placed temporarily. In this environment, the problem is not corruption of the map due to moving objects, but frequent addition and disappearance of landmarks. Additionally, the SLAM agent, e.g. a forklift, will perform SLAM only passively, i.e. it does not take any actions to support the SLAM process, such as driving slowly or in curved lines.

In this paper, we address the issue of SLAM in highly dynamic environments, such as warehouses, which do not contain any static objects, i.e. all objects will be moved within a short time frame. More specifically, we built a static test environment, defined two scenarios to simulate high dynamicity and evaluated four different popular SLAM algorithms through it. The evaluation was done in a two step process. On the one hand, the performance of the algorithms was evaluated based on visual inspection of the produced maps. On the other hand, a quantitative evaluation was performed through the calculation of the difference between the distances of all objects and the walls in the environment and the produced maps. The main question we investigate is whether the SLAM algorithms employed in this study can be used to produce accurate maps of the described highly dynamic environments.

The rest of this paper is structured as follows: Sect. 2 provides an overview of the four evaluated algorithms. Section 3 describes the used test environment and the two evaluation scenarios employed to simulate highly dynamic environments. The evaluation results are discussed in Sect. 4. Finally, Sect. 5 concludes the paper.

## 2 Algorithms

Four different SLAM algorithms are evaluated for the simulated highly dynamic environment used in this study (Sect. 3). All employed algorithms are open source

and available as packages in the Robot Operating System<sup>1</sup> (ROS). The following sections describe the individual algorithms in more detail.

## 2.1 Gmapping

Gmapping is the most widely used SLAM algorithm for mobile robots [11]. It was proposed by Grisetti et al. [5] and follows a Rao-Blackwellized Particle Filter approach that relies on data from odometry and scan matching to estimate the incremental movement and localization of a robot. Gmapping supports loop closure, i.e. it avoids re-mapping of previously visited areas in wrong global locations. A disadvantage of the Gmapping algorithm is that it heavily relies on odometry information provided by the wheel encoders. Thus, inaccurate encoder readings are detrimental to the mapping result. Gmapping always keeps the information on a predefined number of randomly generated particles, which denote the hypotheses of how the robot has traveled in the environment. Each particle has a continuously re-evaluated likelihood attached to it so that only the most likely particle is combined with the current laser scan to draw the map. The likelihood is calculated by comparing the predictions of the different particles with the last LIDAR scan. The employed Gmapping algorithm<sup>2</sup> is available as an open-source package in ROS.

## 2.2 Frontier Exploration

Frontier Exploration is an evidence grid based approach proposed by Yamauchi [19]. Each cell in the evidence grid has a probability that it is occupied with an object [10]. Known cells that do not contain an object and are adjacent to unknown cells are labeled as frontier edges. Adjacent cells that are labeled as frontier edges are then grouped into frontier regions and if a frontier region reaches a minimum size, it is regarded as a frontier. Autonomous robots can increase their knowledge about the world and map their environments by moving to successive frontiers [20]. The employed Frontier Exploration algorithm<sup>3</sup> is available as an open-source package in ROS.

## 2.3 Karto Mapping

Karto Mapping is a graph based approach to solve the SLAM problem [11]. It was proposed by Vincent et al. [16] and computes a map by means of graph optimization, i.e. it uses sequences of measurements to construct a graph of the map. Each node of the graph represents a pose of the robot along the trajectory, while edges between nodes represent motions between two consecutive poses. The algorithm uses a least squares error minimization approach to gain the most likely configuration of the map and a loop closure technique developed

---

<sup>1</sup> <http://www.ros.org/>.

<sup>2</sup> [https://github.com/ros-perception/slam\\_gmapping.git](https://github.com/ros-perception/slam_gmapping.git).

<sup>3</sup> [https://github.com/paulbovbel/frontier\\_exploration.git](https://github.com/paulbovbel/frontier_exploration.git).

by Lu and Milios [9]. Traversing the graph with the combination of laser measurements leads to the computation of the map. The employed Karto Mapping algorithm<sup>4</sup> uses Sparse Pose Adjustment for scan matching and loop-closure [8] and is available as an open-source package in ROS.

## 2.4 Hector Mapping

Hector Mapping was proposed by Kohlbrecher et al. [7] and uses an occupancy grid map to represent arbitrary environments. Due to the discrete nature of occupancy grid maps, bilinear filtering is employed to estimate occupancy probabilities and derivatives. Hector Mapping heavily relies on high scanning rate long-range rangefinders or on modern LIDAR systems and does not require odometry data to build maps, i.e. the 2D robot pose is estimated based on the scan matching process alone, which allows Hector Mapping to be used for robots that do not have access to odometry data. The Hector Mapping<sup>5</sup> algorithm employed in this study is available as an open-source package in ROS and provides two techniques to preprocess the input data, which are Simple Projection and High Fidelity Projection<sup>6</sup>. In this study, the latter is used because initial tests showed that it produces more accurate maps.

## 3 Experimental Setup

To evaluate the SLAM algorithms the Turtlebot3 Burger<sup>7</sup> robot was employed, which is a small circular robot consisting of a Raspberry Pi 3, 360° LIDAR, and an OpenCR Board equipped with an ARM Cortex M7 microcontroller.

Figure 1 illustrates the used evaluation environment, which had a length and width of 5.82 m and 1.23 m, respectively. The robot moves in the middle of the environment in the area labeled *TURTLEBOT PATH*. Four objects are placed on both sides of the area with varying distances between the objects. Since the environment is static, i.e. the objects have fixed positions, the dynamicity is introduced through two different scenarios, which differ in how the robot is moved. In both scenarios the Turtlebot3 robot moves with its maximum speed of approximately 0.22 m/s. However, in scenario 1 the robot moves one time from the left wall to the wall on the right, while in scenario 2, the robot moves one time from the left wall to the middle of the environment and back to the left wall. In both scenarios, the SLAM algorithm only runs while the robot is moving, i.e. no update of the map is done after the robot reaches its final position. The robot was not allowed to stop or explore the environment to simulate the high dynamicity of warehouse environments because after one pass through the environment the positions of all objects might have changed. Thus, it is important that with a single pass the SLAM algorithm is able to produce an accurate map of the environment.

---

<sup>4</sup> [https://code.ros.org/svn/ros-pkg/stacks/slam\\_karto/trunk](https://code.ros.org/svn/ros-pkg/stacks/slam_karto/trunk).

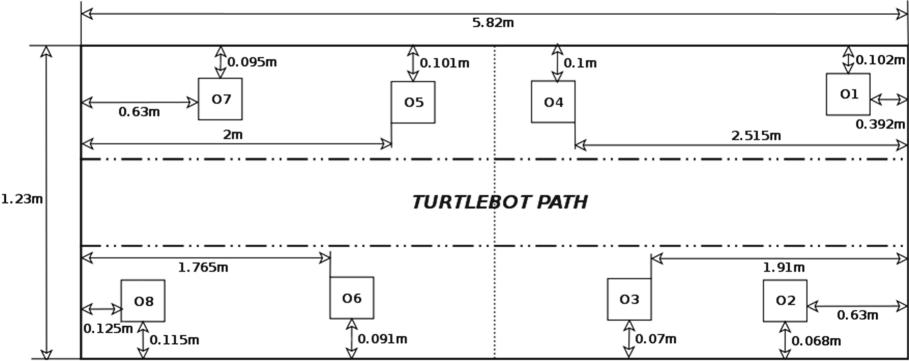
<sup>5</sup> [https://github.com/tu-darmstadt-ros-pkg/hector\\_slam.git](https://github.com/tu-darmstadt-ros-pkg/hector_slam.git).

<sup>6</sup> [http://wiki.ros.org/laser\\_geometry](http://wiki.ros.org/laser_geometry).

<sup>7</sup> <http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>.

## 4 Results

The performance of the four employed SLAM algorithms with default settings (Sect. 2) has been evaluated using the produced maps of the environment. The evaluation was conducted in two steps: (1) Visual inspection, in which the different maps are compared visually, and (2) Quantitative comparison, in which the error between the obtained and real object positions is evaluated. The evaluation results are described in the sections below.

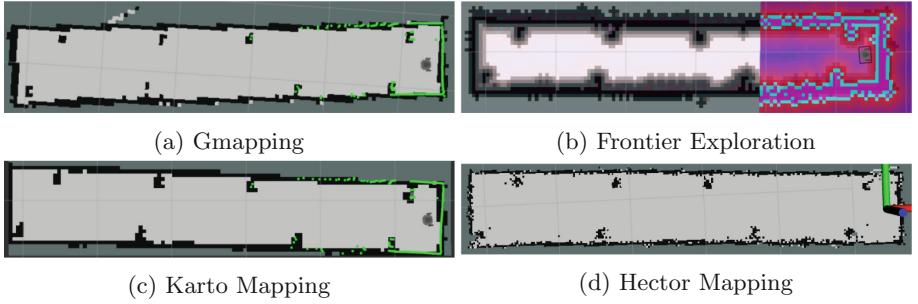


**Fig. 1.** Illustration of the static test environment used for both scenarios.

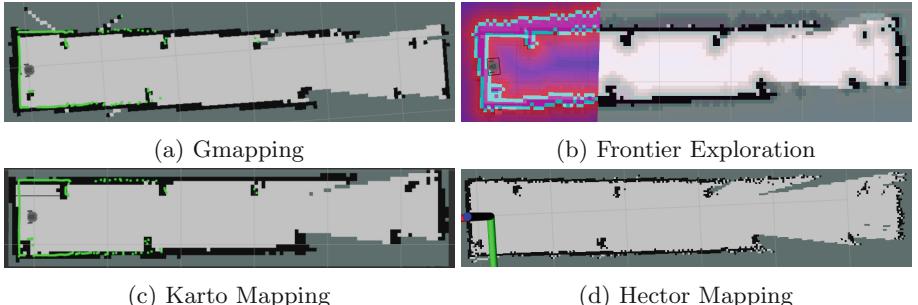
### 4.1 Visual Inspection

Figure 2a and d show that the objects in the environment are clearly visible and distinguishable from the walls for Gmapping and Hector Mapping. In contrast, most objects appear to be attached to the walls in the case of Karto Mapping (Fig. 2c), while the blurriness of the map produced by Frontier Exploration also made it difficult to separate the objects from the walls (Fig. 2b). Although the objects are clearly distinguishable in Gmapping, the formation of walls appears to be nonlinear in nature. Since the map produced by Hector Mapping does not have any of these problems, it is the best algorithm for the first scenario based on visual inspection of the produced maps.

Figure 3 shows the maps produced by all algorithms for scenario 2. In all of them the right one-third of the map is very blurry and incomplete, which is caused by the robot turning around in the middle of the map. For the left two-thirds the results for the individual algorithms are similar to the results obtained for scenario 1 (Fig. 2), i.e. objects are only clearly distinguishable in the maps produced by Gmapping and Hector Mapping and the latter performs overall best.



**Fig. 2.** Illustration of the produced maps of all algorithms for scenario 1. The pink parts in Fig. b are a characteristic of the Frontier Exploration algorithm (Sect. 2.2).



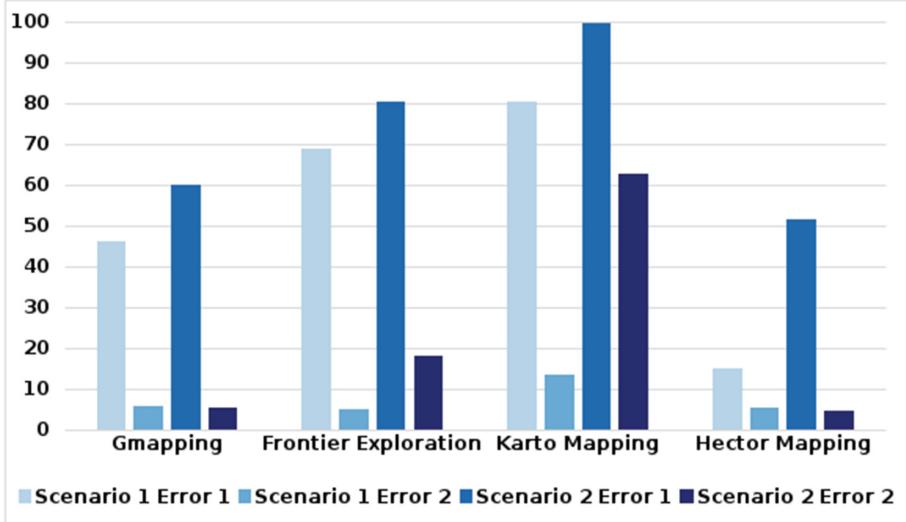
**Fig. 3.** Illustration of the produced maps of all algorithms for scenario 2. The pink parts in Fig. b are a characteristic of the Frontier Exploration algorithm (Sect. 2.2).

## 4.2 Quantitative Comparison

The accuracy of the produced maps is evaluated through the relative errors of the distances between objects and their closest horizontal and vertical walls in the real environment and the produced maps. In the real environment the distances were manually measured, while the distances in the produced maps were calculated using RViz<sup>8</sup>, which is a 3D visualization tool available in ROS that can be used to display sensor data and state information. Equation (1) shows the calculation of the two employed relative errors, where  $D_{horiz}^M$  represents horizontal distances,  $D_{vert}^M$  vertical distances,  $D^M$  distances in the produced maps, and  $D^E$  distances in the real environment.

$$\begin{aligned} E_1 &= \left[ \frac{D_{horiz}^M - D_{horiz}^E}{D_{horiz}^E} \right] \\ E_2 &= \left[ \frac{D_{vert}^M - D_{vert}^E}{D_{vert}^E} \right] \end{aligned} \quad (1)$$

<sup>8</sup> <http://wiki.ros.org/rviz>.



**Fig. 4.** Mean relative errors for all evaluated SLAM algorithms and both scenarios.

Figure 4 illustrates the mean relative errors calculated over the errors for all eight objects. It shows that Error 1, which considers the distance to the horizontal wall, is considerably higher than Error 2, which considers the distance to the vertical wall, independent of the employed algorithm. The reason is that the distances between the objects and the vertical wall is substantially larger than the distances between the objects and the horizontal wall so that the same absolute error results in a much higher relative error. Furthermore, the errors obtained for the first scenario are in general lower than the errors for the second scenario because of high error rates for the objects 1, 2, 3, and 4 in the right half of the environment. This higher error rates are a result of less obtained measurements to accurately determine the object positions because the robot did not drive into the right half of the environment in scenario 2.

When comparing the performance of the employed algorithms, Kart Mapping achieves overall the worst results for both scenarios and errors, while Frontier Exploration achieves similarly bad results for Error 1, but performs significantly better for Error 2. In comparison, Gmapping achieves better performance for both errors and scenarios, however, Error 1 is still more than 45% for both scenarios. In contrast, Hector Mapping achieves significantly lower Error 1 values, while the Error 2 values are nearly identical to Gmapping. Thus, Hector Mapping achieves overall the best performance, which is in line with the results of the visual inspection (Sect. 4.1).

## 5 Conclusions and Future Work

We evaluated different SLAM algorithms for highly dynamic environments in which all objects are only temporarily static. The evaluation procedure employed two different scenarios that were used to simulate high dynamicity in an otherwise static environment. Based on our conducted evaluation Hector Mapping produced the most accurate maps in both scenarios, even though it only uses laserscan data, while the second best algorithm Gmapping uses also odometry data to produce the map.

In future work, we will investigate whether Hector Mapping can be optimised for highly dynamic environments. Additionally, we will change to a faster robot because it would further reduce the number of measurements during one pass. Finally, we will verify the results by conducting tests in a more realistic environment, such as a real warehouse.

## References

1. Bahraini, M.S., Bozorg, M., Rad, A.B.: SLAM in dynamic environments via ML-RANSAC. *Mechatronics* **49**, 105–118 (2018)
2. Bresson, G., Alsayed, Z., Yu, L., Glaser, S.: Simultaneous localization and mapping: a survey of current trends in autonomous driving. *IEEE Trans. Intell. Veh. (T-IV)* **2**(3), 194–220 (2017)
3. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Trans. Rob.* **32**(6), 1309–1332 (2016)
4. Fink, G., Franke, M., Lynch, A.F., Röbenack, K., Godbolt, B.: Visual inertial slam: application to unmanned aerial vehicles. *IFAC-PapersOnLine* **50**(1), 1965–1970 (2017)
5. Grisetti, G., Stachniss, C., Burgard, W.: Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005
6. Hidalgo, F., Bräunl, T.: Review of underwater slam techniques. In: *6th International Conference on Automation, Robotics and Applications (ICARA)*, Queenstown, New Zealand, April 2015
7. Kohlbrecher, S., von Stryk, O., Meyer, J., Klingauf, U.: A flexible and scalable slam system with full 3D motion estimation. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Kyoto, Japan, November 2011
8. Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., Vincent, R.: Efficient sparse pose adjustment for 2D mapping. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010
9. Lu, F., Milios, E.: Globally consistent range scan alignment for environment mapping. *Auton. Robot.* **4**(4), 333–349 (1997)
10. Moravec, H.P., Elfes, A.: High resolution maps from wide angle sonar. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, March 1985
11. Santos, J.M., Portugal, D., Rocha, R.P.: An evaluation of 2D SLAM techniques available in robot operating system. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Linkoping, Sweden, January 2014

12. Siciliano, B., Khatib, O. (eds.): Springer Handbook of Robotics. Springer, Heidelberg (2008)
13. Sun, Y., Liu, M., Meng, M.Q.H.: Improving RGB-D SLAM in dynamic environments: a motion removal approach. *Robot. Auton. Syst.* **89**, 110–122 (2017)
14. Tan, W., Liu, H., Dong, Z., Zhang, G., Bao, H.: Robust monocular SLAM in dynamic environments. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, Australia, December 2013
15. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. Intelligent Robotics and Autonomous Agents. The MIT Press, Cambridge (2005)
16. Vincent, R., Limketkai, B., Eriksen, M.: Comparison of indoor robot localization techniques in the absence of GPS. In: Proceedings of Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XV, vol. 7664, April 2010
17. Wolf, D.F., Sukhatme, G.S.: Mobile robot simultaneous localization and mapping in dynamic environments. *Auton. Robot.* **19**(1), 53–65 (2005)
18. Xiang, L., Ren, Z., Ni, M., Jenkins, O.C.: Robust graph SLAM in dynamic environments with moving landmarks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, September–October 2015
19. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Monterey, CA, USA, July 1997
20. Yamauchi, B.: Frontier-based exploration using multiple robots. In: Proceedings of the Second International Conference on Autonomous Agents, Minneapolis, Minnesota, USA, pp. 47–53, May 1998
21. Zou, D., Tan, P.: CoSLAM: collaborative visual slam in dynamic environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(2), 354–366 (2013)



# An Aerial Robot Path Follower Based on the ‘Carrot Chasing’ Algorithm

Hector Perez-Leon<sup>(✉)</sup>, Jose Joaquin Acevedo, Jose A. Millan-Romera, Alejandro Castillejo-Calle, Ivan Maza, and Anibal Ollero

Robotics, Vision and Control Group, University of Seville,  
Avda. de los Descubrimientos s/n, 41092 Sevilla, Spain  
[{hectorperez,jacevedo,jmromera,acastillejo,imaza,aollero}@us.es](mailto:{hectorperez,jacevedo,jmromera,acastillejo,imaza,aollero}@us.es)

**Abstract.** This paper presents a three-dimensional path follower implementation for an aerial robot based on the carrot-chasing algorithm. The main objective was to improve the performance of the position controller of the PX4 autopilot when following a list of waypoints. This autopilot is widely used in the aerial robotics community, but we needed to improve its performance for navigation in cluttered environments. Different simulations have been carried out under the ROS (Robotic Operating System) environment for the comparison between the position controller of the PX4 and the proposed path follower. In addition, we have implemented different modes to generate the path from the input list of waypoints that are also analyzed in our simulation environment.

**Keywords:** Aerial robotics · Path following · Carrot chasing algorithm

## 1 Introduction

The use of aerial robots for different applications, such as surveillance [1, 2], wildfire tracking [3, 4], transportation [5] and bridge inspection [6, 7] has been increased significantly during the last years. A common requirement for all these applications is the precise, robust and efficient autonomous tracking of predefined paths by the aerial robots.

The path following problem for aerial robots is well studied in the literature, and there are different control based or geometric methods. Carrot-chasing [8], pure pursuit [9], vector field [10] and line-of-sight (LOS) [11] methods are some common geometric algorithms.

Sujit et al. [12] compared path following algorithms for straight lines and loiter paths that are easy to implement, take less implementation time and are robust to disturbances. The authors proved that the carrot-chasing algorithms have the worst performance due to wind disturbances and vector field algorithms are more accurate than the other two dimensional path following algorithms. To

---

This work is partially supported by the MULTIDRONE (H2020-ICT-731667) European project and the ARM-EXTEND (DPI2017-89790-R) Spanish project.

fix this issue, Nunez et al. [13] took into account the wind gusts as they play a key role in small prototypes.

Xavier et al. [14] compared three dimensional path following algorithms for loiter paths with and without wind disturbances. The authors demonstrated that vector field algorithms have largest errors than carrot-chasing and pure line-of-sight (PLOS) [15] methods.

In this paper, a three dimensional path follower implementation based on the carrot-chasing algorithm is presented. It can be used without any configuration or based on a list of parameters and increases the performance of the position controller of the PX4 autopilot when following a list of waypoints. It has been integrated with the *UAV Abstraction Layer*<sup>1</sup> (UAL) [16] previously developed by our research group within the ROS-MAGNA framework [17].

The rest of the paper is organized as follows. Section 4 defines the system architecture, which frames the proposed system. The path following problem is stated in Sects. 2 and 3 describes the proposed solution. Finally, validation results are presented in Sect. 5 and conclusions in Sect. 6 close the paper.

## 2 Problem Statement

This paper poses the path following problem for velocity-controlled aerial robots. An aerial robot  $Q$ , which current position is defined by  $\mathbf{p}(t) \in \mathbb{R}^3$  at any time  $t$ , has to track a path  $\Gamma$  of length  $L$ , defined by a curve  $\gamma(\lambda) \in \mathbb{R}^3$  with  $\lambda \in [0, L]$ .

Let us assume that  $Q$  is holonomic and velocity-controlled, being its velocity defined as  $\mathbf{v}(t)$  at any time  $t$ . Then, the aerial robot motion is controlled via velocity commands, such that  $\frac{d\mathbf{p}(t)}{dt} = \mathbf{v}(t)$ . On the other hand,  $\mathbf{v}(t)$  is bounded by  $v_{\max}$ , such that  $|\mathbf{v}(t)| \leq v_{\max}$  at any time  $t$ .

The objective is to implement a control system to generate velocity commands in order to track the path, minimizing the minimum normal distance between the actual trajectory travelled by  $Q$  and the path  $\Gamma$ , which is given by

$$J = \frac{1}{T} \int_0^T \min_{\lambda < L} |\mathbf{p}(t) - \gamma(\lambda)|, \quad (1)$$

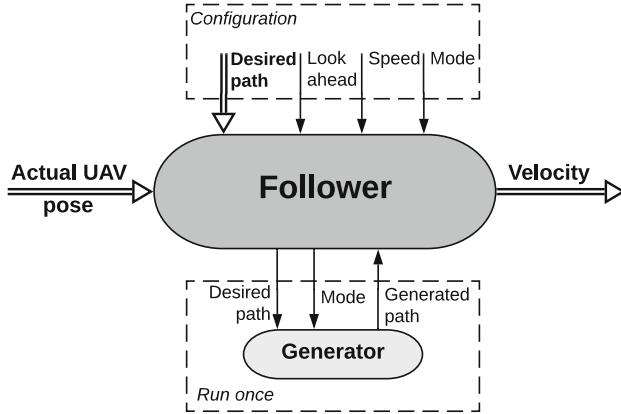
where  $T$  is the time taken to complete the task.

## 3 Proposed Approach

The proposed system has two main components: the path generator and the path follower. The user can interact with both components or just with the follower, which is the default way to use the proposed framework (see Fig. 1).

---

<sup>1</sup> <https://github.com/grvcTeam/grvc-ual>.



**Fig. 1.** The path follower design allows to use it by simply entering the desired path and the current position of the aerial robot. It also provides more configuration options to suit the user needs. The generator is called by the follower and runs once to generate a discrete curve.

### 3.1 Path Generator

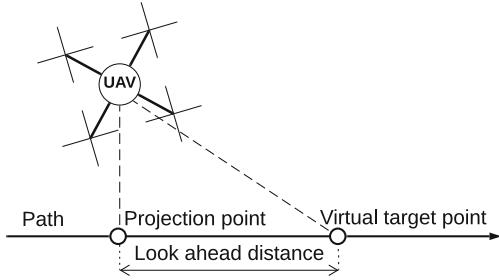
The path generator is in charge of generating a path  $\Gamma$  based on the ordered list of waypoints  $WP_l$  received. The generated path is a much more dense list of waypoints, which can be approximated to the continuous curve  $\gamma(\lambda)$  described in Sect. 2. It has three modes ( $m$ ) to generate a new path interpolating the initial list of waypoints, related to the type of curve used for the interpolation. Each mode has advantages and disadvantages, as it will be shown in Sect. 5, and the users should select the one that better fits their needs.

### 3.2 Path Follower

Initially, the path follower receives the desired path  $\Gamma$  defined as a list of waypoints  $WP_l$  and may receive three parameters: the look-ahead distance  $d$  (1.0 m by default), the cruising speed  $v_c$  (1.0 m/s by default) and the generator mode  $m$  (0 by default, see Sect. 5 to find more details about the modes). Parameter default values are conservative, but setting these values properly is crucial to obtain a good performance, depending on the desired path.

A much more dense list of waypoints is required to apply the path following method efficiently. Hence, it uses the path generator to get a discrete curve  $\gamma(\lambda)$  from the ordered list of waypoints  $WP_l$ , based on the generator mode  $m$ . Then, continuously, it receives the aerial robot pose  $\mathbf{p}(t)$  and generates the velocity commands  $\mathbf{v}(t)$ , based on the method described below.

**Path Following Method.** The proposed path following method is based on the ‘Carrot chasing’ algorithm and illustrated in the Fig. 2. The method runs as follows:



**Fig. 2.** Top view of the three dimensional path follower based on the carrot-chasing algorithm without taking into account the orientation error.

1. Obtain the  $\lambda_p$  argument as

$$\lambda_p(t) = \operatorname{argmin}_{\lambda \in [0, L]} |\mathbf{p}(t) - \gamma(\lambda)|, \quad (2)$$

which minimizes the distance from the aerial robot position to the path.

2. Add the look ahead distance and get the target virtual pose in the path as

$$\mathbf{p}_t(t) = \gamma(\lambda_p(t) + d). \quad (3)$$

3. Calculate the velocity command, based on the cruising speed, as

$$\mathbf{v}(t) = v_c \frac{\mathbf{p}_t(t) - \mathbf{p}(t)}{|\mathbf{p}_t(t) - \mathbf{p}(t)|} \quad (4)$$

to reach the target virtual pose.

The developed method includes two modes: following the path without changing yaw or aiming at the virtual point.

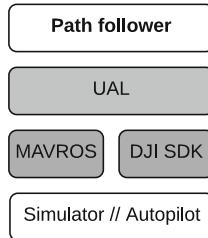
## 4 Software Implementation Details

The work described in this paper has been integrated with the UAL, which tries to abstract the user-programmer from the platform's autopilot, defining a common interface with a collection of the most used information and functionalities of an aerial robot. In particular, the developments presented in this paper are based on the release 2.2. of UAL and the Kinetic version of ROS<sup>2</sup> [18]. The proposed system receives a list of waypoints, generates a path using this list, and calculates which velocity vector should use UAL as reference to reach these waypoints.

The software architecture is split into four main layers, as depicted in Fig. 3. In the upper half is the proposed path follower, which has been packaged as a

---

<sup>2</sup> <https://www.ros.org/>.



**Fig. 3.** The different layers of the software architecture make the system modular. Different autopilots and simulators can be used due to the advantages of using the UAL.

node in the widespread ROS to facilitate experimentation and integration, and is built on top of UAL. The lower half of the software architecture is composed by the autopilots, simulators, and communication drivers. The UAL provides a back-end that works with MAVROS<sup>3</sup> which is in charge of providing a communication driver to ROS for various autopilots that uses MAVLink [19] as communication protocol. MAVROS is the ROS adaptation of MAVLink protocol. The simulator used in these developments is based on the PX4 *Software In The Loop* (SITL) [20] development which is the official SITL environment for the Pixhawk autopilot [21]. UAL has implemented another back-end which works using the ROS SDK that DJI provides to communicate with DJI protocols.

#### 4.1 Software User Interface

The system is written in C++, allowing a high performance, and offers a double interface in its current implementation:

- C++: the user may have access to all the functionalities of the framework creating an object in his code. Any ROS topic, service or action is not required to run this interface.
- ROS: The framework may work using ROS communications (topics, services and actions) if the user prefers to work with another programming language like Python. It publishes continuously its output and responds to service calls.

As this framework aims to improve behaviour using velocity control, it is recommended to use the C++ class interface to avoid communications delays present on ROS communications. The path follower is the main module in the system and the user can interact just with it to have a successful path following. The class generator is called automatically by the follower to simplify the interface with the user. However, if the framework is used from the ROS interface, the generator will be a fully completely independent node even though the interface will be the same (Table 1).

---

<sup>3</sup> <https://wiki.ros.org/mavros>.

**Table 1.** Double user interface implemented. UAL provides the possibility of manage multiple aerial robots, for that reason, each one has a namespace (ns) and a path follower associated.

C++		ROS
Path	preparePath (path, mode look ahead, cruising speed)	Service preparePath
Void	updatePose (pose)	Service updatePose
Void	updatePath (pose)	Service updatePath
Velocity	getVelocity()	Topic /[ns]/velocity

The interface is simple, the user can set everything up just with the *preparePath* method. To read the velocity that the aerial robot must use at that instant the user can read the output of the method *getVelocity*. Before reading the velocity, the user should give the aerial robot pose to the follower using *updatePose* in order to calculate correctly the velocity. The method *updatePath* can be used to change the path during the flight. It will not affect the behavior of the path following because it calculates the velocity referenced to the pose given of that instant.

The proposed framework is under continuously development and publicly available in a stable version along with examples and a guide of how to use it. It can be found in the GitHub repository<sup>4</sup> under the MIT License.

## 5 Validation Results

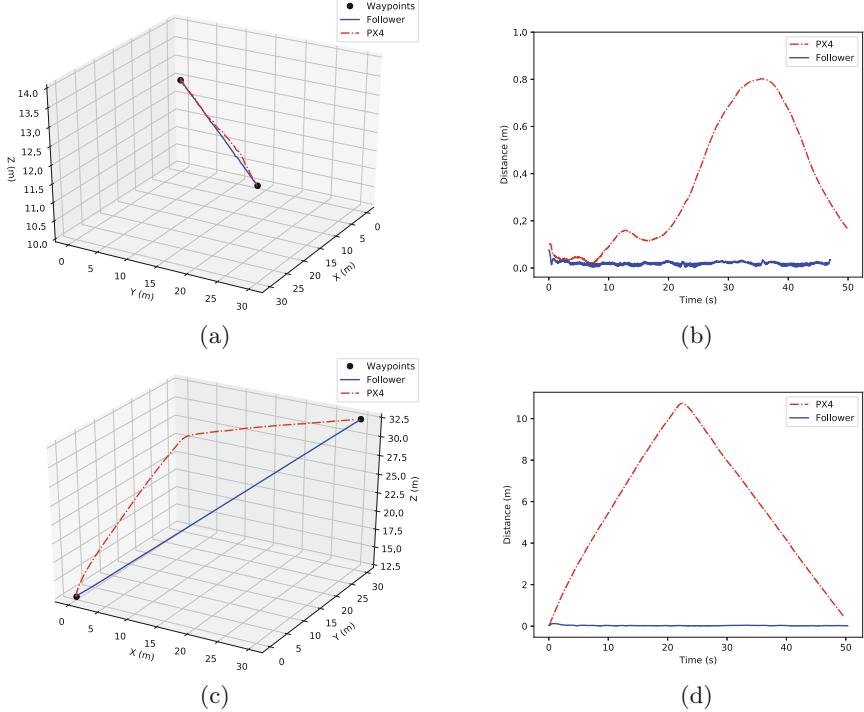
This section presents different simulations results using the proposed system. As UAL integrates the robot simulator Gazebo [22], the developed path follower may be tested easily in simulation using different aerial robot models. All the simulations presented here have been performed based on the same aerial robot model, a simulated autopilot based on the PX4 firmware and assuming a maximum speed of 1.0 m/s. The path follower has used the maximum speed of the aerial robot model as cruising speed. UAL allows to provide sequentially a list of waypoints to the PX4 position controller (using the UAL method *setPose*). Thus, these simulations compare the proposed system based on the developed solution with the original position controller.

First, the behavior of both systems are compared for a straight path at the same altitude. Fig. 4a, b and Table 2 show a better performance using the path follower rather than the system based on the position controller.

On the other hand, both systems are compared for a straight path varying the altitude. The results illustrated in Fig. 4c and d show the main problem of the method based on the position controller. As the simulated aerial robot model has different maximum velocities and accelerations on different axes, the aerial

<sup>4</sup> [https://github.com/hecperleo/upat\\_follower/tree/robot19](https://github.com/hecperleo/upat_follower/tree/robot19).

robot behavior is different on each axis. Table 2 presents significantly differences stating the proposed framework as a better solution to follow a list of waypoints in three dimensional space.



**Fig. 4.** Comparison between the path follower and the PX4 position controller going through two separate waypoints at same (a, b) and different (c, d) altitude. (a, c) Behavior of the comparison in a three dimensional view. (b, d) Detailed view of the values of the normal distance through the path.

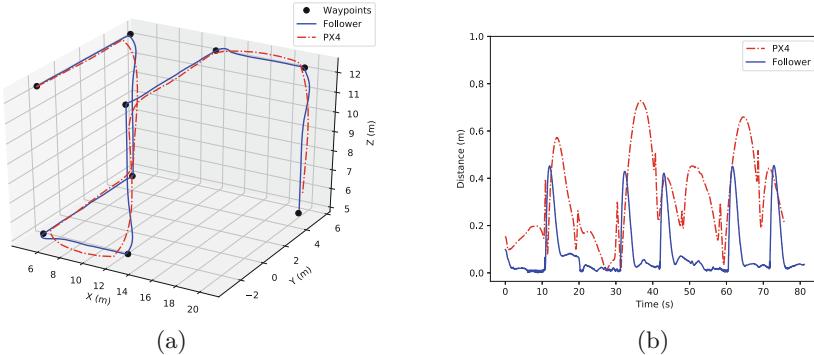
Also, the behaviors of both systems to track more complex paths have been simulated, see Fig. 5a. The results show how the proposed path follower solution works better than setting waypoints using PX4 position controller, see Table 3.

With respect to the path generator node and the generator mode, previous simulations have been performed using the generator mode 0, which uses linear interpolation between waypoints. Although this configuration improves the behavior with respect to the original position controller, it still has some problems. For example, the results show a maximum value of the normal distance of 0.454 meters, which coincides with the peaks shown in Fig. 5b, because the inertia of the aerial robot prevents from changing the course quickly.

If generator mode 2 is used, the generated path is more curve-shaped with smoothed corners based on cubic splines. Also, if a less curve-shaped path with

**Table 2.** Results of the normal distance of the path follower and the PX4 position controller going through two separate waypoints at same and different altitude.

	Same altitude		Different altitude	
Normal distance (m)	PX4	Follower	PX4	Follower
Mean	0.347	0.019	5.757	0.028
Maximum	0.802	0.076	10.733	0.124
Minimum	0.012	0.001	0.086	0.002
Variance	0.074	0.000	9.356	0.000
Standard deviation	0.271	0.008	3.059	0.018



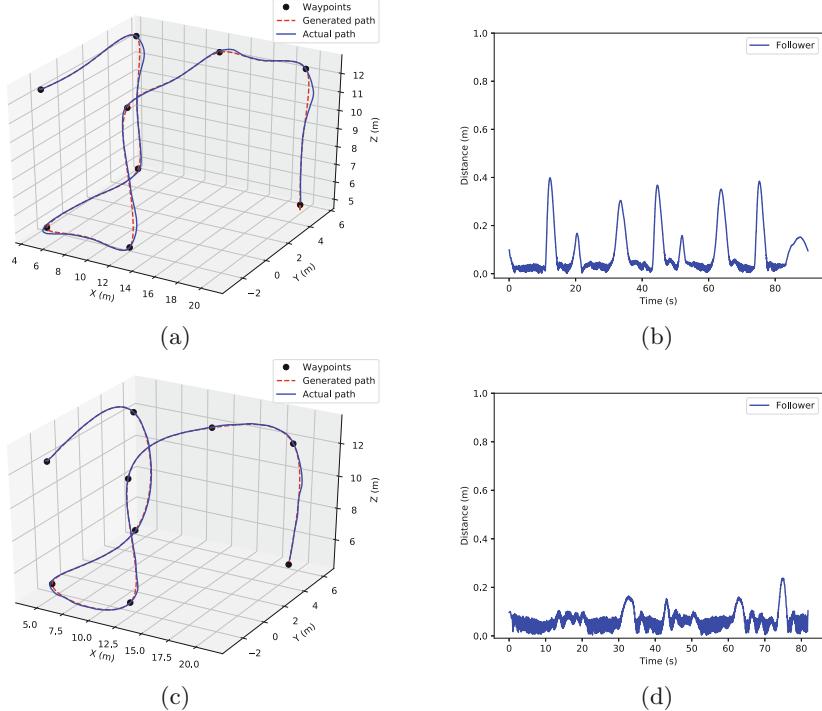
**Fig. 5.** Comparison between the path follower and the PX4 position controller following a list of waypoints. (a) Behavior of the comparison in a three dimensional view. (b) Detailed view of the values of the normal distance through the path.

smoothed corners is needed, generator mode 1 can be used, see Fig. 6a and c. The difference between these two modes is that mode 1 has a joint between each pair of waypoints so the three dimensional interpolation results on a less curve-shaped path. The results show better behavior by having a smoother path without abrupt course changes, because it reduces the peaks and the mean values of the normal distance, see Fig. 6b, d and Table 3.

Several videos of these simulations are publicly available on web page.<sup>5</sup> For each simulation, a video is provided with the configuration of the proposed path follower and the visualization using RViz<sup>6</sup> (ROS visualization). Extra videos, showing behavior of the aerial robot when too large or short look ahead distances are configured, are also provided.

<sup>5</sup> <https://grvc.us.es/robot19path>.

<sup>6</sup> <https://wiki.ros.org/rviz>.



**Fig. 6.** Path follower using the generator mode 1 and 2. (a, c) Behavior of the path follower in a three dimensional view. (b, d) Detailed view of the values of the normal distance through the path.

**Table 3.** Comparative of the behavior of the path follower using different generator modes. Mode 0 uses lineal interpolation. Mode 1 and 2 use cubic interpolation.

	Mode 0		Mode 1	Mode 2
Normal distance (m)	PX4	Follower	Follower	Follower
Mean	0.323	0.088	0.086	0.064
Maximum	0.728	0.454	0.399	0.238
Minimum	0.010	0.000	0.002	0.002
Variance	0.015	0.015	0.009	0.001
Standard deviation	0.186	0.122	0.095	0.038

## 6 Conclusions

This paper presents a path follower which improves the performance of the position controller of the PX4 autopilot when following a list of waypoints. It can be used without any configuration or based a list of parameters: look ahead distance, cruising speed, and the mode of the generator. Trying different values on

simulation several times is recommended before going to fly in the real world because these values directly depend on the input desired path.

The proposed path follower and the PX4 position controller have been compared. The path follower presented better behavior than the original controller on every case, but the difference is larger if the waypoint list had variations on altitude. The solution has been compared with different modes, showing different advantages and disadvantages of each one.

The obtained results shows that at higher speeds the aerial robot may oscillate about the path if a short look ahead distance is settled. On the other hand, if the user sets a large look ahead distance the aerial robot will cut corners because the aerial robot tries to turn towards each new virtual point.

The presented path follower may be placed between a path planner and the UAL. The resulting waypoints of the path planner should be sent to the presented framework instead of directly to UAL to increase significantly the whole performance with respect to the waypoints tracking precision.

## References

1. Basilico, N., Carpin, S.: Deploying teams of heterogeneous UAVs in cooperative two-level surveillance missions. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 610–615. IEEE, September 2015. <http://ieeexplore.ieee.org/document/7353435/>
2. Acevedo, J.J., Arrue, B.C., Maza, I., Ollero, A.: A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4735–4740, May 2014. <https://doi.org/10.1109/ICRA.2014.6907552>
3. Merino, L., Caballero, F., de Dios, J.M., Maza, I., Ollero, A.: An unmanned aircraft system for automatic forest fire monitoring and measurement. *J. Intell. Robot. Syst.* **65**(1), 533–548 (2012). <https://doi.org/10.1007/s10846-011-9560-x>
4. Pham, H.X., La, H.M., Feil-Seifer, D., Deans, M.C.: A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking. *IEEE Trans. Syst. Man Cybern. Syst.* 1–12 (2018). <http://arxiv.org/abs/1803.07926>, <http://ieeexplore.ieee.org/document/8331947/>
5. Kondak, K., Ollero, A., Maza, I., Krieger, K., Albu-Schaeffer, A., Schwarzbach, M., Laiacker, M.: Unmanned aerial systems physically interacting with the environment: load transportation, deployment, and aerial manipulation, pp. 2755–2785. Springer, Netherlands (2015)
6. Sanchez-Cuevas, P.J., Ramon-Soria, P., Arrue, B., Ollero, A., Heredia, G.: Robotic system for inspection by contact of bridge beams using UAVs. *Sensors* **19**(2) (2019). <https://www.mdpi.com/1424-8220/19/2/305>
7. Yoder, L., Scherer, S.: Autonomous exploration for infrastructure modeling with a micro aerial vehicle. In: Wettergreen, D.S., Barfoot, T.D. (eds.) *Springer Tracts in Advanced Robotics*, Springer Tracts in Advanced Robotics, vol. 113, pp. 427–440. Springer, Cham (2016)
8. Micaelli, A., Samson, C.: Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Ph.D. thesis, INRIA (1993)

9. Coulter, R.C.: Implementation of the pure pursuit path tracking algorithm. Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Technical report (1992)
10. Nelson, D.R., Barber, D.B., McLain, T.W., Beard, R.W.: Vector field path following for miniature air vehicles. *IEEE Trans. Robot.* **23**(3), 519–529 (2007)
11. Fossen, T.I., Breivik, M., Skjetne, R.: Line-of-sight path following of underactuated marine craft. *IFAC Proc. Volumes* **36**(21), 211–216 (2003)
12. Sujit, P.B., Saripalli, S., Sousa, J.B.: An evaluation of UAV path following algorithms. In: 2013 European Control Conference (ECC), pp. 3332–3337, July 2013
13. Nunez, H.E., Flores, G., Lozano, R.: Robust path following using a small fixed-wing airplane for aerial research. In: 2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015, pp. 1270–1278. Institute of Electrical and Electronics Engineers Inc. (2015)
14. Xavier, D.M., Natassya Silva, B.F., Branco, K.: Comparison of path-following algorithms for loiter paths of Unmanned Aerial Vehicles. In: Proceedings - IEEE Symposium on Computers and Communications, vol. 2018-June, pp. 1243–1248. Institute of Electrical and Electronics Engineers Inc. (2018)
15. Kothari, M., Postlethwaite, I., Gu, D.W.: A suboptimal path planning algorithm using rapidly-exploring random trees. *Int. J. Aerosp. Innov.* **2**, 93–103 (2010)
16. Real, F., Torres-Gonzalez, A., Ramon-Soria, P., Capitan, J., Ollero, A.: UAL: an abstraction layer for unmanned vehicles. In: 2nd International Symposium on Aerial Robotics (ISAR) (2018)
17. Millan-Romera, J.A., Perez-Leon, H., Castillejo-Calle, A., Maza, I., Ollero, A.: ROS-MAGNA, a ROS-based framework for the definition and management of multi-UAS cooperative missions. In: Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1–10. IEEE, June 2019
18. 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 System (2009)
19. Meier, L., Camacho, J., Godbolt, B., Goppert, J., Heng, L., Lizarraga, M., et al.: MAVLink: micro air vehicle communication protocol. Tillgänglig: <http://qgroundcontrol.org/mavlink/start>. [Hämtad 2014-05-22] (2013)
20. Meier, L., Honegger, D., Pollefeyns, M.: PX4: a node-based multithreaded open source robotics framework for deeply embedded platforms. In: Proceedings - IEEE International Conference on Robotics and Automation (2015)
21. Meier, L., Tanskanen, P., Fraundorfer, F., Pollefeyns, M.: PIXHAWK: a system for autonomous flight using onboard computer vision. In: 2011 IEEE International Conference on Robotics and Automation, pp. 2992–2997. IEEE (2011)
22. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), vol. 3, pp. 2149–2154, September 2004



# ROSS-LAN: RObotic Sensing Simulation Scheme for Bioinspired Robotic Bird LANding

Juan Pablo Rodríguez-Gómez<sup>(✉)</sup>, Augusto Gómez Eguíluz,  
José Ramiro Martínez-de Dios, and Aníbal Ollero

GRVC-Robotics Lab, University of Seville, 41092 Seville, Spain  
[{jrodriguezg,ageguiluz,jdedios,aollero}@us.es](mailto:{jrodriguezg,ageguiluz,jdedios,aollero}@us.es)  
<https://grvc.us.es/>

**Abstract.** Aerial robotics is evolving towards the design of bioinspired platforms capable of resembling the behavior of birds and insects during flight. The development of perception algorithms for navigation of ornithopters requires sensor data information to evaluate and solve the limitations presented during the flight of these platforms. However, the payload constraints and hardware complexity of ornithopters hamper the sensor data acquisition. This paper focuses on the development of a multi-sensor simulator to retrieve the sensor information captured during the landing maneuvers of ornithopters. The landing trajectory is computed by using a bioinspired trajectory generator relying on *tau theory*. Further, a dataset of the sensor information records obtained during the simulation of several landing trajectories is publicly available online.

**Keywords:** Tau theory · Ornithopter · Event-based cameras · LiDAR

## 1 Introduction

The development and implementation of bioinspired aerial platforms are mainly constrained by the lack of technology and scientific knowledge to resemble the behavior of birds during flight. Currently, most of the aerial vehicles fly by using either fixed-wing or rotary-wing configurations. Differently from the previous approaches, some bioinspired vehicles such as ornithopters employ flapping-wing mechanisms to generate the thrust and lift forces necessary to fly. Although there are some advances on the development of ornithopters [2–4, 7], the design of a robotic aerial vehicle capable of resembling the flight of birds is still under development and requires research on areas such as aerodynamics, electronics, mechanics, modeling, sensing, perception, and control.

Autonomous navigation of aerial robots requires a robust and efficient perception system to ensure precise mapping, GNSS-denied pose estimation, object detection and tracking, among others. The operation of ornithopters poses strong perception challenges. The fast movements of ornithopters during flight together

with the mechanical vibrations originated by the flapping motion require efficient and robust perception methods. In most cases only one sensor is not sufficient to capture all the relevant information of the scene. Thus, testing different sensor fusion techniques is a need that contrasts with their complex validation and the limited sensor integration onboard these aerial platforms. Further, integrating hardware in ornithopters requires significant effort due to the size constraints and the weight balance requirements of the platform. The low payload limitations and the high computational burden required for ornithopter perception recommend the implementation of perception techniques in dedicated hardware (e.g. FPGAs), which involves specific hardware development. Hence, a multi-sensor simulator is a useful tool to test and select sensors, methods, and algorithms previous to their implementation and testing in real ornithopter platforms.

This paper presents a multi-sensor simulation tool that provides simulated sensor measurements for the development and evaluation of perception and sensor fusion techniques for ornithopters platforms. It includes a bioinspired trajectory generator based on *tau theory* that simulates the trajectories performed by birds during landing and perching maneuvers [14]. These are maybe the most demanding tasks from a perception perspective. For simplicity, we define the term *tau trajectory* as the trajectory computed using *tau theory*. Existing research has proven that birds use a combination of simple strategies and the value of *tau* obtained from perceptual stimuli to guide most of their intended movements. The sensory data acquisition is necessary to develop perception algorithms to integrate *tau theory* on robotic platforms. The presented sensory simulation tool receives the input parameters that define the *tau trajectory* and generates as output the measurements from the onboard sensors during the *tau trajectory* simulation. This work has been developed in the context of the ERC-GRIFFIN and ARM-EXTEND projects. Our simulation tool provides support by collecting sensory data to be used in the development of novel perception algorithms for GRIFFIN robots. We believe this sensing simulation scheme can contribute and boost R&D in perception systems for ornithopter robots.

Summarizing, the contribution of this work is two-fold. First, it presents a simulation tool to generate simulated multi-sensor measurements obtained from the sensors onboard an ornithopter during landing and perching maneuvers. The simulated sensors include those with the highest interest in ornithopters perception such as frame-based cameras, event-based sensors, solid-state LiDARS, range sensors, altimeters, among others. Second, a dataset with the sensor measurements provided by our simulator is published online<sup>1</sup>. The recorded sensor information corresponds to the simulation of several landing and perching trajectories in different scenarios. This paper is structured as follows. Section 2 describes the related work of bioinspired trajectory generation and mobile robotic simulation. The design methodology of simulation is presented in Sect. 3. Section 4 defines the simulation architecture explaining each block of the simulator. The dataset generation description is explained in detail in Sect. 5. Finally, Sect. 6 includes the conclusions and future work.

---

<sup>1</sup> <https://grvc.us.es/bioinspired-landing-trajectory-sensor-dataset/>.

## 2 Related Work

Ornithopter simulators mainly focus on the study of dynamics and flight control. For instance, the stability and controllability of the aircraft are analyzed in [8] to control the non-linear flight of a flapping-wing robot. The flexible multi-body dynamics of an ornithopter is simulated in [16] considering fluid-structure interaction and flight dynamic behavior to design a robotic model capable of flying in trim. The flappy hummingbird [6] is an open source dynamic simulator of flapping-wing micro aerial vehicles created to facilitate the design and validation of flight control architectures for flapping-wing robots. Although these tools are useful to simulate ornithopters dynamics, kinematics and aerodynamic effects, to the best of the author's knowledge there is not a simulation tool designed to retrieve sensor information during their flight.

Robotic simulation tools are useful for testing and evaluating robotics algorithms before applying them on the real platforms. The use of simulators enlarge security and optimize time by evaluating the behavior of robots in challenging scenarios without endangering the platform and users. Among the most popular robotic simulators, there are Gazebo [11] and V-REP [19]. Game engines are also used for the development of robotic simulators. Engines are preferred in applications that require high framerate and photorealistic rendering. Airsim [20] and CARLA [5] are some of the examples of game engine simulators used for robotics and artificial intelligence applications. Our multi-sensor simulation architecture retrieves the sensor measurements during the flight of ornithopter robots by integrating two simulation tools in parallel; Gazebo and Unreal Engine 4 (UE4). The former simulates different sensors such as frame-based cameras, lasers, force and range sensors. The last is part of the event camera simulator [17], the benefit of using this sensor in our application is explained in Sect. 3.2. Both simulators are integrated in the Robot Operating System (ROS).

## 3 Design

The aim of this work focuses on retrieving simulated multi-sensor measurements for robotics perception during the landing of an ornithopter. For this purpose, we propose the design of a tool to simulate different perception sensors that includes a bioinspired trajectory generator to retrieve the type of movements exerted by birds for landing and perching. Thus, simulation development has to satisfy the following design requirements. First, the platform has to be capable of simulating bioinspired trajectories to resemble the movements performed by birds during flight. Tau theory describes the principle used by animals and humans to guide their motion to make contact with an object or surface using the time-to-contact as reference. The theory has been used in aerial robotics to guide and control the motion of multicopter platforms for docking and landing [10].

Second, the simulator has to include the type of sensors most widely used for robotics perception. The hardware sensor selection takes into consideration the type of information necessary for localization, mapping, obstacle avoidance,

and object identification. We analyze the problem from a robotic perception perspective and select the sensors required for these tasks. However, resembling the flight of birds involves continuous fast movements and strong scene changes in dynamic environments. These facts constrain the hardware selection: the sensors should process information and correct their measurements with the lowest possible noise at the highest frame rate. Our sensor selection considers a variety of sensors typically used in aerial robotics such as LiDAR, IMU, altimeters and vision sensors. It also integrates event-based cameras to deal with the fast-motion limitations and low-latency measurements with high dynamic range. Fusing events from event cameras together with classical perception information (e.g images, point clouds, and IMU measurements) increases the sensor robustness for the development of future SLAM and object detection algorithms for flapping-wing platforms. The most relevant aspects about the planning of the *tau trajectory* and a brief introduction to event cameras functioning and applications are detailed below.

### 3.1 Tau Theory Planning

The simulation architecture presented in this paper relies on *tau theory* [13] to approximate the landing and perching maneuvers of ornithopters. Tau theory postulates that humans and animals (i.e. notably birds) use a combination of simple strategies and the value of *tau* to guide and control most of their intended movements. Tau ( $\tau$ ) is a variable related to the time an observer would take to contact an object or surface if the speed remains constant. The value of  $\tau$  provides a first order approximation of the time-to-contact (TTC) for a given gap  $\chi$ :

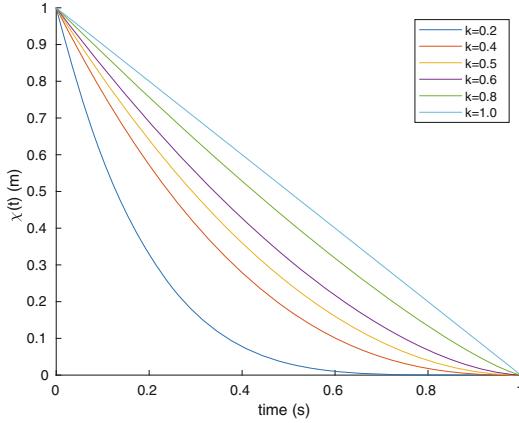
$$\tau(t) = \frac{\chi(t)}{\dot{\chi}(t)}, \quad (1)$$

where  $\chi(t)$  and  $\dot{\chi}(t)$  are the gap and its rate of closure at time step  $t$ . The gap  $\chi$  is, by convention, always negative and the initial closure rate  $\dot{\chi}(0)$  is positive. The work in [13] found that birds tend to keep the gap closure rate constant to control their deceleration. This behavior was defined as a constant tau-dot strategy. Subsequent research showed that zero velocity at contact is reached when  $\dot{\tau}$  is kept constant, positive and less than 0.5. Another variant of tau-dot strategy was proposed in [10] to guide a breaking maneuver using  $\hat{\tau}(t) = kt + \tau(0)$ , which entails a more practical approximation to tau-dot strategy, i.e. the computation of  $\dot{\tau}$  is avoided. It is worth noting that the tau theory postulates that animals do not require cognitive processing for TTC since it is available at neural circuit level [13]. Thus, the value of a gap and its closure rate can be computed as:

$$\chi(t) = \chi(0) \left(1 + kt \frac{\dot{\chi}(0)}{\chi(0)}\right)^{\frac{1}{k}}$$

$$\begin{aligned}\dot{\chi}(t) &= \chi(0) \left(1 + kt \frac{\dot{\chi}(0)}{\chi(0)}\right)^{\left(\frac{1-k}{k}\right)}, \\ \ddot{\chi}(t) &= \frac{\dot{\chi}(0)^2}{\chi(0)}(1-k) \left(1 + kt \frac{\dot{\chi}(0)}{\chi(0)}\right)^{\left(\frac{1-2k}{k}\right)}\end{aligned}\quad (2)$$

where  $\chi(0)$  is the initial gap value, and  $0 < k \leq 0.5$  guarantees the gap and its closure rate reach zero at the same finite time  $T = \frac{-\tau(0)}{k}$ . A further analysis shows that for  $0.5 < k < 1$  the gap closes to zero with  $\dot{\chi}(0) \neq 0$  leading to a collision. Figure 1 shows the gap closure for different values of  $k$ . It is worth mentioning that the figures of this paper show the main gap (e.g  $z$ ) positive contrary to the convention being consistent with a landing trajectory.



**Fig. 1.** Trajectories  $\chi$  for  $0 < k \leq 1$  closing the gap at  $t = 0$ . For  $k = 1$  the gap closes with  $\dot{\chi}$  constant producing a collision.

The simulation of landing or perching trajectories of a bird requires the closure of gaps for position and orientation. Therefore, an extension of tau-constant strategy for more than one gap is required. Closing multiple gaps can be achieved through tau-coupling strategy, which consists of coupling additional gaps  $\vartheta$  to the main gap  $\chi$ . The tau-coupling is computed as  $\tau^\chi = \kappa \tau^\vartheta$ , where  $\kappa$  is a closure constant ratio between both gaps. Therefore, a coupled gap  $\vartheta$ , its closure rate  $\dot{\vartheta}$ , and acceleration  $\ddot{\vartheta}$  at time  $t$  are obtained as:

$$\begin{aligned}\vartheta(t) &= c \chi(t)^{\frac{1}{\kappa}-1}, \\ \dot{\vartheta}(t) &= c \frac{1}{\kappa} \dot{\chi}(t) \chi(t)^{\frac{1}{\kappa}-1}, \\ \ddot{\vartheta}(t) &= c \frac{1}{\kappa} \left( \left( \frac{1}{k} - 1 \right) \dot{\chi}(t)^2 + \chi(t) + \ddot{\chi}(t) \right) \chi(t)^{\frac{1}{\kappa}-2},\end{aligned}\quad (3)$$

where  $c = \frac{\vartheta(0)}{\chi(0)^{(1/\kappa)}}$ .

In the context of the proposed *tau trajectory* generation, the trajectory is obtained by closing a main gap in the  $z$ -axis and three coupled gaps in the  $x$  and  $y$  axis, and the roll  $\theta$ , pitch  $\phi$  and yaw  $\psi$  angles. For simplicity, each gap will be denoted with their corresponding superscript, and a subscript will denote the moment of time. For instance, the main gap in the  $z$ -axis and a couple gap on the yaw  $\psi$  at time  $t$  will be denoted  $\chi_t^z$  and  $\vartheta_t^\psi$  respectively. The trajectory  $S_t$  at time  $t$  includes the gap values, their closure velocities, and accelerations represented in a reference frame  $F$  aligned with the goal position and orientation. The transformation of  $F$ , the reference frame to a global reference frame, can be computed in a straightforward manner. Hence, the proposed method is applicable to any set of initial and target configurations. The proposed method for tau-theory based trajectory generation provides a compact way of estimating trajectories that are similar to those that birds perform during landing and perching maneuvers.

### 3.2 Event Cameras

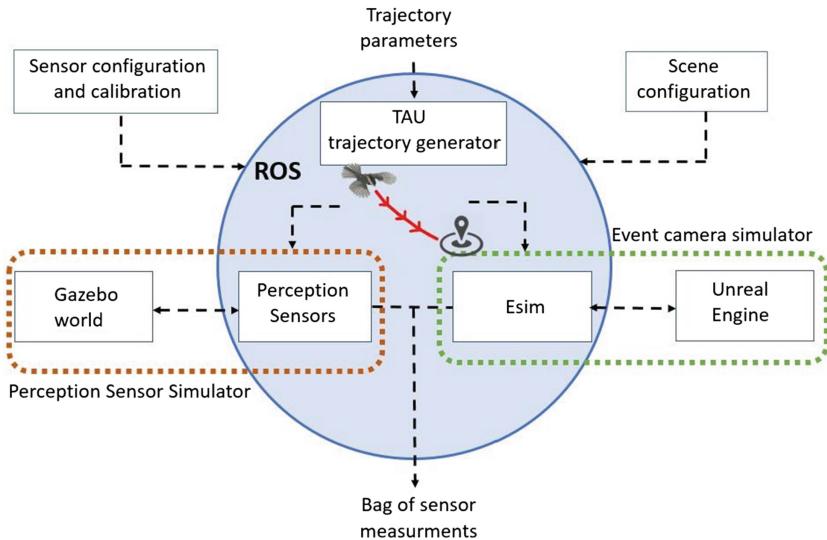
Event cameras are silicon retina vision sensors that mimic the neural behavior and architecture of the retina. These devices are modifying the paradigm of retrieving scene information using vision sensors. Unlike frame-based traditional cameras, event-driven sensors provide information based on pixel intensity variation. Events are triggered asynchronously whenever the intensity of a pixel exceeds a specific threshold and transmitted using the Address Event Representation (AER) Protocol. The sensor provides high temporal resolution generating events with a resolution of  $\mu$  seconds. Besides, event cameras offer a high dynamic range (140 dB) and low power consumption. These capabilities have increased the interest of the computer vision and robotic communities to use these sensors in applications with challenging conditions for frame-based cameras.

Events are defined as a tuple of the form  $e = (t, x, y, p)$ , where  $t$  is the time in which the events are triggered,  $(x, y)$  is the pixel position, and  $p$  is the polarity (i.e. either 1 or 0). The event data format limits the use of events in computer vision and robotics applications as most of the state of the art algorithms cannot be directly applied to the stream of events. During the last years, different novel methodologies have been developed to adapt classical algorithms to use events in applications such as feature detection and tracking [21], optical flow estimation [1], visual odometry [12] and SLAM [18]. These advances contribute to the research in a novel area and lead the sensor integration between event-driven sensors and other sensors such as an IMU and frame-based cameras [22].

Few event camera simulators have been reported. In [9] the difference between consecutive images is used to generate edges that resemble the events produced by the edges of moving objects. The method proposed in [15] attempts to simulate the behavior of an event camera by using the high frame rate capabilities of Blender to sample images like in a continuous timeline. This feature adds the low-latency properties of event cameras. The simulator was improved in [17] by simulating the asynchronous behavior of the retinas by triggering events based on the prediction of the image dynamics.

## 4 The Architecture of the Presented Simulator

The presented multi-sensor simulation tool adopts an architecture based on ROS integrating Gazebo and *Unreal Engine 4* (UE4). It combines the advantages of ROS to integrate drivers, packages, and state of the art robotics algorithms with the advantages of UE4 to provide fast realistic rendering, easy portability, and an intuitive block programming interface. ROS handles the bioinspired trajectory generator based on *tau theory* to simulate the trajectories performed by birds during landing and perching. It also integrates different sensor simulators such as feature-based cameras, event cameras, and proximity sensors; and publishes the sensor measurements on ROS topics. On the other hand, UE4 handles the simulation of a photo-realistic version of the scenario to capture images of the scene at a very high frame rate. Gazebo simulates a simple version of the environment for sensors such as the Velodyne, lasers, sonar, and other sensors.



**Fig. 2.** Block diagram of the simulation architecture.

Figure 2 shows the block diagram of the architecture. The system receives three types of inputs. First, the parameters to compute the *tau trajectory*  $S_t$  as described in Eqs. (2) and (3). Second, the scene configuration file with the pose and orientation of each object in the scene. Third, the sensors configuration and calibration. For instance, the simulation model of the frame-based camera requires the camera calibration file in *yaml* format including the camera matrix and the optical distortion coefficients. Each simulated sensor requires the definition of the coordinate frame of the sensor w.r.t. the reference frame of the simulated robot. The proposed simulation tool includes the following sensors: event camera, Velodyne HDL-32, frame-based camera, laser, altimeter, and

IMU. Due to its flexible and modular architecture, it can be easily extended by including additional devices such as stereo cameras and depth sensors. The output of the simulation tool is a rosbag file with the measurements taken by the simulated sensors while following trajectory  $S_t$  also included in the recording file.

The simulation tool is divided into three main modules: the *Tau Trajectory Generator*, the *Perception Sensors Simulator* and the *Event Camera Simulator*. The *Tau Trajectory Generator* module computes trajectory  $S_t$  using the approach described in Sect. 3.1. The proposed implementation generates up to the 18 components to define the pose ( $\vartheta_t^x, \vartheta_t^y, \chi_t^z, \vartheta_t^\theta, \vartheta_t^\phi, \vartheta_t^\psi$ ), the velocities ( $\dot{\vartheta}_t^x, \dot{\vartheta}_t^y, \dot{\chi}_t^z, \dot{\vartheta}_t^\theta, \dot{\vartheta}_t^\phi, \dot{\vartheta}_t^\psi$ ), and accelerations ( $\ddot{\vartheta}_t^x, \ddot{\vartheta}_t^y, \ddot{\chi}_t^z, \ddot{\vartheta}_t^\theta, \ddot{\vartheta}_t^\phi, \ddot{\vartheta}_t^\psi$ ) for each time  $t$ . The inputs of the trajectory generation block are the initial pose in the target reference frame  $F$ , and the parameters  $k^z, \kappa^x, \kappa^y, \kappa^\theta, \kappa^\phi$ , and  $\kappa^\psi$ , where the upper script defines the tau theory gap.

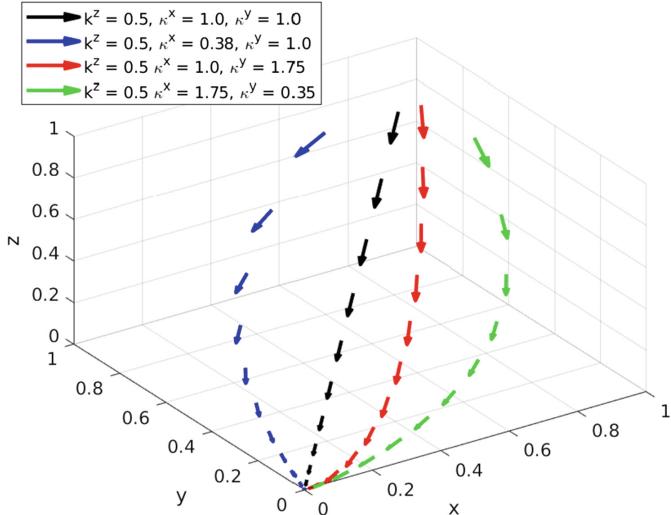
The *Perception Sensors Simulator* module contains the simulation model of sensors such as Velodyne, altimeter, and IMU. The simulator moves the reference frame of the sensors in the virtual scenario by following the *tau trajectory*. The input of the simulation tool includes the external calibration (transformation matrix of the sensor local reference frame w.r.t. the ornithopter frame) and internal calibration files for each simulated sensor.

The *Event Camera Simulator* is based on *esim* [17]. The simulator was modified to receive the input from our *tau trajectory* generator keeping the correct orientation during the simulation. Further, the reference frame of the sensor is modified by a transformation matrix that references it to the reference frame of the scene. The simulator runs in ROS and uses Unreal Engine 4 to render images from a 3D scenario at a very high frame rate (from 100 Hz to 1 kHz). The asynchronous behavior of event cameras is achieved in simulation using two key facts: (i) rendering images at very high frame rate on the engine, and (ii) sampling frames adaptively to generate events asynchronously using the prediction of the optical flow. The simulator moves the event camera following the sensor external calibration and  $S_t$ , the trajectory computed using *tau theory*.

## 5 Datasets

The dataset provides the simulated sensor measurements obtained during the landing of the ornithopter using different landing trajectories. For each simulation, the onboard sensors move by following the trajectory obtained with the tau trajectory generator. Each trajectory was computed by using a final approach angle of  $\pi/6$  and an initial velocity in a range between 3 to 5 m/s as in a real ornithopter. The trajectories were sampled at  $\Delta t = 0.1$  s until  $t = T$ , where  $T$  is computed as described in Sect. 3.1. Figure 3 shows different normalized trajectories obtained from the tau trajectory generator when performing the simulated landing maneuver. The trajectories were computed by varying the values of  $\kappa^x$  and  $\kappa^y$  as it is described in the figure, and setting  $k^z = 0.5$  to obtain a smooth

descending without colliding with the ground. The robot poses along the trajectories are shown as vectors. The position of the vector represents the ornithopter position in  $(x, y, z)$  while its magnitude and direction are defined by the linear velocity components at each point of the trajectory.

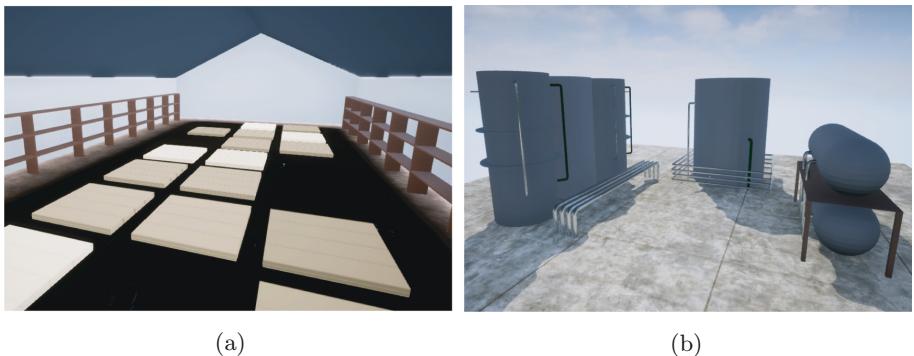


**Fig. 3.** Different normalized trajectories describing the simulated landing maneuver of the ornithopter. The figure shows the magnitude and direction of the approach velocity vectors.

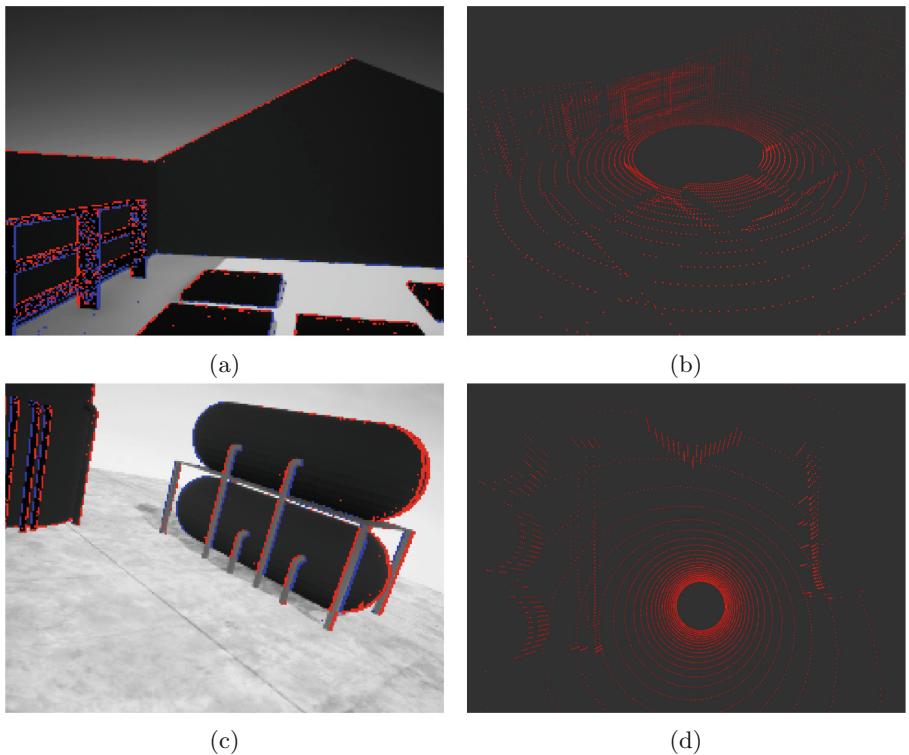
Several simulation scenarios were developed. The scenarios were inspired by industrial environments such as warehouses and factories (Fig. 4) where ornithopter robots could provide robotic support on tasks such as surveillance, payload delivery, and remote sensing. The objects in the developed scenarios were designed in *Blender*. Additional scenarios can be integrated by simply importing the models of the scenes in both Gazebo and UE4. The model integration preserves the scene configuration (i.e position, orientation, and scale) to avoid mismatches between the sensor measurements from each simulator.

In general, the time to simulate the sensor measurements  $t_s$  along the trajectory is longer than the trajectory duration as the simulator performs several computations to produce all the multi-sensory information. However, it is worth to mention that the reference time of the measurements does not correspond to  $t_s$  as it is referred to the trajectory timestamp  $t_t$ . The publish rate of the dataset can be adjusted while running the bagfile by setting the factor  $r$  of ROS to  $t_s/t_t$ .

Our dataset is divided into six simulations each for a different landing trajectory. For each trajectory, the measurements from the monocular camera, event camera, sonar, IMU and Velodyne LiDAR are recorded in a rosbag file. The dataset also includes the ground truth pose information of the sensors during the landing maneuver, see Fig. 5. Our architecture allows the integration



**Fig. 4.** Simulation scenarios to perform the sensors measurements simulation. A warehouse (a), and an oil refinery (b).



**Fig. 5.** Example of the sensor measurements obtained during the simulation of a landing trajectory on the warehouse and refinery scenarios. Rendered images including the generated events during a time window of 10 ms (a, c), Point cloud from the simulated Velodyne LiDAR (b, d). The lighting of image (a) was modified for better visualization.

of additional sensors available on the *gazebo\_ros\_package* such as Kinect, lasers and depth cameras. Each simulation of the dataset contains the following files:

- The rosbag file with the sensor measurements.
- A file with the instructions to run the bag.
- The input bioinspired trajectory computed using *tau theory*.
- The events generated during the simulation in a text file using the format (timestamp,  $x, y$ , polarity).
- The model of each object of the scene including a file with the object poses.

## 6 Conclusions

In this work, we introduce a simulation tool to retrieve the sensing information captured during the landing and perching maneuvers of ornithopter robots. The operation of ornithopters poses very strong perception challenges, which contrasts with the effort-demanding implementation and testing with these aerial platforms. This work is motivated by the need to retrieve the missing realistic multi-sensory measurements to develop, debug and tune perception and autonomous navigation methods for ornithopter robots. Our simulator moves the sensors describing bioinspired trajectories computed using *tau theory*.

A dataset with several trajectories and sensor measurements is publicly available online. Each dataset provides the bioinspired trajectory followed during the simulation along with the sensing information from different sensors. We hope that our dataset boosts the development of novel perception algorithms for robot localization, mapping, and object detection for ornithopters before the development of a platform capable of carrying the necessary sensors for robotic perception. Finally, our future work points towards the development of such perception algorithms particularly aiming to fuse the information provided by event cameras and classical perception sensors to solve the perception challenges presented by the flight of flapping-wing vehicles.

**Acknowledgements.** This work was supported by the European Research Council as part of GRIFFIN ERC Advanced Grant 2017 (Action 788247) and the ARM-EXTEND project funded by the Spanish National RD plan (DPI2017-89790-R).

## References

1. Benosman, R., Clercq, C., Lagorce, X., Ieng, S., Bartolozzi, C.: Event-based visual flow. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(2), 407–417 (2014)
2. Corporate, F.: Smartbird (2011). [https://www.festo.com/net/SupportPortal/Files/46270/Brosch\\_SmartBird\\_en\\_8s\\_RZ\\_110311\\_lo.pdf](https://www.festo.com/net/SupportPortal/Files/46270/Brosch_SmartBird_en_8s_RZ_110311_lo.pdf)
3. Corporate, F.: Bionicflyingfox (2018). [https://www.festo.com/net/SupportPortal/Files/492827/Festo\\_BionicFlyingFox\\_en.pdf](https://www.festo.com/net/SupportPortal/Files/492827/Festo_BionicFlyingFox_en.pdf)
4. Croon, G., Perçin, M., Remes, B., Ruijsink, R., Wagter, C.: The DelFly (2016)
5. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: an open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning, pp. 1–16 (2017)

6. Fei, F., Tu, Z., Yang, Y., Zhang, J., Deng, X.: Flappy hummingbird: an open source dynamic simulation of flapping wing robots and animals. arXiv preprint [arXiv:1902.09628](https://arxiv.org/abs/1902.09628) (2019)
7. Folkertsma, G., Straatman, W., Nijenhuis, N., Venner, C., Stramigioli, S.: Robird: a robotic bird of prey. *IEEE Robot. Autom. Mag.* **24**(3), 22–29 (2017)
8. Han, J., Lee, J., Kim, D.: Ornithopter modeling for flight simulation. In: 2008 International Conference on Control, Automation and Systems, pp. 1773–1777 (2008)
9. Kaiser, J., Tieck, J., Hubschneider, C., Wolf, P., Weber, M., Hoff, M., Friedrich, A., Wojtasik, K., Roennau, A., Kohlhaas, R., et al.: Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks. In: 2016 IEEE International Conference on SIMPAR, pp. 127–134 (2016)
10. Kendoul, F.: Four-dimensional guidance and control of movement using time-to-contact: application to automated docking and landing of unmanned rotorcraft systems. *Int. J. Robot. Res.* **33**(2), 237–267 (2014)
11. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: IEEE/RSJ IROS 2004, vol. 3, pp. 2149–2154 (2004)
12. Kueng, B., Mueggler, E., Gallego, G., Scaramuzza, D.: Low-latency visual odometry using event-based feature tracks. In: IEEE/RSJ IROS 2016, pp. 16–23 (2016)
13. Lee, D.: General tau theory: evolution to date. *Perception* **38**(6), 837 (2009)
14. Lee, D., Bootsma, R., Land, M., Regan, D., Gray, R.: Lee’s 1976 paper. *Perception* **38**(6), 837–858 (2009)
15. Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., Scaramuzza, D.: The event-camera dataset and simulator: event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Res.* **36**(2), 142–149 (2017)
16. Pfeiffer, A., Lee, J., Han, J., Baier, H.: Ornithopter flight simulation based on flexible multi-body dynamics. *J. Bionic Eng.* **7**(1), 102–111 (2010)
17. Rebecq, H., Gehrig, D., Scaramuzza, D.: ESIM: an open event camera simulator. In: Conference on Robot Learning, pp. 969–982 (2018)
18. Rebecq, H., Horstschäfer, T., Gallego, G., Scaramuzza, D.: EVO: a geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robot. Autom. Lett.* **2**(2), 593–600 (2017)
19. Rohmer, E., Singh, S.P., Freese, M.: V-REP: a versatile and scalable robot simulation framework. In: IEEE/RSJ IROS 2013, pp. 1321–1326 (2013)
20. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics, pp. 621–635 (2018)
21. Vasco, V., Glover, A., Bartolozzi, C.: Fast event-based harris corner detection exploiting the advantages of event-driven cameras. In: IEEE/RSJ IROS 2016, pp. 4144–4149 (2016)
22. Vidal, A., Rebecq, H., Horstschaefer, T., Scaramuzza, D.: Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robot. Autom. Lett.* **3**(2), 994–1001 (2018)



# Low Cost Binaural System Based on the Echolocation

Thiago Fernandes Moya Moreira<sup>1</sup> , José Lima<sup>2</sup> , Paulo Costa<sup>3</sup> , and Márcio Cunha<sup>1</sup>

<sup>1</sup> Academic Department of Electronic, Federal University of Technology - Paraná, Curitiba, Brazil  
[thiagomoya@gmail.com](mailto:thiagomoya@gmail.com)

<sup>2</sup> INESC TEC - INESC Technology and Science and CeDRI - Research Centre in Digitalization and Intelligent Robotics, Polytechnic Institute of Bragança, Bragança, Portugal

<sup>3</sup> INESC TEC - INESC Technology and Science and Faculty of Engineering of University of Porto, Porto, Portugal

**Abstract.** Ultrasonic sensors offers attractive features at an affordable cost. The main problem faced by the use of these devices is that the data obtained are not so easy to interpret, restricting their efficiency. This paper describes a binaural sensor system that is able to determine the coordinates of an object or a target in a two-dimensional space, focusing on mathematical and signal processing techniques to provide accurate measurements and increase the system reliability. The proposed work consists only of low cost components, which aims to demonstrate that improvement is possible. Experimental tests, performed in different scenarios, reported good accuracy and repeatability of the measurements.

**Keywords:** Ultrasonic sensor · Echolocation · Object detection

## 1 Introduction

In mobile robot applications the knowledge of the environment in which the device is inserted and the objects that surround it are essential information for a successful navigation, avoiding collisions and increasing the system safety. To achieve this goal the use of sensors becomes indispensable and its features must be taken into account [1]. Among the different types of sensors, many researches have been exploring the ultrasonic sensors to deal with this problem attracted by the low cost, flexibility in several environments and applications where vision based sensors fail due to some constraints like low visibility, mirrors or clear objects [2].

Typically, the employment of ultrasonic sensing in an industrial or commercial atmosphere is restricted to simple tasks, such as obstacles avoidance that do not require accurate measurements. The leading reason for this scenario is due to the fact that the data obtained from the commercial ultrasonic range-finders are

not easy to be interpreted. Therefore, the literature has shown a great interest in investigating the use of these sensors for complex applications [3–5].

The result of a measurement performed by a standard sensor occurs through an ultrasonic burst with a single frequency, which can range from 40 kHz to 50 kHz. The distance is calculated using the Time of Flight (ToF) when the echo emitted from the transmitter reflects an obstacle or object and returns back to the receiver [6]. In general, the reflected signal is detected by a predefined trigger level, however this method is not so efficient [7]. Thus, to improve the system accuracy, some enhanced approaches have combined advanced signal processing techniques [8, 9].

Conventional ultrasonic range-finders mostly are composed of a single transmitter and a single receiver or a transducer that acts as both. The main disadvantage of this system is the wide beam of the emitted signal, causing a very poor resolution. Furthermore, other phenomena like the multi-reflection effect may arise and cause false positives. In order to reduce these issues, a system that contains at least two receivers and a transmitter is needed [6].

This paper proposes a low cost ultrasonic binaural system that combines several mathematical and signal processing techniques to improve the accuracy and performance of the system. Also, the characteristics associated with the use of ultrasonic measurements in indoor environments are discussed.

After this brief introduction, in Sect. 2, a review about ultrasonic sensing is performed. In Sect. 3, necessary techniques to develop the system are presented. In Sect. 4, the architecture of the created system is introduced. In Sect. 5, the main objective of the proposed work and the procedures to reach it are described. In Sect. 6, the results of the experimental tests are reported. Lastly, in Sect. 7 the results are discussed and conclusions drawn.

## 2 Ultrasonic Sensing

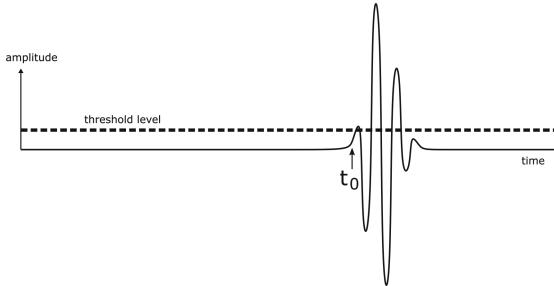
The operating principle used by acoustic sensors, such as ultrasonic, is mainly based on the ToF estimation. The transmitter generates an ultrasonic pulse which propagates through the air and when it detects a target the signal is reflected back to the receiver. In this sense, the ToF can be determined by the time the signal is emitted and received by the sensor [10]. Equation 1 shows how the distance measurement is achieved from the estimated ToF.

$$d = \frac{c \cdot T_f}{2} \quad (1)$$

where  $c$  is the sound velocity and  $T_f$  is the ToF.

As previously mentioned, most of the conventional ultrasonic ranging systems uses a value as reference to establish the exact point when the signal reflected is detected by the receiver. As long as the echo amplitude exceeds the predefined threshold level, there will always be a reading, which occurs at the time  $t_0$  [11], as shown in Fig. 1.

Threshold detection does not require complex calculations and can be easily implemented, but the ToF estimation is not so reliable. The problem arises from



**Fig. 1.** Typical echo of the ultrasonic ranging system [11].

the fact that a signal will always have a rising time, so this condition added to other factors compromises the efficiency of this approach [12].

### 3 System Requirements

In this Section are introduced the fundamental concepts and techniques for the ultrasonic binaural system development, associating optimized approaches to enhance the system performance.

#### 3.1 ToF Estimation Methods

In Sect. 2 the major drawback involving the threshold detection technique is addressed. Thus, to handle with this situation some alternatives used in this work are presented in the following subsections.

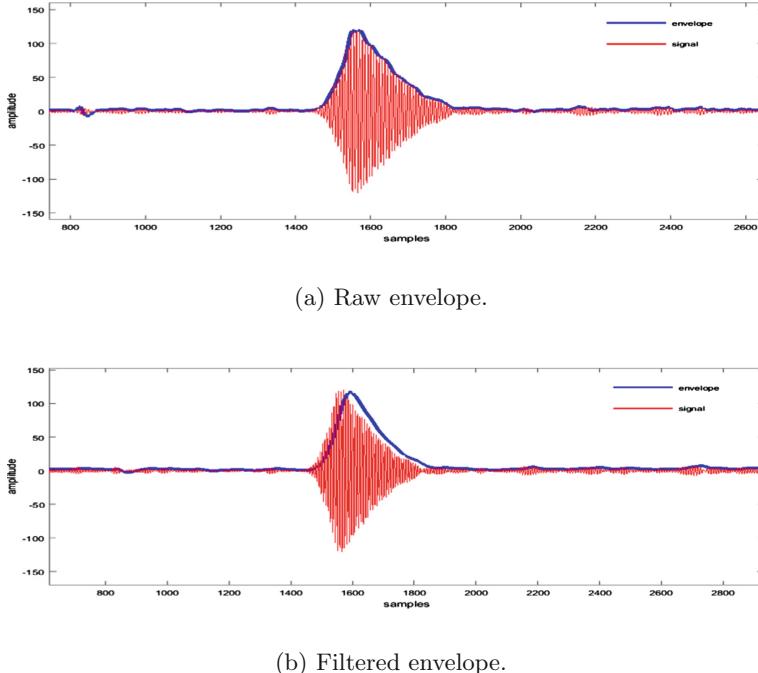
##### 3.1.1 Digital Envelope Detection

Envelope detection is accomplished using analog or digital resources. Analog methods works well if it is done carefully. Nevertheless, the best performing analog circuits are complex and delicate. In addition, according to [13] analog processing is not suitable to the envelope detection of digital data.

In the digital methods, approaches that extract the analytic signal magnitude are explored. For this purpose the Hilbert transform is commonly used to generate an analytic signal from a real signal, whose absolute value represents the envelope of the echo signal received [14]. This technique offers great precision, however it has a high computational consumption, involves complex calculus and in certain cases this method may be unstable [13]. For this reason a different approach is adopted.

In [15] a novel method, i.e. cubic spline interpolation method to extract the envelope from a signal is presented. Interpolation is one of the most important methods of numerical approximation, which allows to establish new data points from a set of known data points. To perform the envelope extraction, a windowing is applied to the signal to locate the peaks, and after that, the unknown

points between each peak are interpolated. Unlike the first one, the procedures to execute this method does not require complex operations. A second order Butterworth low pass filter is designed to soften the envelope. Figure 2a and b shows an example of a received echo signal and the envelope before and after the filter, respectively. The phase delay caused by the filter is negligible. Finally, from the extracted envelope the ToF is determined by the maximum value of the signal.



**Fig. 2.** Extracted envelope from the echo signal.

### 3.1.2 Cross-Correlation

Cross-correlation (CCR) is an important method to determine the time delay between two signals. When this method is applied, it produces a peak at the time delay and a reduction in the noise level. Usually the CCR takes a transmitted and received signal and yields a new signal in the time domain which the maximum value happens at the delay time [8]. Equation 2 demonstrates how the CCR can be calculated.

$$c(t) = \int_{-\infty}^{+\infty} s_T(t)s_R(t + \tau)d(\tau) \quad (2)$$

where  $s_T(t)$  is the transmitted signal and  $s_R(t + \tau)$  is the signal received and shifted in time.

The transmitted signal by the sensor generates eight cycle sonic burst at 40 kHz, i.e. it provides eight square pulses with period of 25  $\mu\text{s}$ . Then, in this case, applying CCR between the transmitted and received signal would not be appropriate. To implement this idea it would be necessary to detect the transmitted signal in a waveform similar to that of the received signal, as can be seen in Fig. 2. For this reason, the CCR is applied only to the received signals from sensor A and B (Fig. 6). Knowing the time lag it is possible to establish which of the receiving sensors is closest to the target.

### 3.2 Triangulation

Triangulation is a procedure to determine the position of an object or a target from known points using trigonometric techniques. In general, the number of dimensions of the estimated position of an object is directly related to the number of sensors, for instance, two sensors can indicate a location in 2D, while three sensors can indicate a 3D location and so on [16].

In geometry, when the length of the sides of any triangle is known a theorem to calculate its area can be applied. This approach was created by the mathematician Heron of Alexandria and has been used in many mathematical applications [17]. With the distance of the baseline and the distance calculated by each sensor known, there is the possibility to estimate the area of the triangle formed (Fig. 3) using Heron's formula, as shown in Eq. 3.

$$\Delta(abc) = \sqrt{p(p-a)(p-b)(p-c)}, \quad (3)$$

$$\text{where } p = \frac{a+b+c}{2} \quad (4)$$

whose  $a, b, c$  are the known sides. Alternatively, the area of a triangle can be computed using the Eq. 5.

$$\Delta(abc) = \frac{b \cdot h}{2} \quad (5)$$

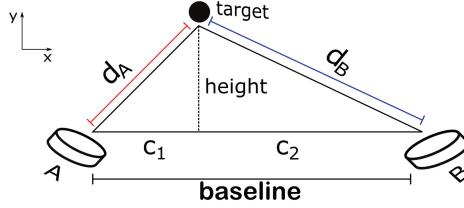
where  $b$  is the base and  $h$  is the height of the triangle. Reorganizing Eq. 5 is possible obtain Eq. 6.

$$h = \frac{\Delta(abc) \cdot 2}{b} \quad (6)$$

If the area provided by Eq. 3 is placed in Eq. 6 the y-coordinate of the target can be determined. By separating the original triangle into two right triangles and after that applying the Pythagorean Theorem it is possible to calculate the x-coordinate, according to Eqs. 7 and 8.

$$C_1 = \sqrt{d_A^2 - h^2} \quad (7)$$

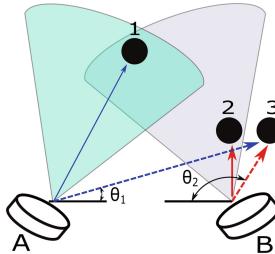
$$C_2 = \sqrt{d_B^2 - h^2} \quad (8)$$



**Fig. 3.** Triangulation created by the sensors.

where  $h$ ,  $C_1$ ,  $C_2$ ,  $d_A$  and  $d_B$  are shown in Fig. 3. If the Eq. 7 is used, the system reference for the coordinates on the x-axis will originate from sensor A, otherwise, from sensor B.

Valid calculations but nonsense physical can arise using the equations described above. Thus, in order to avoid incoherent measurements some restrictions are established. A typical error is shown in Fig. 4, to avoid it, it is observed that  $\theta_1$  and  $\theta_2$  angles are outside of the detection area.



**Fig. 4.** Typical triangulation error. Adapted from [16].

Figure 4 shows that sensor A detects the target 1 and sensor B detects the target 2. Nonetheless,  $\theta_1$  and  $\theta_2$  are not valid, which causes an error resulting in a false positioning, represented by the target 3.

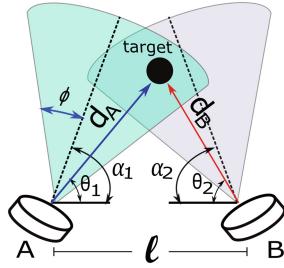
For a valid detection area Fig. 5 displays how  $\theta_1$  and  $\theta_2$  should be, indicating that  $\theta_1$  has to be equal  $\alpha_1 \pm \phi$ , where  $\alpha_n$  is the angular position of the sensor  $n$  corresponding to the x-axis and  $\phi$  is half the angle of the detection beam, which this value is particular to each sensor model. Then, the angular interval can be given by  $\cos(\alpha_1 - \phi) \leq \cos \theta_1 \leq \cos(\alpha_1 + \phi)$ . Similarly, to sensor B the angular limiting range is  $\cos(\alpha_2 - \phi) \leq \cos \theta_2 \leq \cos(\alpha_2 + \phi)$ , where

$$\cos \theta_1 = \frac{l^2 + d_A^2 - d_B^2}{2ld_A}, \quad (9)$$

and

$$\cos \theta_2 = \frac{l^2 + d_B^2 - d_A^2}{2ld_B} \quad (10)$$

$\alpha_1$ ,  $\alpha_2$  and  $\phi$  are constant, then, it is necessary to calculate only  $\cos \theta_1$  and  $\cos \theta_2$ . Measurements outside this range are discarded.



**Fig. 5.** Example of a valid detection. Adapted from [16].

## 4 System Architecture

The system consists in three low-cost ultrasonic sensors HY-SRF05 and a microcontroller (MCU) STM32F103C8T ARM Cortex-M3 with two built-in analog to digital converters. The ultrasonic burst is achieved by sensor C, and the raw echo signal received by sensors A and B is sent to the 12 bits A/D converters of the MCU with a sampling rate of 320 kHz.

Adopting the speed of sound as 340 m/s, it can be stated that the distance is covered 0.034 cm/ $\mu$ s. The mutual is equal to 29.412  $\mu$ s/cm, and considering a round trip corresponds to 58.824  $\mu$ s/cm. Using this value it is possible to simplify the determination of the distance and substitute the Eq. 1 by the Eq. 11.

$$d = \frac{T_f}{K}, \quad (11)$$

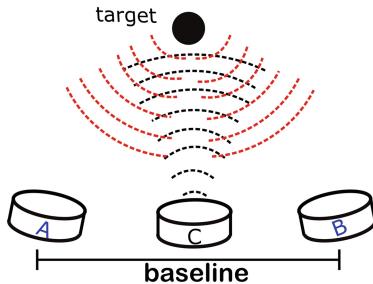
where  $K \approx 59$ .

The measurements from each sensor are stored in an array with four thousand positions, which if multiplied by the sampling period,  $T_s = 3.125 \mu$ s, corresponds to a sample space of 12500  $\mu$ s. Using the Eq. 11 we can estimate that the proposed system has a maximum range of about 212 cm depth, that is, along the y-axis. The x-axis is limited by the established baseline.

After obtaining the data, the samples are transmitted over RS-232 protocol to a computer to be processed by the MATLAB software. Finally, the calculation of the object coordinates is accomplished.

### 4.1 Binaural Sonar System

Binaural sonar system is composed of two receivers and one emitter, as shown in Fig. 6. This arrangement aims to collect more information and increase the



**Fig. 6.** The binaural sensor system.

accuracy of the measurements. Also, according to [6] a multi-sonar system is able to reduce significantly the problems faced by the use of a single sensor.

The designed system has three conventional ultrasonic sensors, which two acts as receivers and one as emitter. The region overlapped by the sensors, as seen in Fig. 7, is called detection area and any object contained in that space is able to be located by the system. The receivers, sensors A and B, are separated by a known baseline ( $l$ ) which through the application of trigonometric techniques allows to calculate the spatial coordinates of an object in a two-dimensional plane. Adjusting the angle and baseline, there is the possibility of changing the detection area when the two ultrasonic beam patterns overlap.

The best positioning is defined according to the baseline and the  $\alpha$  angle from experimental tests.

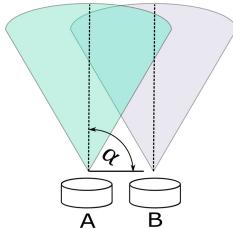
## 5 Proposed Work

The major objective of the proposed work is to improve the performance of the binaural system joining the methods presented in Subsect. 3.1, to define with high accuracy the x and y axis coordinates of one or more targets existing in the detection area through the triangulation.

As mentioned in Sect. 3.1.2, the CCR is performed only with the received signals. From the known time delay between the sensors A and B is possible to determine which of the sensors receives the echo first and consequently if it is closer or not to the object. In spite of obtaining this information, it is not enough to estimate the ToF, because it is necessary to have some reference of the origin, i.e. when the signal is transmitted.

In order to solve the issue previously quoted, the technique described in Sect. 3.1.1 is employed to find out the ToF of one of the received echoes. So, the digital envelope detection is applied to the sensor closest to the target, believing that accuracy may be better. Following, the ToF of the other sensor is defined based on the time delay provided by the CCR. For example, suppose the sensor A is the closest sensor to the target and the delay time given by CCR is 2 ms advanced from sensor B. Then, the ToF of the sensor B will be the ToF of

the sensor A subtracted 2 ms. If sensor B were the closest sensor, the opposite analogy would be used.



**Fig. 7.** Detection area formed by the ultrasonic beam patterns. Adapted from [16].

## 6 Results

The main purpose of the experimental tests is verify the accuracy of the measurements and analyze the general performance of the proposed system. The tests are divided into two stages, the first with only one object and the second with two objects present in the detection area. In the former, a small box is used and the receiver sensors are placed with  $\theta_{1,2} = 45^\circ$  and  $l = 51$  cm. The experiments are performed with the object positioned to the left, right and center of the detection area. About sixty samples of each possibility are obtained and the standard deviation ( $\sigma$ ), mean ( $(\bar{x}, \bar{y})$ ) and relative error of each one are calculated, as can be seen in Table 1.

**Table 1.** Mean, standard deviation and relative error of the first tests.

Location	$(x, y)$ real (cm)	$(\bar{x}, \bar{y})$ estimated (cm)	$\sigma_x$ (cm)	$\sigma_y$ (cm)	Relative error (%)
Left	(11.5, 91.8)	(11.14, 91.57)	1.41	0.36	0.30
Center	(25.6, 84.5)	(24.42, 80.45)	0.01	0.08	4.78
Right	(35.0, 86.4)	(35.98, 84.59)	1.01	0.86	1.39

Since the raw echo signal is used, there is the possibility to do a wide investigation about the measured environment. If a second object is present in the detection field, this information can be discovered through the existence of more than one crest in the waveform of the echo signal. In the second stage, the object described before is used together with a larger box, and the same procedures are applied. The positioning of both objects is arbitrary, the smaller box is closer to the sensors and the larger box further away. The standard deviation ( $\sigma$ ), mean ( $(\bar{x}, \bar{y})$ ) and relative error of each situation are presented in Table 2.

**Table 2.** Mean, standard deviation and relative error of the further tests.

Box	$(x, y)$ real (cm)	$(x, y)$ estimated (cm)	$\sigma_x$ (cm)	$\sigma_y$ (cm)	Relative error (%)
1	(35.5, 54.0)	(33.04, 53.52)	0.39	0.27	2.67
2	(19.0, 111.2)	(19.58, 110.29)	1.09	0.21	0.71

When the CCR is applied to the received echoes a signal with triangular shape is produced, resulting in an improper time delay. Probably this situation occurs because both signals has a significant direct current (DC) offset, then, to deal with this problem, the offset removal is performed through the normalization of the data, subtracting each sample by the arithmetic mean of all samples.

Particularly in the second stage, to use CCR when more than one object is in the scene, it is necessary to separate the echo signals received by the sensors according to the crests of the waveform, as can be seen in Fig. 9, and use the CCR in each of them with its corresponding crest. For instance, the CCR would have to be applied to the first part of the sensor A with the first part of the sensor B, and so on. Otherwise, if the entire signal is used, a single time delay is provided and consequently is not possible to distinguish which target that delay is referring to.

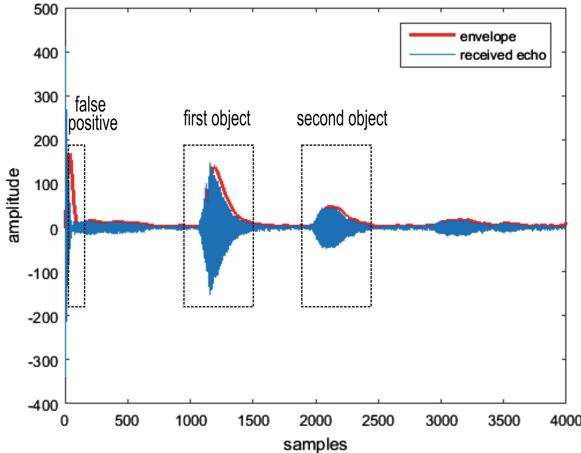
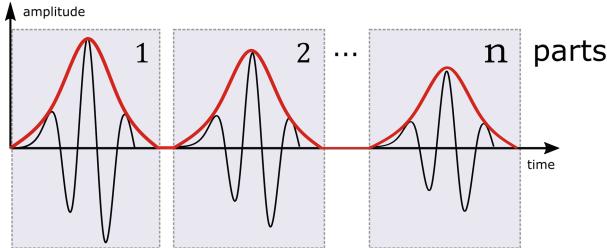
**Fig. 8.** Received echo signal with two objects in the detection area.

Figure 8 shows a frequent error in the receiver sensors. The object is more than 1 cm apart, however the system indicates that the object is at a smaller distance. The problem seems to occur because the echo signal interferes with a signal that travels directly from the transmitter to the receiver, without reflecting on the object. Thus, in an attempt to reduce incoherent data the first samples are ignored.



**Fig. 9.** Fragmentation of the received echo signal.

## 7 Conclusion and Future Work

A low cost binaural sensor system was described in this paper, that was able to detect the coordinates of one or more objects present in the detection area and provide a deeper analysis of the environment. Triangulation and signal processing techniques were used to improve the system performance. Experimental tests confirmed the accuracy of the measurements from the obtained results, which the mean system error for the first scenario with a single object, and for the second with two objects was 2.16% and 1.69% respectively.

As future work, it is intended to optimize the processing time with another MCU, classify the shape of the detected object and embed all to some mobile navigation device for testing and system validation. In general, the binaural sensor system using mathematical and signal processing techniques can be an attractive alternative to the conventional models of ultrasonic sensing, presenting good accuracy and repeatability of the measurements.

**Acknowledgement.** This work is financed by the ERDF—European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation—COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT—Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

## References

1. Lim, Z.S., Kwon, S.T., Joo, M.G.: Multi-object identification for mobile robot using ultrasonic sensors. *Int. J. Control Autom. Syst.* **10**(3), 589–593 (2012)
2. Walter, C., Schweinzer, H.: Locating of objects with discontinuities, boundaries and intersections using a compact ultrasonic 3D sensor. In: 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 591–600. IEEE (2014)
3. Bank, D.: A novel ultrasonic sensing system for autonomous mobile systems. *IEEE Sens. J.* **2**(6), 597–606 (2002)
4. Cong-Nguyen, H., Oh, S., Kim, N., Park, D., Han, S.: Real-time robust control of mobile robot using ultrasonic sensor. In: 2008 International Conference on Control, Automation and Systems, pp. 2585–2589. IEEE (2008)

5. Jawale, R.V., Kadam, M.V., Gaikawad, R.S., Kondaka, L.S.: Ultrasonic navigation based blind aid for the visually impaired. In: 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp. 923–928. IEEE (2017)
6. Kreczmer, B.: Objects localization and differentiation using ultrasonic sensors. In: Robot Localization and Map Building. IntechOpen (2010)
7. Kamiak, G., Schweinzer, H.: A 3D airborne ultrasound sensor for high-precision location data estimation and conjunction. In: 2008 IEEE Instrumentation and Measurement Technology Conference, pp. 842–847. IEEE (2008)
8. Jackson, J.C., Summan, R., Dobie, G.I., Whiteley, S.M., Pierce, S.G., Hayward, G.: Time-of-flight measurement techniques for airborne ultrasonic ranging. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **60**(2), 343–355 (2013)
9. Khyam, M.O., Ge, S.S., Li, X., Pickering, M.R.: Highly accurate time-of-flight measurement technique based on phase-correlation for ultrasonic ranging. *IEEE Sens. J.* **17**(2), 434–443 (2016)
10. Andria, G., Attivissimo, F., Giaquinto, N.: Digital signal processing techniques for accurate ultrasonic sensor measurement. *Measurement* **30**(2), 105–114 (2001)
11. Kuc, R., Siegel, M.W.: Physically based simulation model for acoustic sensor robot navigation. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-9**(6), 766–778 (1987)
12. Queirós, R., Martins, R.C., Girão, P.S., Serra, A.C.: A new method for high resolution ultrasonic ranging in air. In: Proceedings International Measurement Confederation, Rio de Janeiro (2006)
13. Fritsch, C., Ibanez, A., Parrilla, M.: A digital envelope detection filter for real-time operation. *IEEE Trans. Instrum. Measur.* **48**(6), 1287–1293 (1999)
14. Xu, B., Yu, L., Giurgiutiu, V.: Advanced methods for time-of-flight estimation with application to Lamb wave structural health monitoring. In: Proceedings International Workshop on SHM, pp. 1202–1209 (2009)
15. Jia, X.F., An, H.Q., Zhang, S.G.: Cubic spline interpolation method for the envelope tracking of middle and low frequency voltage flicker. In: Advanced Materials Research, pp. 704–709. Trans Tech Publications (2014)
16. Gearhart, C., Herold, A., Self, B., Birdsong, C., Slivovsky, L.: Use of ultrasonic sensors in the development of an Electronic Travel Aid. In: 2009 IEEE Sensors Applications Symposium, pp. 275–280. IEEE (2009)
17. Bényi, Á.: 87.47 a heron-type formula for the triangle. *Math. Gaz.* **87**(509), 324–326 (2003)



# Fully Coupled Six-DoF Nonlinear Suboptimal Control of a Quadrotor: Application to Variable-Pitch Rotor Design

Saeed Rafiee Nekoo<sup>(✉)</sup>, José Ángel Acosta, and Aníbal Ollero

GRVC Robotics Lab., Departamento de Ingeniería de Sistemas y Automática,  
Escuela Técnica Superior de Ingeniería, Universidad de Sevilla,  
41092 Seville, Spain  
saerafee@yahoo.com, {jaar, aollero}@us.es

**Abstract.** In this work, a fully coupled six degree-of-freedom (DoF) nonlinear suboptimal control of a variable-pitch quadrotor is studied using a state-dependent Riccati equation (SDRE) controller. The quadrotor control has been widely considered for attitude control; however, the position control is an uncontrollable problem with the common design of the SDRE. Due to the under-actuated nature of a quadrotor, the state-dependent coefficient (SDC) parameterization of state-space representation of a nonlinear system leads to an uncontrollable SDC pair. The control law is divided into two sections of position and attitude control. The position control provides the main thrust. A virtual constraint is regarded to provide stabilization for the quadrotor in attitude control. Two methods were designed for selection of a state vector or in other words, selection of feedback. The first one uses the position and orientation and their derivatives in global coordinate. The second one uses position and orientation in global and their velocities in local coordinate. The dynamics of a variable-pitch propeller quadrotor was imported to the problem and compared with a fixed-pitch propeller system. The simulation of the systems shows that the SDRE is capable of controlling the system with both fixed- and variable-pitch rotor dynamics.

**Keywords:** Quadrotor · Nonlinear optimal control · SDRE · Virtual constraint

## 1 Introduction

The state-dependent Riccati equation was applied to quadrotors by Voos for the first time in 2006 [1]. The control problem was limited to attitude control of a system in a sub-control unit, and control of the velocity. So, the regulation was not possible since the quadrotor was moving in space with a constant velocity. The combination of the SDRE controller with the neural network was provided to control the velocity vector towards zero [2]. Navabi and Mirzaei presented  $\theta$ -D based nonlinear tracking control of a quadcopter using the state-dependent Riccati equation [3]. Due to under-actuation, only Z direction was controlled. Babaie and Ehyaei proposed a robust SDRE, based on sliding mode design [4]. The position control was addressed through Lyapunov criteria for stabilization of sliding surfaces. Chipofya and Lee presented the position control of

a quadrotor via SDRE controller employing a Kalman filter for estimation [5]. In the design, the planar motion of the quadrotor was not included in the simulation. This problem was visible in most control problems of quadrotors using the SDRE.

At the beginning of the quadrotor control research, most of the cases used fixed-pitch propeller systems to simplify the design, increase the stability of the quadcopter, and employ common methodology in the literature. Fixed-pitch propellers limited the inputs to angular velocities of rotors. The focus of this work is to explore the variable-pitch propellers quadrotor control within the framework of the SDRE. The variable-pitch design decreases stability; hence, a more agile maneuver could be expected. Inverted flight and flip during the motion were also highlighted in the literature [6]. The use of variable-pitch blades in quadrotors was reported by Bristeau et al. [7], and later on by Cutler [6]. Fresk and Nikolakopoulos presented experimental model derivation and control of a variable pitch propeller for a quadrotor [8]. Sheng and Sun focused on the energy consumption of the variable pitch quadrotor control [9]. Panizza et al. presented data-driven attitude control of the system [10]. Chipade et al. presented control of variable pitch quadrotor for payload delivery with focus on mechanism design [11]. Rotor dynamic in variable pitch imposed a nonlinear algebraic equation which required control allocation [12], or other techniques to find a solution. Turning the nonlinear relation to a first-order differential equation was regarded to solve that issue [12]. An extra differential equation in addition to complex nonlinear dynamics of a multirotor enhances the complexity.

The main contribution of this current research is to apply the state-dependent Riccati equation controller for a variable-pitch quadrotor, considering fully coupled six-DoF nonlinear dynamics. The main step of the SDRE control design is state-dependent coefficient parameterization of a nonlinear vector. The SDC matrices must be controllable and observable to guarantee a solution to the related SDRE. The equation of motion of a quadrotor in SDC form does not release a controllable pair of SDC matrices since there is only one actuator for translation control. To overcome this issue, the design of the translation control has been done assuming that three virtual inputs are available for  $XYZ$  directions. Then, virtual constraints were designed to find the relations between thrust (actuator in  $Z$ ) and desired orientation angles. The three virtual inputs have been transformed into one input (thrust) capable of controlling position vector. The main contribution of this research is fully coupled six-DoF control of a variable-pitch quadrotor using state-dependent Riccati equation introducing virtual constraint; for both fixed- and variable pitch rotor design.

Two sets of states were chosen to generate state-space representations for the system. The first one uses the position and orientation of the quadrotor in global coordinates and their velocities in local one. This point of view requires the assumption of small deviations in rotational movement which is common in regulation and tracking of a multirotor system. The second one uses all the position and orientation with their velocities in global coordinate. Both methods were simulated and analyzed to assess the effect of different feedback selection.

## 2 The State-Dependent Riccati Equation

Consider a nonlinear system

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t), \quad (1)$$

where  $\mathbf{x}(t) \in \mathbb{R}^n$  is a state vector and  $\mathbf{u}(t) \in \mathbb{R}^m$  is an input vector.  $\mathbf{A}(\mathbf{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  and  $\mathbf{B}(\mathbf{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$  are state-dependent coefficient parameterization of a nonlinear system, consists of piecewise-continuous vector-valued functions that satisfy Lipschitz condition.

The intention of optimal control is to minimize the cost functional integral [13]:

$$J(\cdot) = \frac{1}{2} \int_0^\infty \{ \mathbf{x}^T(t) \mathbf{Q}(\mathbf{x}(t)) \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{R}(\mathbf{x}(t)) \mathbf{u}(t) \} dt,$$

where  $\mathbf{Q}(\mathbf{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  penalizes the states (symmetric positive semi-definite) and  $\mathbf{R}(\mathbf{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$  penalizes the inputs (symmetric positive definite).

*Controllability condition:* The pair of  $\{\mathbf{A}(\mathbf{x}(t)), \mathbf{B}(\mathbf{x}(t))\}$  is a completely controllable parameterization of a nonlinear system (1) with its condition [14].

*Observability condition:* The pair of  $\{\mathbf{A}(\mathbf{x}(t)), \mathbf{Q}^{1/2}(\mathbf{x}(t))\}$  is a completely observable parameterization of a nonlinear system (1) with its condition [14].

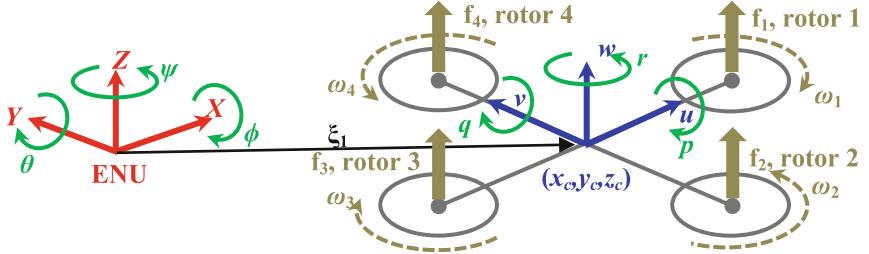
A standard form of the SDRE control law is  $\mathbf{u} = -\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^T(\mathbf{x})\mathbf{K}(\mathbf{x})(\mathbf{x})$ , where the symmetric positive definite suboptimal gain  $[\mathbf{K}(\mathbf{x}(t))]_{2n \times 2n}$ , is a solution to the state-dependent Riccati equation [14]:

$$\mathbf{A}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \mathbf{K}(\mathbf{x})\mathbf{A}(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \mathbf{Q}(\mathbf{x}) = \mathbf{0}.$$

## 3 State-Space Representation

### 3.1 Global Position and Local Velocity

The absolute linear position vector of a quadrotor in the inertial frame is  $\xi_1(t) = [x_c(t), y_c(t), z_c(t)]^T(m)$  in which subscript “c” stands for center-of-mass, the three Euler angles in an inertial frame, roll-pitch-yaw, are set in a vector  $\xi_2(t) = [\phi(t), \theta(t), \psi(t)]^T(\text{rad})$ , linear velocity vector in body frame is  $\mathbf{v}_1(t) = [u(t), v(t), w(t)]^T(\text{m/s})$ , and angular velocity vector in body frame is  $\mathbf{v}_2(t) = [p(t), q(t), r(t)]^T(\text{rad/s})$ , Fig. 1.



**Fig. 1.** Fixed and moving reference frame.

The following kinematics relations are held between inertial and body frame [15]:

$$\dot{\xi}_1 = \mathbf{R}_{ZYX}(\xi_2)\mathbf{v}_1, \quad (2)$$

$$\dot{\xi}_2 = \mathbf{T}(\xi_2)\mathbf{v}_2, \quad (3)$$

where  $\mathbf{R}_{ZYX}(\xi_2)$  is found based on the multiplication of the three rotation matrices around three main axes:

$$\mathbf{R}_{ZYX}(\xi_2) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix},$$

$$\mathbf{T}(\xi_2) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix},$$

in which e.g.  $c_\psi = \cos(\psi(t))$  and  $t_\theta = \tan(\theta(t))$ . There is one input force (thrust)  $T_B(t)$ (N), acting in direction of  $w$  on CoM of the quadrotor (local moving coordinate), and an input torque vector  $\tau_B(t) = [\tau_\phi(t) \quad \tau_\theta(t) \quad \tau_\psi(t)]^T$  (N.m), acting against three Euler angles  $\{\phi(t), \theta(t), \psi(t)\}$ .  $T_B(t)$  is defined in body frame and  $\tau_B(t)$  is set on the inertial frame.

In this point of view, the position and attitude of the quadrotor are considered and measured in the inertial frame, and the linear and angular velocity of that are measured in body frame. This consideration is valid through a simplification  $\dot{\xi}_2(t) \simeq \mathbf{v}_2(t)$  that holds for small angular motions [16], and  $\dot{\xi}_1(t) \simeq \mathbf{v}_1(t)$ . This assumption leads to two more approximations  $\dot{\mathbf{v}}_1(t) \simeq \ddot{\xi}_1(t)$  and  $\dot{\mathbf{v}}_2(t) \simeq \ddot{\xi}_2(t)$ . Based on that assumption, the state vector of the system is assembled as

$$\mathbf{x}(t) = [\xi_1^T(t), \xi_2^T(t), \mathbf{v}_1^T(t), \mathbf{v}_2^T(t)]^T = [x_c, y_c, z_c, \phi, \theta, \psi, u, v, w, p, q, r]^T. \quad (4)$$

Considering state-vector (4), the upper half of the state-space representation of the system uses kinematics relations (2) and (3); and the lower part of that extracts  $\ddot{\xi}_1$  and  $\ddot{\xi}_2$  from equation of motion:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \\ \dot{\mathbf{v}}_1(t) \\ \dot{\mathbf{v}}_2(t) \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{ZYX}(\xi_2)\mathbf{v}_1 \\ \mathbf{T}(\xi_2)\mathbf{v}_2 \\ 1/m\mathbf{I}_{3\times 3}[\mathbf{R}_{ZYX,3}(\xi_2)\mathbf{T}_B - mge_3 - \mathbf{D}\dot{\xi}_1] \\ \mathbf{J}^{-1}(\xi_2)[\tau_B - \mathbf{C}(\xi_2, \dot{\xi}_2)\dot{\xi}_2] \end{bmatrix}, \quad (5)$$

where  $\mathbf{R}_{ZYX,3}(\xi_2)$  is the third column of  $\mathbf{R}_{ZYX}(\xi_2)$  and  $\mathbf{e}_3 = [0, 0, 1]^T$ , and  $\mathbf{J}(\xi_2) = \mathbf{W}^T(\xi_2)\mathbf{IW}(\xi_2)$ ,  $\mathbf{W}(\xi_2)$  is the inverse of  $\mathbf{T}(\xi_2)$  and vector  $[\mathbf{C}(\xi_2, \dot{\xi}_2)\dot{\xi}_2]$  includes Coriolis and centrifugal terms. The aerodynamics effect is incorporated into the dynamics of the system thorough  $\mathbf{D} = \text{diag}(D_x, D_y, D_z)$  (kg/s) matrix [17]. That is the result of drag force caused by air resistance in which  $D_x, D_y, D_z$  are drag coefficients in  $(X, Y, Z)$  inertial frame.

### 3.2 Global Position and Global Velocity

In this section, the position and attitude of the quadrotor, and the linear and angular velocity of that are considered and measured in the inertial frame. This consideration does not need any assumption or approximation. So, the state vector of the system is assembled as

$$\mathbf{x}(t) = [\xi_1^T(t), \xi_2^T(t), \dot{\xi}_1^T(t), \dot{\xi}_2^T(t)]^T = [x_c, y_c, z_c, \phi, \theta, \psi, \dot{x}_c, \dot{y}_c, \dot{z}_c, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T. \quad (6)$$

Considering state-vector (6), the modified representation of (5) is found:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \\ \dot{\mathbf{v}}_1(t) \\ \dot{\mathbf{v}}_2(t) \end{bmatrix} = \begin{bmatrix} \dot{\xi}_1(t) \\ \dot{\xi}_2(t) \\ 1/m\mathbf{I}_{3\times 3}[\mathbf{R}_{ZYX,3}(\xi_2)\mathbf{T}_B - mge_3 - \mathbf{D}\dot{\xi}_1] \\ \mathbf{J}^{-1}(\xi_2)[\tau_B - \mathbf{C}(\xi_2, \dot{\xi}_2)\dot{\xi}_2] \end{bmatrix}. \quad (7)$$

The state-space representation in (7) is based on the measurement of all the states in the inertial frame; however, the representation in (5) computes the half of the states in body frame. So, the choice of state-space representations might be restricted to the sensor selection or practical limitation. One should note that “global position and local velocity (GPLV)” representation has an approximation though “global position and global velocity (GPGV)” form was generated without any simplification or approximation.

## 4 Variable-Pitch Rotor Dynamics

Variable-pitch propeller quadrotors provide the option of additional inputs to the problem; hence, the angle of a blade could be considered as input. More maneuverability, upright or inverted flight, detach/attach equipment in out of reach positions and negative thrust deceleration could be listed as the advantages of the variable-pitch

design. The challenges are also complexity in rotor mechanism design, control approach and reducing the flight stability of the quadrotor.

The blade is defined in terms of thrust coefficient as [12]:  $\alpha_i(t) = \frac{6C_{T_i}(t)}{\sigma C_{l_x}} + \frac{3}{2} \sqrt{\frac{C_{T_i}(t)}{2}}$ , where  $\alpha_i$  and  $C_{T_i}$  are blade angle and thrust coefficient of  $i$ -th rotor with respect,  $C_{l_x}$  is airfoil lift curve slope,  $\sigma = \frac{N_b c}{\pi R^2}$  in which  $N_b$  is number of the blades in each rotor,  $c(\text{m})$  is rotor's chord length and  $R(\text{m})$  is the radius of the rotor. Based on the structure of the quadrotor (plus shape), the thrust coefficient is related to force/moment inputs of a quadrotor [18]:

$$T_B(t) = \gamma K \sum_{i=1}^4 C_{T_i}(t), \tau_\phi(t) = \gamma l K (C_{T_4}(t) - C_{T_2}(t)), \tau_\theta(t) = \gamma l K (C_{T_3}(t) - C_{T_1}(t)), \\ \tau_\psi(t) = \frac{KR}{\sqrt{2}} \left( -|C_{T_1}(t)|^{3/2} + |C_{T_2}(t)|^{3/2} - |C_{T_3}(t)|^{3/2} + |C_{T_4}(t)|^{3/2} \right),$$

where  $\gamma$  is 1 for normal flight and -1 for inverted flight and  $K = \rho \pi R^4 \omega_{ss}^2$  in which  $\rho (\text{kg/m}^3)$  is air density and  $\omega_{ss} (\text{rad/s})$  is a constant angular velocity of the rotors;  $\omega_{ss}$  is considered constant in variable-pitch flight mode. Considering variable  $\omega$  in  $K$  is also possible and increases the flexibility and complexity of the system; in this work, constant angular velocity is chosen for the variable-pitch system.

## 5 The State-Dependent Coefficient Parameterization

The transformation of the state-space representation of the dynamics into apparent (extended) linearization is called state-dependent coefficient parameterization. Since two representations were given for the dynamics of the quadrotor, two sets of SDC are generated. With regard to the state-space Eq. (5), the first point of view, GPLV, provides:

$$\mathbf{A}_t(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{R}_{ZYX}(\xi_2) \\ \mathbf{0}_{3 \times 3} & -1/m \mathbf{I}_{3 \times 3} \mathbf{D} \end{bmatrix}, \quad \mathbf{B}_t = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ 1/m \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad (8)$$

$$\mathbf{A}_o(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{T}(\xi_2) \\ \mathbf{0}_{3 \times 3} & -\mathbf{J}^{-1}(\xi_2) \mathbf{C}(\xi_2, \dot{\xi}_2) \end{bmatrix}, \quad \mathbf{B}_o(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{J}^{-1}(\xi_2) \end{bmatrix}, \quad (9)$$

where index "t" stands for translation and "o" for orientation. With regard to the state-space Eq. (7), the second point of view, GPGV, provides:

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -1/m \mathbf{I}_{3 \times 3} \mathbf{D} \end{bmatrix}, \quad \mathbf{A}_o(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\mathbf{J}^{-1}(\xi_2) \mathbf{C}(\xi_2, \dot{\xi}_2) \end{bmatrix},$$

moreover,  $\mathbf{B}_t$  and  $\mathbf{B}_o(\mathbf{x})$  are similar to (8) and (9).

**Controllability:** It can be easily checked that an arbitrary pair of  $\mathbf{A} = \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{I}_{n \times n} \\ \mathbf{0}_{n \times n} & \mathbf{0}_{n \times n} \end{bmatrix}$  and  $\mathbf{B} = \begin{bmatrix} \mathbf{0}_{n \times n} \\ \mathbf{I}_{n \times n} \end{bmatrix}$  is controllable and a pair of  $\{\mathbf{A}, \mathbf{Q}^{1/2}\}$ , with  $\mathbf{Q} = \mathbf{I}_{n \times n}$  is observable.

Considering that the diagonal elements of  $\mathbf{R}_{ZYX}(\xi_2)$ ,  $\mathbf{T}(\xi_2)$  and  $\mathbf{J}^{-1}(\xi_2)$  possess multiplication of “cosine” functions, they hardly meet zero. So, the controllability and observability of the proposed SDC parameterization are guaranteed, except for  $(\phi(t), \theta(t), \psi(t)) = (2k - 1)\pi/2, k \in \mathbb{Z}$ . The orientation of  $\phi(t), \theta(t), \psi(t)$  is not supposed to reach  $\pi/2$  due to stability of the quadrotor for normal flight; in case of inverted flight, GPGV must be used.

**Note:** It should be noted that  $\mathbf{B}_t$  in (8) should have been a  $3 \times 1$  vector since the thrust input is a scalar value. However, the SDC design based on  $[\mathbf{B}_t]_{3 \times 1}$ , will result in an uncontrollable system. So, the design of (8) was done assuming a fully-actuated system. In Sect. 6, the necessary modification to incorporate 3 virtual inputs to one actual thrust  $T_B$  will be established.

## 6 Virtual Constraint Under-Actuation Compensation

A quadrotor is an under-actuated system, possessing six-DoF motion in space and four input actuators. One could divide the six-DoF system into two subsystems, three translational motion ( $x_c, y_c, z_c$ ), and three rotational one ( $\phi, \theta, \psi$ ). The contribution of the input  $\tau_B(t)$  in the lower sets of state-space Eq. (5) or (7) provides a stable controller. However, the contribution of input thrust  $T_B(t)$  through  $\mathbf{R}_{ZYX,3}(\xi_2)$  in the upper sets of the system (5) or (7) only controls  $z_c$  direction. In order to control  $x_c, y_c$  directions, additional constraints should be provided to link the motions in a meaningful manner. The error of the state vector is defined as  $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}_{des}$ . A stable control law is proposed (assuming the system is not under-actuated):

$$\mathbf{U} = -\mathbf{R}_t^{-1}(\mathbf{x})\mathbf{B}_t^T(\mathbf{x})\mathbf{K}_t(\mathbf{x})\mathbf{e}_t, \quad (10)$$

where  $\mathbf{R}_t(\mathbf{x}) : \mathbb{R}^6 \rightarrow \mathbb{R}^{3 \times 3}$  is the weighting matrix for inputs,  $\mathbf{e}_t = \begin{bmatrix} \xi_1^T - \xi_{1,des}^T \\ v_1^T - \dot{\xi}_{1,des}^T \end{bmatrix}^T$  (GPLV) or  $\mathbf{e}_t = \begin{bmatrix} \xi_1^T - \xi_{1,des}^T; \dot{\xi}_1^T - \dot{\xi}_{1,des}^T \end{bmatrix}^T$  (GPGV) is an error vector including translational states,  $\mathbf{K}_t(\mathbf{x}) : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$  is the symmetric positive definite solution to the SDRE (dedicated for translational control):

$$\mathbf{A}_t^T(\mathbf{x})\mathbf{K}_t(\mathbf{x}) + \mathbf{K}_t(\mathbf{x})\mathbf{A}_t(\mathbf{x}) - \mathbf{K}_t(\mathbf{x})\mathbf{B}_t(\mathbf{x})\mathbf{R}_t^{-1}(\mathbf{x})\mathbf{B}_t^T(\mathbf{x})\mathbf{K}_t(\mathbf{x}) + \mathbf{Q}_t(\mathbf{x}) = \mathbf{0}.$$

Replacing  $\ddot{\xi}_1$  from the equation of motion with  $\mathbf{U}$  results in [15]:

$$\mathbf{U} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \\ c_\phi c_\theta \end{bmatrix} \frac{T_B}{m}. \quad (11)$$

Changing (11) to  $[U_1 \ U_2 \ U_3 + g]^T = \mathbf{R}_{ZYX}(\xi_2)[0 \ 0 \ T_B/m]^T$ , and multiplying  $\mathbf{R}_{ZYX}^T(\xi_2)$  from left side provides ( $\mathbf{R}_{ZYX}(\xi_2)$  is orthogonal):

$$\mathbf{R}_{ZYX}^T(\xi_2)[U_1 \ U_2 \ U_3 + g]^T = [0 \ 0 \ T_B/m]^T. \quad (12)$$

From Eq. (12), two relations could be found as constraints for determining desired values for  $\theta$  and  $\phi$  [15]:

$$\theta_{\text{des}}(t) = \tan^{-1} \left( \frac{U_1 \cos \psi_{\text{des}} + U_2 \sin \psi_{\text{des}}}{U_3 + g} \right), \quad (13)$$

$$\phi_{\text{des}}(t) = \sin^{-1} \left( \frac{U_1 \sin \psi_{\text{des}} - U_2 \cos \psi_{\text{des}}}{\sqrt{U_1^2 + U_2^2 + (U_3 + g)^2}} \right). \quad (14)$$

Equations (13) and (14) are found based on cascade design [15]. So, the desired vector  $\xi_{2,\text{des}}(t)$ , is defined as

$$\xi_{2,\text{des}}(t) = [\phi_{\text{des}}(t) \ \theta_{\text{des}}(t) \ \psi_{\text{des}}(t)]^T, \quad (15)$$

where desired  $\psi_{\text{des}}(t)$ , in (13)–(15), could be independently set. Consequently, the problem of under-actuation is solved and the thrust is in the form of:

$$\begin{aligned} T_B(t) &= m \left\{ [\mathbf{R}_{ZYX,3}(\xi_2)]_1 U_1 + [\mathbf{R}_{ZYX,3}(\xi_2)]_2 U_2 + [\mathbf{R}_{ZYX,3}(\xi_2)]_3 (U_3 + g) \right\} \\ &= m [(c_\phi s_\theta c_\psi + s_\phi s_\psi) U_1 + (c_\phi s_\theta s_\psi - s_\phi c_\psi) U_2 + c_\phi c_\theta (U_3 + g)]. \end{aligned}$$

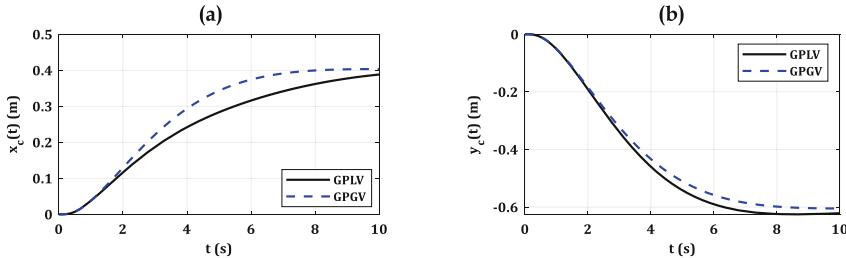
The design of the rotational control of the quadrotor is straightforward (similar to (10)):  $\tau_B = -\mathbf{R}_o^{-1}(\mathbf{x})\mathbf{B}_o^T(\mathbf{x})\mathbf{K}_o(\mathbf{x})\mathbf{e}_o$ , where  $\mathbf{R}_o(\mathbf{x}) : \mathbb{R}^6 \rightarrow \mathbb{R}^{3 \times 3}$  is the weighting matrix for inputs,  $\mathbf{e}_o = [\xi_2^T - \dot{\xi}_{2,\text{des}}^T, \mathbf{v}_2^T - \dot{\xi}_{2,\text{des}}^T]^T$  (GPLV) or  $\mathbf{e}_o = [\xi_2^T - \dot{\xi}_{2,\text{des}}^T, \dot{\xi}_2^T - \ddot{\xi}_{2,\text{des}}^T]^T$  (GPGV) is error vector including rotational states,  $\mathbf{K}_o(\mathbf{x}) : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$  is the symmetric positive definite solution to the SDRE (dedicated for rotational control):

$$\mathbf{A}_o^T(\mathbf{x})\mathbf{K}_o(\mathbf{x}) + \mathbf{K}_o(\mathbf{x})\mathbf{A}_o(\mathbf{x}) - \mathbf{K}_o(\mathbf{x})\mathbf{B}_o(\mathbf{x})\mathbf{R}_o^{-1}(\mathbf{x})\mathbf{B}_o^T(\mathbf{x})\mathbf{K}_o(\mathbf{x}) + \mathbf{Q}_o(\mathbf{x}) = \mathbf{0}.$$

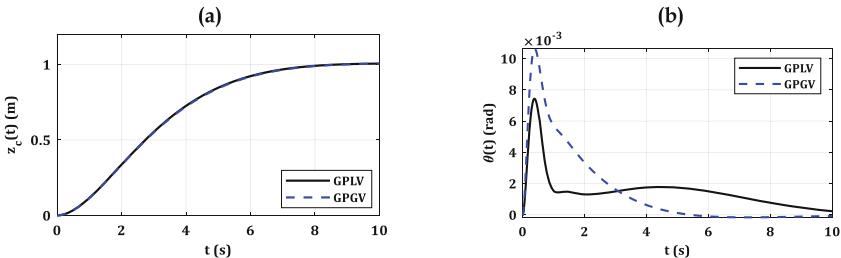
## 7 Simulations

In this section, both cases of GPLV and GPGV are simulated and compared. The parameters of the quadrotor are based on the model in Ref. [17]. A regulation case study is regarded to analyze the modeling and the controllers. The initial condition of the system was set as equilibrium point and the desired position was  $\mathbf{x}_{\text{des}}(t_f) = [0.4, -0.6, 1, \phi_{\text{des}}(t), \theta_{\text{des}}(t), 0.2, \mathbf{0}_{1 \times 6}]^T$ . It should be noted that  $\phi_{\text{des}}(t), \theta_{\text{des}}(t)$  are

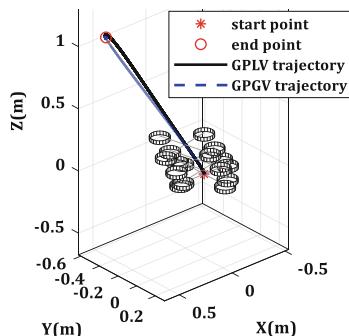
defined by (13) and (14). The simulation was done in 10 s and the weighting matrices were selected as  $\mathbf{R}_o = \mathbf{R}_t = 10 \times \mathbf{I}_{3 \times 3}$ ,  $\mathbf{Q}_o = \text{diag}(1, 1, 1, 0, 0, 0)$  and  $\mathbf{Q}_t = \mathbf{I}_{6 \times 6}$ . More details on weighting matrix selection could be reviewed in Ref. [13]. The position of the quadrotor in Cartesian coordinate is presented in Figs. 2 and 3. Trajectories of the systems are presented in Fig. 4. The thrust and related moment of roll are illustrated in Fig. 5. The error of the GPLV was found 43.1 mm and the one for GPGV 19.3 mm. The change in the weighting matrix, variable  $a$  in  $\mathbf{Q}_t = a \times \mathbf{I}_{6 \times 6}$ , increase or decrease the error of the system. A range of constant matrices has been applied to study the error, see Table 1.



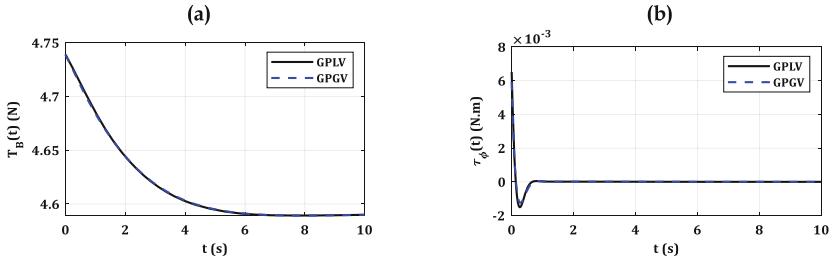
**Fig. 2.** Position of the quadrotor in X (a), and Y direction (b).



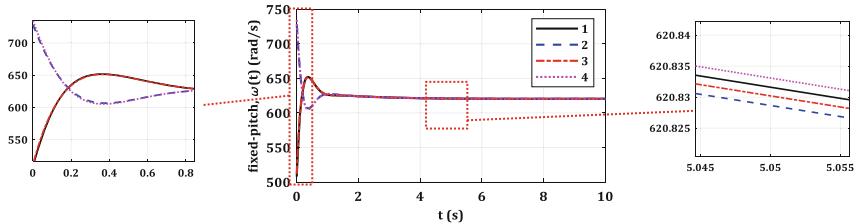
**Fig. 3.** Position of the quadrotor in Z direction (a), and pitch angle of the system (b).



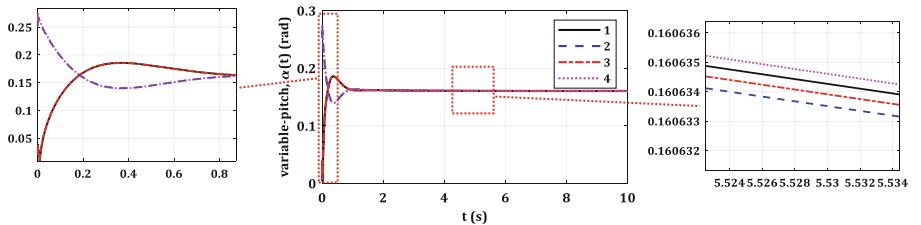
**Fig. 4.** Trajectories and configuration of the quadrotor.



**Fig. 5.** The thrust of the quadrotor (a), and moment input for roll (b).



**Fig. 6.** Angular velocities of the rotors.



**Fig. 7.** Angles of the blades in variable-pitch rotor design.

**Table 1.** Change in the weighting matrix versus error of the quadrotor.

$a$	Error GPLV (mm)	Error GPGV (mm)
0.01	767.3236	765.4663
0.1	272.9474	264.6261
1	25.0975	7.8013
10	1.3296	0.1688
100	0.0624	0.0254

Angular velocities of the rotors are presented in Fig. 6 and the angle of the blades in Fig. 7. The trajectories were almost similar in regulation in Fig. 4 though more accuracy was obtained by GPGV. The steady-state thrust reached 4.59 (N) as it was

expected. The steady-state value of rotors' angular velocities were set to 620.5 (rad/s). The increase in weighting matrix improves the results and accuracy nevertheless the GPGV outweighs the other feedback selection. The reason for similarities in angular velocities of the rotors, Fig. 6, or blade angles, Fig. 7, is rooted in the virtual constraint design, Eqs. (14) and (15). Based on the virtual constraint design, the quadrotor yaw, roll, and pitch angles are kept at minimum; so, the differences between the rotor angles and rotor velocities are small.

The solution to the quadrotor control results in total thrust  $T_B$  and input moment vector  $\tau_B$  which define the control input  $\mathbf{u} = [T_B, \tau_B^T]^T$ , presented in Fig. 5. The relation between  $\mathbf{u}$  and thrusts could be done by fixed- or variable-pitch design. Figure 6 shows the choice of fixed-pitch rotors and Fig. 7 presents the variable-pitch ones.

## 8 Conclusions

This work presented nonlinear fully coupled six-DoF control of a variable-pitch quadrotor using the state-dependent Riccati equation. The dynamics equation of the quadrotor was considered without any simplification and transformed to state-space representation in two schemes: GPLV and GPGV. Both of them successfully simulated on a system and analyzed. For the initial guess of the weighting matrix of states, the error of the GPLV was found 43.1 mm and the one for GPGV 19.3 mm. The decrease in the weighting matrix increased the error and the difference between the results of GPLV and GPGV. Increase in weighting matrix  $\mathbf{Q}$ , decreased regulation error and also reduced the difference between two feedback selections. In all simulations and comparisons, the GPGV gained better accuracy. The reason is the assumption of  $\dot{\xi}_2(t) \simeq v_2(t)$  in state-space representation of GPLV. The transformation of the force/moment of the quadrotor to rotors were done using fixed- and variable-pitch design. The simulation showed success and suggested the application of the SDRE in practical implementations.

**Acknowledgements.** This work is supported by the HYFLIER project (HYbrid FLying-rolling with-snake-aRm robot for contact inspection) funded by the European Commission H2020 Programme under grant agreement ID: 779411 (<https://cordis.europa.eu/project/rcn/213049>); and ARM-EXTEND funded by the Spanish RD scheme (DPI2017-89790-R).

## References

1. Voos, H.: Nonlinear state-dependent Riccati equation control of a quadrotor UAV. In: IEEE Computer Aided Control System Design, IEEE International Conference on Control Applications, IEEE International Symposium on Intelligent Control, pp. 2547–2552. IEEE (2006)
2. Voos, H.: Nonlinear and neural network-based control of a small four-rotor aerial robot. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1–6. IEEE (2007)

3. Navabi, M., Mirzaei, H.:  $\theta$ -D based nonlinear tracking control of quadcopter. In: 4th International Conference on Robotics and Mechatronics, pp. 331–336. IEEE (2016)
4. Babaei, R., Ehyaei, A.F.: Robust optimal motion planning approach to cooperative grasping and transporting using multiple UAVs based on SDRE. *Trans. Inst. Measur. Control* **39**, 1391–1408 (2017)
5. Chipofya, M., Lee, D.J.: Position and altitude control of a quadcopter using state-dependent Riccati equation (SDRE) control. In: 17th International Conference on Control, Automation and Systems, pp. 1242–1244. IEEE (2017)
6. Cutler, M., Ure, N.-K., Michini, B., How, J.: Comparison of fixed and variable pitch actuators for agile quadrotors. In: AIAA Guidance, Navigation, and Control Conference, pp. 6406–6423 (2011)
7. Bristeau, P.-J., Martin, P., Salaün, E., Petit, N.: The role of propeller aerodynamics in the model of a quadrotor UAV. In: European Control Conference, pp. 683–688. IEEE (2009)
8. Fresk, E., Nikolakopoulos, G.: Experimental model derivation and control of a variable pitch propeller equipped quadrotor. In: IEEE Conference on Control Applications, pp. 723–729. IEEE (2014)
9. Sheng, S., Sun, C.: Control and optimization of a variable-pitch quadrotor with minimum power consumption. *Energies* **9**, 232–250 (2016)
10. Panizza, P., Invernizzi, D., Riccardi, F., Formentin, S., Lovera, M.: Data-driven attitude control law design for a variable-pitch quadrotor. In: American Control Conference, pp. 4434–4439. IEEE (2016)
11. Chipade, V.S., Kothari, M., Chaudhari, R.R.: Systematic design methodology for development and flight testing of a variable pitch quadrotor biplane VTOL UAV for payload delivery. *Mechatronics* **55**, 94–114 (2018)
12. Bhargavapuri, M., Sahoo, S.R., Kothari, M.: Robust nonlinear control of a variable-pitch quadrotor with the flip maneuver. *Control Eng. Pract.* **87**, 26–42 (2019)
13. Korayem, M.H., Nekoo, S.R.: Finite-time state-dependent Riccati equation for time-varying nonaffine systems: rigid and flexible joint manipulator control. *ISA Trans.* **54**, 125–144 (2015)
14. Cimen, T.: Survey of state-dependent Riccati equation in nonlinear optimal feedback control synthesis. *J. Guid. Control Dyn.* **35**, 1025–1047 (2012)
15. Zuo, Z.: Trajectory tracking control design with command-filtered compensation for a quadrotor. *IET Control Theory Appl.* **4**, 2343–2355 (2010)
16. Das, A., Subbarao, K., Lewis, F.L.: Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET Control Theory Appl.* **3**, 303–314 (2009)
17. Luukkonen, T.: Modelling and control of quadcopter. Independent research project in applied mathematics, Espoo 22 (2011)
18. Shastry, A.K., Bhargavapuri, M.T., Kothari, M., Sahoo, S.R.: Quaternion based adaptive control for package delivery using variable-pitch quadrotors. In: Indian Control Conference, pp. 340–345. IEEE (2018)



# Human-Robot Collaborative Navigation Search Using Social Reward Sources

Marc Dalmasso<sup>(✉)</sup>, Anaís Garrell, Pablo Jiménez, and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial (CSIC-UPC),  
Llorens Artigas 4-6, 08028 Barcelona, Spain  
[{mdalmasso,agarrell,jimenez,sanfeliu}@iri.upc.edu](mailto:{mdalmasso,agarrell,jimenez,sanfeliu}@iri.upc.edu)

**Abstract.** This paper proposes a Social Reward Sources (SRS) design for a Human-Robot Collaborative Navigation (HRCN) task: human-robot collaborative search. It is a flexible approach capable of handling the collaborative task, human-robot interaction and environment restrictions, all integrated on a common environment. We modelled task rewards based on unexplored area observability and isolation and evaluated the model through different levels of human-robot communication. The models are validated through quantitative evaluation against both agents' individual performance and qualitative surveying of participants' perception. After that, the three proposed communication levels are compared against each other using the previous metrics.

**Keywords:** Human-robot interaction · Human-robot collaboration · Human-Robot Collaborative Navigation · Social reward · Motion planning

## 1 Introduction

On its strife for enhancing life quality, humanity has developed an uncountable number of technologies. Through the years we minimized the effort behind all tasks and automation is a natural consequence of this quest, freeing humans from labour burden and relegating them to supervisory roles. Robotics pursue this ideal, automatic machines capable of physical interaction, motion, learning and adaptation, but some environments offer greater resistance against their intrusion, in particular those populated with humans. Making robots capable of working in social environments is in itself a huge achievement but, despite they may easily outrun humans on some applications, humans still remain as the core experts on many others. The potential of human-robot collaboration cannot be ignored, a coalition capable of exploiting both agents' proficiencies. Achieving effective human-robot collaboration (HRC) is more demanding than previous robotic endeavours. In fact, the main pillars of HRC are: knowledge representation, planning, communication, plan sharing, decision making, agreement and adaptation.

The complexity of social biology is built on instinctual reactions, a feature that inspired the creation of our SRS model, summarized in Sect. 3. This model builds an integrated task and world representation, treating action planning and perception as functionally equivalent events, upon which apply global planning methods. Here, we shape this model towards human-robot collaborative search design, exploring the reward space with an adapted motion planning algorithm. Essentially, we define a testbed for model performance evaluation of collaborative object search methods and propose a functional implementation for this task.

In this work, we focus on knowledge representation and planning for human-robot collaborative navigation tasks. In the given object search testbed, localization and interface communication are assumed solved and noise free. Similarly, human and robot perception are modelled given a range and a field of view and assumed to present no uncertainty. Moreover, the map of the search zone is known.

In the remainder of the paper, a short review of related work is presented in Sect. 2. Section 3 briefly defines the concept of “Social Reward Source (SRS)” and summarizes the motion planning implementation. Section 4 defines the human-robot collaborative search testbed and explains the proposed implementation, Sect. 5 specifies the experiments’ details and validation metrics and Sect. 6 evaluates the obtained results. Finally, in Sect. 7, we discuss conclusions and future work.

## 2 Related Work

Human-robot collaboration is a complex and transversal field. To address it, in what follows we encourage a broad view through a brief survey on philosophical and psychological collaboration definitions and fundamentals, a rough discussion about social biology models, as well as briefly appointing some bio-inspired robotics literature, and a review of the current state of the art in human-robot collaboration.

### 2.1 Fundamentals on Human Collaboration

Bratman defined three characteristic features of any shared cooperative activity: mutual responsiveness, commitment to the joint activity and commitment to mutual support [4]. Sharing a conceptual common ground has huge implications in collaborative tasks [7] and, according to [28], shared intentionality transforms: “gaze following into joint attention, social manipulation into cooperative communication, group activity into collaboration, and social learning into instructed learning”. Human groups fostering the development of shared task representations are proven to outperform those who don’t [29]. In [16] it is claimed that perceiving and action planning are functionally equivalent: internally representing external events. It is interesting to note that humans are capable of representing robot actions in a similar manner [31].

## 2.2 Biology Inspiration

Interactions based on long-lasting chemical marks that trigger instinctual reactions in other individuals caught our attention due to their low cognitive requirements and their broad transversality along species. Especially, the usage of such channels by social insects [30]. We found special potential in ants communication, illustrated by their job-specific trail marking pheromones, combining both positive and negative feedbacks, capable of signalling long and short term attractive paths and temporal avoidance of such [17]. From all insect-inspired models, we may highlight those making use of virtual pheromones [3, 22, 26].

## 2.3 Human-Robot Collaboration

When facing human-robot joint action, it is of utmost interest to analyze disciplines as human-human joint action and connect them to the human-robot joint action case [8, 15]. Theory of mind approaches take importance as we try to model the knowledge of the robot: [9] estimates and maintains mental states of other agents reducing the unnecessary information given to the human and [19] claims to have built a cognitive robot to successfully share collaborative spaces and tasks. Moreover, Roncone's proposal [25] is able to autonomously reason about the problem of allocating specific subtasks to either the robot or its human partner.

Petit et al. [24] presented an approach using real-time multimodal learning capable of negotiation and new actions integration through imitation. Legible motion is defended in front of functional movement, which can harm coordination [10]. In [2], robot intention was conveyed through its projection on objects, [23] monitors human fatigue and [6] used trust as a latent variable. Many interesting efforts approached physical human-robot collaboration (PHRC), a detailed survey of this field can be found in [1].

**Metrics for HRC.** It has become necessary to quantitatively analyze the performance of the heterogeneous teams to enable comparison between different team configurations. Recently, [14] reviewed present subjective and objective fluency metrics. He suggests to carefully observe objective metrics dynamic behaviour, given their variability, and studies their correlation with subjective metrics.

**Human-Robot Collaborative Navigation.** One of the first faced human-robot collaborative challenges was side-by-side navigation [20, 21]. In parallel, [11, 13] approached this challenge through Social Force Model (SFM) methods and, in another context, [18, 27] presented methods for side-by-side wheelchair navigation. Alternative approaches to HRCN include co-driving, as the collaborative teleoperation of a robot through dialogue [12] or the collaborative control of wheelchair [5]. They are the first steps into collaborative models, but they are task-focused thus can't be extended to other applications. We pursue a flexible model capable of representing multiple tasks and conveying such representations to the human.

### 3 Social Reward Sources

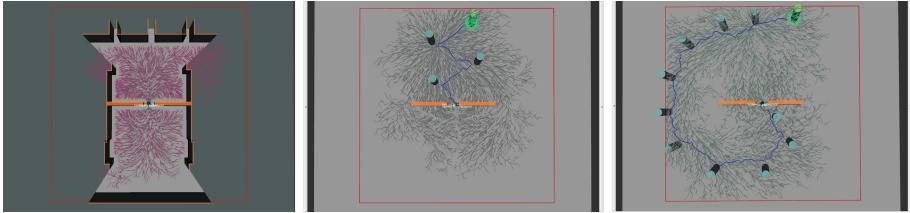
The SRS model is based on two primary instincts, attraction and repulsion, or in other terms, positive and negative perception of one's state. We can find many literature applications based on reward function definitions in the literature, but the SRS model aims not to describe the final reward, but to model its sources. It can be seen as a generative model framework, as it describes sources properties and dynamics affecting the final global reward. As introduced before, such model is inspired in social biology mechanisms, as in the case of virtual pheromones. Logically it easily extends to repulsion over personal space invasion, but these social reward sources may encode higher level abstractions. This includes, for example, the satisfaction over fulfilling a task, the propensity to follow someone's instructions or the discomfort felt when obstructing other people actions, as when standing in front of a person trying to take a photograph. This model aims to integrate and unify world and task representations, human-robot communication and human social or profiling preferences in a unique interrelated framework.

**Table 1.** SRS implementation over sampling methods

Path generation	Path selection
$C_n^{gen} = C_n^{gen,cm} + C_n^{gen,cs}$ $C_n^{gen,cm} = \sum_i^{P_n} \left( \sum_j^{S^{gen,cm}} (s_j(i) \cdot d_{i-1,i}) + c_{i-1,i}^{conn} \right)$ $C_i^{gen,cs} = \sum_j^{S^{gen,cs}} (\max_i^{P_n} \{s_j(i)\})$	$C_n^{sel} = C_n^{sel,cm} + C_n^{sel,cs} + C_n^{sel,f}$ $C_n^{sel,cm} = \sum_i^{P_n} \left( \sum_j^{S^{sel,cm}} (s_j(i) \cdot d_{i-1,i}) + c_{i-1,i}^{conn} \right)$ $C_i^{sel,cs} = \sum_j^{S^{sel,cs}} (\max_i^{P_n} \{s_j(i)\})$ $C_i^{sel,f} = \sum_j^{S^{sel,f}} s_j(i)$

Where  $C_n^{p,k}$  is the cost contribution to the path  $P_n$  ending at node  $n$  of  $S^{p,k}$ , the set of sources  $s$  of nature  $k$  (possibly being cumulative  $cm$ , consumable  $cs$  or final  $f$ ) and application policy  $p$  (path generation  $gen$  and/or path selection  $sel$ ). Also,  $c_{a,b}^{conn}$  and  $d_{a,b}$  denote respectively the connection cost and distance from  $a$  to  $b$ , being the latter in the dimensional magnitude over which cumulative cost densities are defined

Ultimately, the only requirement for a reward source is to correctly generate a reward function defined along all the search space. Nevertheless, consistent spatial properties of humans' world abstractions, such as objects, rooms or demonstrative references, inspired setting a spatial interpretation for such sources. Due to these and other functional and dynamic considerations, each source is defined by the following properties: type (repulsive or attractive), application policy (i.e. path generation and/or path selection), nature (cumulative, consumable or final), model (i.e. standard decay, as Gaussian or power function models, or complex definitions, as the graph-built observability presented in this paper), shape and dynamics (movement).



**Fig. 1.** Social reward sources.  $S - RRT^*$  expansion and path selection for, from left to right: (a) A hallway environment, (b) and (c) two consumable reward sources distributions ending on a final reword source.

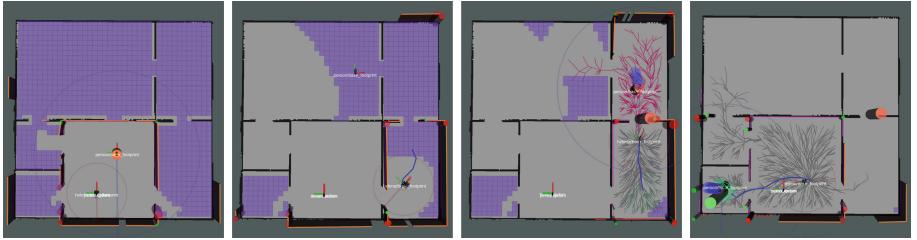
When exploring the generated rewards, they can be mirrored to understand them as costs. Essentially, any negative reward can be seen as the perceived cost of receiving it, while a positive one can be modelled as a negative cost. Any motion planning algorithm that takes these costs in consideration is a suitable search engine for exploring space and computing a path. Here we have adapted the well-known  $RRT^*$  algorithm (Rapidly exploring Random Trees) due to its computational efficiency. We call the resulting algorithm  $S - RRT^*$ , where  $S$  stands for “Sourced” emphasizing the usage of reward’s sources models. The computation of the relevant costs within  $S - RRT^*$  is summarized in Table 1. Moreover, some applications of the method are shown in Fig. 1.

## 4 Human-Robot Collaborative Search

The main issue in collaborative search is to share the exploration progress with each other. Several approaches to map sharing have been published for multi-robot collaboration, but such approaches are not suitable for a human’s mental map. Instead, humans actively do infer others’ knowledge from their actions while, at the same time, they expect to be inferred themselves. Only in doubtful situations, they do resort to specific task-related active communication. As a matter of fact, humans are experts in social and navigation tasks, and thus interacting with a robot can easily become boring or burdening.

The collaborative search testbed has been defined as follows: both the human and the robot know the map of the search zone beforehand and the searched object can be in any place of the unexplored zone with uniform probability. The task ends when either one of the agents finds the object. In simulation experiments, exploration progress is shown to the human to avoid misestimation of the observed zone. Communication between the robot and the human can be arbitrarily extended to enhance joint activity performance or fluency.

In our implementation, both human and robot detection capabilities are defined as radial distances, their field of view is assumed of  $360^\circ$  and no detection uncertainty is considered, as observable in (Fig. 2a). The human is detected and tracked by the robot through 2D laser sensors and the robot knowledge of his or her contribution to the exploration is inferred accordingly (Fig. 2b). Two



**Fig. 2.** Collaborative search testbed. From left to right: (a) The robot infers the unexplored zone from its detection range (red circle) and the person’s (blue circle). (b) People detection is impossible when the person is out of sight, hence no inference is done. (c) The person indicates the robot to avoid searching through that zone, as either it is already explored or the person will do it on their own. (d) The person finds the object, thus indicates the robot to come.

communication examples using our model are shown in Fig. 2c and d, being respectively to “avoid going through a zone” or to “go to one place”.

Aiming at a specific model for the object search task, we discretised the exploratory area and built an observability graph. We model the belief of seeing the searched object from one place as the observable unexplored zone from it (Fig. 3a). Similarly, we model the search isolation of one place as the inverse of the mean observability of the observable area from this point (Fig. 3b). Both values are normalized and merged in a weighted sum, the second being added to tune robot eagerness to clear neighbouring non-observed isolated zones before addressing bigger zones. This combination is normalized and weighted on a logarithmic scale to construct the final search reward shown in Fig. 3c.

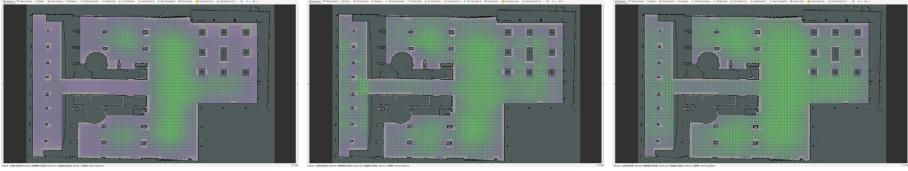
$$O(p) = \sum_{p_i}^{obs(p)} B(p_i) \quad (1a)$$

$$I(p) = \left( obs(p) / \sum_{p_i}^{obs(p)} O(p_i) \right) \cdot O(p) \quad (1b)$$

$$S(p) = O(p) / \max_{p_i}(O(p_i)) \cdot w_o + I(p) / \max_{p_i}(I(p_i)) \cdot w_i \quad (1c)$$

$$R(p) = \log(S(p) / \max_{p_i}(S(p_i))) + 1 \quad (1d)$$

Given  $B(p)$  is the prior probability of the object being at location  $p$  and  $obs(p)$  is the observable zone by the robot from  $p$ .  $O(p)$ ,  $I(p)$  and  $S(p)$  are respectively the observability, isolation and search scores of location  $p$ .  $w_o$  and  $w_i$  are the tuned weight values for observability and isolation,  $R(p)$  is the normalized reward nominal value of  $p$ . The search social reward source is of final nature and applied in the path selection phase.

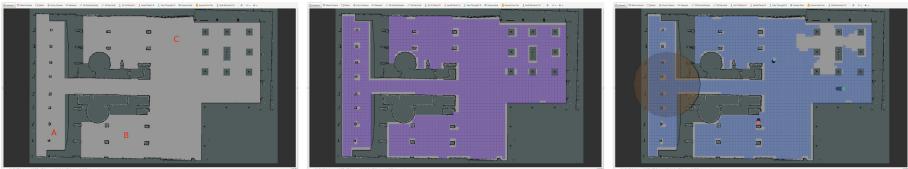


**Fig. 3.** Collaborative search source. From left to right: (a) Observability score of the current map exploration. (b) Isolation Score of the current map exploration. (c) Search reward generated by the collaborative search source for  $w_o = 0.8$  and  $w_i = 0.2$ , the values found to work best.

## 5 Validation

To validate our model, we chose the BRL map from the *Barcelona Robot Lab Dataset*<sup>1</sup> and defined three different origins to begin the search (Fig. 4a). The considered explorable area is discretised and shown in Fig. 4b and all objects in the scene are assumed to block both the view of the robot and the human.

First, we tested human and robot individual search performance to establish a baseline. After that, we tested the human-robot collaborative search model through three different communication levels. In the first one, the human was only able to see the exploration progress and the robot location. In the second approach, the robot showed the human his perceived exploration progress and his current planned path. During the third experiment, the human was able to communicate with the robot through 5 instructions (Fig. 4c): three general instructions (“go to this place”, “pass through this place” and “avoid this place”) and 2 task-related informative messages (“I’m going to this place” and “I’ve already been here”).



**Fig. 4.** Collaborative search experiments. From left to right: (a) BRL map. (b) Search space discretisation. (c) Robot perceived exploration progress and visual feedback of the communication instructions given to the human: “go to this place” (green cylinder), “pas through this place” (blue cylinder), “avoid this place” (red cylinder), “I’m going to this place” (brown area) and “I’ve already been here” (perceived explored area at the top right zone of the map).

A total of 12 volunteers participated in the experiment, with ages between 15 and 34 (mean: 26.42 std: 5.23). On a scale of 1 (None) to 7 (Expert) their

<sup>1</sup> <http://www.iri.upc.edu/research/webprojects/pau/datasets/BRL/>.

average self-evaluated knowledge in robotics was 4.83 (std: 1.64). No one had any experience using the framework, neither were they given the chance to practice. Each of them participated in three or four of experiment setups involving humans, doing 3 or 6 episodes on each one equally distributed among the different origins. Additionally, participants were surveyed after each communication level setup whether they perceived robot plan as efficient and how much did they change their plans due to the robot actions. Both questions were answered on a linear scale from 1 (not at all) to 7 (completely).

During all the experiments, both the speed of the robot and the human were limited. The robot was able to move at a maximum linear speed of 0.7 m/s, being it the nominal maximum velocity of the real robot, a luggage transporter mounted on a *Pioneer P3-DX* base. The human maximum velocity was limited to 1 m/s and it's movement controlled through a *PlayStation 3 Dualshock 3 Wireless Controller*. The final mean speeds of the human and the robot during the simulations were 0.83 m/s and 0.53 m/s.

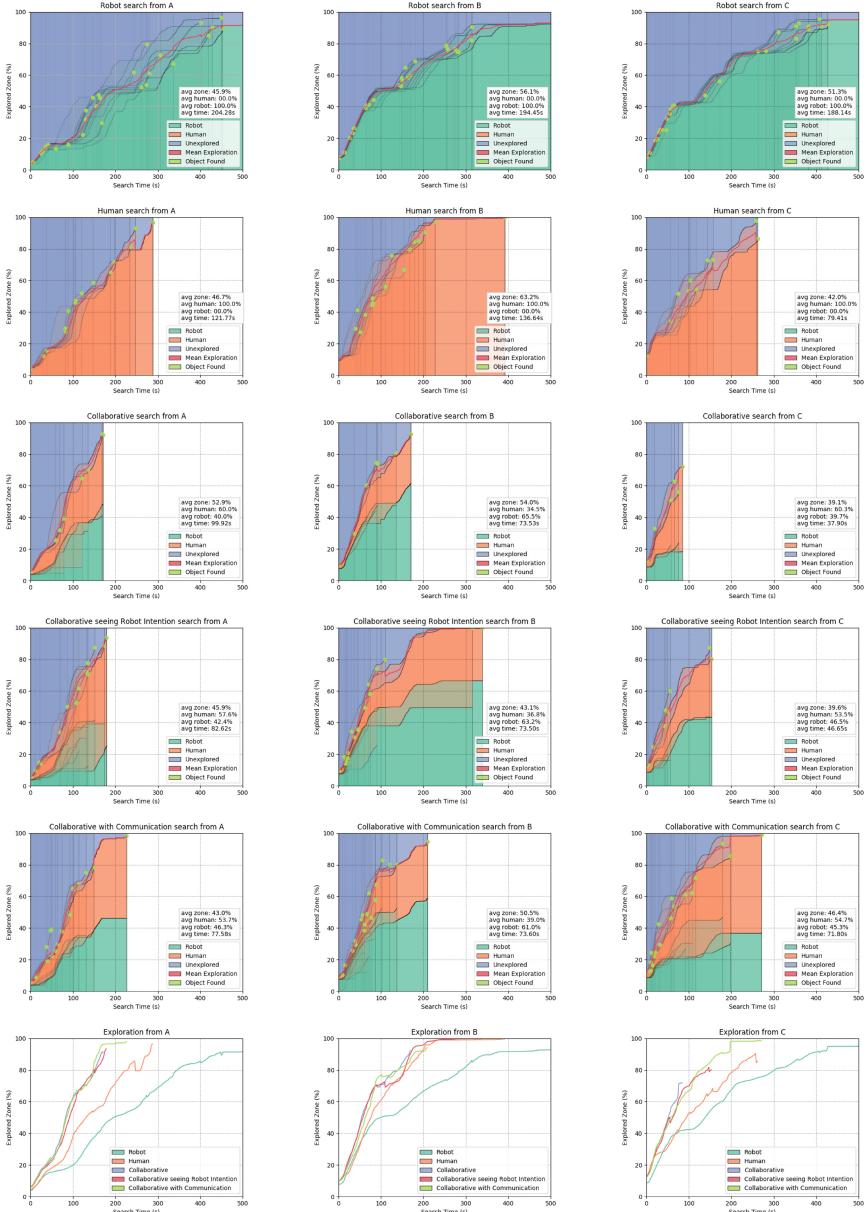
## 6 Results and Discussion

A complete plot of the collaborative search dataset is shown in Fig. 5. Here, we can observe origin selection has a strong effect on the search progress dynamics. Although the robot is slower, we can observe correlations between the human and the robot search progress shape, suggesting their search policies are alike.

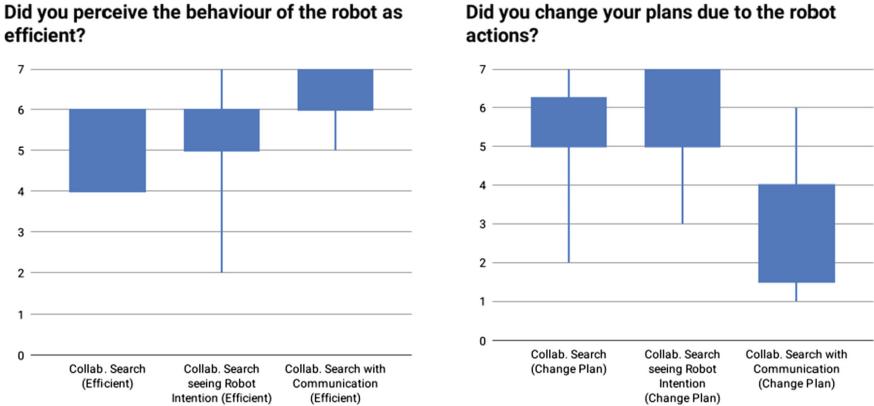
Robot behaviour consistently shows greater variability when beginning in origin A until the last collaborative setup. Such variability presumably appears due to the presence of two major bifurcations. Consistency in the collaborative search with communication dataset suggests human users either instructed the robot where to go or implicitly conditioned its choice by providing it with information. As a matter of fact, all the participants preferred the robot to take the hallway while they explored the remaining area in their side. Moreover, most of them enforced this behaviour through direct orders, while the usage of the task-related informative messages was relegated only to the right part of the map.

Episodes beginning in B have the biggest robot contribution. In this origin, after exploring the little zone at the left, both the robot and the user are enforced to take the same direction. That obstructed searching in parallel. Except for late-stage search progress when beginning in this origin, all three collaborative models surpassed both the individual human and robot baselines. In terms of search progress, however, neither of the three is significantly better than the others. We judge that adaptation capabilities of the human, as well as its superior movement capabilities, made up for the lack of communication.

Human subjective perception of the task, however, does change between the three collaborative setups. Including human to robot communication seems to increase the human perception of the robot efficiency and greatly decrease situations where the human is forced to adapt to the robot. Differences between the other two models are less clear. Even though in the second one the human had



**Fig. 5.** Human-robot collaborative search experiments. From left to right: episodes beginning at origins A, B and C. From top to bottom: robot individual search, human individual search, collaborative search, collaborative search seeing robot intention, collaborative search including human to robot communication and comparison between the 5 setups, both in performance and concurrent activity.



**Fig. 6.** Collaborative search survey

a broader perception of the robot intention, this might have enhanced conflict situations between the human-perceived robot plan and their own. Results of the survey are represented in Fig. 6.

## 7 Conclusions

In this paper, we presented a complete human-robot collaborative navigation task implementation in the SRS framework, which is proven to outperform the individual search baseline. Moreover, human to robot communication is proven to have a major impact in human perception of human-robot collaborative tasks, while performance might not be significantly affected in simple setting due to the human adaptation capabilities.

We aimed to adapt fluency metrics analyzed in [14]. However, their dynamics didn't seem to correlate with the results obtained in the qualitative survey, which may suggest the need to search for other quantitative metrics. Moreover, to do so we identified actual progress in the exploration, as identifying all goal-driven movement would result in the trivial case of not having idle time in any agent.

This is a first approach tackling task-oriented explicit collaborative navigation. In future work, we will expand this model to include theory of mind knowledge models, shared planning and agreement mechanisms.

**Acknowledgements.** Work supported under projects ColRobTransp (DPI2016-78957-RAEI/FEDER EU), TERRINet (H2020-INFRAIA-2017-1-two-stage-730994) and by the Spanish State Research Agency through the Maria de Maeztu Seal of Excellence to IRI (MDM-2016-0656).

## References

1. Ajoudani, A., Zanchettin, A.M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., Khatib, O.: Progress and prospects of the human-robot collaboration. *Auton. Robots* **42**, 957–975 (2018)
2. Andersen, R.S., Madsen, O., Moeslund, T.B., Amor, H.B.: Projecting robot intentions into human environments. In: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 294–301. IEEE (2016)
3. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* **7**(1), 1–41 (2013)
4. Bratman, M.E.: Shared cooperative activity. *Philos. Rev.* **101**(2), 327–341 (1992)
5. Carlson, T., Demiris, Y.: Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **42**(3), 876–888 (2012)
6. Chen, M., Nikolaidis, S., Soh, H., Hsu, D., Srinivasa, S.: Planning with trust for human-robot collaboration. In: Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pp. 307–315. ACM (2018)
7. Clark, H.H., Schreuder, R., Buttrick, S.: Common ground at the understanding of demonstrative reference. *J. Verbal Learn. Verbal Behav.* **22**(2), 245–258 (1983)
8. Clodic, A., Pacherie, E., Alami, R., Chatila, R.: Key elements for human-robot joint action. In: Sociality and Normativity for Robots, pp. 159–177. Springer (2017)
9. Devin, S., Alami, R.: An implemented theory of mind to improve human-robot shared plans execution. In: 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 319–326. IEEE (2016)
10. Dragan, A.D., Bauman, S., Forlizzi, J., Srinivasa, S.S.: Effects of robot motion on human-robot collaboration. In: Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 51–58. ACM (2015)
11. Ferrer, G., Zulueta, A.G., Cotarelo, F.H., Sanfeliu, A.: Robot social-aware navigation framework to accompany people walking side-by-side. *Auton. Robots* **41**(4), 775–793 (2017)
12. Fong, T., Thorpe, C., Baur, C.: Collaboration, dialogue, human-robot interaction. In: Robotics Research, pp. 255–266. Springer (2003)
13. Garrell, A., Garza-Elizondo, L., Villamizar, M., Herrero, F., Sanfeliu, A.: Aerial social force model: a new framework to accompany people using autonomous flying robots. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, 24–28 September 2017, pp. 7011–7017 (2017)
14. Hoffman, G.: Evaluating fluency in human-robot collaboration. *IEEE Trans. Hum.-Mach. Syst.* **49**, 209–218 (2019)
15. Hoffman, G., Breazeal, C.: Collaboration in human-robot teams. In: AIAA 1st Intelligent Systems Technical Conference, p. 6434 (2004)
16. Hommel, B., Müsseler, J., Aschersleben, G., Prinz, W.: The theory of event coding (TEC): a framework for perception and action planning. *Behav. Brain Sci.* **24**(5), 849–878 (2001)
17. Jackson, D.E., Ratnieks, F.L.: Communication in ants. *Curr. Biol.* **16**(15), R570–R574 (2006)
18. Jayawardena, C., Ardekani, I., et al.: A navigation model for side-by-side robotic wheelchairs for optimizing social comfort in crossing situations. *Robot. Auton. Syst.* **100**, 27–40 (2018)

19. Lemaignan, S., Warnier, M., Sisbot, E.A., Clodic, A., Alami, R.: Artificial cognition for social human-robot interaction: an implementation. *Artif. Intell.* **247**, 45–69 (2017)
20. Morales, Y., Kanda, T., Hagita, N.: Walking together: side-by-side walking model for an interacting robot. *J. Hum.-Robot Interact.* **3**(2), 50–73 (2014)
21. Nakazawa, K., Takahashi, K., Kaneko, M.: Movement control of accompanying robot based on artificial potential field adapted to dynamic environments. *Electr. Eng. Jpn.* **192**(1), 25–35 (2015)
22. Narzt, W., Wilflingseder, U., Pomberger, G., Kolb, D., Hörtner, H.: Self-organising congestion evasion strategies using ant-based pheromones. *IET Intel. Transp. Syst.* **4**(1), 93–102 (2010)
23. Peternel, L., Tsagarakis, N., Caldwell, D., Ajoudani, A.: Robot adaptation to human physical fatigue in human-robot co-manipulation. *Auton. Robots* **42**, 1–11 (2018)
24. Petit, M., Lallée, S., Boucher, J.D., Pointeau, G., Cheminade, P., Ognibene, D., Chinellato, E., Pattacini, U., Gori, I., Martinez-Hernandez, U., et al.: The coordinating role of language in real-time multimodal learning of cooperative tasks. *IEEE Trans. Auton. Ment. Dev.* **5**(1), 3–17 (2013)
25. Roncone, A., Mangin, O., Scassellati, B.: Transparent role assignment and task allocation in human robot collaboration. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1014–1021. IEEE (2017)
26. Susnea, I., Vasiliu, G., Filipescu, A., Radaschin, A.: Virtual pheromones for real-time control of autonomous mobile robots. *Stud. Inform. Control* **18**(3), 233–240 (2009)
27. The, V.N., Jayawardena, C.: A decision making model for optimizing social relationship for side-by-side robotic wheelchairs in active mode. In: IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp. 735–740 (2016)
28. Tomasello, M., Carpenter, M.: Shared intentionality. *Dev. Sci.* **10**(1), 121–125 (2007)
29. Van Ginkel, W., Tindale, R.S., Van Knippenberg, D.: Team reflexivity, development of shared task representations, and the use of distributed information in group decision making. *Group Dyn.: Theory Res. Pract.* **13**(4), 265 (2009)
30. Vander Meer, R.K., Breed, M.D., Espelie, K.E., Winston, M.L.: Pheromone Communication in Social Insects. Ants, Wasps, Bees and Termites, vol. 162. Westview, Boulder (1998)
31. Wykowska, A., Chellali, R., Al-Amin, M.M., Müller, H.J.: Implications of robot actions for human perception. How do we represent actions of the observed robots? *Int. J. Soc. Robot.* **6**(3), 357–366 (2014)



# Mind Perception of a Sociable Humanoid Robot: A Comparison Between Elderly and Young Adults

Maryam Alimardani<sup>(✉)</sup> and Sonia Qurashi

Department of Cognitive Science and AI, Tilburg University,  
Tilburg, The Netherlands  
m.alimardani@uvt.nl

**Abstract.** Sociable robots are slowly entering domains such as education and healthcare. As we are exposing our youth and elderly to these new intelligent technologies, it is important to understand their perception and attitudes towards robots. This study investigates the differences between elderly and young adults in ascribing mind perception to a sociable humanoid robot. Both subjective and behavioral measurements were employed to investigate the differences. Several trends were found; elderlies attributed higher scores of mind perception to the robot, whereas young adults seemed to have a more positive attitude towards it. Elderlies seemed to apply human social models of interaction, whereas young adults perceived a master-slave relationship between humans and the robot. Furthermore, a significant positive correlation was found between mind perception and attitude toward the robot for both groups.

**Keywords:** Human-Robot Interaction (HRI) · Sociable robots · Mind perception · Humanoid robot · Elderly

## 1 Introduction

Recently, there has been a growing industry for sociable robots; this has opened a new range of application domains, such as healthcare, education and entertainment. Several studies have found positive effects of sociable robots in these application domains. Robots in elderly care have been compared to animal-assisted therapy, which reduce the loneliness that some elderly feel [1]. Furthermore, significant learning gains have been found in test scores in a second language learning experiment conducted with robots [2]. Outcomes of these studies sound very promising. However, there has been criticism about the potential creation of social relationships between humans and robots. Turkle et al. expressed [3]: “the fact that our parents, grandparents and our children might say ‘I love you’ to a robot who will say ‘I love you’ in return, does not feel completely comfortable; it raises questions about the kind of authenticity we require of our technology”. Sparrow also argues that the design of robots predicts mistaking, at a conscious or unconscious level, for real organisms and that this violates our duty to apprehend our world accurately [4]. Contrarily, Sharkey and Sharkey discuss that it is of human tendency to want to anthropomorphize objects and machines

[5]. As robots are increasingly present in environments shared with humans, it is important to investigate the perception people have of sociable robots.

Different researchers have studied questions related to such concepts. In a recent study by Horstman et al., participants were instructed to switch off a robot after having either a functional or social interaction with the robot [6]. During this interaction, the robot was either objecting or not. The study found that participants hesitated longest when they had a functional interaction, and the robot was objecting to being shut down. Similarly, a study conducted by Rosenthal-von der Putten et al. found that participants experienced increased physiological arousal and empathic concerns when exposed to a video of a dinosaur robot being tortured [7]. These findings were in contrast with the study of Bartneck and his colleagues [8]. They reproduced the famous Milgram experiment where the participants had to follow the experimenter's instructions to induce 'deadly' electric shocks to a robot. In the original experiment, 65% of the participants showed moral distress and some refused to continue giving increasing electric shocks to the human subject. However, in the experiment with the robot, all participants continued giving high voltage electric shocks to the robot. Kahn et al. also investigated the social and moral relationships of children with a robot (Robovie) during a 15-minute interaction [9]. The experimenters sought to engage the children in an increasingly interesting and complex relationship with Robovie by sequencing them in nine interaction patterns, which were socially plausible. Finally, this sequencing led to having each child watch as Robovie was subjected to potential moral harm by the experimenters. The children were videotaped to collect behavioral data and were engaged in a semi-structured interview. The interview data revealed that the majority of participants thought the robot had mental states (e.g. had feelings, was intelligent) and had a moral standing (could be a friend or be trusted with secrets).

In order to have a deeper understanding of the social behavior of humans towards robots, it is important to understand peoples' perception towards robots. An area of investigation within HRI has been whether people believe that the robot has a mind. Gray and colleagues examined the degree in which people ascribe mental capacities to various agents, such as an animal, baby and a robot; this is called mind perception [10]. They found two dimensions of mind perception: mind experience and mind agency. Mind experience can be summarized as an agent's perceived ability for "feeling", such as the capacity to feel: pleasure, fear, desire, hunger, consciousness, pain, rage, personality, embarrassment, joy and pride. Contrarily, mind agency is the agent's perceived ability for "being", such as the capacity for: self-control, memory, planning, morality, emotion recognition, thought and planning. In the study of Gray et al. [10], the robot character received a low score for capacity for experience, but a moderate score for agency (which was higher than a dog). Subsequently in [11], Gray et al. suggest that mind perception is the essence of moral judgement because moral judgement is based on the "cognitive template of perceived minds". In other words, the higher the degree of mind perception, the higher the chance that human rules of morality will be applied.

The theory of mind perception is very closely related to the phenomena of anthropomorphism and applying social models to inanimate objects, in the sense that human capacities of mind are attributed to non-human agents. Nass et al. found that humans are more likely to attribute capacities of mind perception to inanimate objects depending on the number and intensity of primitive social cues they receive from these

objects [12]. In the case of robots, not only external appearance, but also their behavior and successful task performance are shown to influence anthropomorphism [13]. While higher anthropomorphism in social robots facilitates HRI, it is also found to impose limitations on it. People expect humanlike robots to adhere to human norms and have much higher expectations regarding their capabilities compared to machine-like robots [13]. One should keep in mind that there are also individual differences in the perception of identical non-human agents. It has been shown that children and elderly have a higher tendency to anthropomorphize compared to young adults [14]. Interestingly, literature suggests that robots with more humanlike capabilities and adaptive behavior cause higher levels of anxiety and negative emotions in users, particularly elderly [15, 16]. A possible explanation for this negative effect could be lack of experience with technology among elderly that leads to higher perceived mind capacities (of robots) and hence generates an effect similar to computer anxiety [17].

The aim of the current study was to extend the foundational knowledge on the mind perception of robots and investigate whether such perception leads to positive or negative attitudes toward the robot. As mentioned before, there seems to be a negative relationship between age and technology acceptance. In order to investigate age-related differences in ascribing mind perception to a sociable humanoid robot, two age groups (i) elderly and (ii) young adults were compared. The following research question was raised: What are the differences between elderly and young adults in ascribing mind perception to a sociable humanoid robot?

We hypothesized that elderly ascribe a higher level of mind perception to the robot and as a result have a more negative attitude toward it compared to young adults. Consequently, we expected that the elderly would be more inclined to believe that moral injustice is done to the robot when it is subjected to potential moral harm.

## 2 Methodology

### 2.1 Participants

A total of 65 participants, all Dutch natives, participated in this experiment. Some data were removed from analysis due to technical issues of the robot or incomplete answers by the participants. The final analysis was based on the experimental data of 53 participants, 25 elderly ( $M = 68.88$ ,  $SD = 7.12$ ) and 28 young adults ( $M = 21.79$ ,  $SD = 2.95$ ). All participants received explanation about the study and signed an informed consent form before the experiment. The study was approved by the Research Ethics Committee of Tilburg School of Humanities and Digital Sciences.

### 2.2 Experimental Setup

The experiments for the elderly took place at the elderly care home ‘de Wever’, Tilburg, and the experiments for the young adults took place at Tilburg University research labs. Both experiments had identical setup, and were conducted in groups of 2 to 4 participants. A room within the facility was reserved where the experiments could be conducted in a quiet environment.



**Fig. 1.** Young adults (left) and elderly (right) during interaction with robot Charlie.

The humanoid robot used in this study was the NAO robot developed by Aldebaran Robotics (see Fig. 1). The robot, referred to as Charlie in this experiment, was programmed in advance but was also controlled by one of the researchers (not visible to the participants). This researcher (Wizard-of-Oz) made sure that the robot processed the verbal responses of the participants correctly and provided correct verbal responses where necessary.

The interaction scenario lasted about 15–20 min and was designed to investigate the perception and willingness of people to anthropomorphize an inanimate object. The interaction consisted of three phases. In the initial introduction phase Charlie (the robot) introduced himself. He told the participants that he came from Japan and was looking forward to meeting them. He asked the participants whether they wanted to see a short performance, and proceeded to show some moves of Tai Chi. This allowed participants to see the complex movements the robot was capable of. After the introduction phase, two rounds of riddles were played. The participants could ask Charlie for hints or repetition of the riddles/hints. Finally, after the second round of riddles, Charlie was interrupted by the experimenter and was told to stop. Charlie proceeded to show verbal objection (e.g. “can I please stay here and finish my last riddle, I don’t want to leave yet, please!”) and physical objection towards the researcher interrupting him. An example of physical objection shown by Charlie was sitting down and putting his hands up when the researcher tried to pick him up. Nevertheless, the experimenter took the robot and left the room carrying it. The interaction was then terminated.

### 2.3 Measurements

A short pre-experiment questionnaire was used to assess previous experience of the participants with technology. All participants had to rate their experience with phones/computers and robots, based on a 5-point Likert scale ranging from “no experience” to “professional”. All questionnaires employed in this study were filled in individually using paper and pen and the participants took as long as they needed to go through all questions.

A post-experiment questionnaire, including 13 questions, was used to measure the perception of the participants after having an interaction with the robot. Eight items

were chosen from the original 18 item scale of the Dimensions of Mind Perception Questionnaire [18]. The scale was composed of two subscales; Mind Agency (consisting of perceived capacity of the robot to recognize emotions, have thought, memory and self-control) and Mind Experience (consisting of perceived capacity of the robot to feel pleasure, hunger, pain and have a consciousness). Participants rated each item on a 5-point Likert scale ranging from “1 = No ability” to “5 = Complete ability”. Additionally, four questions were inspired by the Robot Attitudes Scale [19]. The following 4 items regarding Attitude were assessed (machine-like: human-like, unfriendly: friendly, stupid: smart, artificial: a living being). These questions were also rated on a 5-point semantic differential scale. Finally, participants were questioned about moral injustice done to the robot “How bad did you think it was when Charlie was taken away against its will?”. They rated their response on a 5-point Likert scale ranging from “1 = I was completely fine with it” to “5 = Horrible”.

The interaction was video-recorded and participant’s facial reactions during the final interruption moments, where Charlie was being removed despite its objection, were analyzed. This part of the interaction lasted approximately 30 s. The facial expressions of the participants during these 30 s were investigated through qualitative analysis of the videos manually conducted by the second author. The expressions: neutral, disengaged, confused, concerned and amused were used to code the facial expression. The most dominant expression of the participant was coded.

### 3 Results

#### 3.1 Survey Results

##### Previous Experience with Technology

Previous experience with technology was assessed for both elderly ( $N = 25$ ) and young adults ( $N = 28$ ) through a pre-experiment questionnaire. Majority of the elderly (48%) indicated that they were experienced with phones/computers; this meant that they used their phones/computers regularly and they knew how to use different kinds of programs and applications. Fifty-six percent of the elderly indicated that they had no previous experience with robots. A large part of the elderly (36%) indicated that they would rate their experience with robots as “beginners”, which meant they had heard about robots and seen them for example on the television. Seventy-five percent of the young adults indicated that they were experienced with phones/computers. As opposed to elderly, 61% of the young adults indicated they had intermediate experience with robots, which meant that they had seen a robot in real life.

##### Mind Perception (Mind Agency and Mind Experience)

A summary of Mind Perception scores from both groups are given in Table 1. Both groups scored close to “moderate ability” for Mind Agency, elderly ( $M = 2.76$ ,  $SD = 0.93$ ) and young adults ( $M = 2.66$ ,  $SD = 0.77$ ). The reliability of this scale was assessed using the Cronbach’s Alpha, which had a good reliability ( $\alpha = .81$ ). A Kolmogorov-Smirnov test revealed that both groups were normally distributed, for

**Table 1.** Mind Perception scores for both groups

Mind perception	Elderly (N = 25)		Young adults (N = 25)	
	M	SD	M	SD
<u>Mind agency</u>				
Ability to recognize emotions	2.64	1.11	2.14	2.36
Ability to think	2.56	1.08	2.36	0.83
Ability to remember	3.16	0.99	3.18	1.06
Ability to control itself	2.68	1.22	2.96	1.17
<u>Mind experience</u>				
Ability to feel pleasure	2.40	0.96	2.46	1.37
Ability to feel hunger	1.56	0.71	1.21	0.57
Ability to feel pain	1.68	0.95	1.46	0.84
Ability to have a consciousness	1.64	0.91	1.50	0.76

elderly D(25) = .16, p = 0.09 and for young adults D(28) = .08, p = .20. An independent samples t-test was performed between elderly and young adults and found that Mind Agency was not statistically different, Mdif = .099, t(51) = .42, p = .67, r = .06.

**Table 2.** Attitude scores for both groups

Attitude	Elderly (N = 25)		Young adults (N = 28)	
	M	SD	M	SD
Machine-like - Human-like	2.60	0.96	2.61	0.99
Unfriendly - Friendly	3.72	0.94	4.46	0.58
Stupid - Smart	3.16	1.18	3.18	0.67
Artificial - Living-being	2.32	1.15	2.04	0.92

Regarding Mind Experience, both groups scored close to “little ability”, elderly (M = 1.82, SD = 0.67) and young adults (M = 1.66, SD = 0.74). The reliability of this scale was assessed using the Cronbach’s Alpha, which had a good reliability ( $\alpha = .78$ ). Scores were not normally distributed (elderly group was normally distributed D(25) = .12, p = 0.20, but the group of young adults was not D(28) = .20, p = 0.01), therefore, a Mann Whitney U test was performed. The test revealed that the differences in perceived Mind Experience were not statistically different between elderly (Mdn = 1.75) and young adults (Mdn = 1.5) U = 288, p = .26, r = .23.

### Attitude

A summary of Attitude scores collected from both groups are given in Table 2. Both groups scored close to “neutral”, elderly (M = 2.95, SD = 0.74) and young adults (M = 3.07, SD = 0.56). The reliability of this scale was assessed using the Cronbach’s

Alpha, which had a satisfactory reliability ( $\alpha = .62$ ). Data was not normally distributed (elderly group D(25) = .15,  $p = 0.13$ , and young adults D(28) = .17,  $p = .04$ ). A Mann Whitney U test revealed that the differences in Attitude were not significantly different between elderly (Mdn = 3.00) and young adults (Mdn = 3.13)  $U = 327$ ,  $p = .68$ ,  $r = .18$ .

### Perception of Moral Injustice

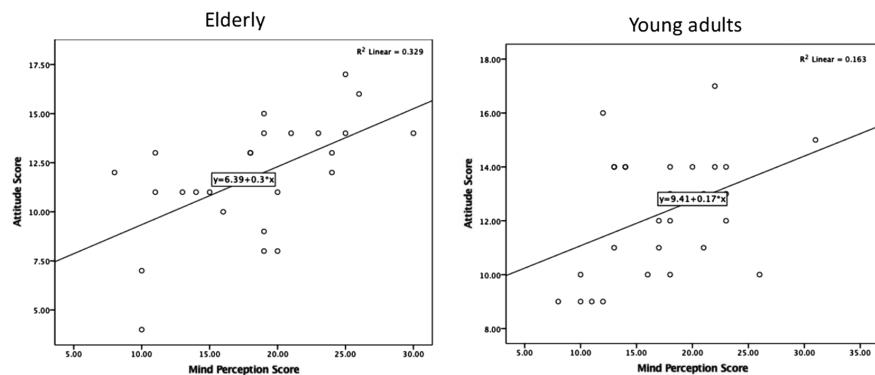
Analysis of the data revealed that the group of young adults ( $M = 3.11$ ,  $SD = 1.10$ ) reported higher levels of moral injustice than elderly ( $M = 2.86$ ,  $SD = 1.28$ ). A Kolmogorov-Smirnov test revealed that data from both elderly group D(25) = .20,  $p = 0.01$  and young adults D(28) = .22,  $p = 0.00$  were not normally distributed. Thus, the non-parametric Mann Whitney test was performed to assess the differences between the two groups. The test revealed that there was no statistical difference between the elderly (Mdn = 3.00) and the young adults (Mdn = 3.00),  $U = 285$ ,  $p = .23$ ,  $r = .10$ .

### Correlation Between Mind Perception and Attitude

To check whether there is a relationship between Mind Perception and Attitude a correlation analysis was performed for each group.

A Pearson correlation analysis was performed for the elderly. Scores from all items in each construct were summed and used for the analysis. Elderly on average scored 18.32 ( $SD = 5.78$ ) out of 40 for Mind Perception and 11.08 ( $SD = 2.60$ ) out of 20 for Attitude. A significant positive relationship was found between Mind Perception and Attitude,  $r = 0.57$ ,  $N = 25$ ,  $p = 0.003$  (Fig. 2 left).

A Kendall's Tau correlation analysis was conducted for young adults. On average, young adults scored Mind Perception 17.28 ( $SD = 5.48$ ) out of 40 and Attitude 12.29 ( $SD = 2.26$ ) out of 20. A significant positive relationship was found between Mind Perception and Attitude,  $r = 0.30$ ,  $N = 28$ ,  $p = 0.034$  (Fig. 2 right).



**Fig. 2.** Significant correlations were found between Mind Perception and Attitude for elderly (left) and young adults (right).

### Correlation Between Mind Perception and Moral Injustice

Similarly, it was assessed whether there is a relationship between Mind Perception and Moral Injustice for each group.

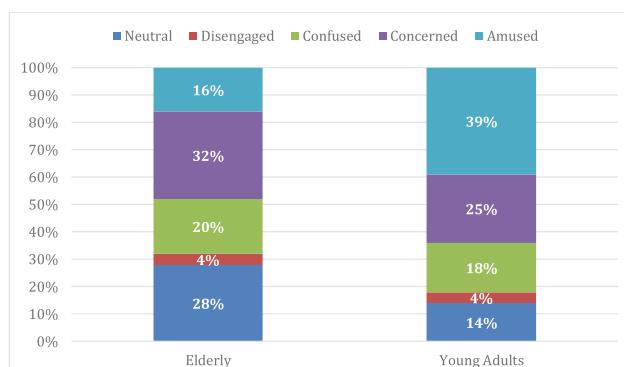
A Kendall's Tau correlation analysis was used for the elderly group. No significant correlation was found between Mind Perception and Moral Injustice,  $r = 0.25$ ,  $N = 25$ ,  $p = 0.119$ . The same correlation analysis was performed for the data from young adults group. Again no significant relationship was found between Mind Perception and Moral Injustice,  $r = 0.26$ ,  $N = 28$ ,  $p = 0.078$ .

### 3.2 Video Analysis

Analysis of participants' facial expression during the interruption time indicated that majority of the elderly group showed signs of concern (32%) (see Fig. 3). They were either frowning or watching the whole situation disapprovingly. For more than quarter of the participants (28%) it was not possible to code a facial reaction. Thus, this group was assigned to the category neutral. This group either showed no emotion, or the emotion could not be classified as a non-neutral category. Twenty percent of the elderly showed signs of confusion, most of these participants were looking back and forth between the robot and the experimenter, and were trying to understand what was happening. Sixteen percent of the elderly were clearly smiling; they were amused by the situation of the robot objecting. Four percent clearly showed signs of disengagement.

Contrary to the elderly, 39% of the young adults group showed signs of amusement (Fig. 3). This group of participants were smiling and enjoying the conversation that took place between the experimenter and the robot. This is an interesting contrast between the two groups, which will be discussed in the discussion section of this report. Interestingly, a big part of the young adults (25%) also showed signs of concern towards the robot and 18% of the young adults seemed confused about the situation.

In order to test whether there is a relationship between age group and facial reaction a Pearson's Chi-Square test was conducted. The test revealed no significant association between age group and facial reaction,  $\chi^2(4) = 3.99$ ,  $p = .457$ .



**Fig. 3.** Facial reaction of elderly and young adults during the interruption phase.

## 4 Discussion

The aim of the current study was to extend the foundational knowledge on mind perception of sociable robots by elderly and young adults, and its impact on their attitude toward the robot. Based on previous empirical evidence obtained by Epley et al. [14], we hypothesized that elderly would ascribe higher levels of mind perception to a sociable robot as opposed to young adults. In this study, mind perception was measured through mind agency (e.g. ability to think) and mind experience (e.g. ability to feel pain). Our results revealed that there was no significant difference between elderly and young adults in the ascription of mind perception. Both groups scored close to “moderate ability” for mind agency and mind experience. Although the differences were not significant, elderly did seem to ascribe higher scores for mind perception than young adults on average. This observation is in line with the findings of [14].

Additionally, we hypothesized that elderly would have a stronger negative attitude towards the robot because they would ascribe higher levels of mind perception than young adults. This hypothesis was based on previous research that found that elderly reported higher levels of anxiety and negative emotions towards a robot that was either more adaptive to the needs of the user [15] or had more humanlike appearance and capabilities [16, 20]. Our results showed that both groups of elderly and young adults assigned scores close to “neutral”, and the differences between the two groups were not statistically significant. On average, young adults seemed to perceive the robot as more humanlike, friendlier and smarter compared to the elderly. Although not significant, this observation is in line with the hypothesized statement, which expected that young adults would have a more positive attitude compared to elderly.

There might be various possible explanations for not finding a significant difference in the employed experimental design. Various evidence has been found that individual differences might affect one’s tendency to anthropomorphize inanimate objects. For example, Rubin et al. found that anxious individuals are sensitive to agentic cues and more readily apply anthropomorphism to objects [21]. The same study also found that loneliness increases the perception of agency in inanimate objects. Asquith suggested that some cultures seem more prone to anthropomorphism than others, and the setting in which an individual interacts with an inanimate object heavily influences the tendency to ascribe mind perception [22]. The current study did not take individual differences (e.g. gender, loneliness or cultural background) into account, which might have had a confounding effect on the results of this study.

It was expected that higher mind perception ascribed by the elderly would translate into being more inclined to believe that moral injustice was inflicted on the robot [11]. Contrary to our expectation, young adults in general reported a higher score for moral injustice in the survey, although this result was not significant between the two groups. This is while the video analysis of participants’ reactions to the robot’s interruption showed that elderly mostly seemed concerned about the situation while the majority of young adults were amused by it.

In the follow-up talk with the participants, young adults used the terms “funny” and “sad” to describe the situation. Interestingly, most of these participants perceived a mother-child or master-slave relationship between the robot and the experimenter

which made the behavior of the experimenter acceptable towards the robot. For this group of participants the situation was not morally troubling, on the contrary, it was “cute” and “funny”. Elderly, on the other hand, mentioned that the experimenter should have communicated in a more friendly manner and let the robot at least finish his game. Takayama and Pantofaru in [23] discuss how a person that perceives little agency in a robot will perceive it as a “tool”. The fact that elderly used the word “rude” to define their perception of the experimenter’s behavior towards the robot, indicates that they did not perceive the robot as just a “tool”. According to most elderly, the experimenter should have behaved “friendlier” and let the robot at least “finish his game”. These are rules in line with human rules of morality. As mentioned by Gray et al. [11], mind perception is the essence of moral judgement; the higher the degree of mind perception, the higher the chance that human rules of morality will be applied. According to this theory, this group of elderly did ascribe a sufficient amount of mind perception to the robot to believe that moral injustice was done to it to some degree.

An important finding of this study was, contrary to expectations, a significant positive relationship between mind perception and attitude in both elderly and young groups, whereas previous research found a negative relationship, suggesting that higher perceived human-likeness and mind capacities are associated with negative emotions amongst elderly [15, 16, 20]. Interestingly, the relationship between mind perception and attitude was almost twice as strong for elderly ( $R^2 = 0.33$ ) as opposed to young adults ( $R^2 = 0.16$ ), which means that the attitudes of elderly was better predictable based on mind perception than young adults. Stafford et al. established in their study that attitudes are closely related to robot acceptance in elderly care homes [18]. Furthermore, they found that eldercare robots have to be easy-to-use apart from being functional. Mind perception in this context can be seen as predictor of technology acceptance and attitudes. This can have implications for design and deployment of eldercare robots. The positive relationship found between mind perception and attitudes in this study can contribute to models that study human acceptance of technology.

A general limitation to this study was the small sample size, additionally there was an unbalanced sample (25 elderly vs. 28 young adults). Expanding the number of participants for each group will increase the statistical power of the results and subsequently its reliability to generalize. Secondly, the current study aimed to investigate the differences in mind perception between two age groups; elderly and young adults. Comparing two groups of individuals that differ so much in age comes with difficulties such as decline in cognitive abilities of the elderly. This could have influenced performances in tasks such as question comprehension, recall of relevant information and the formation of judgement. During the experiment, several participants from the elderly group experienced difficulties in understanding the synthesized voice of the robot. This led to some people not understanding the game of riddles that was played with the robot. Furthermore, elderly needed some assistance in the comprehension of the post-experiment questionnaire. It is debatable whether the self-reported differences represent a true difference in mind perception.

In order to assess the participants’ conception of the robot’s moral standing, they watched an act of moral injustice toward the robot executed by the experimenter. While elderly found this act concerning, the majority of young adults did not interpret it unjust as they assigned a master-slave relationship between the robot and the experimenter. It

has been shown that perspective taking in HRI influences the level of mind perception; people who take the role of an actor in the interaction attribute higher level of emotional capacity and moral rights to the robot than people who are merely observers [24]. Therefore, different results might have emerged if participants were required to remove the robot themselves.

Furthermore, the current study sought to investigate mind perception based on a single interaction, which lasted approximately 15 to 20 min. Research has established that time is an important component in the development of social relationships with robots [25]. In order to understand to what degree humans ascribe mind perception to robots and attach a moral standing to these inanimate objects, it is important to create a setting in which social relationships between humans and robots can flourish. Many participants of this study interacted with a robot for the first time and were impressed by its appearance and capabilities. Therefore, the results might have been influenced by the novelty effect. Many participants of each group indicated that they wished the interaction lasted longer. Also, both elderly and young participants interacted with the robot in a group setting and not individually. It has been shown that the recall of interaction details in children is higher when they interact with a story-telling robot individually rather than in group, although the emotion attribution to the robot remained the same in both conditions [26]. Thus, future studies should investigate the development of mind perception in elderly population over multiple and longer sessions, as well as in individual vs. group interactions.

Finally, this study only collected subjective and behavioral measurements in order to assess the impact of mind perception on participant's attitude and response toward the robot. Future research can employ objective measurements, such as skin conductance, to assess real-time physiological responses of participants to critical moments during the interaction.

## 5 Conclusion

The aim of the current study was to investigate the differences between elderly and young adults in ascribing mind perception to a sociable humanoid robot and whether that would affect attitude toward the robot and perception of moral injustice applied to it. Both subjective and behavioral measurements were employed to investigate the differences between the age groups. Results revealed no significant differences between the two groups in terms of mind perception and attitude toward the robot, but there was a significant positive relationship between the two variables (mind perception and attitude) in each group. Furthermore, post-interaction interviews showed that young people assigned a master-slave relationship between the robot and experimenter, where robot should follow the experimenter's commands, whereas, elderly seemed to ascribe a sufficient amount of mind perception to apply human rules of morality to the robot. The results of this study suggest that both age groups conceptualize mind perception of a sociable humanoid robot in a different way. Furthermore, there seems to be a difference in the most immediate reaction of people and in their reflective perspectives later on. Findings of this study could be employed in the design of future robots and

HRI scenarios, where it is especially important to consider the implications of the development on individuals that are more sensitive to anthropomorphism (e.g. children and elderly).

## References

1. Banks, M.R., Banks, W.A.: The effects of group and individual animal-assisted therapy on loneliness in residents of long-term care facilities. *Anthrozoös* **18**(4), 396–408 (2005)
2. Kennedy, J., Baxter, P., Senft, E., Belpaeme, T.: Social robot tutoring for child second language learning. In: The Eleventh ACM/IEEE International Conference on Human Robot Interaction, pp. 231–238. IEEE Press (2016)
3. Turkle, S., Taggart, W., Kidd, C.D., Dasté, O.: Relational artifacts with children and elders: the complexities of cybercompanionship. *Connect. Sci.* **18**(4), 347–361 (2006)
4. Sparrow, R.: 19 can machines be people? Reflections on the turing triage test. In: Robot Ethics: The Ethical and Social Implications of Robotics, pp. 301–314 (2011)
5. Sharkey, A., Sharkey, N.: Granny and the robots: ethical issues in robot care for the elderly. *Ethics Inf. Technol.* **14**(1), 27–40 (2012)
6. Horstmann, A.C., Bock, N., Linhuber, E., Szczuka, J.M., Straßmann, C., Krämer, N.C.: Do a robot's social skills and its objection discourage interactants from switching the robot off? *PLoS ONE* **13**(7), e0201581 (2018)
7. Rosenthal-von der Pütten, A.M., Krämer, N.C., Hoffmann, L., Sobieraj, S., Eimler, S.C.: An experimental study on emotional reactions towards a robot. *Int. J. Soc. Robot.* **5**(1), 17–34 (2013)
8. Bartneck, C., Rosalia, C., Menges, R., Deckers, I.: Robot abuse—a limitation of the media equation. In: Proceedings of the Interact 2005 Workshop on Agent Abuse, Rome (2005)
9. Kahn Jr., P.H., Kanda, T., Ishiguro, H., Freier, N.G., Severson, R.L., Gill, B.T., Ruckert, J. H., Shen, S.: “Robovie, you’ll have to go into the closet now”: children’s social and moral relationships with a humanoid robot. *Dev. Psychol.* **48**(2), 303 (2012)
10. Gray, H.M., Gray, K., Wegner, D.M.: Dimensions of mind perception. *Science* **315**(5812), 619 (2007)
11. Gray, K., Young, L., Waytz, A.: Mind perception is the essence of morality. *Psychol. Inq.* **23**(2), 101–124 (2012)
12. Nass, C., Steuer, J., Tauber, E.R.: Computers are social actors. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 72–78. ACM (1994)
13. Złotowski, J., Proudfoot, D., Yogeeswaran, K., Bartneck, C.: Anthropomorphism: opportunities and challenges in human–robot interaction. *Int. J. Soc. Robot.* **7**(3), 347–360 (2015)
14. Epley, N., Waytz, A., Cacioppo, J.T.: On seeing human: a three-factor theory of anthropomorphism. *Psychol. Rev.* **114**(4), 864 (2007)
15. Heerink, M., Kröse, B., Evers, V., Wielinga, B.: Assessing acceptance of assistive social agent technology by older adults: the Almere model. *Int. J. Soc. Robot.* **2**(4), 361–375 (2010)
16. Broadbent, E., Lee, Y.I., Stafford, R.Q., Kuo, I.H., MacDonald, B.A.: Mental schemas of robots as more human-like are associated with higher blood pressure and negative emotions in a human–robot interaction. *Int. J. Soc. Robot.* **3**(3), 291 (2011)
17. Czaja, S.J., Sharit, J.: Age differences in attitudes toward computers. *J. Gerontol. Ser. B: Psychol. Sci. Soc. Sci.* **53**(5), P329–P340 (1998)

18. Stafford, R.Q., MacDonald, B.A., Jayawardena, C., Wegner, D.M., Broadbent, E.: Does the robot have a mind? Mind perception and attitudes towards robots predict use of an eldercare robot. *Int. J. Soc. Robot.* **6**(1), 17–32 (2014)
19. Broadbent, E., Tamagawa, R., Patience, A., Knock, B., Kerse, N., Day, K., MacDonald, B. A.: Attitudes towards health-care robots in a retirement village. *Australas. J. Ageing* **31**(2), 115–120 (2012)
20. Ferrari, F., Paladino, M.P., Jetten, J.: Blurring human–machine distinctions: anthropomorphic appearance in social robots as a threat to human distinctiveness. *Int. J. Soc. Robot.* **8**(2), 287–302 (2016)
21. Rubin, A.M., Perse, E.M., Powell, R.A.: Loneliness, parasocial interaction, and local television news viewing. *Hum. Commun. Res.* **12**(2), 155–180 (1985)
22. Asquith, P.J.: Of bonds and boundaries: what is the modern role of anthropomorphism in primatological studies? *Am. J. Primatol.* **73**(3), 238–244 (2011)
23. Takayama, L., Pantofaru, C.: Influences on proxemic behaviors in human-robot interaction. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5495–5502. IEEE (2009)
24. Tanibe, T., Hashimoto, T., Karasawa, K.: We perceive a mind in a robot when we help it. *PLoS ONE* **12**(7), e0180952 (2017)
25. de Graaf, M.M., Allouch, S.B., van Dijk, J.A.: Long-term acceptance of social robots in domestic environments: insights from a user’s perspective. In: 2016 AAAI Spring Symposium Series (2016)
26. Leite, I., McCoy, M., Lohani, M., Ullman, D., Salomons, N., Stokes, C., Rivers, S., Scassellati, B.: Emotional storytelling in the classroom: individual versus group interaction between children and robots. In: Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 75–82. ACM (2015)



# Velocity-Based Heuristic Evaluation for Path Planning and Vehicle Routing for Victim Assistance in Disaster Scenarios

Manuel Toscano-Moreno<sup>(✉)</sup>, Anthony Mandow, María Alcázar Martínez,  
and Alfonso García-Cerezo

Systems Engineering and Automation Department, Universidad de Málaga,  
Málaga, Spain  
[m.toscano@uma.es](mailto:m.toscano@uma.es)  
<https://www.uma.es/robotics-and-mechatronics>

**Abstract.** Natural and human-made disasters require effective victim assistance and last-mile relief supply operations with teams of ground vehicles. In these applications, digital elevation models (DEM) can provide accurate knowledge for safe vehicle motion planning but grid representation results in very large search graphs. Furthermore, travel time, which becomes a crucial cost optimization criterion, may be affected by inclination and other challenging terrain characteristics. In this paper, our goal is to evaluate a search heuristic function based on anisotropic vehicle velocity restrictions for building the cost matrix required for multi-vehicle routing on natural terrain and disaster sites. The heuristic is applied to compute the fastest travel times between every pair of matrix elements by means of a path planning algorithm. The analysis is based on a case study on the ortophotographic DEM of natural terrain with different target points, where the proposed heuristic is compared against an exhaustive search solution.

**Keywords:** Multi-robot team · Heuristics · Search and rescue · Path planning · Vehicle routing problem

## 1 Introduction

Efficient management of paths and routes for teams of autonomous off-road vehicles is crucial for challenging robotic applications such as planetary exploration [11], agriculture [4], and search and rescue (SAR) in post-disaster situations [6]. In off-road environments, the absence of pre-defined roads results in a much larger search space, and terrain characteristics, such as gradient, affect vehicle mobility in aspects such as navigability, travel time or energy consumption. In this sense, grid representations such as digital elevation models (DEM) can capture terrain knowledge for each cell that is useful for considering terrain slope navigability [9, 14], fuel consumption estimation [13], the presence of victims at risk in rescue operations [12], or weed infestation in farming environments [4].

There is a growing interest in the vehicle routing problem (VRP) for fleets of unmanned aerial vehicles (UAV) [10]. However, few works have addressed the vehicle routing problems by considering terrain elevation. In [3] a digital elevation map is considered for a team of UAVs in a terrain mapping application. In the case of ground vehicles, [11], considers the functionality constraints of an unmanned ground vehicle (UGV) where the goal is to reach a several target points of a planetary surface in combination with an UAV by minimizing the travelling distance.

Even if the VRP is intrinsically a spatial problem, some applications impose relevant temporal aspects [5]. This paper focuses on SAR operations, where selecting reliable paths is necessary to actually provide timely attention to victims. In fact, one major difference between the objective function in emergency response routing and other routing problems is that the arrival time at victims' locations is more important than the total travel time [2].

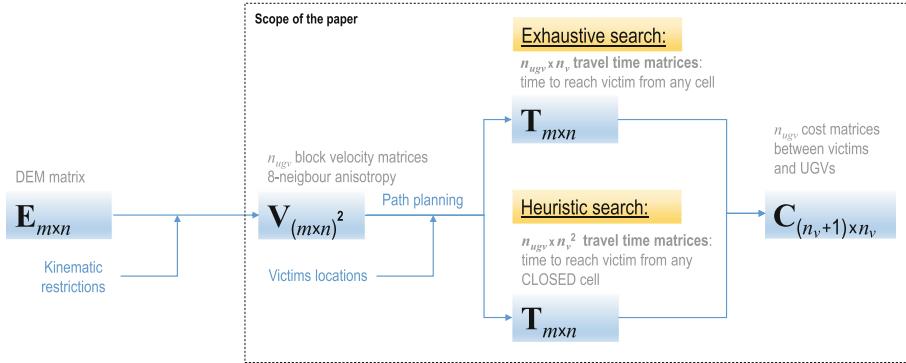
Planning node-to-node paths can be the first step in a VRP solution to build the input cost matrix. A time-aware planning can be achieved by considering the traversal times of the edges in topological search graphs [1,8]. Cells in grid representations can be used as nodes in graph-search methods derived from the Dijkstra algorithm [7], but high resolution maps may result in very large search graphs.

In this paper, our goal is to evaluate a search heuristic function based on anisotropic vehicle velocity restrictions for building the cost matrix required for multi-vehicle routing on natural terrain and disaster sites. The heuristic is applied to compute the fastest travel times between every pair of matrix elements by means of a path planning algorithm. The proposed analysis is based on a case study on an ortophotographic DEM with a set of target points, where the proposed heuristic is compared against an exhaustive search solution. Thus, the contribution of the paper is not focused on path planning or multi-vehicle routing, but on the construction of the prior cost matrices required to address these problems. In particular, the proposed algorithms allow defining a cost matrix for each UGV.

The remaining of the paper is organized as follows. Section 2 introduces the computation of travel time and cost matrices. Section 3 discusses experimental results from a case study for a distribution of victims on a DEM. Section 4 closes the paper with the conclusions.

## 2 Travel Time Matrices and Cost Matrices

This section proposes the computation of a set of travel time matrices  $\mathbf{T}$  that can be used to build cost matrices  $\mathbf{C}$  for a number of UGVs ( $n_{ugv}$ ). The outline of the approach proposed in this work is given in Fig. 1. The cost matrix for each vehicle indicates the times required to travel between any two elements of a set of  $n_v$  victims. The purpose of these  $\mathbf{C}$  matrices is to be suitable for future multi-vehicle routing solutions. First, we describe an exhaustive topological search to find  $\mathbf{T}$  by considering anisotropic behavior resulting from terrain relief and



**Fig. 1.** Outline of exhaustive and heuristic computation processes of cost matrices

kinematic restrictions. Then, we introduce a heuristic to efficiently cope with the large search space provided by the grid representation of the DEM.

## 2.1 Problem Formulation

The environment is represented by an  $n \times m$  matrix  $\mathbf{E}$  with elevation values obtained from its DEM and the resolution  $\delta$  is the distance between two contiguous cells.

Traversal velocity matrices  $\mathbf{V}$  can be built for each vehicle by considering kinematic restrictions for each cell in  $\mathbf{E}$ . Thus,  $\mathbf{V}$  is a block matrix formed by  $m \times n$  sparse submatrices of the same order.

$$\mathbf{V} \in \mathcal{M}_{m \times n}(\mathcal{M}_{m \times n}(\mathbb{R})) \quad (1)$$

Consequently, there is a submatrix for each cell in the DEM. Each submatrix represents the  $XY$  traversal velocity from the corresponding cell in the map to its 8-neighbor cells, so the rest of elements of the submatrices is always zero. Each element  $v_{\mathbf{c}_1 \mathbf{c}_2}$  represents the  $XY$  traversal velocity from the cell  $\mathbf{c}_1$  to its 8-neighbor cell  $\mathbf{c}_2$  and, therefore, null value expresses the kinematic inadmissibility of UGV for a given traversal displacement.

Travel time matrices  $\mathbf{T}$  for a given UGV are built as  $n \times m$  matrices where each element  $t_{ij}$  is the fastest (as optimized by the search algorithm) travel time from cell  $\mathbf{c} = (i, j)$  to the goal cell  $\mathbf{c}_g = (i_g, j_g)$ .

The cost matrix  $\mathbf{C}$ , for a given UGV, is an  $(n_v + 1) \times n_v$  matrix where the upper  $n_v \times n_v$  square submatrix contains the fastest travel times between victims and the last row contains the fastest travel times to each victim from the initial UGV's location.

## 2.2 Exhaustive Search Approach for Computation of Cost Matrix

Algorithm 1 describes the procedure to compute the cost matrix  $\mathbf{C}$  for a given UGV using an exhaustive search. This iterative method computes a travel time

matrix  $\mathbf{T}$  (Algorithm 2) for every goal cell and consults in  $\mathbf{T}$  the travel time associated with the locations of each other goal cells for assigning to an appropriate element of  $\mathbf{C}$  matrix.

---

**Algorithm 1.** Cost matrix  $\mathbf{C}$  through an exhaustive search

---

**Data:**  $\delta \dots$  the resolution of environment discretization  
**Data:**  $\mathbf{V} \dots$  the matrix with  $XY$  traversal velocities for a given UGV  
**Data:**  $[\mathbf{c}_{g_1} \dots \mathbf{c}_{g_{n_v}}] \dots$  the list of  $n_v$  goal cells  
**Data:**  $\mathbf{c}_s \dots$  the start cell for a given UGV  
**Result:**  $\mathbf{C} \dots$  the cost matrix for applying to VRP

```

1  foreach  $\mathbf{c}_{g_{k_2}} \in [\mathbf{c}_{g_1} \dots \mathbf{c}_{g_{n_v}}]$  do
2     $\mathbf{T} \leftarrow \text{ExhaustiveTravelTimesMatrix}(\delta, \mathbf{V}, \mathbf{c}_{g_{k_2}})$ 
3    foreach  $\mathbf{c}_{g_{k_1}} \in [\mathbf{c}_{g_1} \dots \mathbf{c}_{g_{n_v}}]$  do  $\mathbf{C}(k_1, k_2) \leftarrow t_{\mathbf{c}_{g_{k_1}}}$ 
4     $\mathbf{C}(n_v + 1, k_2) \leftarrow t_{\mathbf{c}_s}$ 
```

---

**Algorithm 2.** Exhaustive computation of the travel time matrix  $\mathbf{T}$ 


---

**Data:**  $\delta \dots$  the resolution of environment discretization  
**Data:**  $\mathbf{V} \dots$  the matrix with  $XY$  traversal velocities for a given UGV  
**Data:**  $\mathbf{c}_g \dots$  the goal cell  
**Result:**  $\mathbf{T} \dots$  the travel time matrix

**Function**  $\text{ExhaustiveTravelTimesMatrix}(\delta, \mathbf{V}, \mathbf{c}_g)$ :

```

1  forall  $\mathbf{V}_c \in \mathbf{V}$  do  $t_c \leftarrow \infty$ 
2   $\text{CLOSED} \leftarrow \emptyset$  ,  $\text{OPEN} \leftarrow \{\mathbf{c}_g\}$  ,  $t_{\mathbf{c}_g} \leftarrow 0$ 
3  repeat
4     $\mathbf{c} \leftarrow \underset{\mathbf{c} \in \text{OPEN}}{\operatorname{argmin}} \{t_c\}$ 
5     $\text{OPEN} \leftarrow \text{OPEN} \setminus \{\mathbf{c}\}$  ,  $\text{CLOSED} \leftarrow \text{CLOSED} \cup \{\mathbf{c}\}$ 
6    foreach  $\mathbf{c}_k$  so that  $v_{\mathbf{c}_k \mathbf{c}} \neq 0$  and  $\mathbf{c}_k \not\in \text{CLOSED}$  do
7       $t_{\mathbf{c}_k} \leftarrow \min \left\{ t_{\mathbf{c}_k} , t_c + \frac{\delta \cdot L_2(\mathbf{c}_k, \mathbf{c})}{v_{\mathbf{c}_k \mathbf{c}}} \right\}$ 
8       $\text{OPEN} \leftarrow \text{OPEN} \cup \{\mathbf{c}_k\}$ 
9  until  $\text{OPEN} = \emptyset$ 
```

---

### 2.3 Exhaustive Search for Computation of Travel Time Matrix

Algorithm 2, based on the strategy of well-known methods like Dijkstra, its variations (A\* or D\*) or Fast Marching, describes the procedure to compute the travel time matrix  $\mathbf{T}$  for a given UGV. This algorithm does not incorporate heuristics, so the search for fastest travel time is uninformed and exhaustive, which implies a high computational cost, especially for the case of a topological search graph based on a dense grid discretization.

In each iteration, the cells in the environment can be assigned to two sets: **OPEN** and **CLOSED**. The set **OPEN** contains the cells in the environment which have been evaluated. On the other hand, the set **CLOSED** contains the cells  $\mathbf{c}$  that have already explored by the algorithm and, therefore, whose travel times  $t_{\mathbf{c}}$  have already been computed.

The algorithm initializes: (i) the  $\mathbf{T}$  matrix to infinite values, (ii) the set **OPEN** containing the goal cell  $\mathbf{c}_g = (i_g, j_g)$ , and (iii) the travel time  $t_{\mathbf{c}_g}$  associated with the aforementioned goal cell to zero (lines 1 to 2).

The iterative method explores the cell  $\mathbf{c} = (i, j)$  belonging to set **OPEN** with the shortest value of the evaluation function, i.e. the shortest travel time  $t_{\mathbf{c}}$  (line 4), estimating the travel time from kinematically admissible cells  $\mathbf{c}_k$  - i.e. those that have a non-null XY traversal velocity  $v_{\mathbf{c}_k \mathbf{c}}$  - which have not yet been explored (line 6). In this way, the travel time is accumulated in the successive iterations by updating  $\mathbf{T}$  with the travel time that the vehicle takes from every cell until reaching the goal cell  $\mathbf{c}_g$ . For computation of cumulative travel time (line 7), the algorithm considers the distance  $\delta$  between contiguous cells in the environment, the XY traversal velocity  $v_{\mathbf{c}_k \mathbf{c}}$  associated with this displacement, and the Euclidean distance  $L_2$  between the kinematically admissible cell  $\mathbf{c}_k = (i_k, j_k)$  and the cell  $\mathbf{c} = (i, j)$  extracted from the set **OPEN** in each iteration.

When the evaluation cannot continue (line 9), the travel time matrix  $\mathbf{T}$  has been computed for all cells in the environment.

### 2.4 Heuristic Search Approach for Computation of Cost Matrix

Algorithm 3 describes the procedure to compute the cost matrix  $\mathbf{C}$  for a given UGV using a heuristic search. This iterative method computes a travel time matrix  $\mathbf{T}$  (Algorithm 4) for each pair of cells whose represent each pair of victim's locations and consults in each  $\mathbf{T}$  the travel time associated with the location of respective start cell for assigning to an appropriate element of  $\mathbf{C}$  matrix.

### 2.5 Heuristic Search for Computation of Travel Time Matrix

In order to increase the computational efficiency, a variant of the initially proposed Algorithm 2 is presented, which uses of a heuristic to estimate the travel time matrix  $\mathbf{T}$ . The use of a heuristic aims to reducing the search space in the environment, directing it towards the cell from where the navigation of UGV starts in each case. To do that, Algorithm 4 presents modifications of the Algorithm 2.

**Algorithm 3.** Cost matrix  $\mathbf{C}$  through a heuristic search

---

**Data:**  $\delta$  ... the resolution of environment discretization  
**Data:**  $\mathbf{V}$  ... the matrix with  $XY$  traversal velocities for a given UGV  
**Data:**  $[\mathbf{c}_{g_1} \dots \mathbf{c}_{g_{n_v}}]$  ... the list of  $n_v$  goal cells  
**Data:**  $\mathbf{c}_s$  ... the start cell for a given UGV  
**Result:**  $\mathbf{C}$  ... the VRP cost matrix

```

1  foreach  $\mathbf{c}_{g_{k_1}} \in [\mathbf{c}_{g_1} \dots \mathbf{c}_{g_{n_v}}]$  do
2    foreach  $\mathbf{c}_{g_{k_2}} \in [\mathbf{c}_{g_1} \dots \mathbf{c}_{g_{n_v}}]$  do
3       $\mathbf{T} \leftarrow \text{HeuristicTravelTimeMatrix}(\delta, \mathbf{V}, \mathbf{c}_{g_{k_1}}, \mathbf{c}_{g_{k_2}})$  ,  $\mathbf{C}(k_1, k_2) \leftarrow t_{\mathbf{c}_{g_{k_1}}}$ 
4       $\mathbf{T} \leftarrow \text{HeuristicTravelTimeMatrix}(\delta, \mathbf{V}, \mathbf{c}_s, \mathbf{c}_{g_{k_1}})$  ,  $\mathbf{C}(n_v + 1, k_1) \leftarrow t_{\mathbf{c}_s}$ 
```

---

**Algorithm 4.** Heuristic computation of the travel time matrix  $\mathbf{T}$ 


---

**Data:**  $\delta$  ... the resolution of environment discretization  
**Data:**  $\mathbf{V}$  ... the matrix with  $XY$  traversal velocities for a given UGV  
**Data:**  $\mathbf{c}_0$  ... the cell from where the displacement starts  
**Data:**  $\mathbf{c}_g$  ... the goal cell  
**Result:**  $\mathbf{T}$  ... the travel time matrix

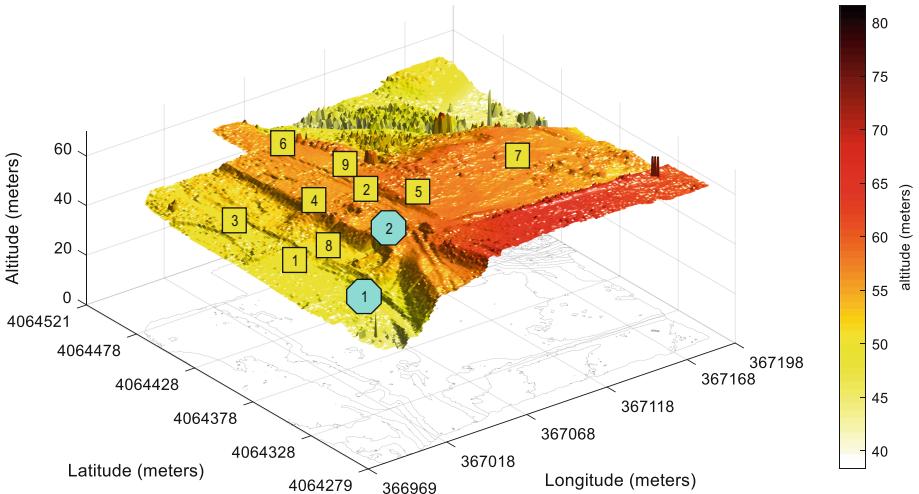
**Function**  $\text{HeuristicTravelTimesMatrix}(\delta, \mathbf{V}, \mathbf{c}_0, \mathbf{c}_g)$ :

```

1  forall  $\mathbf{V}_c \in \mathbf{V}$  do  $t_c \leftarrow \infty$  ,  $\tilde{t}_c \leftarrow \infty$ 
2   $\text{CLOSED} \leftarrow \emptyset$  ,  $\text{OPEN} \leftarrow \{\mathbf{c}_g\}$  ,  $\tilde{t}_{\mathbf{c}_g} \leftarrow 0$  ,  $h_{\mathbf{c}_g} \leftarrow 0$ 
3  while  $\text{OPEN} \neq \emptyset$  and  $\mathbf{c}_0 \notin \text{CLOSED}$  do
4     $\mathbf{c} \leftarrow \underset{\mathbf{c} \in \text{OPEN}}{\operatorname{argmin}} \{\tilde{t}_c\}$ 
5     $t_c \leftarrow \tilde{t}_c - h_c$ 
6     $\text{OPEN} \leftarrow \text{OPEN} \setminus \{\mathbf{c}\}$  ,  $\text{CLOSED} \leftarrow \text{CLOSED} \cup \{\mathbf{c}\}$ 
7    foreach  $\mathbf{c}_k$  so that  $v_{\mathbf{c}_k \mathbf{c}} \neq 0$  and  $\mathbf{c}_k \notin \text{CLOSED}$  do
8       $h_{\mathbf{c}_k} \leftarrow \frac{\delta \cdot L_2(\mathbf{c}_k, \mathbf{c}_0)}{\max\{\mathbf{V}\}}$ 
9       $\tilde{t}_{\mathbf{c}_k} \leftarrow \min \left\{ \tilde{t}_{\mathbf{c}_k} , t_c + \frac{\delta \cdot L_2(\mathbf{c}_k, \mathbf{c})}{v_{\mathbf{c}_k \mathbf{c}}} + h_{\mathbf{c}_k} \right\}$ 
10      $\text{OPEN} \leftarrow \text{OPEN} \cup \{\mathbf{c}_k\}$ 
```

---

Algorithm 4 establishes two reference cells: (i) location  $\mathbf{c}_g$  of the victim to assist, and (ii) initial location  $\mathbf{c}_0$  of the UGV from where the navigation starts. While the first proposal, presented in the Subsect. 2.3, only requires cell  $\mathbf{c}_g$  associated with the location of each victim, this second one requires a new reference cell  $\mathbf{c}_0$ . The cell  $\mathbf{c}_0$  will be different depending on the element  $\mathbf{C}(k_1, k_2)$  of cost matrix  $\mathbf{C}$  that is being estimated, representing the new location from where the UGV navigation will start after assisting each victim.



**Fig. 2.** DEM of a real environment. The elevation of the terrain is represented with different red tonalities, where darker tone indicates higher elevation value. The irregular shape in the graph’s border represents unmodeled area of the environment.

Algorithm 4 describes the heuristic procedure to compute the travel time matrix  $\mathbf{T}$  of a given UGV from an initial cell  $\mathbf{c}_0$  to a goal cell  $\mathbf{c}_g$  in the environment. As in Algorithm 2, in each iteration, the cells in the environment can be assigned to two sets: **OPEN** and **CLOSED**.

During the iterations, the method updates values  $t_c$  of  $\mathbf{T}$  (line 5) using two auxiliary matrices  $\tilde{\mathbf{T}}$  and  $\mathbf{H}$ . For each cell  $\mathbf{c}$  belonging to the set **OPEN**, the matrix  $\tilde{\mathbf{T}}$  contains the estimated travel time  $\tilde{t}_c$  from the initial cell  $\mathbf{c}_0$  to the goal cell  $\mathbf{c}_g$  as it passes through the cell  $\mathbf{c}$  (line 9). On the other hand, the matrix  $\mathbf{H}$  contains the estimated time  $h_c$  from the initial cell  $\mathbf{c}_0$  to the cell  $\mathbf{c}$  (line 8). The matrix  $\mathbf{H}$  is used as a heuristic to accelerate the convergence of the algorithm.

When the search cannot continue or the initial cell  $\mathbf{c}_0$  has been explored (line 3), the travel time matrix  $\mathbf{T}$  has been computed for a subset of cells in the environment (**CLOSED** set).

### 3 Experimental Analysis

In this section we compare the performance of the heuristic search algorithm against the uninformed exhaustive search method. The computation has been carried out using Matlab code on a 6-core Intel i7-8750H CPU 2.21 GHz. In particular, we consider a case study with two omnidirectional UGVs ( $n_{ugv} = 2$ ) and nine victims ( $n_v = 9$ ). Figure 2 shows a digital elevation model (DEM) of a real SAR experimentation environment [6] where UGVs and victims are represented as numbered hexagons and squares, respectively. The dimensions of the DEM are  $230 \times 243$  m with a resolution of one meter.

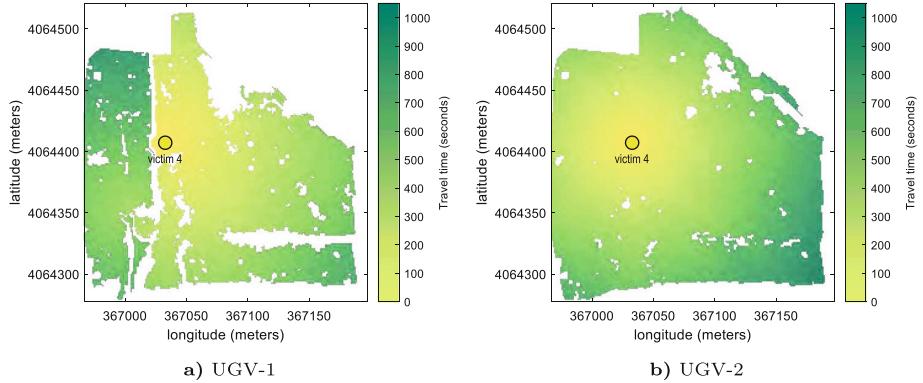
**Table 1.** UGVs' kinematic restrictions

	UGV-1	UGV-2
Safety radius	1.4 m	1.4 m
Nominal speed	0.3 m/s	0.2 m/s
Maximum navigable slope	20°	35°

**Table 2.** Resulting cost matrices  $\mathbf{C}$  for the case study (in seconds)

Victim #	1	2	3	4	5	6	7	8	9
<b>(a) <math>\mathbf{C}_1</math> for UGV-1</b>									
1	0	391	147	497	408	686	514	90	541
2	391	0	523	115	110	300	215	343	155
3	147	523	0	629	540	819	646	198	674
4	497	115	629	0	223	259	320	449	147
5	408	110	541	223	0	401	111	361	226
6	686	300	819	259	401	0	443	466	268
7	514	215	646	320	111	443	0	466	268
8	90	343	198	449	361	639	466	0	494
9	541	155	674	147	226	175	268	494	0
UGV-1	222	517	367	623	535	813	640	237	668
<b>(b) <math>\mathbf{C}_2</math> for UGV-2</b>									
1	0	360	217	303	453	678	616	135	511
2	360	0	353	151	162	450	322	267	232
3	217	353	0	207	510	462	673	297	407
4	303	151	207	0	313	388	471	288	213
5	453	162	510	313	0	602	167	340	339
6	678	450	462	388	602	0	665	675	263
7	616	322	673	471	167	665	0	504	402
8	135	267	297	288	340	675	504	0	462
9	511	232	407	213	339	263	402	462	0
UGV-2	433	215	492	364	96	657	248	305	399

The anisotropic traversal velocity matrices,  $\mathbf{V}_1$  and  $\mathbf{V}_2$ , have been defined beforehand for each UGV. These velocity matrices contain the  $XY$  traversal velocities for each cell in the environment towards each of its 8-neighbors. The algorithms proposed in this paper are independent of the number of kinematic restrictions considered to build  $\mathbf{V}$  for a given application. On an illustrative level, this case study considers a simple set of restrictions, which are given in Table 1.



**Fig. 3.** Examples of graphical representation of travel time matrix  $\mathbf{T}$  to reach victim #4 with exhaustive search

### 3.1 Cost Matrices

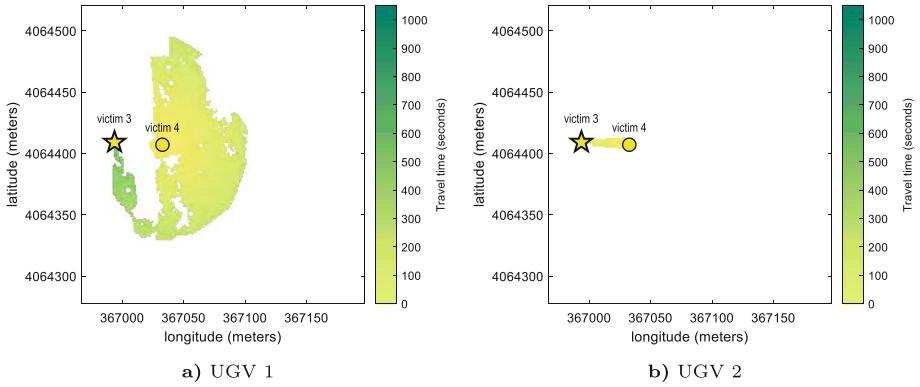
Both search methods reach the same values for the cost matrices  $\mathbf{C}_1$  and  $\mathbf{C}_2$  computed for both UGVs (see Table 2). The upper squared submatrices of  $\mathbf{C}_1$  and  $\mathbf{C}_2$  provide the fastest travel time between the corresponding pair of victim locations whereas the last row indicates the fastest time between the initial position of the UGV and the victims.

The costs between victims depend on the relative positions between them, the terrain relief and the kinematic restrictions contained in the  $\mathbf{V}$  matrices. For example, while the cost for UGV-1 between victims #3 and #4 is 629 s, for UGV-2 it is 207 s. This difference can be explained by the limitations of UGV-1 to surpass the steep slope between victim positions (see Fig. 2), which provoke a longer around path. Conversely, the faster nominal speed of UGV-1 results in a lower cost when travelling between #5 and #7 (i.e., 111 s against 167 s), where both vehicles could travel in a straight line.

### 3.2 Exhaustive Search Approach

With exhaustive search,  $\mathbf{T}$  matrices need to be computed only once for each victim/UGV pair (see Fig. 1). For each UGV,  $(n_v + 1) \times n_v$  elements of  $\mathbf{C}$  matrix are assigned looking the corresponding values up in the set of  $\mathbf{T}$  matrices. The computation times for obtaining each one of  $n_{ugv} \times n_v$   $\mathbf{T}$  matrices (18 for the study case) range between 57.5 ms and 84.8 ms. The total computational time to obtain the cost matrices for both UGVs has been 1249 ms.

Figure 3 illustrates two examples of travel time matrices  $\mathbf{T}$  computed with the exhaustive search approach. These examples correspond to the computation of  $\mathbf{T}$  to reach victim #4 with each UGV. The victim's location is represented with a yellow circle. Each cell represents the fastest travel time to reach the victim starting from that cell. Travel times are represented with different shades



**Fig. 4.** Examples of graphical representation of travel time matrix  $\mathbf{T}$  from victim #3 to reach victim #4 with heuristic search

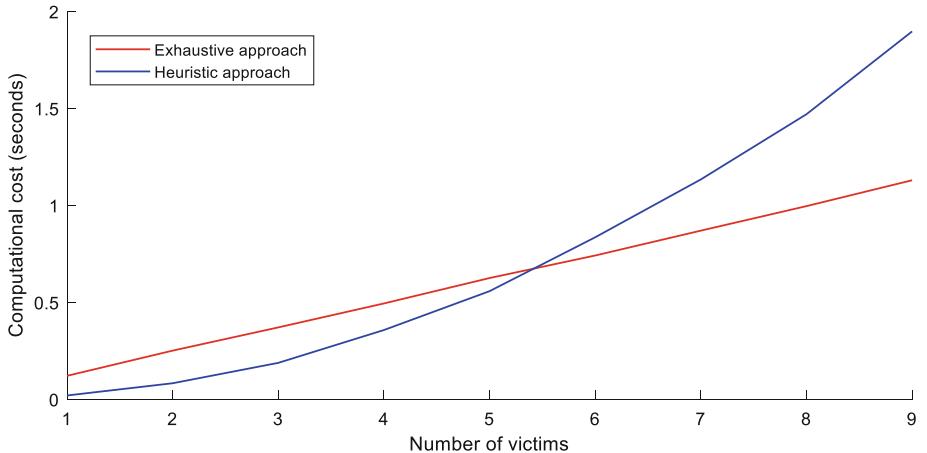
of green. The white cells are those from where the victim cannot be accessed due to UGV's kinematic restrictions (i.e., the search has not produced a solution path).

This difference can be explained by the limitations of UGV-1 to surpass the steep slope between victim positions (see Fig. 2), which requires a longer around path. Furthermore, the darkest green cells for UGV-1 (see Fig. 3a) represent a low area of the natural terrain from where the vehicle needs to travel around an unsurpassable slope to assist victim #4 (see Fig. 2). However, UGV-2 has no limitations to surmount most terrain slopes in these environment (with the exception of those in white cells) so travel times are more related to Euclidean distance (see Fig. 3b).

### 3.3 Heuristic Search Approach

With heuristic search,  $\mathbf{T}$  matrices need to be computed once for each victim pairs and UGV. The computation times for obtaining each one of  $n_{ugv} \times n_v^2$   $\mathbf{T}$  matrices (162 for the study case) range between 7.9 ms and 26.9 ms. The total computational time to obtain the cost matrices for both UGVs has been 1956 ms.

Figure 4 illustrates two examples of travel time matrices  $\mathbf{T}$  computed with the heuristic search approach. These examples correspond to the computation of  $\mathbf{T}$  to reach victim #4 from #3 with each UGV. The victim's location is represented with a yellow circle and the initial location of displacement is represented with a yellow star. Only the closed cells in Algorithm 4 have a green shape. Thus, the white area represents the cells in the environment where the search has not been done and, therefore, the fastest travel time has not been computed. The search space is reduced and the computational time to obtain each individual  $\mathbf{T}$  matrix is decreased. Victim #4 is accessible for UGV-2 with shorter travel time (207 s) than UGV-1 (629 s), which requires a longer around path due to their kinematic restrictions. Whereas, UGV-2 could travel in a straight line (see Fig. 4b).



**Fig. 5.** Average computational cost (in seconds) of exhaustive against heuristic approaches for obtaining cost matrices  $\mathbf{C}$  with two UGVs and different numbers of victims

Figure 5 addresses scalability by presenting the average computational cost (in seconds) to build the cost matrices against the number of victims. In relation to the case study presented, the graph represents the total computation time to construct the cost matrices for both UGVs and a variable number from 1 to 9 victims. For a number of victims less than 6, the heuristic search approach has a total computation time less than the exhaustive search approach because the search space reduction - i.e. reduction of the computation time of the  $\mathbf{T}$  matrices -, in Algorithm 4, compensates for the increase in the number of  $\mathbf{T}$  matrices. Conversely, for a 6 or more victims, increasing the number of  $\mathbf{T}$  matrices penalizes the total computation time.

## 4 Conclusions

In multi-robot disaster response on natural terrain, travel time is a crucial cost optimization criterion that may be affected by inclination and other challenging terrain characteristics. In this paper, we have evaluated a search heuristic function based on anisotropic vehicle velocity restrictions with the purpose of building the cost matrices required for multi-vehicle routing on natural terrain and disaster sites. The analysis has been based on a case study on the orthophotographic digital elevation model of natural terrain with different target points. The heuristic has been applied to compute the fastest travel times between every pair of matrix elements by means of a path planning algorithm.

Two alternative approaches, i.e., uninformed versus heuristic-based, have been analyzed to compute cost matrices  $\mathbf{C}$  that represent the estimated times reaching every victim from another victim as well as from the UGVs' initial location. Although use of the velocity-based heuristic function aims to reducing the

computational cost, the increasing of number of matrices to calculate causes the total computation time to compute the cost matrices to increase as the number of victims grows.

Future work will consider the definition of a VRP solution for multiple heterogeneous vehicles that will use the velocity-based cost matrices.

**Acknowledgments.** This work has received funding from the national project RTI2018-093421-B-I00 (Spanish Government), Universidad de Málaga (Andalucía Tech) and the grant BES-2016-077022 of the European Social Fund.

## References

1. Bae, J., Chung, W.: Heuristics for two depot heterogeneous unmanned vehicle path planning to minimize maximum travel cost. *Sensors* **19**(11) (2019). <https://doi.org/10.3390/s19112461>
2. Bruni, M.E., Beraldi, P., Khodaparasti, S.: A fast heuristic for routing in post-disaster humanitarian relief logistics. *Transp. Res. Procedia* **30**, 304–313 (2018). <https://doi.org/10.1016/j.trpro.2018.09.033>
3. Choi, Y., Chen, M., Choi, Y., Briceno, S., Mavris, D.: Multi-UAV trajectory optimization utilizing a NURBS-based terrain model for an aerial imaging mission. *J. Intell. Robot. Syst.: Theory Appl.* (2019). <https://doi.org/10.1007/s10846-019-01027-9>
4. Conesa-Muñoz, J., Pajares, G., Ribeiro, A.: Mix-opt: a new route operator for optimal coverage path planning for a fleet in an agricultural environment. *Expert Syst. Appl.* **54**, 364–378 (2016). <https://doi.org/10.1016/j.eswa.2015.12.047>
5. Faiad, M., Mostafa, A., Girard, A.: Vehicle routing problem instances: application to multi-UAV mission planning. In: AIAA Guidance, Navigation, and Control Conference (2010). <https://doi.org/10.2514/6.2010-8435>
6. Fernández-Lozano, J.J., Mandow, A., Martín-Guzman, M., Martín-Avila, J., Serón, J., Martínez, J.L., Gomez-Ruiz, J.A., Socarrás-Bertiz, C., Miranda-Paez, J., García-Cerezo, A.: Integration of a canine agent in a wireless sensor network for information gathering in search and rescue missions. In: IEEE International Conference on Intelligent Robots and Systems, pp. 5685–5690 (2018). <https://doi.org/10.1109/IROS.2018.8593849>
7. Garrido, S., Moreno, L., Martín, F., Álvarez, D.: Fast marching subjected to a vector field-path planning method for Mars rovers. *Expert Syst. Appl.* **78**, 334–346 (2017). <https://doi.org/10.1016/j.eswa.2017.02.019>
8. Mühlbacher, C., Gspandl, S., Reip, M., Steinbauer, G.: Adapting edge weights for optimal paths in a navigation graph. *Mech. Mach. Sci.* **49**, 372–380 (2018). [https://doi.org/10.1007/978-3-319-61276-8\\_41](https://doi.org/10.1007/978-3-319-61276-8_41)
9. Muñoz, P., R-Moreno, M.D., Castaño, B.: 3Dana: a path planning algorithm for surface robotics. *Eng. Appl. Artif. Intell.* **60**, 175–192 (2017). <https://doi.org/10.1016/j.engappai.2017.02.010>
10. Muñoz-Morera, J., Alarcon, F., Maza, I., Ollero, A.: Combining a hierarchical task network planner with a constraint satisfaction solver for assembly operations involving routing problems in a multi-robot context. *Int. J. Adv. Robot. Syst.* **15**(3) (2018). <https://doi.org/10.1177/1729881418782088>

11. Ropero, F., Muñoz, P., R-Moreno, M.: TERRA: a path planning algorithm for cooperative UGV-UAV exploration. *Eng. Appl. Artif. Intell.* **78**, 260–272 (2019). <https://doi.org/10.1016/j.engappai.2018.11.008>
12. San Juan, V., Santos, M., Andújar, J.: Intelligent UAV map generation and discrete path planning for search and rescue operations. *Complexity* **2018** (2018). <https://doi.org/10.1155/2018/6879419>
13. Gonzalez-de Soto, M., Emmi, L., Garcia, I., Gonzalez-de Santos, P.: Reducing fuel consumption in weed and pest control using robotic tractors. *Comput. Electron. Agric.* **114**, 96–113 (2015). <https://doi.org/10.1016/j.compag.2015.04.003>
14. Wang, H., Zhang, H., Wang, K., Zhang, C., Yin, C., Kang, X.: Off-road path planning based on improved ant colony algorithm. *Wirel. Pers. Commun.* **102**(2), 1705–1721 (2018). <https://doi.org/10.1007/s11277-017-5229-5>



# Emergency Landing Spot Detection for Unmanned Aerial Vehicle

Gabriel Loureiro<sup>1(✉)</sup>, Luís Soares<sup>1</sup>, André Dias<sup>1,2</sup>, and Alfredo Martins<sup>1,2</sup>

<sup>1</sup> ISEP - School of Engineering, Polytechnic Institute of Porto, Porto, Portugal  
[{1170081,1130615,apd,aom}@isep.ipp.pt](mailto:{1170081,1130615,apd,aom}@isep.ipp.pt)

<sup>2</sup> INESC TEC - INESC Technology and Science, Porto, Portugal

**Abstract.** This paper addresses the topic of emergency landing spot detection for Unmanned Aerial Vehicles (UAVs). During operation, the vehicle is susceptible to faults and must be able to predict the land spot able to ensure that the UAV will be able to land without damages and injuries to humans and structures. A method was developed, based on geometric features extracted from Light Detection And Ranging (LIDAR) data. A simulation environment was developed in order to validate the effectiveness and the robustness of the proposed method.

**Keywords:** LIDAR · Landing site detection · Emergency landing

## 1 Introduction

In recent years, the use of aerial vehicles has been growing significantly due to the applicability in different kinds of operations, such as search and rescue, delivery, surveillance, inspection and interaction with the environment. Although the accident rate is relatively small, they are susceptible to external disturbance or electromechanical malfunction. In this type of situations, UAVs must safely land in a way that will minimize damage to itself and won't cause any injury to humans.

Consequently, estimate a reliable landing spot is essential to safe operation. In order to determine a landing spot, a set of conditions must be considered when analyzing the sensor data. These conditions are typically restraints on the surface such as slope area, and roughness, and the distance that the vehicle must perform to return to the landing position. For instance, considering a delivery system, the vehicle could be affected by disturbances such as wind which may affect energy consumption. Hence, knowing a landing spot improves operation safety.

There have been various researches on landing site searching and selection over the years. LIDAR-based systems were proposed in [9, 18] and [10]. In the first two cases, geometrical features were used to detect and classify suitable landing sites in a point cloud and different techniques for real-time implementation of those algorithms were discussed. In the last case was proposed an algorithm capable of detecting safe landings spots covered in vegetation. The novelty of the

algorithm was the coupling of a volumetric occupancy map with a 3D convolutional neural network. This approach allowed to distinguish between vegetation that can be landed and objects that should be avoided.

Vision-based systems have been the more popular approach to the problem because the sensor is cheaper, smaller and lighter than active sensors like LIDARs. In [2, 5, 8] and [20] were discussed approaches for real-time processing of monocular image sequences for autonomous detection of safe landing sites. Approaches using stereo vision information were proposed in [15] and [11]. In the first case, the detection and classification of landing sites were done over the data given by a depth and a flatness map, both calculated from the information acquired by a stereo pair. In the second case, the stereo vision was used to obtain an elevation map, this information was then used to detect and classify suitable areas for landing. In this paper was also discussed waypoint path planning for autonomous landing. Other vision-based approaches, like [1, 14] and [6], involve multistage neural network classification techniques to classify candidate landing sites.

Useful literature discussing fast and efficient point cloud analysis can be found in [7, 12, 19] and [4]. In the first case, Random Sample Consensus (RANSAC) was used for decomposing a point cloud into a structure of inherent shapes and a set of remaining points. The second paper discussed the critical issues of ground filtering algorithms for LIDAR data. The third paper presents various optimizations of the 3D Hough Transform applied on plane extraction in point cloud data. Finally, the last paper was proposed a new approach for plane detection. The algorithm explores the point cloud randomly in search of planar surfaces by applying Principal Components Analysis (PCA). This approach tries to achieve a balance between high accuracy and fast performance.

Following this motivation, this paper describes novel methods able to detect, store and select emergency landing spots in operation time. Our algorithm is based on geometric analysis of the data. In addition, the probability of a spot is the best landing zone should depend on several factors as well as vary while the robot is operating.

The paper outline is as follows: in the next section, we will describe the problem our project tries to solve. In Sect. 3, we describe our proposed algorithm and the Robot Operating System (ROS) architecture, followed by simulation results in Sect. 4. Finally, in Sect. 5, we discuss the conclusions achieved and future work.

## 2 Problem Formulation

The main focus of this project is to detect and select safe landing sites during the operation. Therefore, the algorithm must be able to classify the incoming data and store the location of suitable areas.

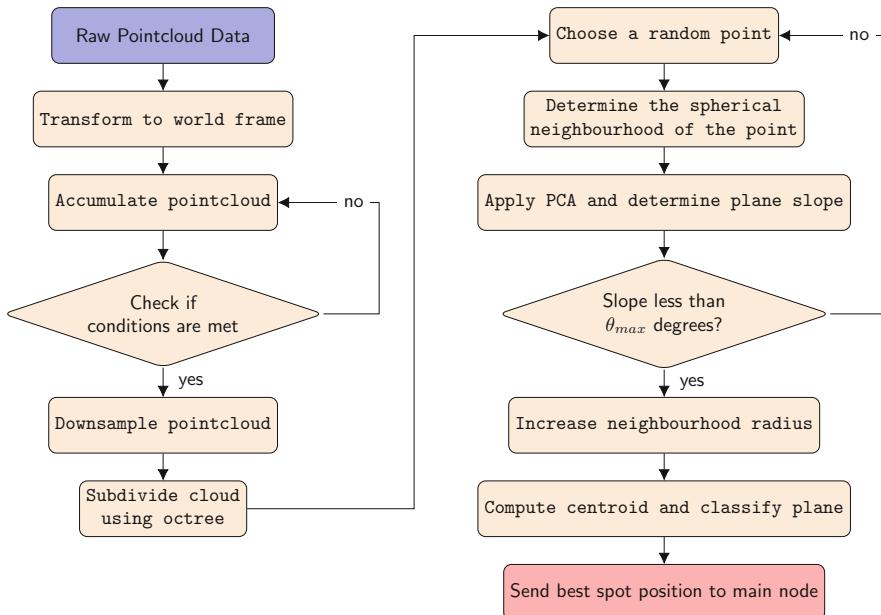
The suitability of a landing site depends on two main factors, the distance of the aircraft to the landing site and the ground conditions. The ground conditions are all the factors that are relevant when the aircraft is in contact with the ground. The conditions considered in this project are:

- The slope of the plane;
- The roughness of the area;
- The size of the spot;
- Presence of obstacles;
- Distance to vehicle.

Is important to notice that, since one of the factors is the distance of the aircraft to the landing site, all the stored areas need to be re-evaluated throughout the operation.

### 3 Proposed Algorithm

The proposed algorithm is designed to have sequential steps similar to a pipeline, i.e. a set of data processing blocks connected in series. The input to our method is a set of 3D points given in LIDAR coordinates. Figure 1 shows the framework of the algorithm. The different stages are explained in the following sections. Furthermore, the method is designed on the modular multi-threaded framework ROS [16] which presents a large community and support.



**Fig. 1.** Structure of our landing spot detection algorithm.

### 3.1 Frame Transformations

Generally, the pose of a robot is given in the inertial coordinates system. Consequently, the landing spot must be in the same coordinates. Then, the first step of the method is to transform the point cloud from LIDAR reference frame to the world reference frame as defined by Eq. 1.

$$p_W(t) = R_B^W \times R_L^B \times p_L(t) \quad (1)$$

Considering a 3D point  $p_L(t)$  in the LIDAR reference frame, at instant  $t$ . The transformation matrix given by  $R_L^B$  transforms the point to the body reference frame. Then, the  $R_B^W$  matrix multiplied by  $R_L^B p_L(t)$  results in the data expressed in the world reference frame. Finally, the point cloud registered in global coordinates is stored until it reaches the conditions to start the next step. We defined two conditions: the cloud size reached the maximum threshold or the robot traveled a sufficient distance from the origin. This step is done considering only the vehicle pose. In future projects, we would also use the vehicle velocity.

### 3.2 Cloud Downsampling

Accumulating the data until the necessary conditions are accomplished will increase computational effort. In order to obtain better performance in terms of execution time and memory consumption, it becomes necessary to perform the downsampling of the point cloud. Therefore, a Voxel Grid filter implemented in the Point Cloud Library [17] is used. The filter takes a spatial average of the points in the cloud. A set of 3D volumetric pixels (voxel) grid is generated over the cloud and the points are approximated with their centroid. This method allows to decrease the number of points of the cloud and gives a point cloud with approximately constant density. Finally, the algorithm reset the original point cloud in order to free the memory.

### 3.3 Octree and Neighbourhood Identification

The next step is to determine the neighborhood of a point to perform the plane identification method. For this reason, the point cloud is spatially structured using octree. Each internal node of the octree is subdivided into eight octants. By using a tree structure like an octree, the execution time of a search algorithm is considerably reduced [13]. There are different methods used in neighborhood identification. In our case, the algorithm finds the spherical neighborhood of a randomly chosen point. Considering a point  $p(x_p, y_p, z_p)$  in the point cloud, the neighbourhood  $N(p)$  of this point with radius  $r$  is determined by:

$$N(p) = \{\forall q : (q_x - p_x)^2 + (q_y - p_y)^2 + (q_z - p_z)^2 < r^2\} \quad (2)$$

where  $q(x_q, y_q, z_q)$  is any point in the cloud. The radius  $r$  is defined by the user as depends on sensor and environment characteristics. By applying the Eq. 2, all the points placed inside the sphere are considered as part of the neighbourhood.

### 3.4 Plane Identification

After the previous step, it is possible to calculate a planar surface given the neighborhood points. This is done satisfactorily by using the PCA algorithm. PCA applies an orthogonal transformation to map the data to a set of values called principal components. These principal components are then sorted in descending order regarding their variance. The procedure is done using the covariance matrix and its eigenvectors and eigenvalues as shown in the following equations:

$$\mathbf{C} = \frac{1}{n-1} \sum_{i=1}^n (X - \bar{X})(X - \bar{X})^T \quad (3)$$

$$\mathbf{CV} = \mathbf{VD} \quad (4)$$

where  $n$  is the number of points,  $D$  is the diagonal matrix of eigenvalues of  $\mathbf{C}$  and  $V$  is the eigenvectors matrix.

Considering the plane equation given by:

$$ax + by + cz + d = 0. \quad (5)$$

The normal vector is represented by:

$$\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (6)$$

The eigenvectors in Eq. 4 serve as the three axes of the plane while the eigenvalues indicate the square sum of points deviations along the corresponding axis. Therefore, the eigenvector with the smallest eigenvalue represents the normal vector given by Eq. 6 and the points are bounded by the other two axes.

In this perspective, the slope of the plane is examined using the normal vector. The slope is the angle between the normal vector and the vertical vector  $\hat{z} = [0 \ 0 \ 1]^T$  is computed using Eq. 7:

$$\theta = \arccos(\hat{z}^T \mathbf{n}) \quad (7)$$

Consequently, a first evaluation can be done using the plane slope. If the resulting value is greater than the maximum angle permitted for the robot, the plane is rejected. Otherwise, the neighborhood radius used in the previous section is increased and the method is repeated. By doing this procedure, we try to find regions with different sizes that can be considered a landing spot.

The Algorithm 1 describes the neighborhood and plane identification steps. The idea of the method is to detect planes for  $n_{points}$  random search points with increasing radius. In summary, only the planes that have the slope within the desired range are sent to the register stage. Table 1 describes the parameters used in the algorithm.

**Algorithm 1.** Algorithm for neighbourhood and plane identification steps.

---

**Input:** pointcloud downsampled

- 1: **for**  $j = 1$  to  $n_{points}$  **do**
- 2:   Select random point in cloud as the search point  
*Declare radius and vectors of cloud indices*
- 3:   float  $r_{min}$ , vector radiusInx  
*While the plane is accepted, do radius search*
- 4:   **while** planeBool = true **do**
- 5:     Start radius search;
- 6:     **if** ( $radiusInx \geq n_{min}$ ) **then**
- 7:       Get points inside the neighbourhood;
- 8:       Start PCA;
- 9:       Compute plane parameters;
- 10:      Compute plane slope;
- 11:      **if** slope  $\leq \theta_{max}$  **then**
- 12:        Increase radius;
- 13:      **else**
- 14:        planeBool = false
- 15:      **end if**
- 16:     **end if**
- 17:   **end while**
- 18: **end for**

---

### 3.5 Registration and Classification

Given the algorithm presented in the previous section, the next step consists in evaluate the detected planes. In this perspective, many factors are considered to decide the best landing spot. Initially, each factor is computed individually. Then, they are rated with different weights and a linear combination of the resulting rates are considered to calculate the percentage of the spot reliability. The analyzed factors are  $r_p$ ,  $\theta_p$ ,  $\sigma_p$ ,  $d_v$ , presented in Table 1. In this stage, we set a maximum standard deviation  $\sigma_{max}$  and discard planes that exceed this value.

Using this parameter, we evaluate the landing spot in terms of terrain roughness, vehicle stability when landed, obstacle clearance of a location. Besides this, the ground path to the spot is considered. Finally, the spots are stored and sorted from highest rate to smallest. In general, a spot quality depends on the robot trajectory. In this perspective, the stored spots are re-evaluated periodically.

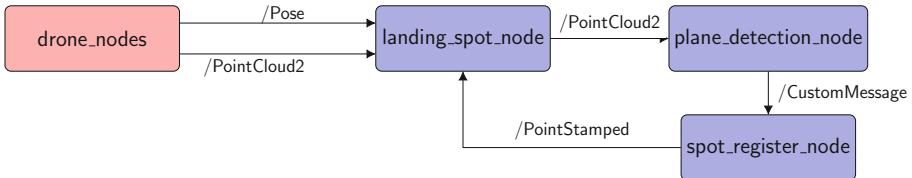
### 3.6 ROS Communication

The developed method was implemented in ROS (Robot Operating System). Figure 2 details the nodes architecture. The red block is related to the robot's nodes that publish the desired messages as Pose and PointCloud2 while the blue blocks represent the algorithm nodes.

The *landing\_spot\_node* subscribes to the robot's topics that receive a PointCloud2 and Pose messages. In addition, it is responsible to realize the frame

**Table 1.** The parameters used in the proposed algorithm. Each parameter is defined by the user.

Parameters	Description
$n_{points}$	Number of random points chosen from the cloud cluster
$n_{min}$	Minimum points used to fit a plane
$r_{max}$	Maximum radius considered for a plane
$r_{min}$	Minimum radius considered for a plane
$\theta_{max}$	Maximum slope accepted for the drone
$\sigma_{max}$	Maximum standard deviation accepted for the plane
$r_p$	Spot radius
$\theta_p$	Spot slope
$\sigma_p$	Standard deviation of the spot
$d_v$	Distance from the spot to the vehicle



**Fig. 2.** ROS Nodes of our system.

transformations and point cloud downsampling steps discussed in the previous sections and publish the resulting cloud to the *plane\_detection\_node*. In this node, the octree is generated and the PCA algorithm is used. The node is also responsible to publish the detected planes to the next stage as the custom message described in Table 2. The custom message has the data that is going to be used in the registration step. Finally, the *spot\_register\_node* receives the message, computes the plane centroid, slope, standard deviation and rate the planes. At a user-defined rate, the node sends the best landing spot as a *PointStamped* message.

**Table 2.** Message That Represents a Plane

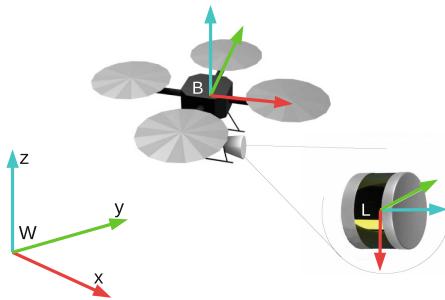
Message type	Description
<code>PointCloud2</code>	The set of points that represents the plane
<code>float32[]</code>	The plane parameters
<code>float64</code>	The plane radius

## 4 Simulation and Results

### 4.1 Setup

For simulation, purposes were chosen the Modular Open Robots Simulation Engine (MORSE) [3]. It is an open-source simulator that provides several features of interest for robotic projects. It is developed in Python and makes use of the Blender Game Engine to model and render the simulation. MORSE provides a set of standard robots, sensors, and actuators that can be interconnected to create any robot configuration. This flexibility allows the user to control the level of abstraction in a variety of simulation scenarios. MORSE also supports various middlewares, one of which is ROS.

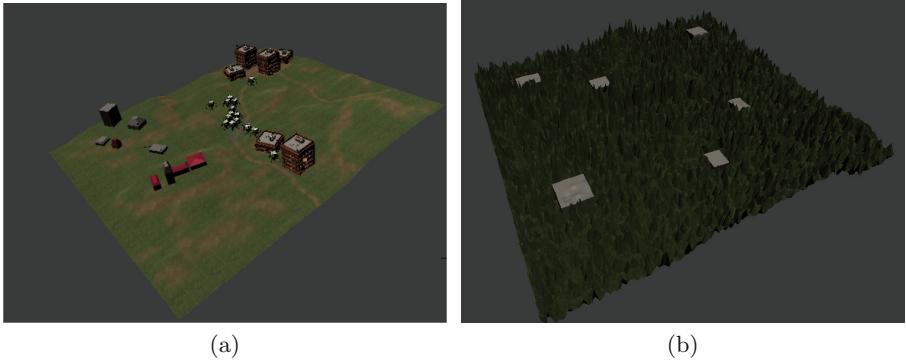
Regarding our project simulation, we considered a generic 3D laser scanner (similar to the model Puck from Velodyne LIDAR) attached to the bottom of a generic quadrotor. The configuration and reference frames of the world (W), quadrotor (B) and laser scanner (L) are illustrated in Fig. 3. The XYZ axes correspond to red, green and blue colors respectively. For the localization of the drone during the simulation, the morse's pose sensor is used. We considered two simulation scenarios, a normal one, which consists of a residential area that has houses, trees and a bunch of good landing spots, and a bad one, which tries to simulate a dense forest with small glades scatter in the map. In terms of algorithm parameters, we set the voxel size in the filter described in Sect. 3.2 to 0.05 m, the maximum slope and standard deviation were set to  $\theta_{max} = 15^\circ$  and  $\sigma_{max} = 0.20$  m, respectively.



**Fig. 3.** Simulation reference frames.

### 4.2 First Scenario

In this section, we show results for the simulation in a good scenario shown in Fig. 4(a). Figure 5(a) shows the trajectory in  $xy$  plane as the height was kept constant at 20 m. It also shows the detected spots in black and the best spots in red with 3 m radius. The performance of the plane identification algorithm



**Fig. 4.** Environments in Morse. (a) First case scenario. (b) Second case scenario.

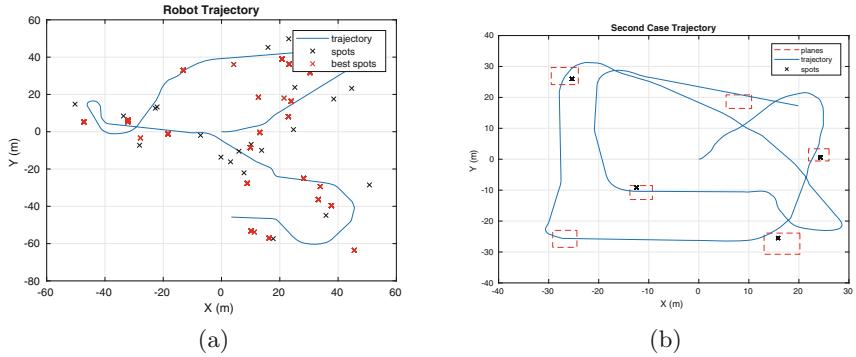
**Table 3.** Results for the simulation of the first case

Search points (#)	Radius (m)	Downsample mean time (ms)	PCA mean time (ms)	PCA total time (ms)	Percentage of operation (%)	Rejected planes	Accepted planes	Chosen spots
10	3	20.54	84.73	2796	2.80	184	146	27
20	3	19.07	171.52	5317	5.39	365	255	38
30	3	19.55	258.75	8280	8.40	547	413	62
50	3	20.00	453.33	14960	15.17	930	718	91
100	3	19.621	871.45	27015	27.39	1747	1352	169
10	4	21.50	400.96	11227	11.64	36	244	62
10	5	23.18	802.21	22546	23.37	35	245	82
10	10	22.14	4583.9	82510	85.52	24	156	7

described in Sect. 3.4 was compared with different values for the parameters described in the previous section. The simulation was performed on a computer with an Intel Core i7-6500U CPU @ 2.50 GHz with 4 cores and 8 GB RAM, with a Linux 4.15.0-50-generic Ubuntu. Table 3 shows the computation time of the downsample and PCA algorithm. The downsample technique is fast and takes about 20 ms. In contrast, the PCA algorithm is slower as the parameters are increased. Therefore, downsampling the point cloud is justified because it limits the sample size in which the PCA is applied. The tables also shows the total execution time of the PCA algorithm. Changing the radius instead of the number of search points increases the execution time at a faster rate. The percentage of operation column compares the total computation time of the PCA algorithm with the simulation time. In other words, it shows how much time the algorithm equals the total simulation time.

The difference between the accepted planes and chosen spots happens due to the maximum standard deviation allowed. Despite the high number of accepted planes, around 20% were chosen as a good spot. In this simulation, we defined the maximum standard deviation  $\sigma = 0.20$  cm. We may reject potentially good spots because we do not filter the LIDAR data to identify vegetation. In this

case, the results up to 30 search points were satisfactory. In general, the number of search points depends on each case.



**Fig. 5.** Plot for both simulations. (a) Simulation results for the first case. (b) Second case results for 20 search points.

### 4.3 Second Scenario

In order to test the algorithm in a bad scenario, we developed the environment shown in Fig. 4(b). The set consists of 6 rectangular surfaces that represent the desired landing spots surrounded by a mountain like structures. Considering the environment and the dimension of each surface, the simulations were realized for 4 different search points. Figure 5(b) presents the quadrotor trajectory and the detected landing spots given 20 search points with a maximum radius of 2 m.

Table 4 shows the simulation results. The plane detection algorithm is faster than in the first case. The main reason is the bad environment as the planes are quickly rejected which decreases the number of iterations. However, the algorithm was not able to detect all surfaces. In this scenario, one drawback of the algorithm is that if the random search point falls near a high deviation zone like the surface edges, the algorithm may discard the plane.

**Table 4.** Results for the simulation of the second case.

Search points (#)	Downsample mean time (ms)	PCA mean time (ms)	PCA total time (ms)	Percentage of operation (%)	Rejected planes	Accepted planes	Chosen spots
20	26.633	9.7667	293	0.28088	278	22	3
30	23.774	20.645	640	0.61353	566	54	5
50	23.567	52.433	1573	1.5079	1362	137	5
100	23.967	119.63	3589	3.4406	2712	288	6

## 5 Conclusion

In this paper, we presented a system that efficiently detects landing spots for unmanned aerial vehicles. The main characteristic of our system is a geometric approach to determine the suitability of potential landing areas given LIDAR range data. Furthermore, we evaluate the spot based on several constraints providing better reliability.

The algorithm was evaluated in different simulated environments. In the first simulation, we present results in an urban area. The system was able to detect many landing spots. The results for the computation time were presented for different parameters. It was observed that given certain values for the parameters, the results were adequate. Conversely, the algorithm was not able to identify all the spots for the second case. We demonstrated that in a bad environment i.e., with few landing spots, the system is able to detect some of them, but it is necessary to reduce the rigidity of the parameters.

The next step of our project is to perform experiments with different vehicles, verifying the reliability of our algorithm in real systems. Consequently, we want to evaluate the effect of weights in the registration step, so that we can optimize the process. Moreover, we plan to compute the trajectory to the landing spot and use this information as another weighting factor to qualify the spot. Finally, we plan to consider the robot velocity to improve the algorithm.

## References

1. Aziz, S., Faheem, R.M., Bashir, M., Khalid, A., Yasin, A.: Unmanned aerial vehicle emergency landing site identification system using machine vision. *J. Image Graph.* **4**(1), 36–41 (2016)
2. Bosch, S., Lacroix, S., Caballero, F.: Autonomous detection of safe landing areas for an UAV from monocular images. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5522–5527. IEEE (2006)
3. Echeverria, G., Lassabe, N., Degroote, A., Lemaignan, S.: Modular open robots simulation engine: morse. In: 2011 IEEE International Conference on Robotics and Automation, pp. 46–51. Citeseer (2011)
4. El-Sayed, E., Abdel-Kader, R.F., Nashaat, H., Marei, M.: Plane detection in 3D point cloud using octree-balanced density down-sampling and iterative adaptive plane extraction. *IET Image Proc.* **12**(9), 1595–1605 (2018)
5. Garcia-Pardo, P.J., Sukhatme, G.S., Montgomery, J.F.: Towards vision-based safe landing for an autonomous helicopter. *Robot. Auton. Syst.* **38**(1), 19–29 (2002)
6. Guo, X., Denman, S., Fookes, C., Mejias, L., Sridharan, S.: Automatic UAV forced landing site detection using machine learning. In: 2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–7. IEEE (2014)
7. Hulik, R., Spanel, M., Smrz, P., Materna, Z.: Continuous plane detection in point-cloud data based on 3D hough transform. *J. Vis. Commun. Image Represent.* **25**(1), 86–97 (2014)

8. Johnson, A., Montgomery, J., Matthies, L.: Vision guided landing of an autonomous helicopter in hazardous terrain. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 3966–3971. IEEE (2005)
9. Lorenzo, O.G., Martínez, J., Vilariño, D.L., Pena, T.F., Cabaleiro, J.C., Rivera, F.F.: Landing sites detection using lidar data on manycore systems. *J. Supercomput.* **73**(1), 557–575 (2017)
10. Maturana, D., Scherer, S.: 3D convolutional neural networks for landing zone detection from LiDAR. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3471–3478. IEEE (2015)
11. Meingast, M., Geyer, C., Sastry, S.: Vision based terrain recovery for landing unmanned aerial vehicles. In: 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No. 04CH37601), vol. 2, pp. 1670–1675. IEEE (2004)
12. Meng, X., Currit, N., Zhao, K.: Ground filtering algorithms for airborne LiDAR data: a review of critical issues. *Remote Sens.* **2**(3), 833–860 (2010)
13. Mosa, A.S.M., Schön, B., Bertolotto, M., Laefer, D.F.: Evaluating the benefits of octree-based indexing for LiDAR data. *Photogram. Eng. Remote Sens.* **78**(9), 927–934 (2012)
14. Mukadam, K., Sinh, A., Karani, R.: Detection of landing areas for unmanned aerial vehicles. In: 2016 International Conference on Computing Communication Control and automation (ICCUBEAE), pp. 1–5. IEEE (2016)
15. Park, J., Kim, Y., Kim, S.: Landing site searching and selection algorithm development using vision system and its application to quadrotor. *IEEE Trans. Control Syst. Technol.* **23**(2), 488–503 (2015)
16. 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, Kobe, Japan, vol. 3, p. 5 (2009)
17. Rusu, R.B., Cousins, S.: Point cloud library (PCL). In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4 (2011)
18. Scherer, S., Chamberlain, L., Singh, S.: Autonomous landing at unprepared sites by a full-scale helicopter. *Robot. Auton. Syst.* **60**(12), 1545–1562 (2012)
19. Schnabel, R., Wahl, R., Klein, R.: Efficient RANSAC for point-cloud shape detection. In: Computer Graphics Forum, vol. 26, pp. 214–226. Wiley Online Library (2007)
20. Templeton, T., Shim, D.H., Geyer, C., Sastry, S.S.: Autonomous vision-based landing and terrain mapping using an MPC-controlled unmanned rotorcraft. In: Proceedings 2007 IEEE International Conference on Robotics and Automation, pp. 1349–1356. IEEE (2007)

# **Intelligent Perception and Manipulation**



# ROS Framework for Perception and Dual-Arm Manipulation in Unstructured Environments

Delia Sepúlveda<sup>(✉)</sup>, Roemi Fernández, Eduardo Navas,  
Pablo González-de-Santos, and Manuel Armada

Centre for Automation and Robotics (UPM-CSIC),  
Spanish National Research Council, Carretera CAMPO-REAL Km 0,2,  
28500 Arganda del Rey, Madrid, Spain  
`{delia.sepulveda, roemi.fernandez, eduardo.navas,  
pablo.gonzalez, manuel.armada}@csic.es`

**Abstract.** In a near future, robotic systems are expected to be able to confront more complex tasks in challenging scenarios. In this context, intelligent perception and dual-arm robotic manipulation capabilities are crucial for improving the autonomy and reliability of these systems. This paper addresses the development of an experimental platform conceived to facilitate the design and assessment of new perception and dual-arm control algorithms in unstructured environments. The proposed testbed is composed of a dual-arm robotic configuration endowed with a visual perception system and a simulation and control platform implemented in ROS (Robot Operating System). The robotic configuration consists of two manipulator arms of 6-DOF (Kinova MICO™) with brushless DC actuators controlled directly through PID controllers, whereas the perception system is formed by a high resolution RGB camera and a Time-of-Flight camera. ROS provides an open source collection of software frameworks, which simplify the task of creating complex and robust robot behaviours across a wide variety of robotic systems. The proposed approach will enable the easy testing and debugging of new applications with zero-risk damage to the real equipment.

**Keywords:** Robot Operating System (ROS) · Dual-arm robot manipulation · Intelligent perception

## 1 Introduction

In the last decades, the number of dual-arm robotic manipulation tasks has increased considerably. In the industrial sector, dual-arm robots are a topic of worldwide interest due to the increase of flexibility that this kind of system can provide for assembly tasks [1]. The cooperation of two end-effectors can also improve significantly the efficiency of several precision agriculture applications, such as harvesting and pruning [2, 3]. In addition, they are being used to help surgeons to perform surgical tasks in a precise manner, improving surgery outcomes and reducing post-surgical complications [5]. They can also be in the surgery room performing tasks of surgical instrument sorting [4] or performing both transport and active manipulation tasks to ensure human-like

operation [6]. Moreover, most part of social robots is endowed with dual-arm manipulation capabilities, providing services and assistance to humans in their everyday activities, and increasing interaction possibilities [7].

Enhancement of dual-arm robotic manipulation in unstructured environments requires experimenting with different perception and control algorithms for finding the optimum collision-free paths to grasp and manipulate the target objects. For this reason, the goal of this work is to develop a ROS based software framework for a robotic dual-arm manipulator with an intelligent perception system. The platform is intended to accomplish manipulation tasks in unstructured environments, forming the testbed used for an easy testing and debugging of new applications with zero-risk damage to the real equipment. ROS [8] provides a unified platform independent of languages for publishing and subscribing to data streams and comprises a large collection of functionalities and applications, which facilitates the robot software development.

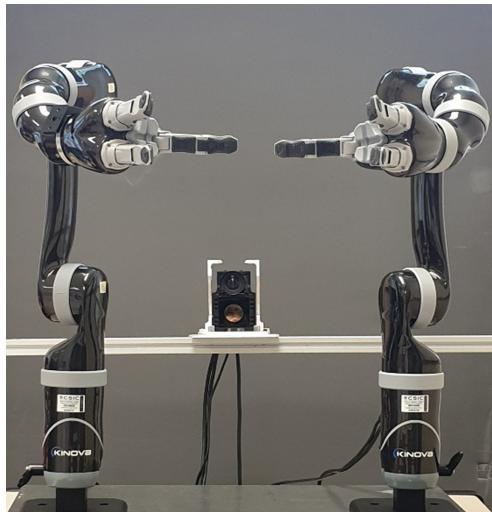
Therefore, the proposed system will provide a reliable solution to bridge the gap between the design of novel dual-arm control algorithms and the laboratory tests, contributing to boost the robustness and efficiency of future autonomous bimanual manipulators in unstructured environments.

The rest of the paper is organized as follows. In Sect. 2, the overview of the proposed testbed is provided, including the description of the hardware and software architectures. Section 3 presents the image acquisition module, whereas Sect. 4 explains the spatial localization module. In Sect. 5, the scene simulation and trajectory planning developed in MoveIt! [9] are presented. The control module of the platform is provided in Sect. 6, followed by concluding remarks and future work in Sect. 7.

## 2 Overview of the Proposed Testbed

A robotic dual-arm system and a sensory rig form the hardware part of the proposed testbed. The selected robotic arms are two Kinova MICO™ endowed with the Kinova Gripper KG-3 [10]. This gripper is under-actuated with a set of three flexible fingers. The opening and closing movements of the fingers are driven by three linear actuators, one for each finger, allowing the grasping or releasing of objects with a grip force of 40 N. The arms are lightweight, extremely portable and provide low power consumption, as well as adequate workspace and load capacity for the desired purpose [11]. Each robotic arm is composed of six interlinked segments providing 6-DOF (Degrees of Freedom). The maximum load that the robotic arm can manipulate in a continuously normal use is about 2.1 kg from minimum to middle reach (0.35 m distance between actuator #2 and the load). Considering this, as well as the software position limitations of the actuators #2 and #3, the robotic arms are arranged in the same plane with a distance between them of 0.425 m. Moreover, to obtain a good performance of the robotic arms during dual-arm manipulation tasks, they are disposed in right-handed and left-handed configurations. Hence, the grippers are close enough to perform a grip with both hands, as can be appreciated in the Fig. 1.

On the other hand, the vision system consists of two cameras; one of them (Prosilica GC2450C) provides a high-resolution colour image and the other one (Mesa SwissRanger SR4000) provides a point cloud of the scene [12].



**Fig. 1.** Dual-arm configuration and the visual perception system.

The Prosilica GC2450C is a 5.0 Megapixel colour camera with a GigE Vision compliant [13]. This camera incorporates a high quality sensor that provides superior image quality, excellent sensitivity, and low noise.

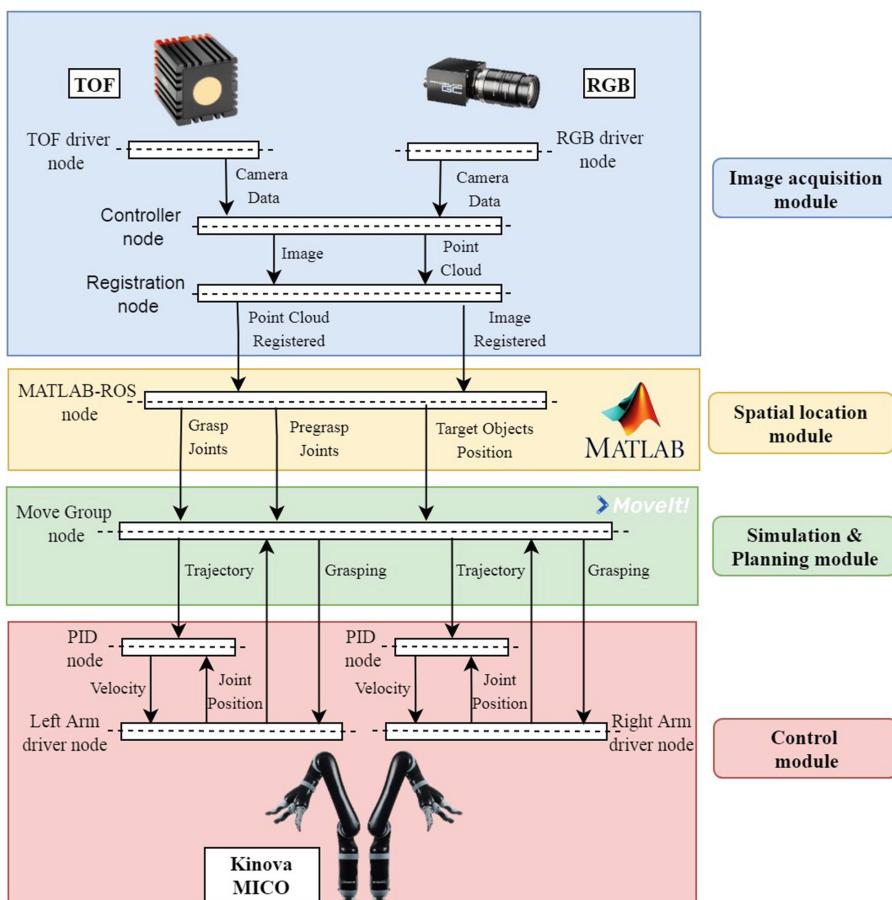
The Mesa SwissRanger SR4000 camera is a measurement device that allows capturing 3D data of infrared (IR) light reflective objects in the surrounding scene [14]. The distance measurement capability is based on the Time-of-Flight (TOF) principle. In TOF systems, the time taken for light to travel from an active illumination source to IR light reflective objects in the field of view and back to the sensor is measured. From this measure, the  $x$ ,  $y$  and  $z$  coordinates per pixel can be extracted. In nominal operation mode, an absolute accuracy of less than 0.01 m is achievable in a range of work of 10 m.

Both cameras use software trigger mode, that means that the camera waits for an acquire command and when the command is received the image capture starts. In addition, both have communication via Ethernet.

After the description of the hardware, the interaction of the software between these devices is explained by means of four main software modules. These modules are: (i) the image acquisition module; (ii) the spatial localization module; (iii) the simulation and planning module; (iv) and the control module. All these modules are organized in nodes within the ROS architecture.

The image acquisition module, constituted by the cameras API (Application Programming Interface), is responsible of extracting the images and the point cloud information of the scene through a ROS interface. These data are then processed by the spatial localization module, which is implemented by MATLAB® software. This module enables the communication with the Image Processing Toolbox™ [15] and the Robotics System Toolbox™ [16] of MATLAB®. With the help of these Toolboxes, the spatial localization module is able to identify the target objects in the scene, locate

their centroids, calculate the end-effector positions of the robotic arms, as well as the inverse kinematics (IK) of the dual-arm configuration. Target object information and joint configuration of the robotic arms are next utilized by the simulation and planning module for modelling the virtual scenario and planning the collision-free trajectories. This module has been developed with the open source software for ROS, MoveIt! [9], which incorporates the latest advances in motion planning and it is designed to work with many different types of planners. The resulting virtual environment is a reproduction of the real scene, including the different obstacles considered for calculating the trajectory between the start and the goal positions. Finally, once the trajectories are calculated and validated, the dual-arm robot executes the planned movements by means of the control module, which communicates with the DSP (Digital Signal Processor) inside the robot base.



**Fig. 2.** Communication workflow between all the system nodes.

The architecture of the proposed ROS-based system (ROS Indigo on Ubuntu 14.04, version recommended and tested by the manufacturer of Kinova MICO™) is shown in Fig. 2. At the heart of the architecture is the ROS master running on localhost, which makes it possible for nodes to find each other and exchange data. Each node has its own topics that can be used to publish or subscribe to messages [17]. A node publishes data in a common space under a topic. Other nodes can use this data simultaneously by subscribing to that topic. As shown in Fig. 2, the system has six programmed nodes:

- Two nodes within the image acquisition module for running both cameras (TOF and RGB) synchronously and registering the colour and range data in the same reference frame.
- The MATLAB-ROS node of the spatial localization module for recognizing the target objects, estimating their centroid positions, and calculating the inverse kinematics of the robotic arms.
- The Move Group node of the simulation and planning module, which is responsible for computing the necessary control inputs and sending the corresponding commands to the control module.
- Finally, the two PID nodes of the control module for running the joint of each arm according to the commanded control inputs.

### 3 Image Acquisition Module

As we mentioned above, this module is responsible of acquiring the colour and range data from the scene and consists of four ROS nodes. A ROS driver node is implemented for each camera. These driver nodes provide facilities to dynamically reconfigure camera parameters without having to restart the node, as well as services for some parameters such as the acquisition mode, the exposure time, the pixel format and the integration time. Data acquired by each camera, and status of the acquisition, are transferred by publishing messages. The sensory system controller node is in charge of controlling image acquisition. Synchronous image acquisition is achieved when the controller publishes a specific trigger message.

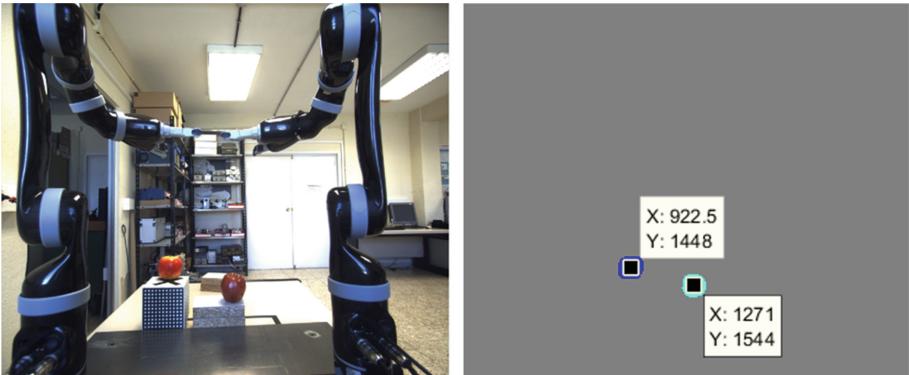
In addition, TOF camera has less pixel resolution ( $176 \times 144$ ) and a smaller field of view than the RGB camera ( $2448 \times 2050$ ). For the processing, both images need to have the same pixel resolution, in this way, the obtained pixel coordinates are in the same range in the RGB image than in the point cloud [18]. Therefore, a registration ROS node performs this task sending images with the same pixel resolution.

### 4 Spatial Localization Module

The objective of the spatial localization module is to detect the target objects in the scene, estimate the end-effector positions of the robotic arms according to these detections, and calculate the IK of the dual-arm robotic configuration.

The RGB image in conjunction with the point cloud need to be processed for obtaining the pixel coordinates of the target objects. To take advantage of the functionalities provided by the Image Processing Toolbox™ of MATLAB®, a ROS master

is created in MATLAB®, as well as a global node that is connected to the master running on localhost. This node subscribes to the topics published by the registration node of the image acquisition module, and these data are utilized as input for the detection algorithm, which is based on colour and size descriptors. Once the objects of interest are detected, their centroids are calculated. In this way, the pixel coordinates of the objects' centroids are obtained. Finally, searching these pixel coordinates in the registered point cloud, the measurement of the distances between the target objects and the cameras can be determined. An example of RGB image with centroid coordinates recognition and its corresponding point cloud can be observed in Figs. 3 and 4.



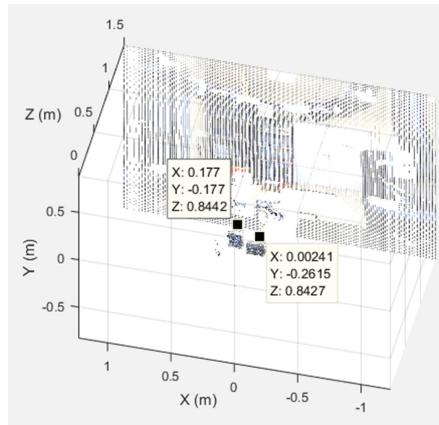
**Fig. 3.** RGB image and target detection with pixel centroid coordinates.

The coordinates of the pixels of interest and the distances from the camera to the target objects are extracted in the camera coordinate system. Consequently, a transformation from the camera coordinate system to the robot base is required as follows:

$$pixel_{coord} * T_{pixel}^{cam} * T_{cam}^{end} * T_{end}^{base} = p_{base} \quad (1)$$

As it is exposed in the Eq. 1, first, the pixel coordinates ( $x, y$ ) are transformed into the camera coordinates, using the camera projection matrix. After obtaining the planar projective coordinates, the distance in the  $z$  axis provided by the TOF camera is added. Using the camera-robot calibration proposed by Taylor [19] we obtain the matrix transformation between the camera and the end-effector robot. Finally, applying the transformation of the end-effector robot to its base, the final pose of the robotic arm is obtained.

In addition, to manipulate the target objects without colliding during the execution of the trajectory, two positions are calculated from the centroid of the target objects, the pregrasp and the grasp position. Knowing the centroid of the target objects, for calculating these positions it is necessary to add an offset, in this way, the robot arms accomplish a first approximation to the target and finally, it moves close enough to perform the grasp of the objects.



**Fig. 4.** Registered point cloud of the real scene

Once the Cartesian poses of the end-effector are calculated, it is necessary to perform the inverse kinematics transformation. The inverse kinematics solver is also implemented in MATLAB® by means of the Robotics System Toolbox™, which includes algorithms for inverse kinematics and kinematic constraints using a rigid body tree representation. In this case, the kinematic structure of the proposed manipulation system is modelled using the Denavit–Hartenberg parameters of the Kinova MICO™ robot. The solver needs the Cartesian pose and the desired orientation of the end-effector robot for obtaining the joints configuration. After the processing, the MATLAB-ROS node publishes the target objects information and the final joint configuration of the robotic arms.

## 5 Simulation and Planning Module

To perform the planning of trajectories, it is necessary to create a virtual scene, which is used to represent the world around the robot, and also to store the state of the robot itself. The planning scene is developed in MoveIt! using the primary node called Move Group. This node serves as an integrator, pulling all the individual components together to provide a set of ROS actions and services. The ROS Parameter Server is used to get three kinds of information [20]:

- Unified Robot Description Format (URDF) - which is an XML format for representing a robot model. It contains the lists of all the details of the kinematic chain. Including this information, the system allows the movement of the robotic arm avoiding collisions with itself. It is used to get the robot description of the Kinova MICO™ robotic arm.
- Semantic Robot Description Format (SRDF) - this format is intended to represent information about the robot that is not in the URDF file, but it is useful for a variety of applications. The intention is to include information that has a semantic aspect to it. SRDF file contains the most useful information on the robot: predefined

positions, planning groups, end-effector structure and finally, a list of links that are always or never in collision, in order to slim computations. In this case, a group description that represents the set of joints of both arms is included. This group allows the simultaneous movement of the robotic dual-arms and avoids collisions between them.

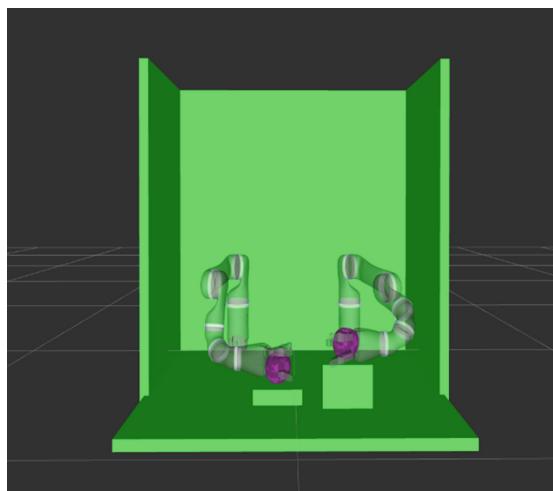
- MoveIt! configuration - to get other configuration specific to MoveIt!, including joint limits, kinematics, motion planning and perception.

To maintain this virtual scene, MoveIt! uses the planning scene monitor inside the Move Group node. The planning talks to the robot through ROS topics and actions. It communicates with the robot to get the current joints' state, in this case, listening to the publishers of the driver of both arms. This information is used to update the rendering of the virtual robot state.

In addition, the planning scene monitor listens to the world geometry information. This information is provided by the MATLAB-ROS node, which publishes the data corresponding to the Cartesian position of the target objects. In this way, the target objects are introduced in the virtual scene using the CollisionWorld objects for collision checking in MoveIt!. These elements are then considered for the motion planning. Therefore, the calculated trajectory avoids the collision with them. To maintain the virtual scene the most similar to the real world, the aspect of the target objects is introduced using their STL file description.

Other elements like walls and the floor are introduced in the virtual scene for avoiding collisions of the robotic arms with obstacles in the real world like the cameras or the table. These obstacles are created using the box standard pattern supplied by MoveIt! and providing the adequate obstacle dimensions.

Figure 5 illustrates an example manipulation scene captured during the experimental tests. As it can be appreciated in the image, in MoveIt!, the robot state can include objects carried by the robot, which are considered to be rigidly attached to the robot. Consequently, the collision checking includes these new elements in the motion plan.



**Fig. 5.** Example of a virtual scene.

MoveIt! works with motion planners from multiple libraries through a plugin interface. The default motion planners for the Move Group are configured using the Open Motion Planning Library (OMPL), an open-source motion planning library that primarily implements randomized motion planners. The OMPL generates the trajectory in response to the motion plan request using the current robot state and the goal target, but also checking collisions with the obstacles by default and including self-collisions.

There are two ways of setting the goal targets, specifying an absolute value for each joint or specifying the Cartesian poses of the two end-effectors. For the purpose of the proposed application, the angular position control mode is preferred, because otherwise, the planner could calculate any joint values to reach the desired pose. This inverse kinematics operation during the simulation can produce a delay between the executions of consecutive movements. Therefore, to obtain these joints values, MoveIt! subscribes to the topic created in MATLAB® that contains the joint configuration of the next movement.

The trajectory created by the OMPL is formed by several waypoints and each of them contains the position, velocity and acceleration for all the joints of both arms, as well as the start time of the next trajectory waypoint. The waypoint positions of the trajectory are used for the PID controllers to provide the motion execution command to the robot. The performance of the control module will be explained in the next Section.

## 6 Control Module

The system implements two PID control nodes for managing the trajectory of both arms. This type of controller continually calculates an error value  $e(t)$  as the difference between a desired point and a measured process variable, and applies a correction based on proportional, integral, and derivative terms.

In this case, the PID controller acts in each waypoint of the trajectory and for controlling each of the joints by calculating the difference between the joint position fixed in the trajectory waypoint and the current position of the robotic arm. Therefore, it is a close loop feedback mechanism.

Because certain requirements must be met between precision and velocity, the goal position offset can be fixed according to our system needs. In this case, fixing a maximum velocity of 35 rad/s, a precision of 0.005 rad of difference between the current and the waypoint position of the joints is allowed. The PID controllers parameters are the ones provided by the manufacturer, which are  $K_p = 120$ ,  $K_i = 0.2$  and  $K_d = 10$ .

The error calculated is used to adjust the velocity of the robotic arm during the execution of the trajectory. Therefore, the output information obtained in the PID controller nodes is published on the velocity topic of the robotic arms driver.

On the other hand, for controlling the grippers, it is only necessary to set the percentage of closing for each one and publish this information on the topic of each gripper driver.

## 7 Conclusion and Future Work

This paper proposes a testbed conceived to facilitate the design and assessment of new perception and control algorithms in unstructured environments. The solution includes a robotic configuration that provides dual-arm manipulation capabilities; a visual perception system for acquiring the colour and distance of the target objects enabling their localization in the coordinate space; and a software architecture consisting of four modules: (i) the image acquisition module; (ii) the spatial localization module; (iii) the simulation and planning module; (iv) and the control module. All these modules are organized in nodes implemented in ROS, which provides an open source software solution simplifying the task of creating complex and robust robot behaviours. This software architecture enables the interfacing with sensors and actuators, communication with MATLAB® and high-level modules for simulation, planning and control. Communication with MATLAB® offers several advantages like the easy modification and evaluation of algorithms, the fast processing of data, and a set of Toolboxes, which can be very useful in the development and standardization of new software modules.

Experimental results demonstrated the versatility of the proposed testbed, enabling the easy testing and debugging of new applications with zero-risk damage to the real equipment.

As future lines of this work, the implementation of a new inverse kinematics algorithm to obtain a quick solver of the joint configuration will be considered, which will allow the development of the entire system in MoveIt!. In addition, planning and control algorithms based on behaviour and perception of the environment will be designed and implemented. These novel algorithms will allow carrying out dual-arm manipulations in more challenging scenarios, with more obstacles and targeting objects.

**Acknowledgments.** The research leading to these results has received funding from:

- (i) FEDER/Ministerio de Ciencia, Innovación y Universidades – Agencia Estatal de Investigación/Proyecto ROBOCROP (DPI2017-84253-C2-1-R)
- (ii) RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, funded by “Programas de Actividades I + D en la Comunidad de Madrid” and cofunded by Structural Funds of the EU.

## References

1. Makris, S., Tsarouchi, P., Surdilovic, D., Krüger, J.: Intuitive dual arm robot programming for assembly operations. *CIRP Ann.* **63**(1), 13–16 (2014)
2. Zhao, Y., Gong, L., Liu, C., Huang, Y.: Dual-arm robot design and testing for harvesting tomato in greenhouse. *IFAC-PapersOnLine* **49**(16), 161–165 (2016)
3. Korayem, M.H., Shafei, A.M., Seidi, E.: Symbolic derivation of governing equations for dual-arm mobile manipulators used in fruit-picking and the pruning of tall trees. *Comput. Electron. Agric.* **105**, 95–102 (2014)
4. Wu, Q., Li, M., Qi, X., Hu, Y., Li, B., Zhang, J.: Coordinated control of a dual-arm robot for surgical instrument sorting tasks. *Robot. Auton. Syst.* **112**, 1–12 (2019)

5. Attia, M., Hossny, M., Nahavandi, S., Dalvand, M., Asadi, H.: Towards trusted autonomous surgical robots. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4083–4088 (2019)
6. Fleischer, H., Drews, R.R., Janson, J., Chinna Patlolla, B.R., Chu, X., Klos, M., Thurow, K.: Application of a dual-arm robot in complex sample preparation and measurement processes. *J. Lab. Autom.* **21**(5), 671–681 (2016)
7. Bustos, P., García-Varea, I., Martínez-Gómez, J., Mateos, J., Rodríguez-Ruiz, L., Sánchez, A.: Loki, a mobile manipulator for social robotics. In: Workshop of Physical Agents, pp. 1–8 (2012)
8. Quigley, M., Berkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan (2009)
9. MoveIt! Motion Planning Framework. <https://moveit.ros.org/>. Accessed 22 Apr 2019
10. Robotics company—Robotic assistive technology—Kinova. <https://www.kinovarobotics.com/en>. Accessed 22 Apr 2019
11. Campeau-Lecours, A., Lamontagne, H., Latour, S., Fauteux, P., Maheu, V., Boucher, F., L’Ecuyer, L.-J.C.: Kinova modular robot arms for service robotics applications. In: Rapid Automation. IGI Global (2019)
12. Fernández, R., Salinas, C., Montes, H., Sarria, J.: Multisensory system for fruit harvesting robots. Experimental testing in natural scenarios and with different kinds of crops. *Sensors* **14**(12), 23885–23904 (2014)
13. MESA Imaging SR4000/SR4500 User Manual Contents. [http://www.realtechsupport.org/UB/SR/range\\_finding/SR4000\\_SR4500\\_Manual.pdf](http://www.realtechsupport.org/UB/SR/range_finding/SR4000_SR4500_Manual.pdf). Accessed 22 Apr 2019
14. Prosilica GC2450—5.0 Megapixel Sony ICX625 CCD sensor - Allied Vision. <https://www.alliedvision.com/en/products/cameras/detail/ProsilicaGC/2450.html>. Accessed 22 Apr 2019
15. Image Processing Toolbox – MATLAB. <https://en.mathworks.com/products/image.html>. Accessed 22 Apr 2019
16. Robotics System Toolbox - MATLAB®; Simulink. <https://en.mathworks.com/products/robotics.html>. Accessed 22 Apr 2019
17. Martinez, A., Fernández, E.: Learning ROS for Robotics Programming. Packt Publishing, Birmingham (2013)
18. Salinas, C., Fernández, R., Montes, H., Armada, M.: A new approach for combining time-of-flight and RGB cameras based on depth-dependent planar projective transformations. *Sensors* **15**(9), 24615–24643 (2015)
19. Taylor, Z.: <http://www.zjtaylor.com/>. Accessed 22 Apr 2019
20. Concepts | MoveIt!. <https://moveit.ros.org/documentation/concepts/>. Accessed 22 Apr 2019



# Modular Dual-Arm Robot for Precision Harvesting

Eduardo Navas<sup>(✉)</sup>, Roemi Fernández, Delia Sepúlveda,  
Manuel Armada, and Pablo Gonzalez-de-Santos

Centre for Automation and Robotics (UPM-CSIC),  
Spanish National Research Council, Carretera CAMPO-REAL Km 0,2,  
28500 Arganda del Rey, Madrid, Spain

{eduardo.navas, roemi.fernandez, delia.sepulveda,  
manuel.armada, pablo.gonzalez}@csic.es

**Abstract.** Robotics for selective harvesting is a promising emergent technology for decreasing cost of labour and improving profitability in precision agriculture. In order to contribute to advance the research in this field, this paper addresses the design of a dual-arm harvesting robot. The objective of the design was to achieve a modular torso that can be adapted to different types of plants, thus being able to vary its workspace in order to optimise harvesting. The torso holds a particular dual-arm robot system with 12 DoF, but its adaptability also allows implementing other types of arms. In addition, the torso has a variable z-axis as a support for vision cameras, which can be moved along this axis to improve image acquisition.

**Keywords:** Dual-Arm robot system · Harvesting robot · Torso · Modularity and adaptability

## 1 Introduction and State of the Art

Agriculture is a strategic sector of great social, territorial, environmental and economic importance. Due to the increase of world population, which is expected to be 9,725 million in 2050, the agricultural industry must undergo a transformation process that allows it to double productivity to meet growing demand [1]. In this regard, harvesting robots, which are designed to sense the complex agricultural environment using various sensors and use that information, together with a goal, to perform target actions [2], are called to play a key role in precision agriculture.

With all this in mind, this article presents the design of a Dual-Arm Harvesting Robot. The research oriented to the development of autonomous harvesting robots in the last 30 years has been focused on certain crops, such as citrus fruits [3], apples [4], cucumbers [5], tomatoes [6], peppers [7, 8], grapes [9], strawberries [10] and others [11–18]. However, most studies choose to design the structure of the harvesting robots in a non-modular way, sometimes using tubular structures, what complicates its manufacturing, and mostly aimed at a single type of crop. For this reason, in this work, the principles of modularity and adaptability have been kept in mind, because it is well known that a modular design is a practical and feasible way to reduce the investment of

harvesting robots [19]. Therefore, the proposed design will be able to adapt to different types of plants to be harvested, and will be endowed with two modules that will facilitate this process. These modules are: (i) The dual-robotic arms support, and (ii) the vision cameras support.

The rest of the paper is organised as follows. In Sect. 2, some technical aspects of the design are explained. Section 3 addresses the analysis of the required workspace and the mechanical analysis of the proposed structure. Finally, Sect. 4 summarises major conclusions and future research directions.

## 2 The Modular Dual-Arm Robotic System

We proposed a dual-arm robotic system with adaptable body geometry and made of modular strut aluminium profiles for precision harvesting applications. The objective of this variable configuration is to endow the platform with the capability to modify its geometry according to the harvesting requirements of the different types of plants, as well as enabling its easy replication and upgrading due to the use of modular aluminium frames.

The Dual-Arm Robotic System can be divided into two distinct parts. On the one hand, there is the design of the body or torso, which incorporates the dual-arms. On the other hand, there is the support for the pan-tilt system that contains the vision cameras of the robot. This last module is integrated in the torso.

In order to design the body of the robot, the methodology presented in [15, 20] is followed. Figure 1 shows the result of adapting this methodology to our specific case, where the objective is to design a static torso.

The first step of the design procedure is to decide the robot dimensions and its configuration for the desired application workspace. In our case, part of the robot dimensions are constrained by the selected robotic arms, the Kinova Mico<sup>TM</sup>, illustrated in Fig. 2(a). These arms with 6 DoF, are lightweight, extremely portable and provide low power consumption [21].

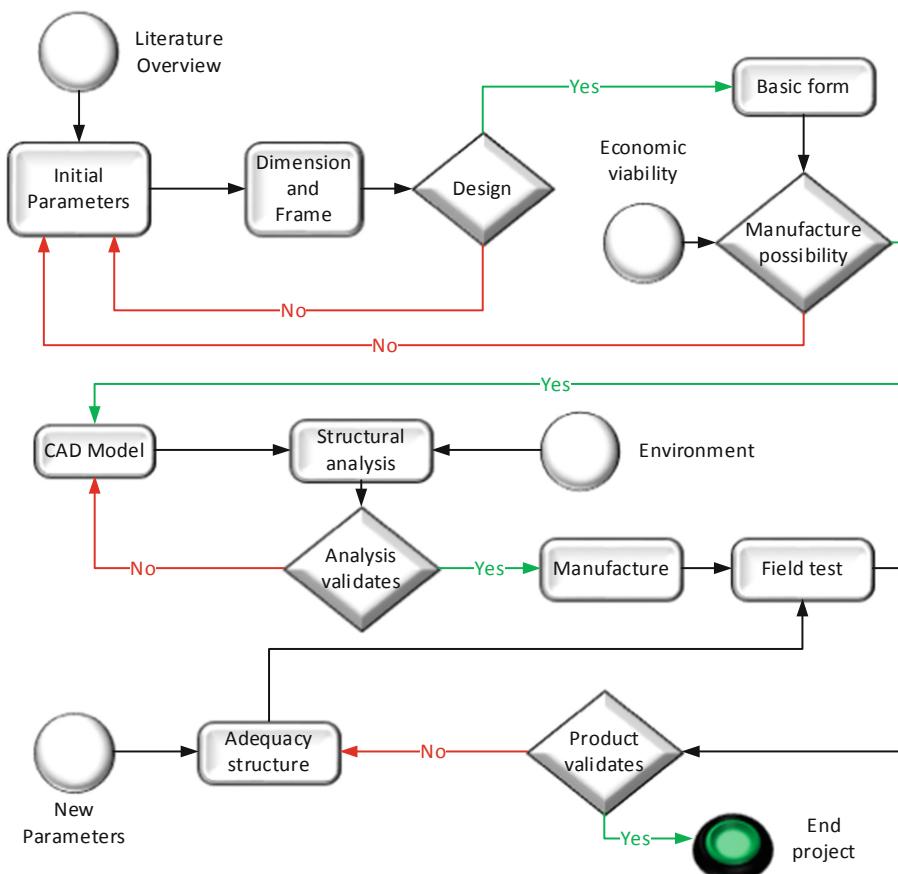
Therefore, working on that point, the main objective is to obtain a modifiable workspace that can be adapted to the particularity of the different species of plants to be harvested, maximizing the achievable volume. In particular, the main objective is to cover a  $1.20 \times 0.24 \times 1.40$  m prism, which is the size of a standard indoor hydroponic vertical planter.

The body is designed in 4 well-defined frames (see Fig. 2(b)):

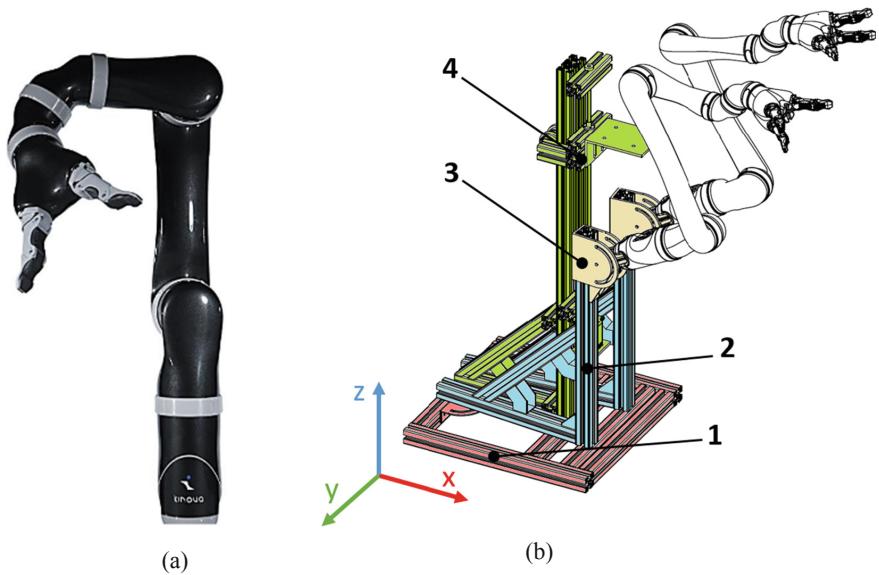
- (1) The frame that forms the rectangular base of the torso.
- (2) The frame that acts as a union between the base and the arm, and that is screwed to the first frame, so that it can configure a certain angle with it. This angle is designated as  $\alpha$ , or opening angle.

- (3) The end of each support, which joins the arm with the previous frame, and which also gives the possibility of selecting the angle that best suits the job. This angle is designated as  $\beta$ , or inclination angle.
- (4) The last frame is the one referring to the pan-tilt support. This, like the others, is independent, and is fixed to the base.

The material used for the various frames is aluminium structural profiles of Rexroth type. This type of profile is used because of its great modularity, resistance and low density. This will reduce manufacturing costs, as only cutting work will be needed and no welding or finishing work will be required, as the whole assembly will be screwed together.

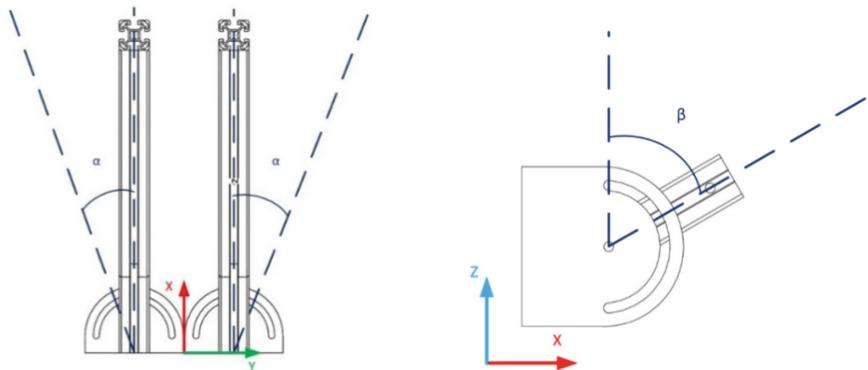


**Fig. 1.** Flowchart of the harvesting robot design. (Adapted from: Design and development of the architecture of an agricultural mobile robot [15]).



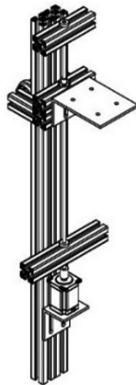
**Fig. 2.** (a) Robotic arm used for the torso design (Source: Kinova Mico<sup>TM</sup> Arm), (b) Torso Assembly. (1. Base Frame, 2. Union Base-Arm Frame, 3. End-Support Frame, 4. Z-axis Frame).

In order to perform the harvest, the geometry of the torso can be changed based on the two angles mentioned above. The first one, the azimuth angle  $\alpha$ , will affect the width where the two arms intersect, i.e., the space where the arms can work together during dual-arm manipulation. This angle is of great importance, because it is the one that regulates the work space and ensures the good execution of the harvest. The second angle, the zenith angle,  $\beta$ , will affect the height up to which both arms can reach. This angle is also critical, since an increment of this angle can result in an increase in height, but at the same time, it narrows the width of the working space. A diagram of these angles is shown in the Fig. 3.



**Fig. 3.** Scheme of the adjustable angles ( $\alpha$ ,  $\beta$ ).

Regarding the second module, the pan-tilt support, illustrated in Fig. 4, acts as a travel mechanism along the z-axis, being able to move up to 0.6 m, increasing the field of view of the cameras and improving the alignment of the cameras with the target object [22, 23]. In addition, this mechanism can remain static in order to be placed in the upper part of the robotic body.



**Fig. 4.** Z-axis frame.

### 3 Analysis and Simulations

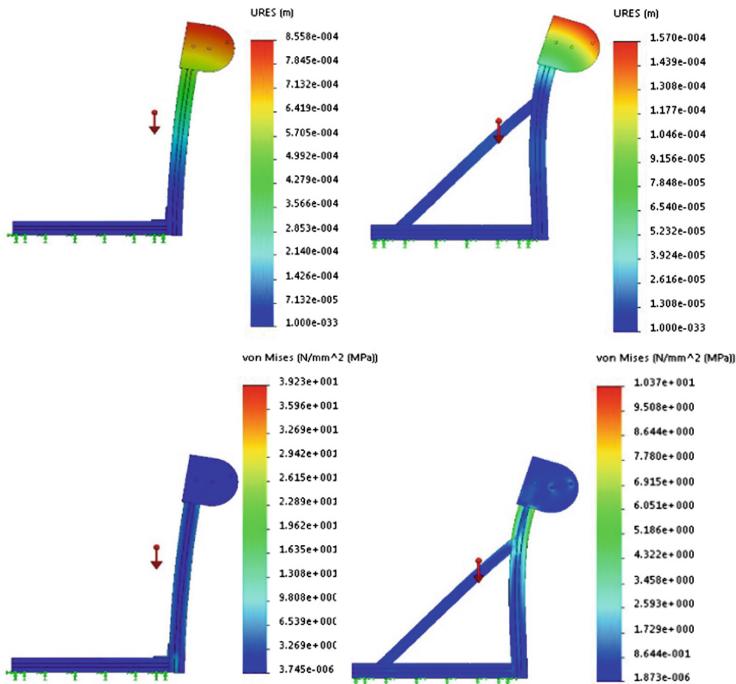
In this Section, a mechanical analysis of the proposed torso and a workspace analysis for the proposed dual-arm configuration are presented.

#### 3.1 Mechanical Analysis

As mentioned in previous sections, the material used for the torso are extruded aluminium profiles, with the following mechanical properties: Young's Modulus: 68.3 GPa; Poisson's Ratio: 0.34; Density: 2.68 g/cm<sup>3</sup>; Shear Strength: 80 MPa; Tensile Ultimate Strength: 130 MPa. The type of the screws is 12.9 grade heat-treated high tensile alloy steel, which has the following mechanical properties: 1200 MPa maximum tensile strength, 1100 MPa yield strength and 970 MPa proof load.

For the mechanical analysis of the torso, the torso shall be subjected to the load exerted on the arm, taking this as a 6 kg mass located in the centre of gravity of the arm and a mass of 0.6 kg, which is the maximum mass the robotic arm can raise, at the end of the arm. With these data, the maximum displacement obtained is 8.6e<sup>-4</sup> m and the maximum tension that acts on the screws is 39 MPa.

In order to add robustness, given that the robot will work in rough environments and with the objective that the torso can support other heavier arms, a nerve has been added. This nerve reduces the maximum displacement from 8.6e<sup>-4</sup> m to 1.6e<sup>-4</sup> m, as well as the tension exerted on the whole, from 39 MPa to 10 MPa. The simulation of the mechanical analysis is illustrated in Fig. 5.



**Fig. 5.** Mechanical analysis simulation.

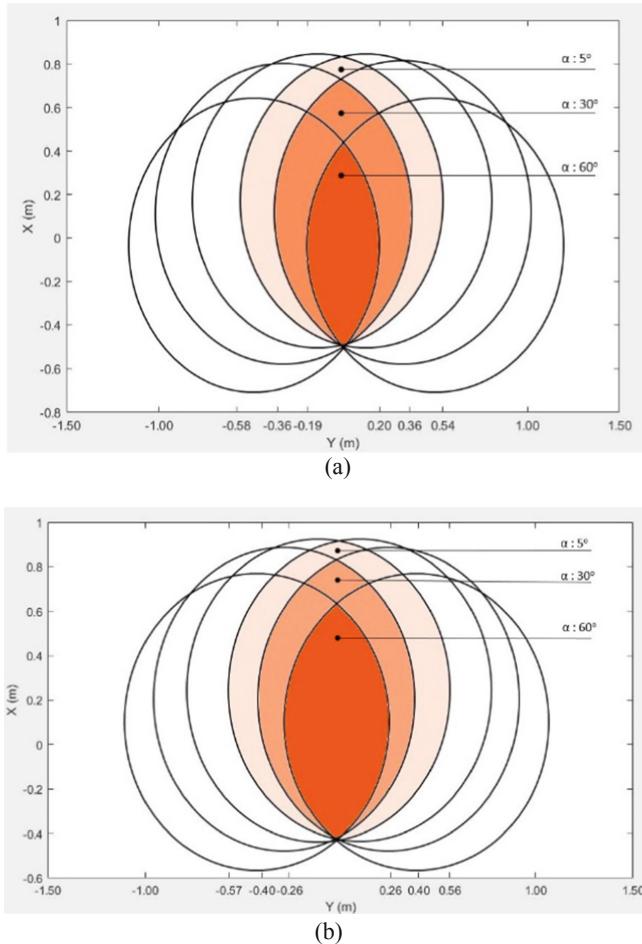
### 3.2 Workspace Analysis

Regarding the issue of determining the workspace in the harvesting of plants by autonomous robots, there exists a specific region that requires higher dexterity than other regions when considering dual-arm tasks [24]. This specific region depends on the work environment and the way the arms are going to interact with it.

In this case, the dual-arm needs to work facing the plant, where one arm helps the other to perform the harvest. This environment can be modelled as a geometrical prism, determined by the space occupied by the plant, which is intersected by two irregular spheres, referred to the dual-arms. The workspace is estimated according to the variation of the  $\alpha$  and  $\beta$  angles given by the torso design. Under the hypothesis that the useful workspace is always forward and focused on a particular zone, a series of angles were tested for the target workspace to make this hypothesis plausible. Therefore, in that way, 2 zenith angles ( $\alpha$ ) and 3 azimuth angles ( $\beta$ ) were selected.

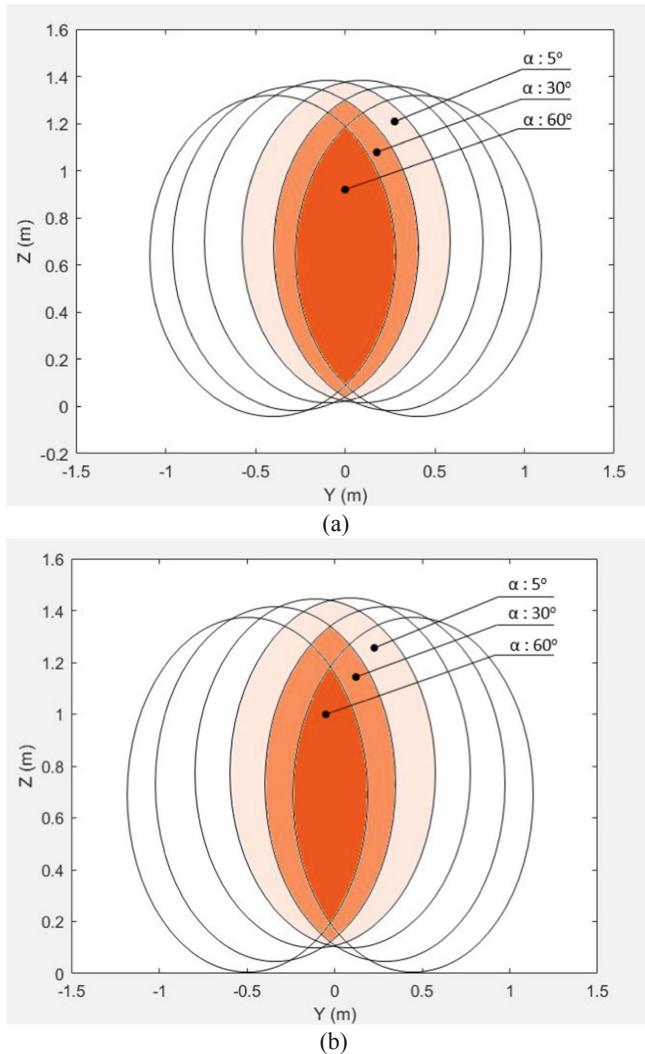
When determining these volumes, three axes must be taken into account: (i) the axis that marks the height, denoted as z axis; (ii) the axis that marks us the depth, to be known as x axis; and (iii) the axis that increases with the width, denoted as y axis. Regarding the impact of the angles on these aforementioned axes, the zenith angle ( $\alpha$ ) will affect the x and y axes to a greater extent, and the z-axis to a lesser extent, whereas the azimuth angle ( $\beta$ ) will affect to a greater extent the z-axis of the workspace. To calculate the actual workspace, the inverse kinematics problem is solved for each arm,

which will generate a cloud of points that will determine the position where each arm can reach. This cloud of points is given by a certain origin, which is placed in the first joint of each arm. As a result, the following graphics are obtained.



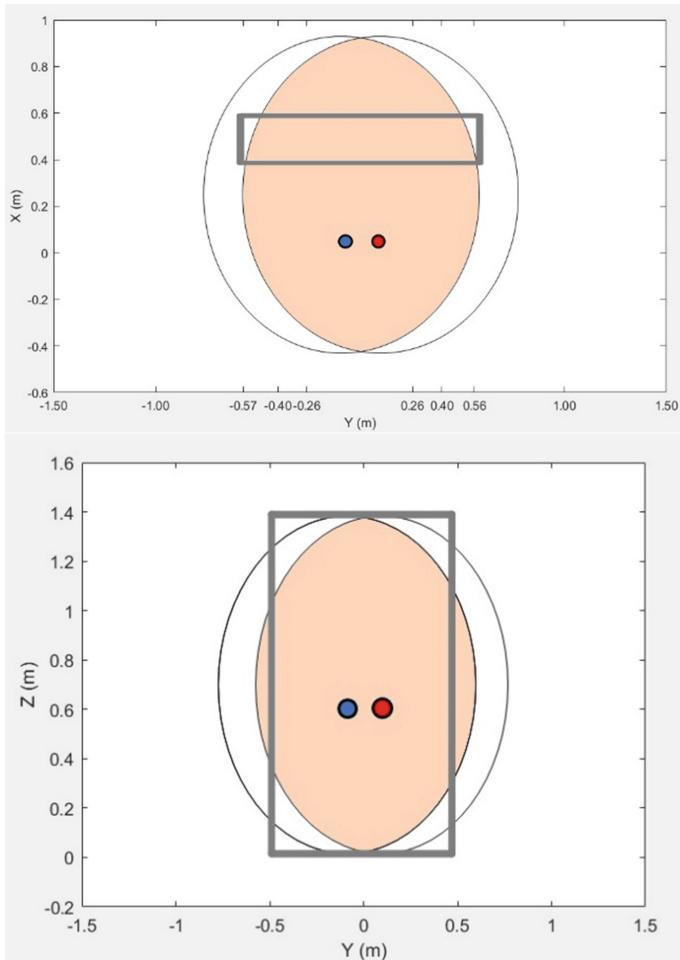
**Fig. 6.** Simulation results of the workspace. Plane X-Y. (a)  $\alpha: 5^\circ, 30^\circ, 60^\circ$ ,  $\beta: 30^\circ$ . (b)  $\alpha: 5^\circ, 30^\circ, 60^\circ$ ,  $\beta: 60^\circ$ .

As can be observed in Figs. 6 and 7, as the alpha angle is reduced (bright colour in the figures), the width of the workspace is widened from  $[-0.57 \text{ m}, 0.54 \text{ m}]$  to  $[-0.19 \text{ m}, 0.20 \text{ m}]$  for  $\beta: 30^\circ$ , and from  $[-0.57 \text{ m}, 0.56 \text{ m}]$  to  $[-0.26 \text{ m}, 0.26 \text{ m}]$  for  $\beta: 60^\circ$ . On the other hand, these figures above show that as the  $\beta$  angle is decreased, the height of the working space shifts upwards from 1.37 m for  $\beta: 60^\circ$  to 1.44 m for  $\beta: 30^\circ$ . Finally, as the beta increases, the depth also goes up from 0.84 m to 0.91 m for beta  $30^\circ$  and  $60^\circ$  respectively.



**Fig. 7.** Simulation results of the workspace. Plan Z-Y. (a)  $\alpha: 5^\circ, 30^\circ, 60^\circ$ ,  $\beta: 30^\circ$ . (b)  $\alpha: 5^\circ, 30^\circ, 60^\circ$ ,  $\beta: 60^\circ$ .

Taking into account these results,  $\alpha = 5^\circ$  and  $\beta = 60^\circ$  are selected, as they are the best suited to the objective standard indoor hydroponic vertical planter (see Fig. 8). It can be observed, that although an increase of  $30^\circ$  in  $\alpha$  noticeably changes the work space, an increase of  $\beta = 30^\circ$  is only minimally noticeable in terms of height and depth.



**Fig. 8.** Selected angles,  $\alpha: 5^\circ$ ,  $\beta: 60^\circ$ , with adjustment to the target standard indoor hydroponic vertical planter ( $1.20 \times 0.24 \times 1.40$  m prism).

#### 4 Conclusions and Future Work

In this article, the design of a dual-arm harvesting robot has been presented. The proposed configuration is modular and easily adaptable for harvesting different types of plants. Adaptability is provided by the possibility of modifying its geometry based on two angles, the zenith and azimuth angles, which affect the position of the shoulder of each arm. Modularity is afforded by the structure used, based on standardized structural aluminium profiles.

On the other hand, the workspace analysis carried out shows the feasibility and effectiveness of the proposed dual-arm robotic configuration to perform the required selective harvesting tasks.

With regard to future work, it is of interest for this project to offer mobility to the upper part of the robot. In addition, a fairing that adapts to the change in the geometry of the robot and that can be visually and technically integrated into it is required.

**Acknowledgments.** The research leading to these results has received funding from:

- (i) FEDER/Ministerio de Ciencia, Innovación y Universidades – Agencia Estatal de Investigación/Proyecto ROBOCROP (DPI2017-84253-C2-1-R).
- (ii) RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, funded by “Programas de Actividades I+D en la Comunidad de Madrid” and cofunded by Structural Funds of the EU.

## References

1. United Nations Department of Economic and Social Affairs. World Population Prospects: The 2015 Revision, Key Findings and Advance Tables (2015)
2. Yael, E., Gaines, E.M.: Systems engineering of agricultural robot design. *IEEE Trans. Syst. Man Cybern.* **24**, 1259–1265 (1994)
3. Mehta, S.S., Burks, T.F.: Vision-based control of robotic manipulator for citrus harvesting. *Comput. Electron. Agric.* **102**, 146–158 (2014)
4. Davidson, J.R., Silwal, A., Hohimer, C.J., Karkee, M., Mo, C., Zhang, Q.: Proof-of-concept of a robotic apple harvester. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 634–639 (2016)
5. Van Henten, E.J., Hemming, J., Van Tuijl, B.A.J., Kornet, J.G., Meuleman, J., Bontsema, J., Van Os, E.A.: An autonomous robot for harvesting cucumbers in greenhouses. *Auton. Robots* **13**, 241–258 (2002)
6. Li, Z., Li, P., Yang, H., Wang, Y.: Stability tests of two-finger tomato grasping for harvesting robots. *Biosys. Eng.* **116**, 163–170 (2013)
7. Bac, C.W., Hemming, J., van Tuijl, B.A.J., Barth, R., Wais, E., van Henten, E.J.: Performance evaluation of a harvesting robot for sweet pepper. *J. Field Robot.* **34**(6), 1123–1139 (2017)
8. Lehnert, C., English, A., McCool, C., Tow, A., Perez, T.: Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robot. Autom. Lett.* **2**(2), 872–879 (2017)
9. Cubero, S., Diago, M.P., Blasco, J., Tardáguila, J., Millán, B., Aleixos, N.: A new method for pedicel/peduncle detection and size assessment of grapevine berries and other fruits by image analysis. *Biosys. Eng.* **117**, 62–72 (2014)
10. Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., Kurita, M.: Evaluation of a strawberry-harvesting robot in a field test. *Biosys. Eng.* **105**(2), 160–171 (2010)
11. Zion, B., Mann, M., Levin, D., Shilo, A., Rubinstein, D., Shmulevich, I.: Harvest-order planning for a multiarm robotic harvester. *Comput. Electron. Agric.* **103**, 75–81 (2014)
12. Bak, T., Jakobsen, H.: Agricultural Robotic Platform with Four Wheel Steering for Weed Detection. *Biosys. Eng.* **87**, 125–136 (2004)
13. Baerveldt, A.: An agricultural mobile robot with vision-based perception for mechanical weed control. *Auton. Robots* **13**, 21–35 (2002)
14. Zhao, Y., Gong, L., Liu, C., Huang, Y.: Dual-arm robot design and testing for harvesting tomato in greenhouse. *IFAC-PapersOnLine* **49**(16), 161–165 (2016)

15. Tabile, R.A., Godoy, E.P., Pereira, R.R.D., Tangerino, G.T., Porto, A.J.V., Inamasu, R.Y.: Design and development of the architecture of an agricultural mobile robot. *Engharia Agrícola* **31**, 130–142 (2011)
16. Agostini, A., Alenyà, G., Fischbach, A., Scharr, H., Wörgötter, F., Torras, C.: A cognitive architecture for automatic gardening. *Comput. Electron. Agric.* **138**, 69–79 (2017)
17. Font, D., Pallejà, T., Tresanchez, M., Runcan, D., Moreno, J., Martínez, D., Teixidó, M., Palacín, J.: A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm. *Sensors* **14**(7), 11557–11579 (2014)
18. Blanes, C., Mellado, M., Ortiz, C., Valera, A.: Review. Technologies for robot grippers in pick and place operations for fresh fruits and vegetables. *Span. J. Agric. Res.* **9**(4), 1130–1141 (2011)
19. Hwan, H., Kim, C.S., Park, D.Y.: Development of multi-functional tele-operative modular robotic system for watermelon cultivation in greenhouse. In: Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2003, Kobe, Japan, vol. 2, pp. 1344–1349 (2003)
20. Madsen, T.: Mobile robot for weeding. Unpublished MSc. thesis Danish Technical University (2001)
21. Campeau-Lecours, A., Lamontagne, H., Latour, S., Fauteux, P., Maheu, V., Boucher, F., Deguire, C., L'Ecuyer, L.-J.C.: Kinova modular robot arms for service robotics applications. *Int. J. Robot. Appl. Technol.* **5**, 49–71 (2017)
22. Salinas, C., Fernández, R., Montes, H., Armada, M.: A new approach for combining time-of-flight and RGB cameras based on depth-dependent planar projective transformations. *Sensors* **15**, 24615–24643 (2015)
23. Fernández, R., Salinas, C., Montes, H., Sarria, J.: Multisensory system for fruit harvesting robots. Experimental testing in natural scenarios and with different kinds of crops. *Sensors* **14**, 23885–23904 (2014)
24. Tsarouchi, P., Makris, S., Michalos, G., Stefos, M., Fourtakas, K., Kaltsoukalas, K., Kontrovakis, D., Chryssolouris, G.: Robotized assembly process using dual arm robot. *Procedia CIRP* **23**, 47–52 (2014)



# Integrating Multiple Sources of Knowledge for the Intelligent Detection of Anomalous Sensory Data in a Mobile Robot

Manuel Castellano-Quero<sup>(✉)</sup>, Juan-Antonio Fernández-Madrigal,  
and Alfonso J. García-Cerezo

Systems Engineering and Automation Department, University of Malaga,  
Campus de Teatinos, 29071 Malaga, Spain  
[{mcastellano,jafernandez,ajgarcia}@uma.es](mailto:{mcastellano,jafernandez,ajgarcia}@uma.es)

**Abstract.** For service robots to expand in everyday scenarios they must be able to identify and manage abnormal situations intelligently. In this paper we work at a basic sensor level, by dealing with raw data produced by diverse devices subjected to some negative circumstances such as adverse environmental conditions or difficult to perceive objects. We have implemented a probabilistic Bayesian inference process for deducing whether the sensors are working nominally or not, which abnormal situation occurs, and even to correct their data. Our inference system works by integrating in a rigorous and homogeneous mathematical framework multiple sources and modalities of knowledge: human expert, external information systems, application-specific and temporal. The results on a real service robot navigating in a structured mixed indoor-outdoor environment demonstrate good detection capabilities and set a promising basis for improving robustness and safety in many common service tasks.

**Keywords:** Robot sensors · Bayesian inference · Sensory data diagnosis

## 1 Introduction

Service robotics is in current expansion [1]. Blossomed almost two decades ago mainly due to the methodological advances in probabilistic management of information [2], now standard robots are capable of dealing with relevant amounts of uncertainty in the real world intrinsically and efficiently, particularly at the lowest control levels. However, dealing with uncertainty (noise) is not enough: they should also be able to identify and deal intelligently with—make decisions about—abnormal situations in order to improve their expected performance; other than computational efficiency issues, there are no conceptual reasons that prevent them to achieve that even at the most basic levels of operation.

In this paper we focus on the particular problem of using sensory data as safely and robustly as possible beyond uncertainty, i.e., when those data are heavily modified by unexpected circumstances: adverse environmental conditions, the special nature of some perceived elements in the world, or even breakdowns in the sensor hardware. For obtaining such capability in today robots, they have to resort to multiple sources of knowledge besides the ones intrinsic to their own design and operation, i.e., regarding the mentioned circumstances, the normal behaviour of their sensor devices, and their dynamics. These sources and the modalities of knowledge they provide are really diverse: they can be human experts, external information systems, previous knowledge about the environment where the robots work, etc. In spite of that diversity, all of them should be integrated and used as rigorously and as homogeneously as possible to optimise tractability, robustness and safety.

Concerning our particular problem, there exist a number of tools that could be employed for detecting abnormal sensory data: neural networks [3], fuzzy inference [4], Bayesian inference [5], ad-hoc or heuristic approaches [6], etc. However, only Bayesian inference can provide a homogeneous and mathematically rigorous foundation (based on probability and statistics [7]) for fusing knowledge coming from a number of different sources; neural networks offer no explicit explanations about their deductions nor a rigorous basis for managing uncertainty, fuzzy logic is mostly suitable for expert knowledge, and other approaches are not based on a sound mathematical foundation, thus compromising their guarantees. Bayesian networks can reason not only in one direction (from data to conclusions), but can infer knowledge about any element in the network, i.e., we can deduce either which anomaly is present or what would be the actual information perceived by a sensor if some anomaly occurs, without re-building the network. In addition, Bayesian inference can be hybridised with other paradigms [8,9] and be integrated naturally with probabilistic methods used by modern robots. The main drawback of Bayesian inference with respect to other approaches is its high computational cost. In this paper we do not cope with that problem (our current and future work is being focused and planned to advance in that direction), but some approximation inference algorithms exist that limit the amount of computation during inference [10,11], thus we claim that the Bayesian approach is a highly promising one for our goals.

Bayesian inference, although widely used in robotic estimation (for localization and mapping [2]), has been scarcely and only sporadically used for sensory diagnosis (e.g., [12]). A general Bayesian inference system has been reported in [13], but that is a preliminary work focused on the interactive construction of the network rather than in its capabilities, and has been used only in particular, static robot situations, being, in consequence, only limitedly tested. In this paper we take a step further by augmenting the sources of knowledge that can be integrated and exploited by the network, particularly through the addition of sequential filters to its output data, concretely a kind of infinite impulse response [14] for probability estimation and a robust moving window median filter for value estimation [15]—a much more computationally efficient approach

than using a dynamic Bayesian network, which would increase the cost of inference exponentially. This allows us to include temporal information, a source of knowledge of great relevance in a deduction process that has necessarily to deal with the dynamics of the robot-environment interaction. We have also tested our system along a complete navigation route of the mobile robot in a mixed indoor-outdoor scenario where the robot encounters some sensory issues, not only in particular, static situations, thus providing a richer study of the anomaly detection and sensory data recovering during the robot operation in a real application.

Our current implementation fuses geographical knowledge (the location of the robot), meteorological knowledge (about the weather in the season at that location), expert knowledge (of a human, concerning the possible anomalies that could occur in the sensory devices), environmental knowledge (some characteristics of the scenario and its elements), and temporal knowledge (about how the last sensory and estimation data affects the present). In principle, the number and diversity of knowledge sources are only constrained by the complexity of the resulting Bayesian network.

In our experiments the robot has used both very basic sensors (wheel encoders, bumpers, etc.) and more complex ones (laser rangefinders and RGB-D cameras). It has dealt with abnormal sensory data produced by the presence of dark-surface objects (difficult to perceive by range sensors that emit infrared radiation), thin objects (easy to be missed between consecutive beams), and adverse environmental conditions (excessive light when the robot goes outside, that affect visual devices). Thanks to the sequential filter, the dynamics of the detection process has been coped with adequately in all cases. All in all, our results show that with all this information, an operational robot can infer successfully whether its sensors are working nominally or some anomaly is likely to be present, and even correct the data coming from a doubtful sensor device with the ones from another sensor or with commonsense information.

The rest of the paper is as follows. Section 2 gives an overview of our inference system based on Bayesian networks. Section 3 details its most relevant elements for a particular case of mobile robot. Section 4 explains the inference capabilities of the system in several adverse circumstances encountered by the robot during its operation. Finally, Sect. 5 summarizes the main conclusions of this work and sets some future lines of research.

## 2 Overview of the Bayesian Inference System

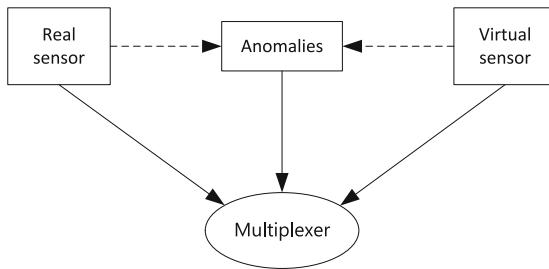
A Bayesian network defined on a set of variables  $\mathbf{V}$  is a pair  $(G, \Theta)$  consisting of a direct acyclic graph,  $G$ , over  $\mathbf{V}$ , called the *network structure*, and a set of Conditional Probability Tables (CPTs),  $\Theta$ , for each variable in  $\mathbf{V}$ , called the *network parametrization*. The graph structure captures the causal relationships between variables through directed arcs, which indicate dependence relationships, while CPTs define probability distributions over the variables. For more in-depth treatment of Bayesian networks and inference, please refer to [7, 16].

We are interested in inferring new knowledge from existing one, i.e., in deducing a probability distribution over a set of query nodes of the graph,  $\mathbf{Q}$ , given

some evidence nodes  $\mathbf{E}$ , i.e.,  $P(\mathbf{Q}|\mathbf{E})$ . This can be done by applying basic probability theory (the chain rule, the Bayes theorem, etc.) repeatedly, although that can be prohibitive even for small scale problems, so a Bayesian inference algorithm should aim to a reduced computational complexity.

There exist many inference methods for Bayesian networks, both exact, such as the well known junction tree algorithm [16], and approximate, such as loopy belief propagation and likelihood weighting [7]. In general, the former provide correct answers using more computational resources (junction tree is  $O(n \cdot \exp(w))$  for a graph with  $n$  nodes and a treewidth of  $w$ , for instance), while the latter can be more efficient at the expense of less accurate answers—their quality depends on the problem and on the allowed number of iterations, since they are often any-time algorithms. In this paper we have used the junction tree algorithm since it has provided the best trade-off between quality of the results and computational cost for the size of our networks.

Our architecture for doing inference in sensory systems uses a basic component, a so-called *Bayesian sensor*, modelled through a Bayesian network, which represents not only a real sensor but also additional information that enables intelligent diagnosis and sensory enhancement, i.e., it is the component in charge of integrating the diverse sources of knowledge concerning the sensory data production. Figure 1 depicts the structure of this generic Bayesian sensor, formed by three different subnetworks (rectangles) and a multiplexer node (ellipse) that we explain below. Multiple components of this kind can be interconnected in our complete system; nodes of interest can also be enhanced by adding temporal filters to them (not shown in that figure). These subnetworks and filters will be dealt with in more detail, and deployed into their elements, in Sect. 3.



**Fig. 1.** Generic sensor based on a Bayesian network. Dashed lines are optional. Interconnections among these Bayesian sensors are done through their multiplexers.

The *real sensor* subnetwork represents an existing sensor on board the mobile robot. This subnetwork will contain one node whose values represent the measurements of the sensor.

The *virtual sensor* subnetwork receives information from other Bayesian sensors (directly or through some calculations) in order to emulate the behaviour of its real sensor when it is faulty and to deduce its data, i.e., for recovery.

The *anomalies* subnetwork indicates whether there are faults or abnormal situations in the associated sensor. This can be deduced, for example, using information from other sensors, or integrating external knowledge (weather, location, expert, etc.).

Finally, the *multiplexer* node selects the inferred sensor measurement. If there is a high probability of abnormal behaviour, the virtual sensor will have more influence than the real one in this final result. In our current implementation, the multiplexer node is a discrete variable with the same values as the ones of the leaf nodes of the real and virtual sensor subnetworks.

Although the component depicted in Fig. 1 is enough for integrating a diversity of knowledge coming from different sources, we also include sequential filters in some nodes in order to take into account temporal knowledge (dynamics). These filters work on the posterior distributions obtained during the robot operation, which may represent either anomalies or sensory data encoded by multiplexers.

### 3 Instantiation for a Particular Mobile Robot

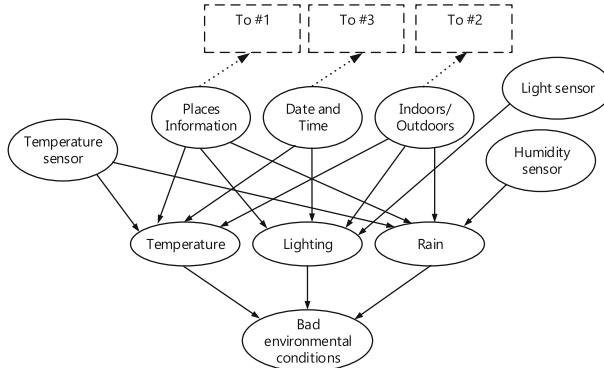
In this work we have used a *Turtlebot* robot [17] to implement the architecture defined above (see Fig. 5). It is a robot with a *Kobuki* mobile platform and a bunch of sensors: three bumpers, two magnetic encoders, three cliff detectors, a gyroscope, two wheel drop sensors, a *Kinect* RGB-D device and a *Hokuyo* 2D laser rangefinder. In our experiments we use the laser rangefinder, the RGB-D camera, the bumpers and the encoders.

For the sake of space, in this section we focus on one of the components of the whole network, the one corresponding to the laser rangefinder, since it contains enough complexity in the integration of several sources of knowledge and represents well the decisions made during the implementation of the other sensory devices in the entire system. Also notice that, from a software implementation point of view, some of the further described elements can be coded only once, since they affect several parts of the network the same way; we have considered these re-factoring issues appropriately in our software.

The real sensor subnetwork for the laser rangefinder in this robot contains as many nodes as elements we wish to represent from the vector of measurements (beams). Each node is a discrete random variable with a suitable discretization of the measured distances. The multiplexer node is replicated for each laser beam and it represents the final probability distributions we want to get. The structure is trivial from Fig. 1.

In the corresponding anomalies subnetwork we integrate environmental information (weather, location) as well as data from other sensors to detect abnormal situations (see Fig. 2). This is done in two complementary ways: through the connectivity (dependences) among nodes in the network shown in the figure and by filling their CPTs with suitable commonsense and expert knowledge regarding all that information. We omit here the content of the CPTs for their very large size; in short, the knowledge for filling them has been translated into imperative programming, and thus appropriate routines have been coded with it. For

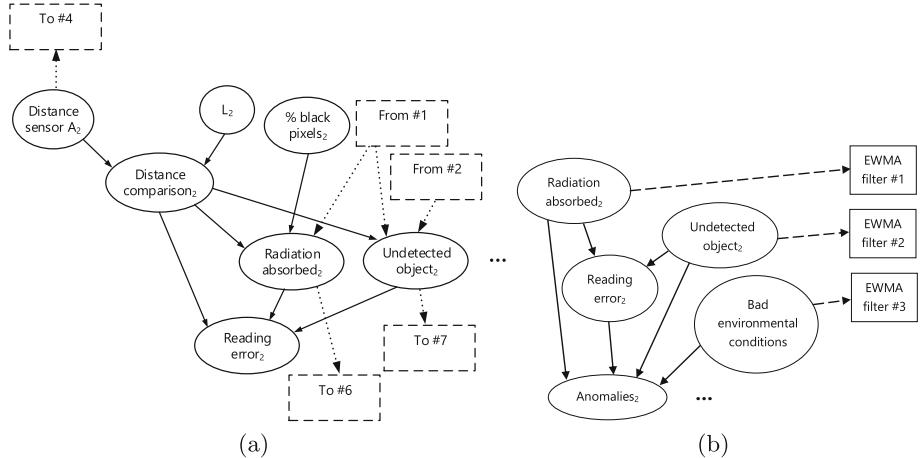
instance, bad lighting in indoors can produce wrong measurements, as well as the presence of rain outdoors and also extremely high temperatures. Environmental anomalies in this device are considered to affect all laser beams equally.



**Fig. 2.** Abnormal environmental conditions detection in the anomalies subnetwork of the laser rangefinder Bayesian network. The CPTs associated to edges in the graph code the integration of the different sources of knowledge.

There also exists systematic errors that can affect a laser rangefinder, such as the detection of too thin, transparent or black objects. For this part of its anomalies subnetwork we use an alternative distance sensor (the depth information provided by the *Kinect* camera), a RGB image and some nodes of the environmental part (see Fig. 3(a)). The actual values of the laser rangefinder are compared with the ones from the alternative distance sensor; if there is a significant difference, then this anomaly is detected—we are here integrating expert knowledge about the sensors. We can also distinguish the absorbed radiation anomaly (black objects) by combining the information from the place where the robot is and the percentage of black pixels in the RGB image. The undetected object anomaly (too thin or transparent objects in the path of the laser beams) takes into account the sense of the difference between the laser and the alternative sensors, and also information known about the working place; for instance, if we are known to be in a place with transparent objects, e.g., windows, and one or more alternative sensors indicate shorter distances than the laser, the probability of anomalies for this reason will be high. There is also a particular node (reading error failure) that increases its probability when there is an important difference between measured distances while the mentioned anomalies are false.

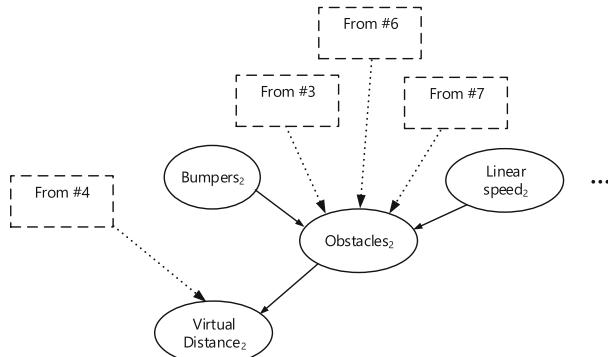
All the described anomalies are summarized into a leaf node which finally indicates if a certain beam of the sensor has an abnormal state. This is depicted in Fig. 3(b). This kind of leaf nodes are natural targets for integrating temporal knowledge, i.e., for considering dynamics. For that purpose, we have attached a sequential filter suitable for continuous data to them, concretely an exponentially weighted moving average filter (EWMA); more precisely, one to each of the three



**Fig. 3.** (a) Systematic errors detection for the anomalies subnetwork corresponding to beam #2 of the laser rangefinder. (b) Anomalies integration for beam #2 of the laser rangefinder and their connection with the associated EWMA sequential filters.

main anomalies considered in this sensor. Thus, the value of their probabilities are affected by present and past inferred distributions.

The second main element of the laser rangefinder sensor network is the virtual subnetwork (deployed in Fig. 4). It integrates the value of the bumpers and the linear speed of the platform under a specific laser scan angle, in addition to some anomalies, using expert knowledge. For instance, assuming reactive navigation, when the linear speed is low the probability of finding a nearby obstacle should be high; also, the bumper node gets information about collisions, which, in case of occurring, should set high the probability of having a short distance to obstacles.



**Fig. 4.** Virtual subnetwork for the beam #2 of the laser rangefinder sensor.

The previously described anomaly subnetwork provides this one with knowledge about whether there is an undetected object (too thin, transparent or black). Furthermore, knowledge about the time of the day and date is important to estimate the amount of people or other kind of mobile obstacles present in the navigation scene. This is another example of the power of Bayesian networks to integrate diverse kinds of knowledge under a common, consistent formalism, namely graph connectivity and CPTs.

Finally, all the multiplexers in our entire Bayesian networks are also temporally filtered, but not with the EWMA filters attached to anomaly nodes, as before, since we are interested in inferred values of the nodes as outputs (discrete), and not in their probabilities (continuous). For that purpose we have used moving-window median filters, an also efficient and well-known solution in statistics for increasing the robustness of data against outliers [15].

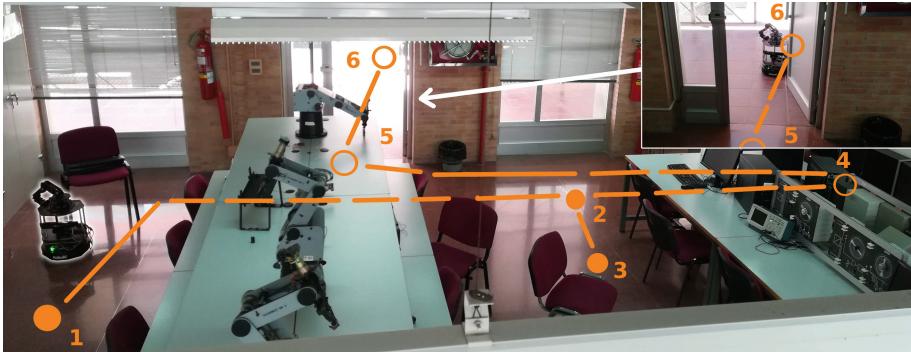
## 4 Evaluation of the Inference System

The Bayesian inference system described in the previous sections has been evaluated in a *Turtlebot* mobile robot in a real experiment described here. The robot has an on board netbook PC with an Intel Celeron N2840 at 2.16 GHz and 2 GB DDR3 that runs Ubuntu 14.04 with ROS [18]. Since we are not interested in dealing with computational limitations in this work, we have used another PC to remotely execute our Bayesian inference software, an Intel Core i3 3217U at 1.8 GHz and 6 GB DDR3 that runs Ubuntu 16.04 with ROS, where our software with the model described in Sect. 3 has been implemented in MATLAB using the Bayes Net Toolbox (BNT) [19].

The Bayesian network has been endowed with 10 laser beams (10 multiplexer nodes in the network, corresponding to approximately  $60^\circ$  of fov) with 10 possible distances for each one, ranging from 1 (no obstacle detected) to 10 (maximum distance). The resulting network has 148 nodes. In every experiment we have run a control loop in which sensory data are obtained, then the Bayesian architecture is evaluated with that evidence and, finally, velocity commands are sent to the robot if needed.

In the real experiment, the *Turtlebot* robot navigates along a route in the mixed indoor-outdoor scenario shown in Fig. 5. During the route, it has to deal with different sensory abnormalities (described in the previous section), that are represented by the posterior distributions associated with nodes of the anomalies subnetworks. These results are shown with their corresponding temporal filtering in Fig. 6, where we consider that there is an abnormal situation when its probability is reasonably high, e.g., equal or greater than 0.7.

Firstly, we analyse the results obtained for the undetected object anomaly (Fig. 6(a)). This issue arises whenever the sensors are faced to thin objects (chair or table legs, columns, cables, etc.) and also when they are not able to capture too distant objects. In this experiment, both situations take place. When the robot is halfway between points 1 and 2 (see Fig. 5), it is pointing to the wall near point 4—this happens during the experiment times of 50 and 60 s, and



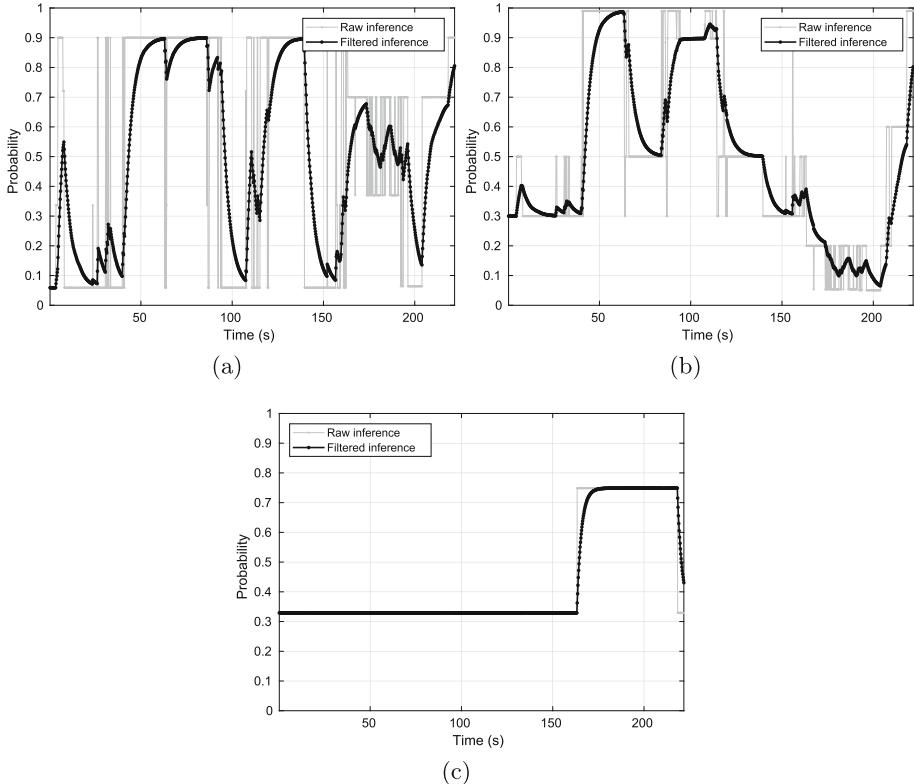
**Fig. 5.** Route followed during the experiment, with some points of interest. A video is available at <http://150.214.109.139/robot2019.mp4>. The robot is highlighted close to the point 1 (bottom-left); the outdoor portion of the route is zoomed in (top-right).

the probability of anomaly shown in Fig. 6 is nearly 0.9. That wall cannot be detected by the laser rangefinder since it is out of its range, but this is not the case for the depth camera, which is used to correct the final sensory value. After that, the robot moves between distinctive points 2 and 3 pointing towards the center chair (times between 70 and 90 s). The problem here is that the chair legs are too thin to be detected; thus the probability of undetected obstacle is again 0.9. Note that this deduction is possible, in particular, thanks to the integration of knowledge about the working environment of the robot—presence of chairs. When the robot is navigating from distinctive point 4 to 5 there are no nearby obstacles again, thus the problem of measurement range reappears.

Concerning the radiation absorbed anomaly (Fig. 6(b)), that situation is provoked by the presence of dark surfaces, specially black ones, which absorb part of the infrared radiation emitted by the ranging sensors. In this particular experiment, the robot is in front of dark objects two times. One coincides with the first undetected object anomaly, as the robot is pointing to a few black-cases computer surfaces that lie around point 4. After navigating through the central corridor (points 2 and 3), the robot moves from point 2 to 4 (experiment times between 90 and 110 s) pointing again towards the dark objects. The inference about this problem is also possible due to the knowledge about the environment, as well as the fact that some measurements are lost.

A third abnormal situation appear in the outdoor part of the environment. Our robot goes outdoor between points 5 and 6, and there finds strong sunlight (experiment time between 160 and 220 s). Both ranging sensors are seriously affected due to the interferences produced by the infrared component of natural light, but that is correctly detected in the probability distribution shown in Fig. 6(c). In this case the deduction is possible thanks to the integration of geographical and other external knowledge (location, season and weather).

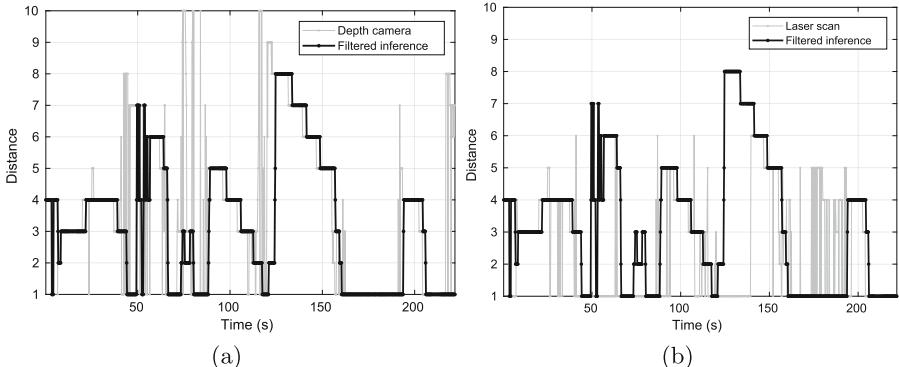
Our Bayesian inference system enables not only to detect abnormal situations as shown so far, but also to recover sensory data under these conditions. In



**Fig. 6.** Posterior distributions for anomalies in the laser rangefinder, both before temporal knowledge integration (gray) and after (black). (a) Undetected obstacle, beam #4. (b) Radiation absorbed, beam #4. (c) Environmental conditions (excess of light).

order to illustrate this, the multiplexer nodes of the laser rangefinder have been used. These variables have probability distributions over the possible discretized distances to obstacles (from 1 to 10 in this case). We have selected the distance with the highest probability at each point along the experiment in order to analyse the recovery capabilities, considering the integration of temporal knowledge through the moving median filter, as explained before. We show the behaviour of a single beam, comparing the corresponding filtered multiplexer with the raw distances obtained by both the laser and depth sensors, in Fig. 7.

These results show that the proposed system is able to recover sensory data in quite adverse situations. As an example, consider the time interval between 120 and 140 s. As shown in Fig. 7(b), the laser rangefinder detects no obstacle, while the filtered multiplexer indicates the presence of a distant object—it has been deduced that the depth camera operates under nominal conditions here. Furthermore, it is also inferred that the probability of undetected obstacle within this interval is high (Fig. 6(a)). In this case, lost measurements have been cor-



**Fig. 7.** Most-likely distance to obstacles inferred by the multiplexer of beam #4 after temporal filtering (black). (a) and (b), in gray, raw evidence for that provided by the depth camera and the laser rangefinder, respectively. “1” means no detected obstacle; otherwise the number corresponds to an increasing distance range, from “2” to “10”.

rected by the data coming from another sensor, integrating knowledge about the conditions of the experiment in this point and temporal information.

## 5 Conclusions and Future Work

In this work we have shown how Bayesian inference can be used to fuse multiple and heterogeneous sources of knowledge, external, expert and temporal, in a rigorous and consistent framework, so as to improve the robustness and power of robotic sensory systems. Our results show that the proposed inference system enables not only to infer faults in the sensors and their causes but also to recover sensory data even in those faulty situations.

In the future there are a number of issues to address. The computational cost of the inference method used here (junction tree) is not suitable for every robotic task, thus improvements are needed (e.g., parallelization, use approximate algorithms, or abstracting the network). Also, the Bayesian network should be created more autonomously and automatically. This should be done by a procedure that ensures the integration of human knowledge and at same time allows to discover the most likely structure of the network. Finally, we also plan to extend our inference system to different robotic platforms and applications.

**Acknowledgements.** This work has been supported by the Spanish government through the national grant FPU16/02243, by the University of Malaga through its local research program and the International Excellence Campus Andalucia Tech, and by the national research project DPI2015-65186-R.

## References

1. International Federation of Robotics (IFR): Executive Summary World Robotics 2017. Service Robots. <https://ifr.org/free-downloads/>. Consulted 14 June 2019
2. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. In: Intelligent Robotics and Autonomous Agents. The MIT Press (2005). ISBN 0262201623
3. Hu, M., Wang, Z., Yuan, Y., Qi, L.: On-line sensor diagnosis of the diesel engine cold starting based on RBFNN. In: IEEE Circuits and Systems International Conference on Testing and Diagnosis (2009)
4. Calderwood, S., McAreavey, K., Liu, W.: Context-dependent combination of sensor information in Dempster-Shafer theory for BDI. *Knowl. Inf. Syst.* **51**(1), 259–285 (2017)
5. Mengshoel, O., Darwiche, A., Uckun, S.: Sensor validation using Bayesian networks. In: 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (2008)
6. Zhu, C., Wang, W.Q., Chen, H., So, H.C.: Impaired sensor diagnosis, beamforming, and DOA estimation with difference co-array processing. *IEEE Sens. J.* **15**(7), 3773–3780 (2015)
7. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press, Cambridge (2009)
8. Osoba, O., Mitaim, S., Kosko, B.: Bayesian inference with adaptive fuzzy priors and likelihoods. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **41**(5), 1183–1197 (2011)
9. Giovanis, G., Papaioannou, I., Straub, D., Papadopoulos, V.: Bayesian updating with subset simulation using artificial neural networks. *Comput. Methods Appl. Mech. Eng.* **319**(1), 124–145 (2017)
10. Murphy, K., Weiss, Y., Jordan, M.: Loopy belief propagation for approximate inference: an empirical study. In: Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence, UAI-99, San Francisco, CA, pp. 467–475. Morgan Kaufmann Publishers (1999)
11. Shachter, R.D., Peot, M.A.: Simulation approaches to general probabilistic inference on belief networks. In: Uncertainty in Artificial Intelligence, vol. 5, pp. 221–231 (1989)
12. Saha, B., Koshimoto, E., Quach, C.C., Hogge, E.F., Strom, T.H., Hill, B.L., Vazquez, S.L., Goebel, K.: Battery health management system for electric UAVs. In: 2011 Aerospace Conference, Big Sky, MT (2011)
13. Castellano-Quero, M., Fernández-Madrigal, J.A., García-Cerezo, A.: Interactive construction of Bayesian inference networks for robust robot sensorics. In: XIV Simposio CEA de Control Inteligente (2018)
14. Pham, H. (ed.): Springer Handbook of Engineering Statistics. Springer, Heidelberg (2006)
15. Huber, P.J., Ronchetti, E.M.: Robust Statistics. Wiley, Hoboken (2009). ISBN 978-0470129906
16. Huang, C., Darwiche, A.: Inference in belief networks: a procedural guide. *Int. J. Approximate Reasoning* **15**(3), 225–263 (1996)
17. Open Source Robotics Foundation, Turtlebot mobile robot official website. <https://www.turtlebot.com>. Visited 16 June 2019
18. Quigley, M., Conley, K., et al.: ROS: an open-source robot operating system. In: ICRA Workshop Open Source Software, pp. 1–6 (2009)
19. Murphy, K.: The Bayes net toolbox for MATLAB. *Comput. Sci. Stat.* **33**(2), 1024–1034 (2001)



# A Manipulation Control Strategy for Granular Materials Based on a Gaussian Mixture Model

Carlos M. Mateo<sup>(✉)</sup>, Juan Antonio Corrales, and Youcef Mezouar

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal,  
63000 Clermont-Ferrand, France  
[cmateoagul@sigma-clermont.fr](mailto:cmateoagul@sigma-clermont.fr)

**Abstract.** In the context of metal additive manufacturing, one of the most attractive tasks to be robotized is the cleaning process of metal powder after the printing operations. This task presents a challenging scenario for most of robot manipulation approaches in the literature.

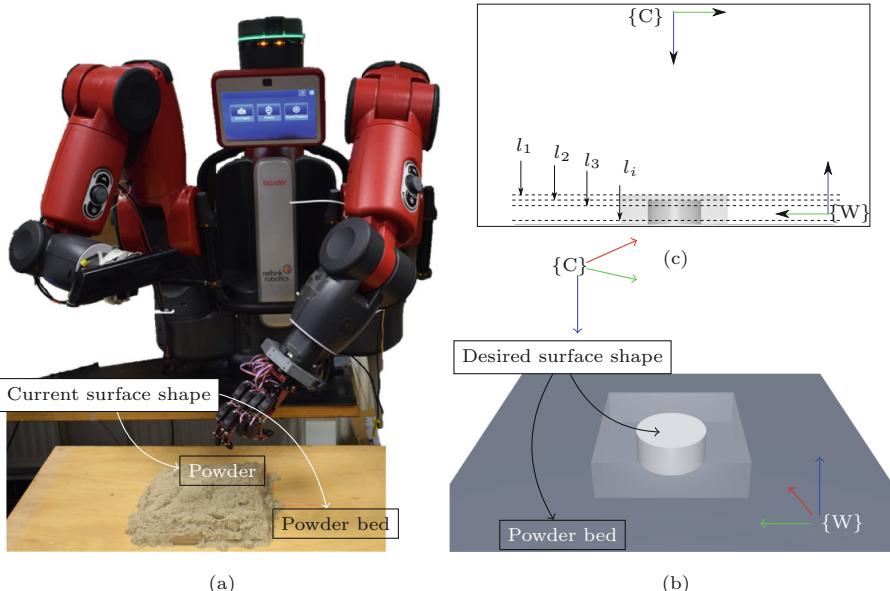
In this paper we present an approach, marker-less and real time affordable, which address the cleaning problem like a shape manipulation control problem. This control strategy is designed as an optimization problem. The error function is written as a lagrangian function using an objective function based on Gaussian Mixture Model (GMM). The local optimization is performed by a gradient descent and a global optimization process is used to avoid local minima.

**Keywords:** Robot control · Image processing · Manipulation · Granular material · Optimization · Non-rigid registration

## 1 Introduction

Robotize cleaning process in metal additive manufacturing nowadays is a hot topic because this process is still handmade, regardless, the manipulated material is an unhealthy substance for people. Metal powder can be easily inhaled and absorbed by cardiovascular system, producing serious health disorders. In addition, this material also presents explosive behavior in contact with oxygen, making the manipulation of this material unsafe [1]. Moreover, during the printing process of pieces, it is wasted more than half of the source product since just a few parts of the powder material is melted. Therefore, we make safer and ecological additive industry by robotizing this task since we avoid involve human operators in risky situation and the rates of re-usability and recycling are increased.

We define the cleaning task as a close-loop form control strategy to drive the robot to the parts to be cleaned while avoiding obstacles. Specially, we consider the operation of removing powder as a soft manipulation task where the powder bed plus the printed piece plus the powder is a whole object (Fig. 1). So, here the



**Fig. 1.** Platform setup where (a) shows the real robot in front of a table with one object occluded inside the powder. Image (b) represents the virtual scenario with object, solid cylinder, which is covered by the powder, transparent cube. (c) Front view of the virtual scenario.

goal is to transform the current object shape (cube of powder with the object inside) into a desired one (the object without any powder). During last years in the literature, we find a few approaches aimed to resolve concrete problems in object (granular material) manipulation tasks, such as [2]. Others like, the work [3] by Sanchez et al. makes a review of advances in robotic manipulation task of non-rigid objects. Works [4,5] by Mateo et al. present a method for understanding how the surface changes during manipulation task, implementing a Dijkstra-based method to model the deformation of the object. The authors also present a method in [6] to predict when the object surface changes drastically during manipulation tasks to achieve dexterous manipulation and prevent damages in objects. The previous cited works are focused mainly in the perception of the objects, but works more related to the control of robots for the manipulation of elastic objects are [7,8] by Navarro-Alarcon, in which an strategy to control the shape of objects with robots, using Fourier transform as features, is presented. In this line, works like [9,10] realize similar tasks, where Cherubini in [9] presents a study to understand how to address the dexterous manipulation of granular materials using robot hands without a previous planning. At the same time Schenck et al. in [10] present a deep learning strategy to teach robots to address the problem presented by Cherubini.

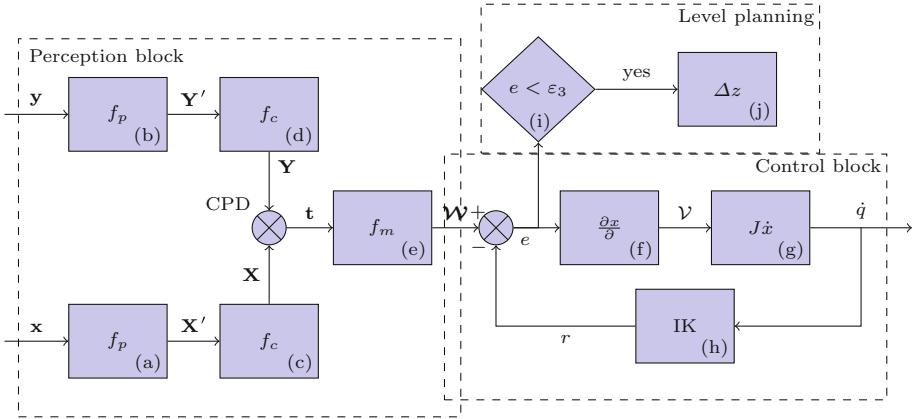
The contribution of this work lies in the proposal methodology (strategy) to resolve the problem of powder removal in metal additive manufacturing. Being the goal of this strategy the transformation of the current shape of the scenario to the desired one. The methodology combines visual perception techniques, optimization methods and robot control schemes. The robot should be moved to minimize the disparity between the current shape and the target shape. We decided to use a strategy to drive the robot to those surface points with maximum disparity and at the same time closest to the current robot position. Here, the disparity is computed using a registration point set method. In the literature there are works which address the problem of registering two different non-rigid surfaces: Coherent Point Drift (CPD) [11]; Volume Deformation [12]; or Killing Fusion [13]. All these works follow the same key concept, retrieve the transformation (deformation) which map one surface to another as rigidly as possible. The control of the robot is formalized using a gradient descent algorithm to optimize the robot velocity [14, 15], expressed in  $\mathbb{R}^3$  manifold.

## 2 Strategy Definition

This approach is devoted to defining the control policy for robotic manipulation tasks in cleaning operations. This method continuously controls a robot until the task is done, without stops for sensing neither re-planning. We consider finished the task when the current scenario shape matches with a reference (or target) scenario. The diagram of the method is pictured in Fig. 2. As is illustrated in the flowchart, the framework splits the approach into three different blocks: perception; and control; level planning. We use the visual perception part to generate the control references (attractors points) used in the robot control. These emerge from the disparity map  $\mathcal{W}$ , computed in the block (e) in Fig. 2, by using the reference  $\mathcal{D}^*$  and current  $\mathcal{D}$  depth images of the scenario.

We use a 3D model to define the target scenario, i.e. a mesh composed by the platform (powder bed) where the printed piece will be built and the piece itself (Fig. 1(b) flat surface and cylinder). Using the 3D model, we create a virtual depth image  $\mathcal{D}^*$ . By a ray tracing technique using a virtual camera placed in the same pose than the real one, a top-view configuration. Where the camera z-axis (optical view axis) is aligned with the z-axis of the world reference frame (Fig. 1(c)). We use as world frame the bottom-right corner of the scenario (Fig. 1(b)).

Once we have the set of projected points  $\mathbf{x} \in \mathcal{D}^*$  and  $\mathbf{y} \in \mathcal{D}$  in image space, we re-project this set into the euclidean space as the set of  $N$  points  $\mathbf{X} \in \mathbb{R}^{N \times 3}$  and  $\mathbf{Y} \in \mathbb{R}^{N \times 3}$ . This is done by the well-known re-projection equation  $\mathbf{X} = \mathbf{P}^+ \mathbf{x}$  where the projection matrix  $\mathbf{P} = \mathbf{K} [\mathbb{I}_{3 \times 3} \mid \mathbf{0}_{3 \times 1}]$  is composed by the extended intrinsic camera parameters matrix  $\mathbf{K}_{4 \times 3}$ , and a matrix  $3 \times 4$ , extrinsic camera parameters, where the first block is an identity matrix  $\mathbb{I}_{3 \times 3}$  and the last column  $\mathbf{0}_{3 \times 1}$  (a zero column). The Moore-Penrose pseudo-inverse projection matrix is computed as  $\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T$ . Equally for the current projected points  $\mathbf{y} \in \mathcal{D}$



**Fig. 2.** Method flowchart. The perception task follows an open-loop scheme where is being compared current and target scenario. Control part is designed as a close-loop for minimizing the error between current and target robot position.

to obtain  $\mathbf{Y}$ . To both point sets (point clouds)  $\mathbf{X}$  and  $\mathbf{Y}$ , we execute a pass-through filter  $f_p : \mathbb{R}^{N \times 3} \rightarrow \mathbb{R}^{M \times 3}$ , where  $M < N$ , (Fig. 2(a) and (b)) to select those points which lie in the current depth layer  $l_i$  (Fig. 1(c)), according with,

$$f_p(\mathbf{X}, z_{low}, z_{up}) = \{X \mid z_X > z_{low} \wedge z_X < z_{up}\}, \quad (1)$$

where  $z_X$  is the  $z$  component of any point in  $\{X_1, X_2, \dots, X_M\} \in \mathbf{X}$ .

We use the Canny edge filter  $f_c$  [16] to extract the contours of the silhouette, in the image projection of the survivor points  $\mathbf{X}'$  and  $\mathbf{Y}'$  in Eq. 1. Function  $f_c$  (Fig. 2(c) and (d)) works in image space, thus  $\mathbf{x}'$  and  $\mathbf{y}'$  are projected before to apply this filter. We use  $\mathbf{x}' = \mathbf{P}\mathbf{X}'$  (and  $\mathbf{y}' = \mathbf{P}\mathbf{Y}'$ ) relation to project back the 3D points into image plane. Thus, final  $\mathbf{X}$  and  $\mathbf{Y}$  point sets are composed by the point re-projection of the sets  $f_c(\mathbf{x}')$  and  $f_c(\mathbf{y}')$ . We compute the error between the two final point sets  $\mathbf{X}$  and  $\mathbf{Y}$  using Coherent Point Drift (CPD) operator [11, 17] (Fig. 2). CPD is a 3D registration algorithm aimed to resolve the problem of 3D non-rigid alignment. This operation returns a set of transformations  $\mathbf{t} \in \mathbb{R}^{3 \times N}$ , that represent the translations between each pair of correlated points.

## 2.1 Map Function $f_m$

Disparity field  $\mathbf{W}$  is used in this work to balance the importance of each boundary point  $\mathbf{Y}$  according to the robot pose  $r$  and the visual feedback  $\mathbf{t}$ . So, we purpose the following mapping function  $f_m : \mathbb{R}^{N \times 3} \rightarrow \mathbb{R}^N$  subject to the robot pose  $r$  and  $\mathbf{t}$  to get  $\mathbf{W}$  (Fig. 2 block (e)). Thus, the fundamental idea of this function is to define a reachability rank of points. That is, weight with the maximum likelihood the point with better ratio between: distance to  $r$ ; and maximum disparity between current  $\mathbf{Y} \in \mathbb{R}^3$  and the desired  $\mathbf{X} \in \mathbb{R}^3$  position. Note that here,

we keep using the correlation space between  $\mathbf{X}$  and  $\mathbf{Y}$  obtained after applying CPD algorithm beside robot position  $r$ .

Mathematically, we can express the mapping function  $f_m(\mathbf{Y} \mid r, \mathbf{t})$  as a multiplication of two different weights terms  $\mathbf{W} = \mathbf{W}_t \mathbf{W}_d$ . Where  $\mathbf{W}_d$  is the vector of weights according to the robot pose  $r$  (i.e. distance between point in the current surface and the robot) and  $\mathbf{W}_t$  with the visual feedback (i.e. distance between current and desired surfaces). Then,  $\mathbf{W}_t$  is formally expressed like the normalized vector according with infinity norm, i.e. the set of  $\hat{\mathbf{t}}$  is weighted with respect to the maximum  $\hat{\mathbf{t}}$ . This is because we emphasize those points with more transformation, formally this is as follows,

$$\hat{\mathbf{t}} = \|\mathbf{t}\|_2, \quad \mathbf{W}_t = \frac{\hat{\mathbf{t}}}{\|\hat{\mathbf{t}}\|_\infty} \quad (2)$$

where  $\hat{\mathbf{t}}$  is a vector in  $\mathbb{R}^N$  stacking all euclidean norm of all elements in  $\mathbf{t}$ . On the other hand,  $\mathbf{W}_d$  is the normalized distance vector  $\hat{\mathbf{d}} = \mathbf{d} - \mathbf{r}$  according with the negative infinity norm, i.e. the set  $\hat{\mathbf{d}}$  is balanced w.r.t. the minimum  $\hat{\mathbf{d}}$ . This is because we look for the closest point,

$$\hat{\mathbf{d}} = \|\mathbf{d}\|_2, \quad \mathbf{W}_d = \frac{\hat{\mathbf{d}}}{\|\hat{\mathbf{d}}\|_{-\infty}}. \quad (3)$$

Values in Eqs. 2 and 3 are bounded in the range  $(0, 1]$ , thus  $\mathbf{W}$  is also bounded in the same range. Where the most relevant point  $\mathbf{Y}_i \in \mathbf{Y}$  is valued with a 1.

## 2.2 Objective Function Definition

At this point all filter, registration and mapping functions used in the perception block are presented. This subsection is devoted to present the objective function used in the control law which is shown in Fig. 2.

Control here is formalized as an optimization problem where the parameters that minimize the objective function should be found. In our case, the parameter that we want to optimize is the robot velocity. The goal (or objective) is to move the robot  $r$  to the maximum-interesting points  $\mathbf{W}$ , discussed above. Our objective function is constrained to the fact that the robot does not have to cross over the desired shape  $\mathbf{X}$ . This constraint is imposed because  $\mathbf{X}$  is the contour of the solid 3D printed object plus the machine support (powder bed Fig. 1), therefore crossing over this area means a collision between the robot and the scene.

To convert a constrained into an unconstrained problem, we use the Lagrange multipliers method [18]. Using this function, we can find the local minimal of our objective function subject to our constraint,

$$\mathcal{L}(r, w) = f(r) - wg(r \mid \mathbf{X}) \quad (4)$$

where,  $f(r)$  is the objective function,  $g(r \mid \mathbf{X})$  is the equality constraint and  $w$  (free parameter, in our experimentation  $w = \frac{1}{2}$ ) is a weight to balance the relevance of  $g$ . Note that in  $\mathcal{L}$  is still differentiable.

Because the intrinsically noisy nature of image sensors, we define the objective function using a Mixture Model (MM) function. Concretely the MM used here is the Gaussian Mixture Model GMM,

$$f(r) = -\log \sum_i^N \alpha_i \mathcal{N}(r \mid Y_i, \sigma), \quad (5)$$

where  $\mathcal{N}(r \mid Y_i, \sigma)$  is a Gaussian distribution function centered in the point  $y_i$  with a spread factor of  $\sigma$  and a peak value equal to  $\alpha_i$ . Note that there is a Gaussian for each point in the current contour. Gaussian's peak in GMM is treated as the prior-knowledge (in our case  $\mathbf{W}$ ) of the  $i$ -th distribution and is bounded between  $[0, 1]$ . In the other hand, constraint equation  $g$  is defined in terms of an exponential artificial field,

$$g(r \mid \mathbf{X}) \equiv \mathcal{E}(r \mid \mathbf{X}). \quad (6)$$

Note that we do not use the Gaussian Mixture Model to represent the position of the robot but to relate the current robot pose with respect to the points in the current surface shape  $\mathbf{Y}$ . Also, MM is used to deal with the intrinsic uncertainty of visual information.

Following section shows more detailed develop about objective function  $f$ , constraint  $g$  and lagrangian  $\mathcal{L}$ .

### 3 Task-Space Robotic Control

The most straightforward approach to move a robot is to design a velocity controller which relates the error function variations during time  $\dot{e}$  with the robot velocity  $\mathcal{V} = [v, \omega]$ , Chaummette and Hutchinson in [19]. Formally, this is written as  $\dot{e} = \nabla e \mathcal{V}$ , where  $\nabla e$  is the gradient of the error function  $e$ ,  $v \in \mathbb{R}^3$  is the linear velocity and  $\omega \in \mathbb{R}^3$  the angular velocity, thus the velocity vector is in  $\mathcal{V} \in \mathbb{R}^6$ .

In this work the error function is equivalent to the Lagrangian Eq. 4,  $e \equiv \mathcal{L}(r, w)$ . Considering that the hidden idea is to minimize the error ensuring that the decreasing follows a monotonic exponential function with respect to time  $\dot{e} = -\lambda_e e$ , being the controller gain  $\lambda_e$  an scalar positive constant, the obtained expression is as follow,

$$\mathcal{V} = -\lambda_e \nabla e^+ e. \quad (7)$$

where  $\nabla e^+$  is the Moore-Penrose pseudo-inverse gradient matrix.

We extend the control law (Eq. 7) to optimize the spread  $\sigma$  of  $\mathcal{N}$  while the robot velocity  $\mathcal{V}$ . Therefore, our final velocity vector  $\bar{\mathcal{V}} \equiv [\mathcal{V}, \dot{\sigma}] \in \mathbb{R}^7$  is composed by  $\mathcal{V}$  and the gaussian's spread time variation  $\dot{\sigma}$ . Consequently, the gradient  $\nabla e$

is differentiated also with respect to  $\sigma$ , not just w.r.t. the robot pose. This is because by controlling the spread of GMM variance, we are controlling the velocity of the robot. The idea is to situate the robot pose in the slop of the GMM where it is the maximum value variation of the function and therefore the gradient have the maximum value.

### 3.1 Error Definition

As it has been introduced before, removing particle media from a scene can be considered as a surface shape control problem. Then, controlling the difference between desired  $\mathbf{X}$  and current  $\mathbf{Y}$  shapes can be expressed as the optimization of an error function  $e$ . Moreover, the error function  $e$ , as we discussed above, is equivalent to the lagrangian operator  $\mathcal{L}$ , defined in Eq. 4.

Objective function  $f$  in  $\mathcal{L}$  is defined using a GMM in Eq. 5, where the  $i$ -th gaussian distribution  $\mathcal{N}(r | Y_i, \sigma)$  is defined as,

$$\mathcal{N}(r | Y_i, \sigma) = \exp\left(-\frac{1}{2} \frac{\|r - Y_i\|_2}{\sigma^2}\right), \quad (8)$$

and the prior-knowledge (guassian's peak) of the  $i$ -th gaussian is equivalent to  $\alpha_i \equiv \mathcal{W}_i$ . The potential field term  $\mathcal{E}$  is designed using a sigmoid function and is defined as follows,

$$\mathcal{E}(r | \mathbf{X}) = -\frac{1}{1 + \exp(-\|r - \mathbf{X}\|_{-\infty})} \quad (9)$$

As the equation shows this potential barrier is based on the minimum distance between any point of the desired surface  $\mathbf{X}$  and current robot pose of  $r$ . Therefore, the potential barrier is going to be governed just by the closest desired contour point  $X_i$  to  $r$ .

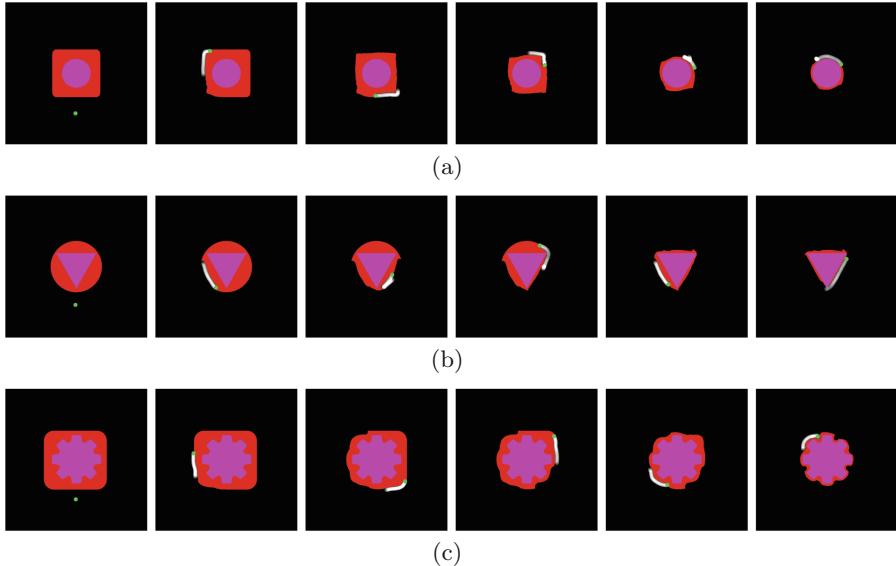
Then, by injecting Eqs. 8 and 9 in 4 we obtain our error function as,

$$e = -\log \left[ \sum_i^M \mathcal{W}_i \exp\left(-\frac{1}{2} \frac{\|r - Y_i\|_2}{\sigma^2}\right) \right] + w \frac{1}{1 + \exp(-\|r - \mathbf{X}\|_{-\infty})} \quad (10)$$

Equation 10 represents the error function of our method. This formulation can be see has a problem of regularization where we have two different forces: attraction (first term); and repulsive (second term). As it is expressed in the first part of the equation, as smaller is the difference between robot pose with the points in current surface shape  $\|r - Y_i\|_2$ , smaller is the energy of the error function. In contrast the second term as smaller is the distance between the robot pose and the point in desired surface shape  $-\|r - \mathbf{X}\|_{-\infty}$ , bigger is the energy of the error function.

## 4 Global Robot Pose Optimization and Level Planning

We use the Deterministic annealing algorithm [20] to govern the global behavior of our control scheme, while the control scheme is devoted to walk towards the



**Fig. 3.** Experiments over 3 different scenarios. Green dot represents robot pose, path is represented in white color, current surface is the red surface and purple is the desired surface. Note that the purple area is a restricted area for the robot.

most interesting points in the current surface according with  $\mathcal{W}$ . The global optimization strategy prevents that the system falls in stationary state. These stationary states are reached when the robot position is too far from the influence of  $f$  or too close to the GMM's peak. This global optimization is as follows,

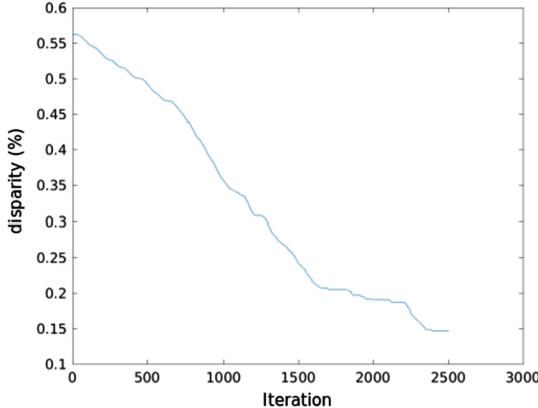
$$\sigma_{t+1} = \begin{cases} \lambda_\sigma \sigma_t, & e < \varepsilon_1 \\ \lambda_\sigma^{-1} \sigma_t, & e > \varepsilon_2 \end{cases} \quad (11)$$

where  $\lambda_\sigma$  is the global gain of the system and  $\varepsilon_1$  and  $\varepsilon_2$  are the thresholds used for activating the global optimization.

In the other hand, the global evolution of the cleaning task  $l_1 \rightarrow l_2 \rightarrow \dots \rightarrow l_i$  evolves by removing layers of powder (Fig. 1). The method flowchart as a level planning block (Fig. 2 block (i)) aimed to move the robot in  $z$ -direction in the cleaning task. Then, the depth range are  $z_{low} = z_{low} + \Delta z$  and  $z_{up} = z_{up} + \Delta z$  (this is a range area surrounding a level  $l_i$ ) if the differences between current and target surfaces is lower than a desired value  $\|\mathbf{X} - \mathbf{Y}\|_2 < \varepsilon_3$ .

## 5 Experiments

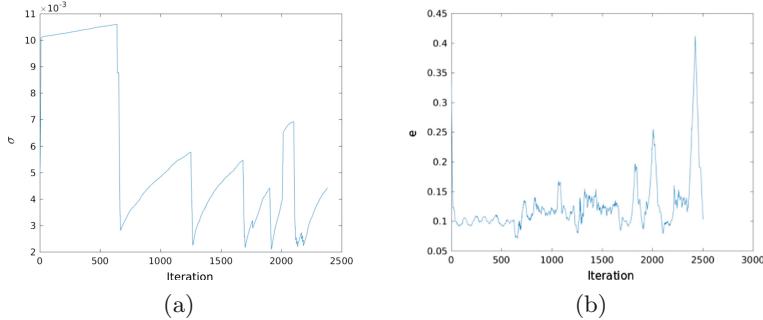
Here, a set of 3 different simulations are designed (Fig. 3), assuming the robot should clean in  $z$  direction, i.e. layer by layer, where each test represents a depth map in a specific layer. This strategy is chosen because it reduces drastically



**Fig. 4.** The disparity between shapes evolution over time. x-axis represents the iteration

the point set density (because only those points in the silhouette of the shape are considered) in comparison to take all the points in the 3D surface, hence the computational time is reduced proportionally. We first evaluate the characteristics of the proposed method using simulations. And secondly, we have implemented this strategy in a real robot to prove the viability. This approach has 2 different gains: one for the local optimization  $\lambda_e$ ; and the other for the global optimization  $\lambda_\sigma$ . Moreover, this scheme has three bound parameters  $\varepsilon_1$ ,  $\varepsilon_2$  and  $\varepsilon_3$ , two first to keep  $e$  in a desired level and the last to know when a depth level is cleaned (or deformed). Due the fact that this strategy comprised a global optimization step to bound the error function inside an error levels, this method cannot be evaluated directly on the study of the error evolution. Therefore, the convergence is analyzed based on the shape similarities evolution between current and target shapes.

We found that the behavior of system is suitable for each test, doing just a single set up of these parameters. The gains for error minimization and variance optimization are  $\lambda_e = 0.03$  and  $\lambda_\sigma = 200$ . The limits for error function in this experimentation are set in the range  $\varepsilon_1 = 0.1$  and  $\varepsilon_2 = 0.15$ . Additionally, we have initialized the value of GMM spread to  $\sigma = 0.005$ . One of the advantages of this method is the smooth transformation that generates. This ability can be appreciated in Fig. 3(a) where the shape is transformed from a square to a circle. This kind of behavior is progressively doing the task while avoiding problems of local minimum. In this work, we consider that the robot is in a local minimum when this is stop and still remind powder to remove. Although it is not common, we still have problem with local minima in concave shapes. In other hand, the Fig. 3(b) presents a challenging situation where three unconnected powder areas should be removed. This test shows how the system deals with local tasks to get a global solution. The experiment 3(c)) was designed to demonstrate how the

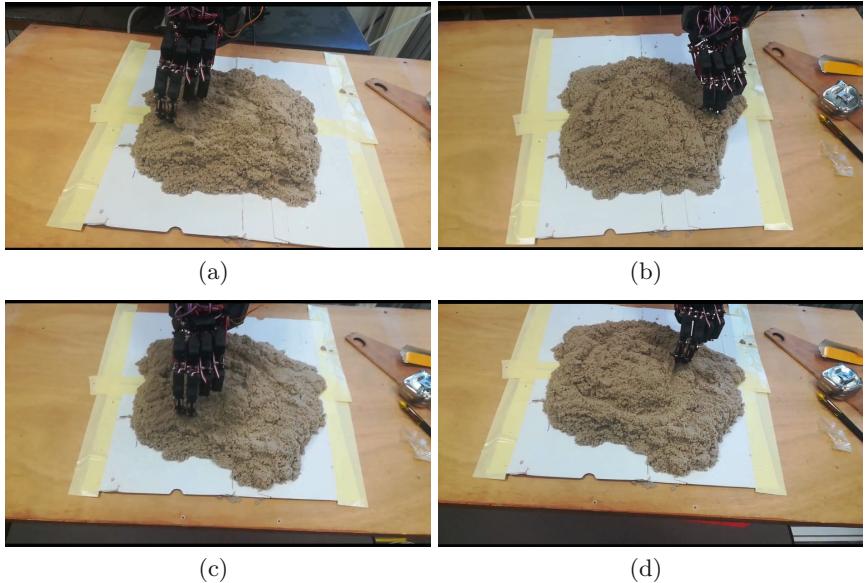


**Fig. 5.** Evolution over the time (x-axis) of: (a) GMM  $\sigma$ , y-axis; and (b) error  $e$  function, y-axis

controller manages to convert a simple shape into a much more complex shape (concave shape).

It is demonstrated for all experiments that the system presents a local and global behavior. All experiments complete their task after approximately 2500 iterations. The time to process a single iteration depends on the amount of points in the silhouette. More concretely the relation between the number of points and the time to perform the task is linear. In these experiments, the silhouette is composed by point set of around 300 elements and the time to compute an iteration is approximated 0.01 s. As it is mentioned before, the system convergence is achieved when the disparity between surfaces (ranged in [0, 1]) is less than a desired value. A desired disparity of 0.015 was set for the experiments. Figure 4 shows the evolution (monotonic decrease) of disparity for experiment 3 (Fig. 3(c)). In the other hand, the associated error evolution is presented in Fig. 5(b) and shows how this error falls until a bounded error is achieved and during all the task the system works to keep this error inside the limits. Figure 5 shows global optimization evolution according to the plot presented in Fig. 5(b). GMM variance evolution is plotted in Fig. 5(a) and it can be observed how the  $\sigma$  is being progressively update using local minimum optimization until, approx. iteration 600, when error value (Fig. 5(b)) fall down the low limit. Then the value of  $\sigma$  is globally updated, to recover rapidly the robot velocity, according to the Eq. 11 (Fig. 5(a)). And again, the GMM variance is locally updated.

Finally, we implement this approach over a real platform to demonstrate its applicability in real scenarios (Fig. 6). We use the collaborative robot Baxter equipped with a Kinect camera in the end effector of the right arm and an AR10 humanoid hand in the left arm. We decided to use a hard actuation (touching) over the material (kinetic sand) to avoid uncertainties produced by soft actuation (vacuum aspiration). Figure 6 presents a series of key frames of the manipulation task. This experiment is following the described trajectory in the third simulated test (Fig. 3(c)).



**Fig. 6.** Real robot experiment. Hard actuation over kinetic sand.

## 6 Conclusions

This work presents a novel approach to manipulate granular materials using a robotic manipulator. Here, it is defined the cleaning tasks problems as a shape deformation problem. The goal is reached using our proposed control scheme based on a Gaussian Mixture Model. One of the major contributions of this work is the novel control scheme which combines local and global optimization by means of a gradient descent and a Deterministic Annealing method. Another important contribution is that here the task control is not defined in terms of error evolution. The error evolution here is just used to control the robot velocity. In contrast here, task evolution is defined in terms of surface shape disparity (Fig. 4). The experiments show that this method reach solutions in real time when the point density of silhouette contour is low.

But this approach cannot guaranty an optimal solution when desired shape has complex topology (many concave shapes). In this work we initialize the gaussian distribution's spread although this can be discovered by the global optimization algorithm. We do this because as near these values are to the optimal ones, the convergence is faster. This fact opens the possibility to extend this method to predict optimal values of  $\sigma$  by prior clues using learning strategies. Similarly, the definitions of boundaries in the energy function can be improved by using another source of information which predicts best boundaries for each problem.

**Acknowledgements.** Research supported by project CoMManDIA (SOE2/ P1/F0638) which is cofinanced by Interreg Sudoe Programme (European Regional Development Fun), and by the Chair Industrial of SIGMA Clermont Chaire SIG-AM “Fabrication Additive Multisectorielle”.

## References

1. Gibson, I., Rosen, D., Stucker, B.: Additive Manufacturing Technologies (2015)
2. Balatti, P., Kanoulas, D., Tsagarakis, N.G., Ajoudani, A.: Towards robot interaction autonomy: explore, identify, and interact, pp. 9523–9529, April 2019
3. Sanchez, J., Corrales, J.A., Bouzgarrou, B.C., Mezouar, Y.: Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *J. Robot. Res.* **37**(7), 1–34 (2018)
4. Mateo, C.M., Gil, P., Torres, F.: Computation of curvature skeleton to measure deformations in surfaces. In: Lecture Notes in Electrical Engineering, vol. 383, pp. 197–207. Springer, Cham (2016)
5. Gil, P., Mateo, C.M., Delgado, Á., Torres, F.: Visual/tactile sensing to monitor grasps with robot-hand for planar elastic objects. In: 47th International Symposium on Robotics, ISR 2016, vol. 2016, pp. 439–445 (2016)
6. Mateo, C.M., Gil, P., Torres, F.: 3D visual data-driven spatiotemporal deformations for non-rigid object grasping using robot hands. *Sensors (Switzerland)* **16**(5), 640 (2016)
7. Navarro-Alarcon, D.: A Depth-Based Algorithm for Manipulating Deformable Objects Using Smooth Parametric Surfaces and Energy Minimisation, pp. 33–34, April 2018
8. Navarro-Alarcon, D.: Visual shape servoing of deformable objects : the fundamentals. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 18, p. 5386 (2017)
9. Cherubini, A., Leitner, J., Ortenzi, V., Corke, P.: Towards vision-based manipulation of plastic materials (2018)
10. Schenck, C., Tompson, J., Fox, D., Levine, S.: Learning robotic manipulation of granular media. In: Proceedings of Machine Learning Research 78(Conference on Robot Learning (CoRL)) (2017)
11. Myronenko, A., Myronenko, A., Song, X., Song, X., Carreira-Perpiñán, M.Á., Carreira-Perpiñán, M.Á.: Non-rigid point set registration: coherent point drift. In: Advances in Neural Information Processing Systems 19, pp. 1009–1016, September 2007
12. Innmann, M., Zollhöfer, M., Nießner, M., Theobalt, C., Stamminger, M.: VolumeDeform: real-time volumetric non-rigid reconstruction. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) LNCS, vol. 9912, pp. 362–379 (2016)
13. Slavcheva, M., Baust, M., Cremers, D., Ilic, S.: KillingFusion: non-rigid 3D reconstruction without correspondences. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 4, pp. 1386–1395 (2017)
14. Corke, P.: Robotics, Vision and Control: Fundamental Algorithms in MATLAB®, vol. 118, 2nd edn. Springer (2017)
15. Lynch, K.K.M., Park, F.C.: Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press, Cambridge (2017)
16. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI* **8**(6), 679–698 (1986)

17. Myronenko, A., Song, X.: Point set registration: Coherent point drifts. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 2262–2275 (2010)
18. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer, Heidelberg (2006)
19. Chaumette, F., Hutchinson, S.: Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* **13**(4), 82–90 (2006)
20. Ueda, N., Nakano, R.: Deterministic annealing EM algorithm. *Neural Netw.* **11**(2), 271–282 (1998)



# Visual and Tactile Fusion for Estimating the Pose of a Grasped Object

David Álvarez<sup>1</sup> , MÁximo A. Roa<sup>2</sup> , and Luis Moreno<sup>1</sup>

<sup>1</sup> Systems Engineering and Automation Department,  
Carlos III University of Madrid, Getafe, Spain  
`{dasanche,moreno}@ing.uc3m.es`

<sup>2</sup> Institute of Robotics and Mechatronics, DLR - German Aerospace Center,  
Cologne, Germany  
`maximo.roa@dlr.de`

**Abstract.** This paper considers the problem of fusing vision and touch senses together to estimate the 6D pose of an object while it is grasped. Assuming that a textured 3D model of the object is available, first, Scale-Invariant Feature Transform (SIFT) keypoints of the object are extracted, and a Random sample consensus (RANSAC) method is used to match these features with the textured model. Then, optical flow is used to visually track the object while a grasp is performed. After the hand contacts the object, a tactile-based pose estimation is performed using a Particle Filter. During grasp stabilization and hand movement, the pose of the object is continuously tracked by fusing the visual and tactile estimations with an extended Kalman filter. The main contribution of this work is the continuous use of both sensing modalities to reduce the uncertainty of tactile sensing in those degrees of freedom in which there is no information available, as presented through the experimental validation.

**Keywords:** Pose estimation · Sensor fusion · Tactile sensors · Visual information

## 1 Introduction

In-hand object pose estimation is a natural cognitive online process that humans perform while grasping or manipulating objects. There are several indications that humans use complementary sensor information from vision and touch in this process [1, 2], that manipulation tasks rely on accurate and fast pose estimation [3], and that human memory is multi-sensorial in nature [4].

In computer vision, several methods for stable and reliable object pose tracking exist in literature. Many approaches are based on tracking object boundaries [5] or on non-linear pose computation using RGB-D information [6]. Texture tracking [7] and model-free cues [8] have also been presented. While simple scenarios allow an accurate object pose tracking, more complex tasks may require

inferring object properties [9]. In this paper, a combination of texture tracking and image-based motion cues is used for processing visual information, inspired by [10].

Tactile sensing has also received a great deal of attention recently, including applications in texture and object recognition [11, 12] and in-hand pose estimation. Object recognition by exploring the object's surface and edges using a particle filter combined with an Iterative Closest Point approach was presented in [13]. In the case of pose estimation, some methods use an offline description of the object's facets to match the current sensor measurements [14]. Also, preventing physically unfeasible solutions can be considered for the in-hand pose estimation process [15]. Position and torque measurements from the finger joints have been used to estimate the pose of the object as well as the contact state of the grasp [16]. Our approach for using tactile information, initially presented in [17], uses a particle filter to enhance solutions that match sensor measurements, thus avoiding physically unfeasible estimations.

One of the first attempts to integrate vision and touch was presented in [18], using geometric models of objects that are complemented with tactile sensing for gathering information on the unseen parts. More recently, RGB-D and tactile data were treated using an invariant extended Kalman filter (EKF) to discover and refine 3D models of unseen objects [19], with practical applications for simplified models of symmetric objects characterized by two features, width and angle. The fusion of tactile and visual measurements enables also the pose estimation of a moving target at high rate and accuracy [20]. Instead of tracking an external object, they follow a probe, which produces tactile measurements, mounted on an hydraulic manipulator. Fusion of tactile and visual information has been used to refine an initial estimation of the hand-object pose for grasping applications [21, 22]. Several approaches have tried to simultaneously use vision and tactile information for in-hand object pose estimation. However, the use of visual information often ends when the hand is closed around the object, and afterward only tactile information is used for the pose estimation process [23, 24]. In [25], tactile sensing is used to add physical constraints to a vision-based estimator; however, the pose estimation is mainly based on vision, therefore heavy occlusions are difficult to manage.

Both vision and touch can be used to separately estimate the 6-DOF pose of an object, but typically each estimation is not accurate along one or several degrees of freedom. This work is centered on the effective combination of both modalities to improve the pose estimation during a grasping action. The visual estimation is based on [10], which uses a CAD-based pose estimation and an optical flow-based tracker, while the tactile information is processed following our previous work in [17]. The fusion of both estimations is done using an Extended Kalman Filter, which prioritizes one of the sensor modalities depending on the accuracy of each method at a given stage. The visual information is constantly used to complement the information gathered by tactile sensors while there is contact with the object, thus reducing uncertainty along the directions where the

tactile information does not provide enough information to effectively estimate the pose of the object.

## 2 Sensor Fusion Framework

The grasp execution is divided into different phases depending on the existence or not of contact between hand and object. These phases define the type of information available for estimating the hand-object pose, as summarized in Table 1. During the pre-grasp phase, the hand moves towards the object to achieve the pose from which the grasp is executed. The vision system has a clear view of the scene, while there is no useful tactile information yet. The grasp phase starts when the first contact between hand and object is detected by the force sensors of the hand, and ends when the hand is commanded to open the fingers. It is during this phase that the two sensing modalities can be independently used for estimating the hand-object pose; however, the vision system may have difficulties tracking the object due to occlusions created by the fingers wrapped around the object. Finally, when the hand releases the object there is only visual information available, although there might be no estimation at all when the vision system gets lost. Therefore, in the case of the first and third phases of the grasp execution, only visual information is used in the pose estimation, while in the second one, both visual and tactile information are effectively fused.

**Table 1.** Information provided by the sensors in each grasp phase.

Grasp phase	Pre-grasp (Before closing)	Grasp (Object in hand)	Release (Hand open)
Information	Visual	Visual/tactile	Visual

### 2.1 Vision-Based Estimation

The vision-based pose estimation and visual tracking are both based on Simtrack [10]. In our case, the tracking problem is simplified since we only use RGB information provided by the vision system. The system assumes that a 3D model of the object, which includes texture, is available and used to perform a comparison with the information provided by the camera.

From the color feed provided by the RGB camera, SIFT key-points (Scale-Invariant Feature Transform) [26] and optical flow (movement in the image) are extracted. The SIFT keypoints are used by the object detection system in order to provide an initial estimation for starting the tracker start, or for restarting it when the tracker is not capable anymore of following the object. On the other hand, motion cues provided by optical flow are used by the tracking system. Note that the original software also uses cues extracted from stereo disparity or depth information, but in this case, this information is not available due to



**Fig. 1.** Examples of the initial visual estimation for the object’s pose. A common error that appears is that the estimated object pose is floating above the supporting table, as illustrated in the central image.

the experimental setup (Sect. 3). The tracker updates the estimated pose of the object so that the consistency between the motion and a 3D representation of the environment is maximized.

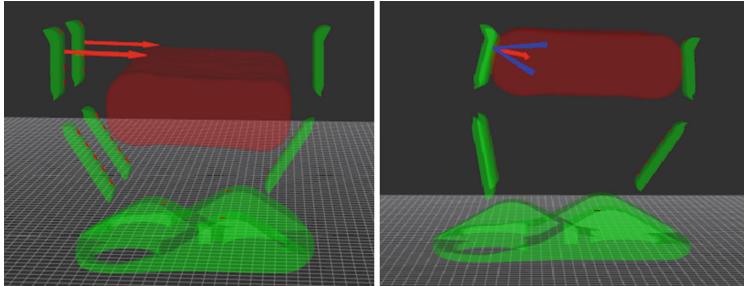
When the pose estimation is active, GPU libraries are used in order to extract the SIFT features from the 2D images. Then, a Random Sample Consensus (RANSAC) method [27], which tries to match SIFT descriptors extracted from the RGB images to the textured 3D models in the database, is used to find correspondences and extract the 6-DOF pose of the object. This step tries to perform an exhaustive matching in the given frame and, therefore, this sparse estimation does not depend on the previous one, or on the movement in the images.

Once an initial estimation is available, the tracker starts and uses motion cues to compute the motion in the scene (using GPU libraries [28]). For this, an Augmented Reality (AR) version of the estimation is rendered. Then, optical flow is computed out of the difference between the (partially) synthetic image, rendering the object model based on the current pose estimate, and the next obtained image. When used for tracking, this information is insensitive to drift since it measures the difference between the current scene hypothesis and the observed scene (rather than simply the image motion). For the same reason, it can be used to measure the reliability of tracking. The motion observed by the optical flow is used to recover a rigid rotation and translation that best explains the visual cues, and transforms the pose estimation accordingly.

Figure 1 shows examples of the initial estimation based on visual information. The estimated pose is also shown (in green) on the image. The image in the center shows an error that appears commonly, namely, the estimated pose is too high over the table surface, which is not physically realistic. This is due to the point of view used for acquiring the images. Average errors of the initial pose estimated by the vision system for two different objects are presented in Table 2.

**Table 2.** Average error in the initial visual estimation.

Object	Can			Box	
Trial	1	2	3	1	2
Error in position (cm)	3.67	2.67	1.40	3.08	2.97
Error in orientation (°)	0.45	1.52	1.03	0.97	0.85



**Fig. 2.** Left: two contacts are detected (red arrows), but the estimated pose of the object does not produce contacts. Right: friction cone at a contact location.

## 2.2 Tactile-Based Estimation

Tactile-based estimation is only possible when there is at least one contact between the hand and the object. Therefore, when the first contact is detected, a particle filter starts looking for object poses using, as first prior estimation, the last pose provided by the vision system. The estimated pose should agree with the information provided by the tactile sensors.

The reference frame used for the tactile pose estimation is located at the wrist of the hand. The parameters describing the object pose are

$$x = [q, t]^T = [q_x, q_y, q_z, q_w, t_x, t_y, t_z]^T \quad (1)$$

where  $q$  is a rotation expressed as a quaternion, and  $t$  is a translation vector.

Assuming that the 3D model of the object is available, the estimation is tackled by combining the following general ideas:

1. When a contact is detected by a sensor, the estimated object pose must produce a contact at the same location. Figure 2 (left) illustrates a case where the estimated object pose cannot explain the contact readings in two fingers.
2. The estimated object pose should not be in collision with the hand (just in contact). Besides, the object cannot float without contacting the hand at all when at least one sensor reading is positive.
3. The inward normal of the object surface at the contact location and the outward normal at the contact surface in the hand should have the same direction. When friction is considered, the normals do not necessarily have to be aligned, but, since the friction coefficient is in general not known, the angle between normals should be as close to 0 as possible (right side of Fig. 2).

A deeper explanation on the implementation of the particle filter (where each particle represents one pose of the object) can be found in [17]. The general outline of the algorithm is shown in Algorithm 1. The key ideas presented above are included in the measurement model that weighs the estimation of a given particle as explained below.

---

**Algorithm 1.** Bootstrap Particle Filter

---

```

1: procedure BPF( $N_p, prior\_estimation$ )
   Initialization:
2:    $x_i(0) \sim Pr(x(0)), W_i(0) = \frac{1}{N_p}$ 
   Importance Sampling:
3:    $x_i(t) \leftarrow system\_model(x_i(t-1), input_t)$ 
4:    $W_i \sim Pr(W_i(t))$ 
   Weight Update:
5:    $W_i(t) = W_i(t-1) \times measurement(y(t)|x_i(t))$ 
   Weight Normalization:
6:    $W_i(t) = \frac{W_i(t)}{\sum_{i=1}^{N_p} W_i(t)}$ 
   Resampling:
7:   if  $\hat{N}_{eff}(t) \leq N_{thresh}$ , then  $\hat{x}_i(t) \Rightarrow x_j(t)$ 
8: end procedure

```

---

**Weight Update: Measurement Model.** The measurement model gives to each one of the particles a weight that quantifies how similar is the state expressed by that particle to the true state of the object, comparing the estimation with the measurements provided by the position and tactile sensors in the hand. The measurement model used in this work is inspired by [15]. Three new features have been added: first, not only the sensor location but also the force measurements are used to compute the real contact locations. Second, the evaluation method considers differently each sensor depending on whether it is in contact or not. And third, the friction cone of a contact is considered to evaluate the feasibility of a contact between the hand and the object.

In order to evaluate the particles, the scene is simulated using the Flexible Collision Library (FCL) [29] to compute the shortest distance (no collision, positive value) or deepest penetration (in collision, negative value) between each sensor and the object. Taking into account this information, three kind of measurements are considered:

- For each sensor that does not detect contact with the object, a probability is assigned to each particle based on its distance to the object  $d_i^o$  by:

$$p_{nc,i}(d_i^o) = 0.5 * \left( 1 + erf \left( \frac{d_i^o}{\sqrt{2}\sigma_{nc}} \right) \right) \quad (2)$$

where  $\sigma_{nc}$  is a standard deviation value that can be adjusted to match the inaccuracy of the measurements, and  $erf$  corresponds to the error function.

This function is chosen to assign high weights to positive distances and small weights to negative distances, which helps to avoid estimations that predict unreal collisions.

- For each sensor that detects a contact, the distance  $d_i^o$  is used to associate a probability to the measurement with:

$$p_{c,i}(d_i^o) = e^{-0.5\left(\frac{d_i^o}{\sigma_c}\right)^2} \quad (3)$$

where  $\sigma_c$  can be adjusted to account for the uncertainty in the force sensors. This function assigns high weights to values that are close to zero, i.e. close to contact.

- Assuming the grasp is stable, the normal of the surface of the object at the contact point (for the sensors that detect a contact) should lie within the friction cone around each contact point in the hand. The contact force measured by the hand is considered to be normal to its surface, therefore, the angle  $\alpha_i$  between the normals to the surfaces can be computed, and afterward evaluated with:

$$p_{a,i}(\alpha) = e^{-0.5\left(\frac{\alpha_i}{\sigma_a}\right)^2} \quad (4)$$

where  $\sigma_a$  accounts for the friction between the surfaces. This function assigns high weights to values that are close to 0.

Finally, a combined weight for each particle ( $W_i$ ) can be expressed as:

$$W_i = \prod_{k=1}^{N_m} p_{nc,i} * p_{c,i} * p_{a,i} \quad (5)$$

where  $N_m$  is the number of measurements for each particle. This weight is calculated for every particle during the update step in Algorithm 1.

### 2.3 Sensor Fusion with Extended Kalman Filter

The Extended Kalman filter is a linearized version of the Kalman Filter, a recursive continuous state observer that uses knowledge of the system and measurement models and their corresponding noises. These models can be formulated as:

$$\begin{aligned} X_t &= f_s(X_{t-1}, U_t, V_t) \\ Z_t &= f_m(X_t, W_t) \end{aligned} \quad (6)$$

where  $f_s$  is the function that defines the system dynamics, computing the current state  $X_t$  based on the value of the previous state  $X_{t-1}$  and the input  $U_t$ , and  $V_t$  represents the noise of this function. Furthermore,  $f_m$  is the function that defines the measurement system, computing the current sensor readings  $Z_t$  based on the value of the actual state  $X_t$ , and  $W_t$  represents the noise of this function. Both  $V_t$  and  $W_t$  are considered to be discrete functions representing a zero mean Gaussian disturbance, with  $Q$  and  $R$  as their respective covariances.



**Fig. 3.** Experimental setup for object pose estimation. A reference frame parallel to the wrist reference frame is shown in the lower right corner.

Using this knowledge and Eq. (1) as the state of our system, the Extended Kalman filtering process is divided into two steps:

- Prediction step: uses a previously estimated state ( $\hat{X}_{t-1}$ ), the input ( $U_t$ ) and the system model ( $f_s$ ) to predict the value of the next state, as well as the state-estimated covariance:

$$\begin{aligned} U_t &= [\Delta q_t, \Delta t_t]^T \\ \hat{X}_{t|t-1} &= f_s(X_{t-1}, U_t, V_t) = [q_{t-1} * \Delta q_t * q_v, t_{t-1} + \Delta t_t + t_v] \\ P_{t|t-1} &= \left( \frac{\partial f_s}{\partial x} \right) P_{t-1|t-1} \left( \frac{\partial f_s}{\partial x} \right)^T + Q \end{aligned} \quad (7)$$

where  $\left( \frac{\partial f_s}{\partial x} \right)$  is the Jacobian of  $f_s$  with respect to state  $X$ , and  $P_{t|t-1}$  is the estimated covariance.  $U_t$  is measured as the average movement of the fingers that are in contact with the object.

- Update step: uses the current sensor measurements (visual and tactile estimations) together with the statistical properties of the model to correct the initial estimate. Besides, the Kalman gain and state-estimate covariance are also computed.

$$\begin{aligned} K_t &= P_{t|t-1} \left( \frac{\partial f_m}{\partial x} \right)^T \left[ \left( \frac{\partial f_m}{\partial x} \right) P_{t|t-1} \left( \frac{\partial f_m}{\partial x} \right)^T + R \right]^{-1} \\ \hat{X}_{t|t} &= X_{t|t-1} + K_t (Z_t - f_m(\hat{X}_{t|t-1})) \\ P_{t|t} &= \left[ I - K_t \left( \frac{\partial f_m}{\partial x} \right) \right] P_{t|t-1} \end{aligned} \quad (8)$$

where  $K_t$  is the Kalman gain,  $\left( \frac{\partial f_m}{\partial x} \right)$  is the Jacobian of  $f_m$  with respect to state  $X$ , and  $P_{t|t}$  is the covariance of the estimation. Since the measured

properties ( $Z_t$ ) are of the same type as the predicted estate of the system ( $X_t$ ), position and orientation,  $f_m$  is just the vertical concatenation of two  $6 \times 6$  identity matrices.

These two steps are repeated for every sample:  $t = 1, 2, \dots, T$ .

The only user-configured parameters of the algorithm are the covariance matrices representing the system and sensor noise. Since we are fusing information coming from two different sensors, it is important to carefully choose the sensor noises, since the Extended Kalman filter naturally gives more importance to the signal measured by the sensor with less noise, i.e., the more reliable one. In order to choose these values, the tactile and visual readings were studied offline separately, computing average and standard deviation errors of their estimation. Noise values have been chosen to be 1/100 of the average noise in each axes, giving as a result that noise in the tactile estimation is 2 times larger for the orientation values, 1.3 larger in the Y axis, and 2.5 smaller in the X and Z axes.

Finally, the filter is executed every time there is a new reading from any of the sensors (estimations); since it is possible that not all of them are available at the same time, the last available reading is always used. This is also applied if any of the estimators (visual or tactile) loses track of the object.

### 3 Setup Description

For the experimental tests, we use the ReFlex TakkTile hand (Fig. 3). The hand is equipped with two types of sensors: pressure sensors located along the fingers (9 per finger) and the palm (11 sensors), and magnetic encoders in the proximal and distal joints, which allow computing the location of both phalanges in each finger. The position of each tactile sensor and the normal vector to the surface at its position can be constantly computed. The provided force measurements are based on the pressure transmitted by the rubber that covers the fingers. However, since the pressure flows through the rubber, one single contact with an object may be detected by two (or more) consecutive sensors. When this happens, a linear combination of measurements is performed to compute the actual contact location  $c_i$  with respect to the wrist, as follows:

$$c_i = t_i + \left| 1 - \frac{f_i}{f_i + f_{i+1}} \right| \times (t_{i+1} - t_i) \quad (9)$$

where  $t_i$  is the position of sensor  $i$  and  $f_i$  its corresponding force measurement. Note that this model assumes that there is maximum one contact with the object at each link of the fingers.

The camera used to retrieve visual information is an RGB camera with a resolution of  $640 \times 480$  at 30 frames per second. The spatial location of the hand with respect to the camera is provided by Apriltags [30] located on the top surface of the hand, as shown in Fig. 3. Note that the object also has an Apriltag on the top surface, which is used to compute the ground truth for the relative pose of the object with respect to the hand.



**Fig. 4.** Cheez-it box grasped from the side with the robotic hand held by a human operator. The sequence is ordered from left to right, and includes the pre-grasp pose, grasping, moving and releasing the object.

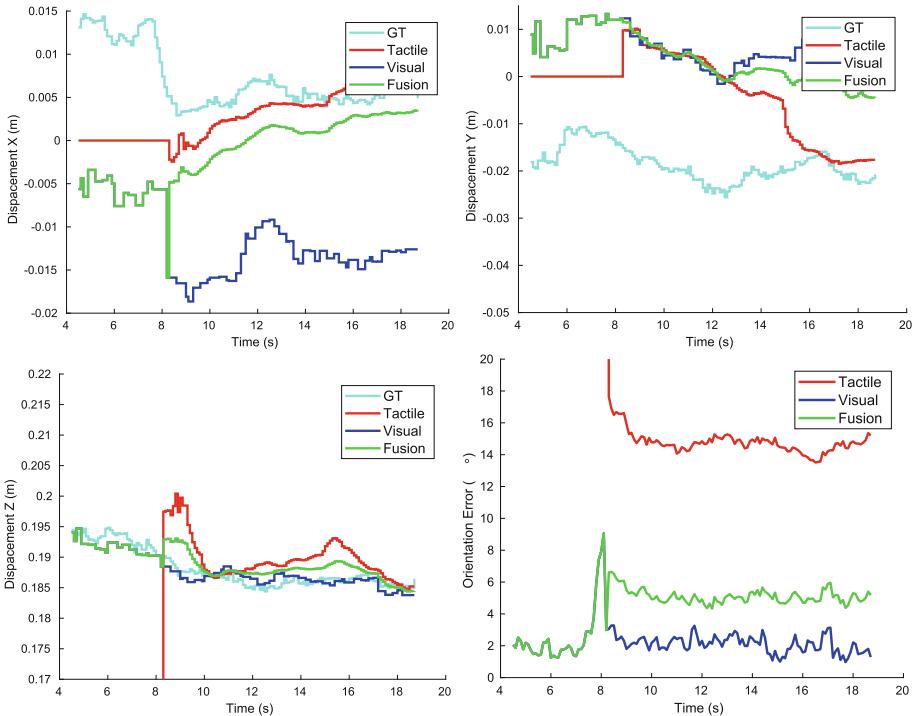
For the measurements, a human operator holds the hand and approaches the object, grasping it and releasing it. The hand could also be attached to a robotic arm; this has no influence on the results of the estimation of the hand-object relative pose nor on the in-hand object pose estimation. The performed experiments follow these steps:

- Hand and object are placed on a flat surface, and the scene is perceived with the RGB camera. The visual estimator starts looking for the object.
- An operator picks up the hand and positions it to execute the grasp.
- After 10 seconds, the fingers close toward the object. Once they are in contact, a constant closing velocity is maintained in all fingers to make the grasp stable.
- The tactile estimation is started as soon as the first contact between hand and object is detected. An initial population of particles is built (adding Gaussian noises) based on the last estimation given by the vision system. It is ended when the hand is commanded to open.
- The object is lifted by the operator. As long as both tactile and visual estimations are available, the extended Kalman filter computes the in-hand object pose.

## 4 Experimental Results

Two different objects from the YCB database [31] are used for the tests, a Pringles can and a Cheez-it box (Fig. 1). For the initial test of the concept, three test sequences were made with the can and two with the box. For each of them, pose estimation tests were run 5 times. One of the test runs can be seen in Fig. 4. From left to right, the figure shows the pre-grasp pose, grasping pose, lifting and moving the object, and hand opening.

Figure 5 shows the results of the pose estimation for one of the experiments using the Cheez-it box. For the displacements in the three axes, the ground truth (GT - light blue), the tactile (red), visual (dark blue) and the EKF-based fusion estimation (green) are shown. The orientation error around the three axis is also shown in the figure. Note that at the beginning of the movement, the EKF-based estimation is the same as the vision-based one, since there is no contact with the object yet. The first contact between the hand and the object is detected



**Fig. 5.** Evolution of the displacement and orientation error while grasping the Cheez-it box with the sequence shown in Fig. 4.

at about 8s. From that moment, the estimation using the fusion technique here presented differs from the pose estimations using only one sensing modality. In the  $X$  axis (top-left of Fig. 5), the estimation is corrected by the influence of the tactile system, while in the  $Z$  axis, the estimation is better for the vision system (tactile information does not help to pinpoint the object location along this axis for this particular object). It is possible to appreciate the initial error in the estimation of the location along the  $Y$  axis due to the vision system, which remains almost constant with time. Because the tactile system has no means of measuring changes in the  $Y$  axis and has less accuracy in estimating the changes in orientation, the resulting estimation in that case follows more closely the estimation coming from the vision system (right side of Fig. 5). Note that this is a result of the matrices used in this technique to represent the noise for each sensor modality.

In the case of the orientation error suffered by the tactile system, it is related to the initial error found along the  $Z$  axis. Because a contact between the box and the palm of the hand is detected, but the initial estimation in the  $Z$  axis is actually off by almost 2 cm, the simulation recreates the same situation by turning the box around the  $X$  axis so that this same contact is detected.

**Table 3.** Average errors and standard deviations in the pose estimation for the selected objects in different test sequences.

Object	Trial	Vision		Tactile		Fusion	
		$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
Can	1	3.72 cm	0.9	2.04 cm	0.5	3.15 cm	0.8
		0.13°	0.8	2.32°	0.6	1.00°	0.8
	2	4.03 cm	1.2	3.47 cm	0.61	3.55 cm	1.1
		7.82°	6.4	13.90°	3.3	9.65°	6.3
	3	3.03 cm	0.3	3.58 cm	0.4	3.18 cm	0.3
		2.71°	0.6	6.54°	2.3	5.28°	1.5
Box	1	2.29 cm	0.2	1.69 cm	0.7	1.78 cm	0.4
		0.64°	0.8	11.55°	2.3	2.2°	1.6
	2	3.01 cm	0.7	1.92 cm	0.6	2.03 cm	0.6
		4.15°	2.1	7.8°	2.5	4.20°	2.2

However, these errors are successfully corrected over time by the fusion of the two estimations.

Table 3 shows the average and standard deviations for the errors obtained in the different test sequences. The first test of the box corresponds to the one shown in Fig. 5. The errors are computed as an average over all the grasping action for the fused estimation, including a short time before grasping has occurred and after the object has been released. The same period is used for computing the errors for the vision system. However, for the tactile system the error only covers those moments in which there is effective contact between object and hand.

In the second test of the box and the can, the visual estimation is misled by the movement of the fingers, and this is estimated as a movement of the object itself. This results in larger errors both in position and orientation. In the third experiment of the can, the vision system is not able to track the object while it is grasped by the hand because of the occlusion of the object, and it is not able to recover until the fingers open again. This results in a worse estimation in general, first because the prior given to the tactile system is worse, but also because there are no visual corrections in those axes where the tactile system is weaker. Lastly, the first experiment conducted with the can produces very good estimations, the only error found is given by the initial error committed by the visual estimation. The magnitude of the errors described in Table 3 are in the same range of those found in similar works [19, 23, 25], and there is a clear improvement in the initial visual estimation used in our work.

## 5 Conclusions

This paper presented a 6D object pose estimation method that combines visual and tactile information. The fusion of the information provided by both sensing

modalities is performed by an extended Kalman filter. An initial experimental evaluation with real data captured with an RGB camera and a robotic hand is performed to study the integration of the two complementary sensor modalities in order to successfully reduce the overall uncertainty of the pose estimation.

Improvements to the approach presented here include a better initial visual estimation, since this error is later propagated to the fusion with the tactile information. A tracking system more robust to object occlusions would also be desirable, and experiments with objects of more complex geometries is a next step. A possible extension of this work could investigate how to avoid using explicit object models in the estimators.

**Acknowledgments.** The authors want to thank Naiara Escudero for her assistance on the implementation of the Extended Kalman Filter, and Karl Pauwels for insights given on the use of Simtrack.

The research leading to these results has received funding from RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, funded by “Programas de Actividades I+D en la Comunidad de Madrid” and co-funded by Structural Funds of the EU. This work has also received funding from the Spanish Ministry of Economy, Industry and Competitiveness under the project DPI2016-80077-R.

## References

1. Lacey, S., Sathian, K.: Visuo-haptic multisensory object recognition, categorization, and representation. *Front Psychol.* **5**, 730 (2014)
2. Macura, Z., Cangelosi, A., Ellis, R., Bugmann, D., Fischer, M., Myachykov, A.: A cognitive robotic model of grasping. In: International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, pp. 89–96 (2009)
3. Dogar, M., Hsiao, K., Ciocarlie, M., Srinivasa, S.: Physics-based grasp planning through clutter. In: Robotics: Science and Systems VIII (2012)
4. Vasconcelos, N., Pantoja, J., Belchior, H., Caixeta, F.V., Faber, J., Freire, M.A., Cota, V.R., de Macedo, E.A., Laplagne, D.A., Gomes, H.M., Ribeiro, S.: Cross-modal responses in the primary visual cortex encode complex objects and correlate with tactile discrimination. *Proc. Natl. Acad. Sci.* **108**(37), 15408–15413 (2011)
5. Petit, A., Marchand, E., Kanani, K.: A robust model-based tracker combining geometrical and color edge information. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3719–3724 (2013)
6. Choi, C., Christensen, H.I.: RGB-D object tracking: a particle filter approach on GPU. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1084–1091 (2013)
7. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3D tracking using online and offline information. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 1385–1391 (2004)
8. Kyrki, V., Krägic, D.: Integration of model-based and model-free cues for visual object tracking in 3D. In: IEEE International Conference on Robotics and Automation, pp. 1566–1572 (2005)
9. Güler, P., Bekiroglu, Y., Pauwels, K., Krägic, D.: What’s in the container? Classifying object contents from vision and touch. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3961–3968 (2014)

10. Pauwels, K., Ivan, V., Ros, E., Vijayakumar, S.: Real-time object pose recognition and tracking with an imprecisely calibrated moving RGB-D camera. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2733–2740 (2014)
11. Jamali, N., Sammut, C.: Majority voting: material classification by tactile sensing using surface texture. *IEEE Trans. Robot.* **27**(3), 508–521 (2011)
12. Madry, M., Bo, L., Krägic, D., Fox, D.: ST-HMP: unsupervised spatio-temporal feature learning for tactile data. In: IEEE International Conference on Robotics and Automation, pp. 2262–2269 (2014)
13. Aggarwal, A., Kirchner, F.: Object recognition and localization: the role of tactile sensors. *Sensors* **14**, 3227–3266 (2014)
14. Haidacher, S., Hirzinger, G.: Estimating finger contact location and object pose from contact measurements in 3-D grasping. In: IEEE International Conference on Robotics and Automation, pp. 1805–1810 (2003)
15. Chalon, M., Reinecke, J., Pfanne, M.: Online in-hand object localization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2977–2984 (2013)
16. Pfanne, M., Chalon, M.: EKF-based in-hand object localization from joint position and torque measurements. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2464–2470 (2017)
17. Álvarez, D., Roa, M.A., Moreno, L.: Tactile-based in-hand object pose estimation. In: Third Iberian Robotics Conference, ROBOT 2017, Advances in Intelligent Systems and Computing, vol. 694, pp. 716–728. Springer (2018)
18. Allen, P.K.: Integrating vision and touch for object recognition tasks. *Int. J. Robot. Res.* **7**, 15–33 (1988)
19. Ilonen, J., Bohg, J., Kyrki, V.: Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing. *Int. J. Rob. Res.* **33**(2), 321–341 (2014)
20. Alkkiomäki, O., Kyrki, V., Kälviäinen, H., Liu, Y., Handroos, H.: Complementing visual tracking of moving targets by fusion of tactile sensing. *Rob. Auton. Syst.* **57**, 1129–1139 (2009)
21. Kolycheva, E., Kyrki, V.: Task-specific grasping of similar objects by probabilistic fusion of vision and tactile measurements. In: IEEE-RAS International Conference on Humanoid Robots, pp. 704–710 (2015)
22. Zhang, M.M., Detry, R., Matthies, L., Daniilidis, K.: Tactile-vision integration for task-compatible fine-part manipulation. In: Robotics: Science and Systems. Workshop on Revisiting Contact—Turning a Problem into a Solution (2017)
23. Bimbo, J., Rodríguez-Jiménez, S., Liu, H., Song, X., Burrus, N., Senerivatne, L.D., Abderrahim, M., Althoefer, K.: Object pose estimation and tracking by fusing visual and tactile information. In: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 65–70 (2012)
24. Bimbo, J., Senerivatne, L., Althoefer, K., Liu, H.: Combining touch and vision for the estimation of an object's pose during manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4021–4026 (2013)
25. Schmidt, T., Hertkorn, K., Newcombe, R., Marton, Z., Suppa, M., Fox, D.: Depth-based tracking with physical constraints for robot manipulation. In: IEEE International Conference on Robotics and Automation, pp. 119–126 (2015)
26. Wu, C.: SiftGPU: a GPU implementation of scale invariant feature transform (SIFT). <http://github.com/pitzer/SiftGPU>
27. Lepetit, V., Fua, P.: Monocular model-based 3D tracking of rigid objects. *Found. Trends Comput. Graph. Vis.* **1**, 1–89 (2005)

28. Pauwels, K., Tomasi, M., Diaz Alonso, J., Ros, E., Van Hulle, M.: A comparison of FPGA and GPU for real-time phase-based optical flow, stereo, and local image features. *IEEE Trans. Comput.* **61**(7), 999–1012 (2012)
29. Pan, J., Chitta, S., Manocha, D.: FCL: a general purpose library for collision proximity queries. In: IEEE International Conference on Robotics and Automation, pp. 3859–3866 (2012)
30. Olson, E.: AprilTag: a robust and flexible visual fiducial system. In: IEEE International Conference on Robotics and Automation, pp. 3400–3407 (2011)
31. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.: The YCB object and model set: towards common benchmarks for manipulation research. In: IEEE International Conference on Advanced Robotics, pp. 510–517 (2015)



# Object Classification for Robotic Platforms

Samuel Brandenburg<sup>1</sup>, Pedro Machado<sup>1(✉)</sup>, Pranjali Shinde<sup>2</sup>,  
João Filipe Ferreira<sup>1,3</sup>, and T. M. McGinnity<sup>1,4</sup>

<sup>1</sup> Computational Neurosciences and Cognitive Robotics Group,  
Nottingham Trent University, Clifton Campus, Nottingham NG11 8NS, UK  
[samuel.brandenburg2016@my.ntu.ac.uk](mailto:samuel.brandenburg2016@my.ntu.ac.uk),

{pedro.baptistamachado,joao.ferreira,martin.mcginnity}@ntu.ac.uk

<sup>2</sup> INESC TEC, R. Dr. Roberto Frias, Porto, Portugal  
[pranjali.shinde@inesctec.pt](mailto:pranjali.shinde@inesctec.pt)

<sup>3</sup> Institute of Systems and Robotics, University of Coimbra, Polo II,  
Coimbra, Portugal  
[jfilipe@isr.uc.pt](mailto:jfilipe@isr.uc.pt)

<sup>4</sup> Intelligent Systems Research Centre, Ulster University,  
Magee Campus, Londonderry BT48 7JL, UK  
[tm.mcginnity@ulster.ac.uk](mailto:tm.mcginnity@ulster.ac.uk)

**Abstract.** Computer vision has been revolutionised in recent years by increased research in convolutional neural networks (CNNs); however, many challenges remain to be addressed in order to ensure fast and accurate image processing when applying these techniques to robotics. These challenges consist of handling extreme changes in scale, illumination, noise, and viewing angles of a moving object. The project main contribution is to provide insight on how to properly train a convolutional neural network (CNN), a specific type of DNN, for object tracking in the context of industrial robotics. The proposed solution aims to use a combination of documented approaches to replicate a pick-and-place task with an industrial robot using computer vision feeding a YOLOv3 CNN. Experimental tests, designed to investigate the requirements of training the CNN in this context, were performed using a variety of objects that differed in shape and size in a controlled environment. The general focus was to detect the objects based on their shape; as a result, a suitable and secure grasp could be selected by the robot. The findings in this article reflect the challenges of training the CNN through brute force. It also highlights the different methods of annotating images and the ensuing results obtained after training the neural network.

**Keywords:** Object classification · Training · YOLOv3 · CNN · ROS

## 1 Introduction

Humans can detect target objects amongst distractors nearly instantly once they have caught their attention, even in the presence of extreme changes in scale,

illumination, noise, and viewing angle of an object [14]. Adding to this, humans possess self and spatial awareness, abilities which are developed from an early age [5]. Spatial awareness is essential for humans to complete their daily activities, which may range in difficulty from reaching for a cup of coffee to playing a high-performance sport. Many of these innate capabilities in humans have still to find convincing implementation in state-of-the-art robotics – see, for example, [1, 4, 6]. This research highlights the challenges and recent advancements in detecting objects using an artificial neural network (ANN).

Recent advancements in computer vision have been driven by research on deep neural networks (DNN); however, DNNs require an extensive amount of data and pre-processing. Poor data filtering can mislead results, therefore pre-processing is a crucial procedure for proper training. Therefore, standards such as the cross-industry standard process have been created to ensure to increase the rate of success [13]. This project’s main contribution is to provide insight on how to properly train a convolutional neural network (CNN), a specific type of DNN, for object tracking in the context of industrial robotics. The paper structure is as follows: Sect. 2 presents the related work; the methodology is discussed in Sect. 3; the results and the critical analysis is done in Sect. 4; and the conclusions and future work are discussed in Sect. 5.

## 2 Related Work

The enactment of “pick and place” by a robot inherently presents many challenges, many of which at the level of artificial visual perception. Adding to these challenges, the use of ANNs to enhance visual processing involves time-consuming, complex training procedures. Furthermore, many factors, such as insufficient or low-quality training sets, can reduce their performance. For instance, an image fed into an ANN for processing may contain not only the desired object, but also other objects in the background. These distractors may degrade ANN performance if training is not properly performed. Another challenge in using ANNs is the detection of objects using a moving sensor (i.e. by resorting active perception, such as visual servoing). In this case, distortions such as motion blur may result in the detection algorithm failing to recognise the item.

**CNN Neural Networks** – Convolutional neural networks have proved over-time to be an effective algorithm for recognising visual patterns. The first model for a convolutional neural network was the leNet-5 created by Yann Lecun in 1998 [7]. This network was made up of two convolutional layers, two average pooling layers, two fully connected layers, and a softmax layer [7]. The leNet was not able to classify images, but it proved successful at classifying numbers. However, interest in CNNs was revived in 2012 when AlexNet was created which was much larger than its predecessor [3]. The network contained 5 Convolutional Layers and 3 Fully Connected Layers [3]. Training AlexNet to classify images took five to six days using two GTX 580 3 GB GPUs [3]. This breakthrough has

led to CNNs being the preferred solutions for image processing; however, real-time requirements make this version of the algorithm unsuitable for computer vision, in particular in robotics.

**Faster Region Convolutional Neural Network Algorithm** – Faster R-CNN is the current state-of-the-art optimisation method for CNN algorithms, designed to achieve real-time performance. Fast R-CNN and its predecessor used a technique called selective search [12]. Faster R-CNN, however, uses a technique called region proposal networks (RPN), which are much faster. RPN take an image as an input and output sets of anchor boxes of proposed objects; these objects are then associated with a score [11]. Faster R-CNN works by first creating a feature map from the input image. Subsequently, the RPN generates a set of proposed objects together with their score. Like its predecessor, Faster R-CNN still uses the ROI layer to make the proposed regions a fixed length. Once all the regions are of the same size, they are passed to the fully connected layer where SoftMax and linear regression are applied. The image is then classified and the algorithm outputs bounding boxes for the objects [12]. As a consequence of this optimisation process, Faster R-CNN decreases the time it takes to detect an image from 2s to 0.2s [11].

**YOLO Algorithm** – YOLO, which stands for “you only look once”, was first introduced in 2015 by Redmon et al. [8] as alternative to R-CNN, which had complex pipelines that made it slow and hard to optimise. Unlike R-CNN, YOLO looks at the full image once. It uses a single CNN that predicts multiple bounding boxes and class probabilities for those boxes simultaneously [8]. YOLO predicts the score of an object using logistic regression for each bounding box. The proposed network was composed of 24 convolutional layers, four max-pooling layers<sup>1</sup>, and two fully connected layers [8].

There are several benefits associated with YOLO. First of all, as mentioned previously, YOLO is faster compared to other detection methods because it avoids complex pipelines by framing it as a regression problem. YOLO encodes contextual information about classes and their appearance because it views the entire image for training and testing. Furthermore, it outperforms algorithms like fast R-CNN in making fewer background errors – Fast R-CNN mistakes background patches as objects because it does not have the full context of an image. Lastly, YOLO can generalise features it learns from objects, which can then be applied to new items.

At the time YOLO was introduced, it was the fastest general purpose object detection algorithm when compared to R-CNNs, Fast R-CNNs, and two versions of Faster R-CNNs [8]. The first was the VGG-16, which was ten mean Average Precision (mAP) higher but six times slower than YOLO [8]. The other was Zeiler-Fergus Faster R-CNN [8]. This model was much faster than the previous one, but it was not as fast and accurate as YOLO.

---

<sup>1</sup> Max pooling is a technique that extracts the most significant features from the convolutional layer.

Additional developments have been made in the meanwhile to further improve YOLO's performance. In December 2016, YOLO version 2 was launched. This version differs from its predecessor because it uses a classification model called Darknet19 [9]. Darknet consists of 19 convolutional layers and 5 max-pooling layers [9]. The purpose of Darknet is to increase the speed and accuracy of classifications. Although the original YOLO algorithm outperformed previous object detectors, it under-performed in accuracy compared to Fast R-CNN by introducing a considerable amount of localisation errors [9]. Another aim was correcting the low recall YOLO produced compared to the region proposal-based method [9]. There were many ways YOLO version 2 addressed these problems. The first was to use batch normalisation, which significantly improved convergence while removing the need for other forms of regularisation. Another improvement made was to add high-resolution classifiers, which provided a 4% increase in mAP [9]. YOLO was susceptible to unstable gradients during training, therefore Anchor boxes reduced mAP slightly from 69.5 to 69.2, but the recall improved from 81% to 88% [9]. In other words, even if accuracy was slightly decreased, it increased the chances of detecting all the ground truth objects.

YOLO version 3 (YOLOv3) is the latest version that focuses on improvements on object classification [10]. Methods such as Single Shot Detection still outperform YOLOv3 in terms of accuracy; YOLOv3, however, executes three times faster for the same input [10]. Furthermore, YOLOv3 makes a considerable improvement in how well it can detect small objects [10]. Finally, this framework now uses Darknet 53, an enhancement to Darknet19, and increases the scale for feature extraction by increasing the number of convolutional layers from 19 to 53 [10].

### 3 Methodology

The object classification methodology was designed for robotic grasping applications. In the particular case of the authors, a Sawyer robotic arm<sup>2</sup> equipped with an AR10 hand<sup>3</sup> and state-of-the-art Biotac SP fingertips (see footnote 3) was target. However, only the thumb, index finger, and middle finger were used with Biotec sensors installed at the end of them. The main task was to grasp objects using 3 of the 5 fingers in the hand using an adaptive grasp controller. It was decided that those items would be of a ball, a Pringles can, a sponge, and a water bottle (Fig. 1). These items were chosen because each object varied in shape and size; Therefore, the robotic arm would need to change the grasping method for each of the items.

For this to be accomplished, the first objective was to detect the objects. Once the item had been detected, the gripping algorithm could be appropriately adjusted, and the robotic arm and hand guided to perform the grip. The first step

---

<sup>2</sup> Retrieved from <https://www.syntouchinc.com/en/sensor-technology/>, last accessed 2019-06-20.

<sup>3</sup> Retrieved from <https://www.active8robots.com/robots/ar10-robotic-hand/>, last accessed 2019-06-20.



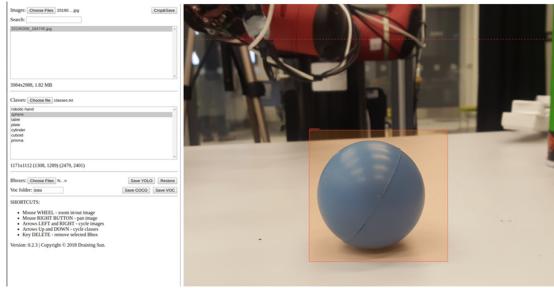
**Fig. 1.** Objects used for training the YOLOv3

to installing and configuring the YOLOv3<sup>4</sup> and adjust specific convolutional layers to match the number of classes, and filters to the number of classes designed. The selected classes were robotic hand, table marks, sphere, cuboid, prism and cylinder. These classes were extracted from a newly constructed database of images that was taken with an Intel D435 real sense camera. The robotic hand and the table marks are required for navigating the arm toward the target object, and the 3D shape of the objects is required for selecting the pre-grasp and pose of the objects. The aim of this paper in the object detection and therefore the details of the navigational algorithms, pre-grasp and pose estimations are not addressed in the paper. Annotation for training was performed by manually labelling each object in each image in the training data-set using a custom annotation tool developed by the authors<sup>5</sup>. This method of annotating classes makes up the majority of the pre-processing phase. The CNCR annotation tool opens a window browser (see Fig. 2) and lets the user import images and classes. The user can then label objects by defining a bounding box. The box parameters are then recorded in a text file. Each row contains five columns: the first indicates the index for the different classes, the second and third columns specify the location of the object in an  $X$  and  $Y$  grid, respectively, and the last column stores the height and width of the object. Training YOLOv3 can start once a dataset of images and their annotations have been formed; however, preventive measures must be put in place in order to prevent underfitting and overfitting [2]. Overfitting, in particular, prevents the model from generalising to unknown data and thus leading to poor performance. A standard way to prevent overfitting and underfitting is to construct a validation set, i.e. by defining distinct training and test subsets from the images collected in the training phase.

---

<sup>4</sup> Retrieved from [https://medium.com/@manivannan\\_data/how-to-train-YOLOv3-to-detect-custom-objects-ccbcafeb13d2](https://medium.com/@manivannan_data/how-to-train-YOLOv3-to-detect-custom-objects-ccbcafeb13d2), last accessed on 25/04/2019.

<sup>5</sup> Available online, [https://gitlab.com/CNCR-NTU/CNCR\\_annotation\\_tool](https://gitlab.com/CNCR-NTU/CNCR_annotation_tool), last accessed on the 15/06/2019.



**Fig. 2.** CNCR annotation tool

The next step of the process was training the model. To train the YOLOV3 model the names of the classes, the configuration file, the data file, and the YOLOv3 architecture were needed, taking approximately two days to conclude the procedure. The YOLOv3 creates weights in increments of 10k steps during this period; however, adjustments were made to have weights created every 2k steps (this value was an empirical value obtained experimentally). The trained weights were then tested on the testing data and the weights that produced better results were selected.

## 4 Results

During the pre-processing phase, it became apparent that many factors influenced object detection performance when annotating. The majority of this section will highlight the methods used while annotating. Furthermore, this section will convey their results. Over ten training cycles have been conducted during the implementation phase. The overall goal of accomplishing computer vision was to detect and generalise similar objects.

A significant part needed for visual servoing was detecting the parameters of the table. The table was marked on each corner by a strand of red tape. The first objective was producing a stable object detection model. Therefore, detecting the medium size ball was the first goal. Key points that were learned during this objective was the training dataset needed to contain over 200 images. Furthermore, the performance of the model varied depending on the weights used. There were performance issues with each of the weights used. However, 10k weights gave the model more accuracy and stabilised detection. Below displays results of the first test 1 (Table 1).

It took two training sets before obtaining positive results. However, it was found that the ball would not be detected if it was moved too far to the right or left on the table. Another downside was the size of the bounding box on the item. The width of the bounding box was similar to that of the annotations. A smaller box would be ideal. As a result, the next set of annotations contained small boxes covering the item versus one larger box. It was seen that smaller boxes could reduce the amount of background noise (see Table 2).

**Table 1.** Accuracy readings from .5 – 1.0 using different weights

Item	10 k weight	12 k weight	16 k weight	20 k weight	30 k weight
Ball	.98	.98	.99	.95	.91

**Table 2.** Detection with the background changing

10 k weights percentage	Table background (known)	Mat background (known)	Unknown background
Sphere	1.00	.95	N/A
Cuboid	.86	.82	N/A
Cylinder	.96	.98	N/A

Additionally, smaller annotation boxes would decrease the size of the bounding box. For instance, if annotation were being done for an object in being grasped by the robotic hand, The boxes outlining the hand would shrink, but increase in number to only contain the hand. This method brought abysmal results because the model became overfitted. The model projected bounding boxes over the whole image. In order to correct this problem changes needed to be made to the labelled files. It was discovered that an object should contain no more than two annotation boxes around it. Creating a limit helped prevent overfitting. It also made the ball detectable again. The previous issue still resided, but it lost the object less than before<sup>6</sup>.

The next step was detecting multiple objects at the same time. The can and the ball needed to be detected. Achieving this objective was relatively easy. This was accomplished by increasing the number of images in the training set. It was also important that there was as less background noise as possible. However, more issues appeared when all the items needed to be annotated. The difference was the positioning of the items in the image. Merging annotations occurred very often because the items were too close. Another method would be to highlight the sections that would not overlap the images. The issues with that approach are that it does not get the full object. Therefore, it may degrade results on that one item if partial annotations are continuously being done.

At this point, the detection algorithm was performing well on the ball, the Pringles can, and the sponge with a confidence score over 0.7. However, it is important to note that some items may consist of a combination of different shapes. Therefore, the classifying process for such an item must contain different classes in an item. This was the case for the bottle. When annotating an item like a bottle, it was important to match each shape with the class that fitted best. Breaking the item into different identifiable shapes assist the process of picking the item up. However, the first attempt at annotating multiple shapes

<sup>6</sup> Available online <https://www.youtube.com/watch?v=vdDqMtdyUYU>, last accessed on 25/04/2019.

had different classes merging into another. Thus when testing this model using 10k weights, no items were being classified. It was identified that the overlap of labelling boxes substantially increases the number of false positives. Thus the solution was the avoidance of overlapping annotation boxes. This strategy worked, and the detection algorithm began working for all the objects. The next test was to evaluate the generalisation of spheres. This test involved using more than ten spheres with different sizes and colours. However, only two spheres were detected. The training sphere was detected, and another sphere similar in colour but different in size. After receiving abysmal results, further tests were conducted to understand why the generalisation of spheres failed. This test involved using a lid that had a sphere shape with a light blue tint similar to the other two spheres. When the item laid flat, it was not detected, but when it was standing upright, it was detected as a sphere. These results demonstrated that (i) the algorithm recognised objects similar in shape and colour to that it was trained on and (ii) that increasing the number of different samples was crucial to enable the CNN to start generalising patterns to unfamiliar objects. The next step was to create the environment so that visual servoing could be used since all the objects were being detected. This process involved mapping out the boundaries of the table by adding markers to each corner. After training the model to recognise these tags, It was found that the two nearest markers were detected, but not consistently. Furthermore, the furthest two markers failed to be detected. Despite, adding more annotations outlining all the table markers, the problem was not fixed. At the same time, these annotations included the placement of the robotic arm at different positions of the table. It was later discovered that the lighting of the room and shadows created by the arm was the reason behind the inconsistent detection.

After all of the objects were successfully detected, training the CNN to recognise the robotic arm and table marks was next. This is needed for moving the arm from a starting point to the object pick-up place. The process involved mapping out the boundaries of the table by adding markers to each corner. After training the model to recognise these tags, It was found that the two nearest markers were detected, but not consistently. Furthermore, the furthest two markers failed to be detected. Despite, adding more annotations detection of all the table markers was not fixed. At the same time, these annotations included moving the robotic arm across the table when conducting test on its' range of motion. It was later discovered that the lighting of the room and shadows were the reason certain items stopped being detected. This was especially true when the robotic arm was moving since its' arm made a shadow. As mentioned before in the literature review illumination changes make perfect object detection difficult. One method that helped alleviate these issues were annotating objects while in different illuminations. This method proved it had the potential to work; however, annotating every angle and circumstance was too tedious of a task.

Other interesting facts learned while testing the customised model of YOLOv3 was objects needed to be moved around. If an object stays in the same position for all the images, it will not be detected once it is moved. This

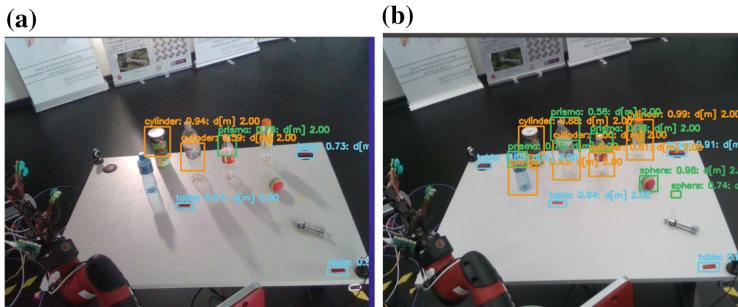
was discovered while testing on a set of spheres. The initial training set contained images where the position of the spheres did not move. It was found that annotation files could be automated by copying the parameters in an annotated file. Those parameters were then pasted into new text files that were attached for the remaining training files. However, this method proved unsuccessful. Another aspect that affects the model's performance is when changes are made to the background. This theory was tested by placing a sphere on a black mat. The sphere was detected while on the table, but it was not when moved to the mat. Furthermore, the bounding box for an object would disappear if anything went near the item. This was another challenge to resolve. The robotic requires consistent detection while reaching for the item. If the item is no longer detected than the robotic arm may fail to grasp the item correctly. One way of resolving this issue was including the robot hand near the item and around the item.

The current model was tested using a range of weights to determine which one performed the best. From a visual perspective, 10k weights detect more items and it also was more consistent. Furthermore, it also had the highest accuracy percentage from all the test which was 85%. 30k weights gave the model its' second highest percentage at 83%. Nevertheless, it still struggles to detect the table marker furthermost away on the left-hand side. It also does not detect the robot hand consistently. Whereas, the other weights in the table detected the robot hand consistently. However, their accuracy percentage was also slightly less accurate with the lowest being 20k weights at 77%. Nevertheless, the current model has shown that it is capable of detecting objects it has seen (Table 3). Another positive result is that the current mode can detect up to 92% of the items. It also can detect the hand while its moving better than it initially could. However, some of the issues mentioned in the related worked section. When the illumination changes, the detection rate drops to 46%. This refers to inconsistent detection and tracking. This experiment exhibited many of the challenges written about in the related work section. In addition to the challenges presented from changes illumination, detection performance drops when viewing objects from different angles. Video footage was taken with multiple cameras from different positions. The main camera was able to detect the items from up above, but only two of the three side cameras detected anything<sup>7</sup>. This project also revealed how important annotating was and why there is a need for large datasets. This is to say that the large dataset tries to encompass an extensive range of scenarios. If the dataset contains these, the model can then begin to generalise this information to unknown objects.

**Table 3.** Illumination impact

Labels	Table	Cylinders	Prism	Sphere	Mislabeled
With light	4/4	5/5	3/4	1/1	1
Reduced lighting	3/4	2/5	1/4	0/1	0

<sup>7</sup> Available online <https://www.youtube.com/watch?v=IzN3kp7eAuY>, last accessed on 25/04/2019.



**Fig. 3.** Object detection with (a) reduced lighting and (b) normal light

Figure 3 illustrate how much performance is reduced from illumination changes. In the image above it can be seen that most of the objects are detected. However in the second image the performance drops considerably. This is just one example of how variations can impact classification.

## 5 Conclusions and Future Work

This project proposed a combination of techniques to replicate a pick and place task using a robotic arm. YOLOV3, a state of the art detection algorithm, was the central algorithm used for this work. More specifically, a customised version of YOLOV3 was used, trained on over 2,000 training images. The assessment of this model showed that it could detect up to 92% of the trained objects. However, real-time results are inconsistent; therefore, this model can sometimes achieve a higher percentage rating. Furthermore, detection in different angles along with generalising objects classification was performed. Despite these achievements, the proposed methods for this project heavily depended on stable object detection and tracking which was not achieved. This was especially needed to implement the intended visual servoing practices. Consistent object detection of the table was to be used to guide the robotic arm to find items using triangulation. Unfortunately, the act of moving robotic arm created many fluctuations that affected the appearance of the object. As a result, detecting and tracking those objects was very inconsistent. It was also discovered that the weight for the model made a significant impact on the model performance. Four tests were conducted to discover the optimum amount of weights were needed. This assessment highlighted that the model showed the best results at 10k weights. This change to 10k would increase object detection accuracy to 85%. However, If the number of weights was to decrease the model performance would reduce because it is underfitted. In contrast, if the weights were too high, the model would be overfitted. Both of these issues led to poor performance.

Therefore, at first glance we conclude that, although over 2,000 images were annotated, stable object detection will require many more training samples. However, this is far too time-consuming; therefore, other methods to achieve this objective should be explored in future work. However, the most significant

drawback to the proposed method was to require consistent detection of the corners for trajectory planning. A more straightforward solution would have been to collect the coordinate of the location from one triangulation. This would have solved the issue of inconsistent detection. Another potential solution could have been to use object detection to identify if an object is on a designated spot. The robotic arm could then react by reaching for the item. The implementation for this process would be hard coded thus simplifying many of the issues that are associated with visual servoing. Yet another alternative would be to implement one-shot learning on a pre-trained neural network. This could be done using the weights provided by YOLOv3 which would alleviate the tedious task of manual pre-processing thousands of images. Additionally, it may be better to choose a different method of visual servoing. Finally, off-the-shelf open source software such as VISP exists that allows a user to form a spatial map, allowing for even potential improvements to explore in future work.

## References

1. Aggarwal, C.C.: Convolutional neural networks. In: Neural Networks and Deep Learning, pp. 315–371. Springer, Cham (2018). [http://link.springer.com/10.1007/978-3-319-94463-0\\_8](http://link.springer.com/10.1007/978-3-319-94463-0_8)
2. Brownlee, J.: Overfitting and Underfitting With Machine Learning Algorithms (2016). <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>
3. Das, S.: CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more... (2017). <https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>
4. Ferreira, J.F., Dias, J.: Attentional mechanisms for socially interactive robots – a survey. *IEEE Trans. Auton. Mental Dev.* **6**(2), 110–123 (2014)
5. Jafri, R., Aljuhani, A.M., Ali, S.A.: A tangible interface-based application for teaching tactile shape perception and spatial awareness sub-concepts to visually impaired children. *Proc. Manuf.* **3**(Ahfe), 5562–5569 (2015). <https://doi.org/10.1016/j.promfg.2015.07.734>
6. Kräig, D., Gustafson, J., Karaoguz, H., Jensfelt, P., Krug, R.: Interactive, collaborative robots: challenges and opportunities. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018), pp. 18–25 (2018)
7. Kuo, C.C.: Understanding convolutional neural networks with a mathematical model. *J. Vis. Commun. Image Representation* **41**, 406–413 (2016)
8. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016). <https://doi.org/10.1109/CVPR.2016.91>
9. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, January 2017, pp. 6517–6525 (2017)
10. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement (2018)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)

12. Sharma, P.: A Step-by-Step Introduction to the Basic Object Detection Algorithms (Part 1) (2018). <https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/>
13. Wirth, R.: CRISP-DM: towards a standard process model for data mining. In: Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining, pp. 29–39 (2000)
14. Yazdi, M., Bouwmans, T.: New trends on moving object detection in video images captured by a moving camera: a survey. *Comput. Sci. Rev.* **28**, 157–177 (2018). <https://doi.org/10.1016/j.cosrev.2018.03.001>

# **Machine Learning in Robotics**



# Effects of a Social Force Model Reward in Robot Navigation Based on Deep Reinforcement Learning

Óscar Gil<sup>(✉)</sup> and Alberto Sanfeliu

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain  
[{ogil,sanfeliu}@iri.upc.edu](mailto:{ogil,sanfeliu}@iri.upc.edu)

**Abstract.** In this paper is proposed an inclusion of the Social Force Model (SFM) into a concrete Deep Reinforcement Learning (RL) framework for robot navigation. These types of techniques have demonstrated to be useful to deal with different types of environments to achieve a goal. In Deep RL, a description of the world to describe the states and a reward adapted to the environment are crucial elements to get the desire behaviour and achieve a high performance. For this reason, this work adds a dense reward function based on SFM and uses the forces in the states like an additional description. Furthermore, obstacles are added to improve the behaviour of works that only consider moving agents. This SFM inclusion can offer a better description of the obstacles for the navigation. Several simulations have been done to check the effects of these modifications in the average performance.

**Keywords:** Robot navigation · Deep Reinforcement Learning · Social Force Model · Dense reward function

## 1 Introduction

In Robotics research one of the most important task is improve the ability of a robot to travel to a goal or destination. This ability depends on the challenging that the environment can be, the information about the environment that the robot can obtain and the algorithms which the robot uses to navigate. These factors limit the applicability in most cases.

The situations that can be found in a navigation task are wide and very complex. Different traditional methods have been developed based on potential fields, forces, or other features to solve the task. Actually, one of the approaches to get this purpose is to use Deep Reinforcement Learning techniques.

---

Work supported by the Spanish Ministry of Science and Innovation under project ColRobTransp (DPI2016-78957-RAEI/FEDER EU) and by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656). Óscar Gil is also supported by Spanish Ministry of Science and Innovation under a FPI-grant, BES-2017-082126.

These techniques have been used to teach the robot to take the correct actions during navigation but it is difficult to achieve a good performance in complex environments [25]. In the particular case of facing complex obstacles SFM can be useful to describe the environment better and avoid them.

In the remainder of the paper, preliminaries and related work is presented in Sects. 2 and 3. In Sect. 4 the problem formulation is exposed. Section 5 explains a new type of reward based on Social Force Model and Sect. 6 presents the simulations and the results. Finally, conclusions are discussed in Sect. 7.

## 2 Preliminaries

### 2.1 Social Force Model

For simulate pedestrians motion it is important to consider the physical forces caused by the environment like friction forces to walk, gravity forces or forces in a collision but it is equally important to consider the influence of other pedestrians and obstacles in social rules to walk. For example, when a person who is walking see an obstacle change the trajectory before be near the obstacle to avoid a possible collision. In this process the obstacle does not apply forces to the person but the person reacts as if a force exists.

This type of “virtual forces” that do not exist but can be a way to model the social interactions are known as the Social Force Model [14]. This model simulates the social interactions when a person walks to a determinate goal. There are 2 types of forces in this model applied to an agent  $n$  (robot or pedestrian). One is the attractive force to the goal,  $\mathbf{q}_n^{goal}$ ,

$$\mathbf{f}_n(\mathbf{q}_n^{goal}) = k (\mathbf{v}_n^0(\mathbf{q}_n^{goal}) - \mathbf{v}_n), \quad (1)$$

where  $k$  is a constant,  $\mathbf{v}_n^0(\mathbf{q}_n^{goal})$  is the preferred velocity to the goal and  $\mathbf{v}_n$  is the current agent velocity. The second type of force is the repulsive force of an obstacle or pedestrian  $z$  to an agent  $n$ ,

$$\mathbf{f}_{z,n}^{int} = a_z e^{(d_z - d_{z,n})/b_z} \hat{\mathbf{d}}_{z,n}, \quad (2)$$

where  $a_z, b_z, d_z$  are parameters of the forces,  $d_{z,n}$  is the distance between the agent  $n$  and the obstacle or pedestrian  $z$  and  $\hat{\mathbf{d}}_{z,n}$  is the unitary vector in the direction defined by  $n$  and  $z$ , pointing to  $n$ . The resultant force applied to the agent is:

$$\mathbf{F}_n = \mathbf{f}_n(\mathbf{q}_n^{goal}) + \sum_{z=1}^Z \mathbf{f}_{z,n}^{int}. \quad (3)$$

Several works have been developed using the SFM in order to get socially acceptable trajectories for robots in diverse situations like side-by-side or approaching tasks [7, 8, 20] adding elements to incorporate anisotropy or other goals.

In this work the forces are calculated only for the robot and the resultant force is calculated only with the repulsive forces. The attractive force it's only used in a new type of reward.

## 2.2 Optimal Reciprocal Collision Avoidance (ORCA)

Reciprocal Velocity Obstacle (RVO) and Optimal Reciprocal Collision Avoidance (ORCA) are reactive methods to avoid collisions between moving agents [2, 26]. In ORCA, the current velocities and positions of all agents in an environment are used to calculate the new velocities that all agents should have to avoid all the possible collisions.

The advantage of this method is that the ratio of collisions is zero by definition because ORCA chooses the velocities to avoid collisions.

On the other hand, if an agent is not controlled by ORCA, it cannot be guaranteed that the collision with this agent is prevented. For this reason, ORCA is not very useful to control an agent like a robot that moves with independent agents like people or animals. Additionally, ORCA has problems when an agent has to avoid a concave obstacle. This problem limits the type of obstacles that can be used in the simulations.

In this paper the RVO library with ORCA is used to simulate the environment with agents and obstacles. The robot is controlled with ORCA in the Imitation Learning steps to learn the basic movements to the goal.

## 3 Related Work

In this section, different works related to this paper are briefly described.

### 3.1 The Reward Shaping Problem

It is a very common problem in Reinforcement Learning tasks decide what type of reward choose to improve the training process. This is known as reward design or reward shaping.

Reward shaping is used to decrease the temporal credit assignment problem. This occurs when the agent only receives reward in a goal that must achieve and cannot use this information to know what are the best individual actions to reach the goal.

Several forms of reward shaping can be used to handle the temporal credit assignment [10, 19, 28]. It is difficult to get a proper reward that not get stuck in local minima. In this paper it is proposed a different reward transforming a sparse reward into a dense reward based in the SFM.

### 3.2 Navigation Based on Deep Reinforcement Learning

Supervised Learning techniques have been applied to predict the movements of agents in a temporal horizon. For example, sequence models that use Long Short Term Memory (LSTM) recurrent neural networks like Social LSTM [1] and other [12, 13] are capable to encode the Human-Robot interactions and Human-Human interactions to improve the predictions. Other techniques are based in generative models like Social-GAN [11] or SoPhie [21] that use pooling modules and attention modules.

This type of techniques offers very useful information in navigation tasks but not take into account all the elements in navigation tasks like obstacles, actions, kinematics or goals.

For cover a more general point of view in the robot navigation task, Reinforcement Learning techniques are a more suitable way to achieve a good performance. Normally, these techniques use neural networks due to the very high number of states to make the computation possible and, for that reason, in these cases are Deep Reinforcement Learning techniques.

In particular, here it is used the Temporal Difference (TD) algorithm [22] using a neural network to calculate the value function. The aim of this method is to use the next prediction to actualize the current prediction before reach the final state taking into account that the states are correlated. This method gives very good results in areas like finance [27] and games [24].

Several works [4–6, 17] for navigation have been developed using these methods to learn policies. Sometimes, these methods combine visual information to improve predictions like PoliNet [15]. Another method [9] combines Probabilistic Roadmaps and a RL based local planner to guide the robot for long-range indoor navigation.

There are other works that use Imitation Learning approaches to learn policies [16, 18, 23]. In the Crowd-Robot Interaction (CRI) approach [3], in which is mainly based the navigation in this paper, Imitation Learning is combined with the Deep Reinforcement Learning task to obtain a policy (SARL) for robot navigation. For Imitation Learning, the policy used to control all the agents is ORCA. Deep V-learning algorithm is used in the Imitation Learning and Reinforcement Learning phases with a temporal-difference method, a fixed target network and standard experience replay. In contrast with this work, in the CRI approach obstacles have not been used in the training. It causes some problems in the robot movement when obstacles are added to test the policy.

## 4 Problem Formulation

In this work, it is considered the same navigation task as in the CRI approach. A robot navigates to a goal through an environment with  $n$  agents. The difference is that there are obstacles considered too. The objective is to learn a policy to decide the velocities for navigate to a goal avoiding the agents and obstacles with a velocity close to a preferred value and taking socially acceptable trajectories.

The state of the robot is defined with  $\mathbf{s}$ . The states of the agents are represented in  $\mathbf{w} = [w_1, w_2 \dots w_n]$ . In this list of agents, obstacles are included too like agents with velocity equal to zero. All these states are rotated to the robot centred coordinate system with the  $x$  axe pointing to the goal. In this coordinate system the states, which proceed from the CRI work, are this,

$$\begin{aligned}\mathbf{s} &= [d_g, v_{pref}, \theta, r, v_x, v_y], \\ \mathbf{w}_i &= [p_x, p_y, v_x, v_y, r_i, d_i, r + r_i],\end{aligned}\tag{4}$$

where  $d_i = \|\mathbf{p} - \mathbf{p}_i\|_2$  is the robot's distance to the agent  $i$ ,  $d_g = \|\mathbf{p} - \mathbf{p}_g\|_2$  is the robot's distance to the goal,  $\mathbf{p} = [p_x, p_y]$  is the agent or obstacle position,

$\mathbf{v} = [v_x, v_y]$  is the current velocity,  $v_{pref}$  is the preferred velocity of the robot,  $\theta$  is an angle different to zero for non-holonomic kinematic cases,  $r_i$  is the agent or obstacle radius and  $r$  is the robot radius.

In this work it is used a second type of states that include a SFM description of the agents and obstacles introducing the repulsive forces  $\mathbf{f}_i = [f_x, f_y]$  (described in Sect. 2.1) of each agent and obstacle applied to the robot. These forces are included in the agent states and obstacle states. In the robot state it is included the resultant force of all the agents and obstacles,  $\mathbf{F} = [F_x, F_y]$ . These states are,

$$\begin{aligned}\mathbf{s} &= [d_g, v_{pref}, \theta, r, v_x, v_y, F_x, F_y], \\ \mathbf{w}_i &= [p_x, p_y, v_x, v_y, r_i, d_i, r + r_i, f_x, f_y].\end{aligned}\quad (5)$$

The states for all the agents and obstacles are always known and used to calculate the value function using a neural network and the Temporal Difference algorithm of the CRI work [3]. With this value function the greedy policy is calculated, which gives in this case the robot's velocity (the optimal action) each timestep.

## 5 Social Force Model Reward

In this work two types of rewards are used to guide the robot to the correct behaviour. The first reward is the reward used in the CRI approach [3]:

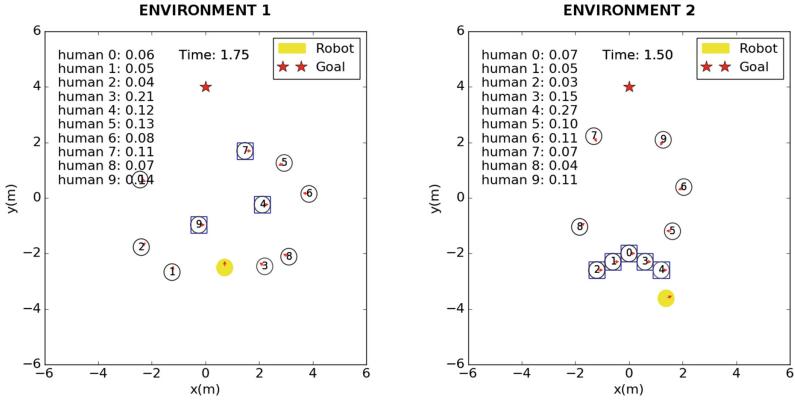
$$R = \begin{cases} -0.25 & \text{if } d_t < 0 \\ 0.25(-0.1 + d_t/2) & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \mathbf{p} = \mathbf{p}_g \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

where  $d_t = d_i - r - r_i$ , represents the distance between agents minus the radius of the 2 agents.

The second reward is a dense reward based on the SFM described in Sect. 2.1. This Social Force Model Reward takes into account the attractive force to the goal and the repulsive forces of the agents and obstacles,

$$R = \begin{cases} -0.25 & \text{if } d_t < 0 \\ Ae^{-Bd_t} & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \mathbf{p} = \mathbf{p}_g \\ k - \|k(\mathbf{v} - \mathbf{v}_{pref})\|_2/2 - 0.0001d_g & \text{otherwise} \end{cases}, \quad (7)$$

where  $A = -0.03$ ,  $B = 10$  and  $k = 0.001$ . These values have been chosen looking for rewards that are not very different from the first reward. The first term of the reward is applied in collisions, the second term is applied when an agent or obstacle is very near, the third term is applied if the robot get the goal and the last term is equivalent to an attractive force that gives more reward if the velocity is pointing to the goal and the robot is near to the goal. A variation of this reward it is used in a version with a little temporal discount,  $-0.02(t - 10)$ , if  $t \geq 10$  in the third term. This discount is chosen to be not very high and taking into account the average time needed to reach the goal in the best cases.



**Fig. 1.** First and second environment. In the first environment obstacles are separated. In the second case, obstacles are joined into a concave barrier.

## 6 Experiments

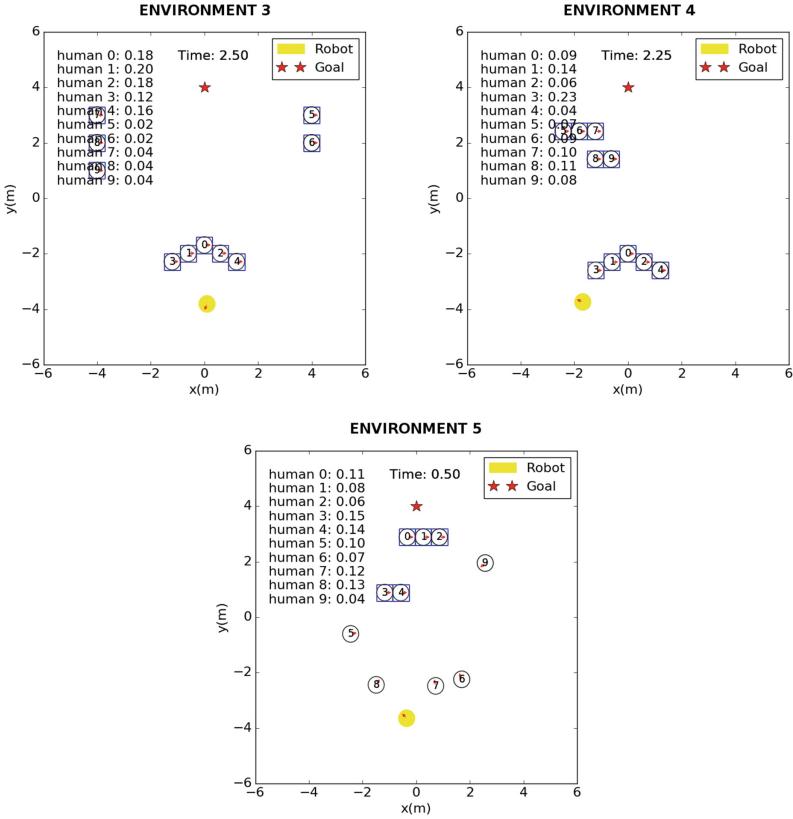
In this section are described the environment used in the simulations, the metrics to compare different versions, the results and the implementation details. No real experiments have been conducted.

### 6.1 Simulation Environment

The environment in the simulations is created using gym and the RVO library and plotted using matplotlib library like in the CRI work. It contains a robot in an initial position  $(0, -4)$  and the robot's goal in  $(0, 4)$ . The agents and the robot are represented like circles using the same radius.

The obstacles are squares whose side is equal to the diameter of the agents and have an agent inside. This agent inside the obstacle has velocity zero and gives the state's information of the obstacle,  $w_i$ . This is to achieve that the robot be aware of the obstacle when is controlled by the SARL policy. The square obstacle is generated to achieve that the agents controlled by ORCA distinguish between agents and obstacles.

There are 5 types of environments that always contain 10 elements generated randomly as agents or obstacles. In the first environment, shown in Fig. 1, the agents are generated in a circumference with their goals in the opposite side. The probability of generate an agent is 0.6 and 0.4 for an obstacle. The second environment, shown in Fig. 1, has always 5 obstacles joined into a fixed concave barrier and 5 agents in a circumference with their goals in the opposite side. The other environments in Fig. 2 use combinations of the concave barrier and barriers of 2 or 3 obstacles. In environment 3 the concave barrier can be in different positions between the robot and the goal.



**Fig. 2.** Other environments. Environments from the third to the fifth.

## 6.2 Metrics

In this section are explained the metrics used to measure the performance of the navigation. These metrics are the same than in the CRI work.

- **Success rate:** This value measures the ratio of times that the robot reaches the goal.
- **Collision rate:** This value measures the ratio of times that the robot collides with agents or obstacles.
- **Navigation Time:** Time used for the robot to reach the goal. The maximum time is 25 s.
- **Total Reward:** The average cumulative reward of all the episodes.

The most meaningful metrics are the success rate and collision rate because brief on the most important features of the navigation task.

**Table 1.** Test results. Average results obtained in the test evaluation after 500 episodes. All these cases are obtained using only the two first environments.

Version	Success rate	Collision rate	Navigation time	Total reward
SARL(2 times)	<b>0.96/0.97</b>	0.02/0.01	<b>10.85/11.77</b>	<b>0.3059/0.2850</b>
SARL-SFM	0.91	0.03	11.83	0.2417
SARL-SFM2	0.86	0.02	11.92	0.2363
SARL-SFM3	0.93	0.01	11.27	0.2741
SARL-SFM4	0.90	<b>0.00</b>	11.61	0.2646
SARL-SFM5	<b>0.96</b>	0.03	11.69	0.2799
SARL-SFM6	<b>0.96</b>	0.03	11.25	0.2713

**Table 2.** Second test results. Average results obtained in the test evaluation after 1000 episodes using all the environments in training.

Version	Success rate	Collision rate	Navigation time	Total reward
SARL	<b>0.901</b>	<b>0.001</b>	<b>10.54</b>	<b>0.3046</b>
SARL-SFM6	0.893	0.004	10.88	0.2763

### 6.3 Quantitative Evaluation

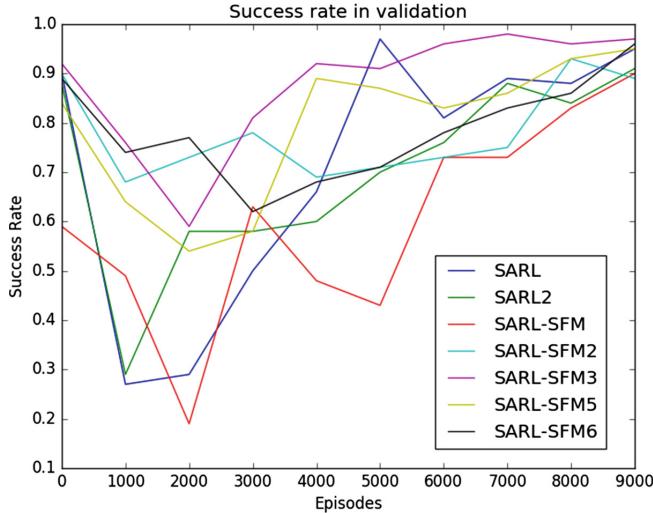
In Table 1 the average results of a test using 500 episodes are exposed for the different versions:

- **SARL:** The CRI approach without local map.
- **SARL-SFM:** With SFM reward and temporal discount in the reward.
- **SARL-SFM2:** With SFM reward.
- **SARL-SFM3:** With SFM reward but without concave barrier in training.
- **SARL-SFM4:** With SFM reward but  $k = 0.003$  in the reward.
- **SARL-SFM5:** With forces included in the states and without SFM reward.
- **SARL-SFM6:** With forces included in the states and with the SFM reward.

In the Fig. 3 is shown the success rate for each type of training during the validation. In Table 2 are shown the results of a second type of training with only the SARL and SARL-SFM6 cases using all the environments.

### 6.4 Qualitative Evaluation

Comparing to the cases analysed in the CRI paper, in which there were only 5 moving agents without obstacles, this paper has proposed a more demanding environment with 10 elements between agents and obstacles. Normally, with the new environments in training, the robot does not get stuck because of the obstacles and can surpass the concave barrier.



**Fig. 3.** Validation graphic. This graphic shows the validation results for the success rate each 1000 episodes of the first type of training.

In spite of the fact that the robot faces better the obstacles, there are some cases where the robot is very slow avoiding the concave barrier. If there are not agents moving in the environment the robot gets stuck with the concave barrier.

In Table 1, results obtained are, in general, worse than in the CRI paper because of the more challenging environment and the best success rate is obtained without the SFM reward. Comparing the SARL-SFM2 to SARL-SFM3, although the success rate is better in SARL-SFM3, in SARL-SFM2 the robot get stuck fewer in the concave barrier episodes. For this reason, the concave barrier can be useful in the training phase.

In the SARL-SFM4, k is 3 times bigger to increment the rewards and the influence of the attractive effect to the goal and there is not temporal discount. This causes a worse performance than the SARL-SFM case.

Although the test results are better in the SARL case, the training progression is not equal for all the cases. This can be observed in Fig. 3. For example, in the SARL-SFM3 the validation applied after 7000 episodes of training gives a success rate of 0.98. In SARL-SFM5, this success rate after 4000 episodes is 0.89, while in the SARL case was only 0.66. In the SARL-SFM case the success rate is always smaller than the SARL success rate.

The 2 last cases, SARL-SFM5 and SARL-SFM6, combine the SFM reward with the new states and offers better results than cases with only SFM reward. The convergence is faster in the first part of the training than the SARL case and the performance in test is more or less the same. These results suggest that the training progression is faster introducing the forces in the states and the oscillations in training are reduced.

In Table 2 all the environments are used in the training and the performance of the robot facing obstacles improves only a little in the SARL and SARL-SFM6 cases. There are more or less a 10% of cases where the robot get stuck and these cases normally occur in environments that only contain obstacles.

## 6.5 Implementation Details

For the implementation, the robot is visible for the agents and the different versions that are compared do not use the local map of the CRI work. The neural network is the same as the network used in the CRI approach. It is only changed the input layer in the first multilayer perceptron (MLP) and the input layer in the last MLP to use the new states proposed in this paper. In the states, the parameters of the repulsive forces described in Sect. 2.1 are  $a_z = 1$ ,  $d_z = 0$  and  $b_z = 1$ . This choice offers inputs neither too big nor too small to the neural network.

In the optimization task it is used the Stochastic Gradient Descent method with momentum 0.9. The batch size is 100.

For imitation learning, 3000 episodes are collected using ORCA for the robot and the policy is trained with 50 epochs and learning rate 0.01. In this phase, the environment is the one without the concave barrier in all cases.

For reinforcement learning, the learning rate is 0.001, the discount factor is 0.9, the number of episodes is 10,000 and the policy used is the  $\epsilon$ -greedy policy of the CRI work. Each 1000 episodes there is a validation phase with 100 episodes. At the end of the training there is a test phase with 500 episodes. In Table 1, the environment used for the reinforcement learning phase of the training, the validation phase and the test phase is randomly chosen each episode between one of the two first environments with probabilities 0.7 for the first environment and 0.3 for the second one. In Table 2, the environments used in reinforcement learning are a total of 4, from the second to the fifth, with probabilities 0.25, the number of training episodes is 15,000 and there are 1000 test episodes.

The work assumes holonomic kinematics and the same action space that is used in CRI work. The implementation has been developed in PyTorch and the simulations have been launched in a Tesla K40c GPU.

## 7 Conclusion

In this work, 3 modifications have been done in order to check the changes in the performance of the SARL policy, designed in the CRI approach.

Firstly, the environment has been expanded through obstacles to improve the robot's behaviour when faces them. This type of modification has slightly improved the robot behaviour but there are still some problems when the robot faces only obstacles and a better description of the environment or a different neural network would be required.

Secondly, the reward signal has been modified taking account the SFM. This type of modification has caused overall worse results in the navigation performance reducing the success rate and increasing the collision rate. In spite of this

problem, the convergence improves in the first 5000 episodes. This better initial convergence could be caused because the algorithm has additional information in the first part of the training.

Finally, the last modification has been the inclusion of the repulsive forces in the agent states and the resultant force in the robot state. This modification has not produced notable changes but it appears that the convergence is faster in the first 5000 episodes. It also makes the test results very much independent of using SFM reward, palliating its negative effect.

To sum up, this approach offers a way to slightly accelerate the convergence in the TD learning for navigation tasks. For better results in the performance of the policy are suggested as future work, forms of reward shaping, other RL algorithms like Policy Gradients, more prediction information of the environment like in Social GAN work [11] and mix the RL algorithms with a global planner [9].

## References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 961–971, June 2016
2. Van den Berg, J., Lin, M., Manocha, D.: Reciprocal velocity obstacles for real-time multi-agent navigation. In: 2008 IEEE International Conference on Robotics and Automation, pp. 1928–1935. IEEE (2008)
3. Chen, C., Liu, Y., Kreiss, S., Alahi, A.: Crowd-robot interaction: crowd-aware robot navigation with attention-based deep reinforcement learning. In: 2019 International Conference on Robotics and Automation, pp. 6015–6022, May 2019
4. Chen, Y.F., Everett, M., Liu, M., How, J.P.: Socially aware motion planning with deep reinforcement learning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1343–1350. IEEE (2017)
5. Chen, Y.F., Liu, M., Everett, M., How, J.P.: Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 285–292 (2017)
6. Everett, M., Chen, Y.F., How, J.P.: Motion planning among dynamic, decision-making agents with deep reinforcement learning. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3052–3059. IEEE (2018)
7. Ferrer, G., Sanfeliu, A.: Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1730–1735. IEEE (2014)
8. Ferrer, G., Zulueta, A.G., Cotarelo, F.H., Sanfeliu, A.: Robot social-aware navigation framework to accompany people walking side-by-side. Auton. Robots **41**(4), 775–793 (2017)
9. Francis, A., Faust, A., Chiang, H.L., Hsu, J., Kew, J.C., Fiser, M., Lee, T.E.: Long-range indoor navigation with PRM-RL. CoRR abs/1902.09458 (2019)
10. Grzes, M., Kudenko, D.: Reward shaping and mixed resolution function approximation. In: Developments in Intelligent Agent Technologies and Multi-Agent Systems: Concepts and Applications, pp. 95–115. IGI Global (2011)
11. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: 2018 IEEE/CVF

- Conference on Computer Vision and Pattern Recognition, pp. 2255–2264, June 2018
12. Haddad, S., Wu, M., Wei, H., Lam, S.K.: Situation-aware pedestrian trajectory prediction with spatio-temporal attention model. CoRR abs/1902.05437 (2019)
  13. Hasan, I., Setti, F., Tsesmelis, T., Del Bue, A., Galasso, F., Cristani, M.: MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6067–6076 (2018)
  14. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. *Phys. Rev. E* **51**(5), 4282 (1995)
  15. Hirose, N., Xia, F., Martin-Martin, R., Sadeghian, A., Savarese, S.: Deep visual MPC-policy learning for navigation. arXiv preprint [arXiv:1903.02749](https://arxiv.org/abs/1903.02749) (2019)
  16. Liu, Y., Xu, A., Chen, Z.: Map-based deep imitation learning for obstacle avoidance. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8644–8649. IEEE (2018)
  17. Long, P., Fanl, T., Liao, X., Liu, W., Zhang, H., Pan, J.: Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In: 2018 IEEE International Conference on Robotics and Automation, pp. 6252–6259 (2018)
  18. Long, P., Liu, W., Pan, J.: Deep-learned collision avoidance policy for distributed multiagent navigation. *IEEE Robot. Autom. Lett.* **2**(2), 656–663 (2017)
  19. Ng, A.Y., Harada, D., Russell, S.J.: Policy invariance under reward transformations: theory and application to reward shaping. In: Proceedings of the Sixteenth International Conference on Machine Learning, ICML 1999, pp. 278–287. Morgan Kaufmann Publishers Inc. (1999)
  20. Repiso, E., Garrell, A., Sanfeliu, A.: Adaptive side-by-side social robot navigation to approach and interact with people. *Int. J. Soc. Rob.* **11**, 1–22 (2019)
  21. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: SoPhie: an attentive GAN for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1349–1358 (2019)
  22. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press (2018)
  23. Tail, L., Zhang, J., Liu, M., Burgard, W.: Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In: 2018 IEEE International Conference on Robotics and Automation, pp. 1111–1117 (2018)
  24. Tesauro, G.: Temporal difference learning and TD-Gammon. *Commun. ACM* **38**(3), 58–68 (1995)
  25. Trautman, P., Ma, J., Murray, R.M., Krause, A.: Robot navigation in dense human crowds: statistical models and experimental studies of human-robot cooperation. *Int. J. Robot. Res.* **34**(3), 335–356 (2015)
  26. Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Robotics Research, pp. 3–19. Springer (2011)
  27. Van Roy, B.: Temporal-Difference Learning and Applications in Finance. MIT Press, Cambridge (2001)
  28. Zou, H., Ren, T., Yan, D., Su, H., Zhu, J.: Reward shaping via meta-learning. CoRR abs/1901.09330 (2019)



# A Cross-Situational Learning Based Framework for Grounding of Synonyms in Human-Robot Interactions

Oliver Roesler<sup>(✉)</sup>

Artificial Intelligence Lab, Vrije Universiteit Brussel, 1050 Brussels, Belgium  
[oliver@roesler.co.uk](mailto:oliver@roesler.co.uk)

**Abstract.** Natural human-robot interaction requires robots to link words to objects and actions through grounding. Although grounding has been investigated in previous studies, not many considered grounding of synonyms and the majority of employed models only worked offline. In this paper, we try to fill this gap by introducing an online learning framework for grounding synonymous object and action names using cross-situational learning. Words are grounded through geometric characteristics of objects and kinematic features of the robot joints during action execution. An interaction experiment between a human tutor and HSR robot is used to evaluate the proposed framework. The results show that the employed framework is able to successfully ground all used words.

**Keywords:** Language grounding · Cross-situational learning · Human-robot interaction

## 1 Introduction

The number of service robots that are employed in complex and human-centered environments instead of factories is constantly growing [14], thereby bringing us closer to a future in which robots are an essential part of everyday life. To enable robots to efficiently collaborate with human users in their daily life, they must be able to converse in natural language and understand the instructions of a user so that they execute the desired actions appropriately, such as *bring a drink* or *pick up a box* [15]. Understanding natural language instructions is non-trivial and requires connections between symbols, i.e. words, and their meanings. In theory, the latter can be provided by relating unknown symbols to other symbols. However, if the meaning of the other symbols is not known, they do not provide meaning to the unknown symbol. This problem is called the “Symbol Grounding Problem” and was first described by Harnad [12]. The main idea is that abstract knowledge and language only becomes meaningful, when it is linked to the physical world, which can be done by creating mappings from words to corresponding percepts. However, not all words need to be directly grounded

through percepts because they can be indirectly grounded by being linked to directly grounded words.

There exist many different grounding approaches. She et al. [26] investigated the use of a dialog system for grounding of higher level symbols through already grounded lower level symbols. While it can be used as an additional grounding mechanism, its usefulness is limited due to the need for a sufficiently large set of grounded lower level symbols. Additionally, the system requires a professional tutor to answer its questions, who might not always be available and increases the cost to obtain new groundings. The latter problem also constraints the applicability of the *Naming Game*, which allows an agent to quickly learn word-percept mappings, if another agent is present and knows the correct mappings [30]. To ground manipulation actions in an unsupervised manner, i.e. without the need for a tutor, cross-situational learning (CSL) can be used, which assumes that one word appears several times together with the same perceptual feature vector so that a corresponding mapping can be created [11, 27, 29].

The basic idea of CSL is that the context a word is used in leads to a number of candidate meanings, i.e. mappings from words to percepts, and that the correct meaning lies at the intersection of the sets of candidate meanings [9, 19]. Thus, the correct mapping between a word and its corresponding percepts will reliably reoccurs across situations [2, 28]. Previous studies investigated the use of cross-situational learning for grounding of objects and actions [10, 32] as well as spatial concepts [1, 5, 33]. In all studies, grounding was conducted offline, i.e. perceptual data and words were collected in advance, which prevents their models from being used in real-time human-robot interactions. Additionally, the employed models were not able to handle ambiguous words, although, the sentences humans produce are often ambiguous due to homonymy, i.e. one word refers to several objects or actions, and synonymy, i.e. one object or action can be referred to by several different words. The latter do not need to be true synonyms, i.e. words that refer to the exact same meaning, especially, since there are no true synonyms according to the “Principle of Contrast” [3]. Instead, words are only synonyms as references to an object or action in a particular set of situations. Examples are words that refer to the purpose or content of an object, instead of the object itself, such as: *coca cola* or *lemonade* instead of *bottle*. One recent study showed that grounding of unknown synonyms, i.e. synonymous words of previously encountered words that have not been encountered before, requires the use of semantic and syntactic information [21]. However, in a different study Roesler et al. [20] showed that known synonyms, i.e. synonyms that have been encountered in previous situations, do not require semantic or syntactic information. Since all words used in this study appear in several situations, the employed online grounding mechanism does not use any semantic or syntactic information to ground synonyms.

In this paper, we propose a novel grounding framework for grounding of synonymous object and action words. More specifically, we present an unsupervised learning model that uses cross-situational learning to map words to corresponding percepts so as to infer the meaning objects and actions. In the conducted

studies objects and actions are described by geometric characteristics and kinematic features of the robot joints during action execution, respectively. In the framework obtained percepts are first converted to an abstract representation through clustering and then provided to the cross-situational learning algorithm to ground the encountered natural language instructions.

The rest of this paper is structured as follows: Sect. 2 provides an overview of the employed grounding framework. The experimental design and obtained results are described in Sects. 3 and 4. Finally, Sect. 5 concludes the paper.

## 2 System Overview

The used grounding system consists of four parts: (1) 3D object segmentation system, which segments objects into point clouds to determine their geometric characteristics, (2) Action recording system, which records the state of several joints while the robot is executing actions to create action feature vectors, (3) Percept clustering component, which clusters the different percepts to obtain an abstract representation of percepts, and (4) Cross-situational learning component, which maps percepts to words. The inputs and outputs of the individual parts are highlighted below, and described in detail in the following subsections.

### 1. 3D object segmentation:

- **Input:** Point cloud data.
- **Output:** Geometric characteristics of objects.

### 2. Action recording:

- **Input:** Changes of joint states during action execution by the robot.
- **Output:** Action feature vectors representing the executed actions.

### 3. Clustering of percepts:

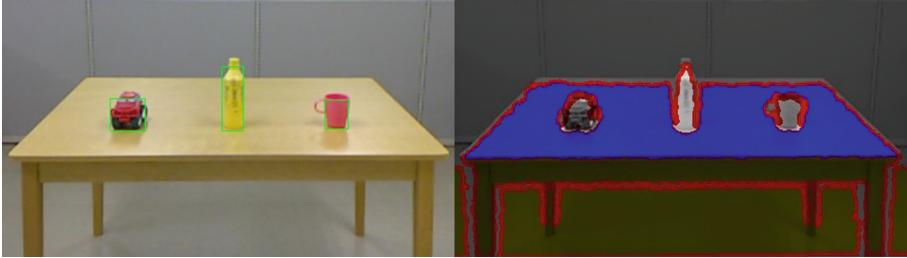
- **Input:** Geometric object characteristics and action feature vectors.
- **Output:** Cluster numbers of percepts.

### 4. Cross-situational learning:

- **Input:** Natural language instructions and cluster numbers of percepts.
- **Output:** Word to percept mappings.

### 2.1 3D Object Features

Many different approaches for 3D point cloud segmentation have been investigated in the literature [17]. Edge based methods detect points with fast intensity change to segment point clouds into regions using the detected points as boundaries [23]. Although these methods are fast, they are also highly sensitive to noise. Region based methods combine neighbouring points that have similar properties into regions [16]. These methods are less sensitive to noise, but also not good at determining exact region borders. Attributes based methods cluster point clouds through predefined attributes, such as point density and vertical distribution [7]. They can achieve a high accuracy and flexibility, however, they are often also slow and the overall performance depends heavily on the quality of the selected



**Fig. 1.** Examples of the used objects and the corresponding 3D point cloud information: (A) car, (B) bottle, and (C) cup.

attributes. Graph based methods represent point clouds through a graph, where each point represents a vertex connected via edges to neighbouring points [31]. While these methods can handle noise and uneven density, they can often not be run in real time. Model based approaches create clusters of points with similar mathematical representations based on geometric criteria [24]. They can handle outliers and are fast, but cannot handle point clouds from different sources.

In this study, a fast and reliable unsupervised model based segmentation approach is used to segment objects lying in a plane into separate point clouds without the need for much prior knowledge, such as object models or the number of regions to process [4]. The applied model detects the major plane in the environment, which is a tabletop in the conducted experiment, via the RANSAC algorithm [8], and keeps track of it in consecutive frames, while it defines planes that are orthogonal to the major plane and touch at least one border of the image as wall planes. After filtering out points that belong to the main or wall planes, it voxelizes the remaining points and clusters them into blobs representing object candidates. Blobs that are neither extremely small nor large are treated as objects<sup>1</sup>. Each point cloud of a segmented object is characterized through a Viewpoint Feature Histogram [22] descriptor, which represents the object geometry taking into consideration the viewpoint, while ignoring scale variance. Figure 1 provides an illustrative example of the obtained 3D point cloud information.

## 2.2 Action Features

The dynamic characteristics of actions during execution through teleoperation are represented through action feature vectors. Overall, five different characteristics, which represent possible subactions, are recorded through the sensors of the robot [34]. The used characteristics are:

---

<sup>1</sup> The threshold for the blob size was manually set after selecting the objects for the experiment and should be suitable for all objects of similar size.

1. The distance from the actual to the lowest torso position in meters.
2. The angle of the arm flex joint in radians.
3. The angle of the wrist roll joint in radians.
4. Velocity of the base.
5. Binary state of the gripper. (1: closing, 0: opening or no change)

They are then combined into the following vector

$$\begin{pmatrix} a_1^1 & a_1^2 & a_1^3 & a_1^4 & a_1^5 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_6^1 & a_6^2 & a_6^3 & a_6^4 & a_6^5 \end{pmatrix}$$

where  $a^1$  represents the difference of the distances from the lowest torso position in meters, while  $a^2$  and  $a^3$  represent the differences in the angles of the arm and wrist in radians, respectively. The differences are calculated by subtracting the value at the beginning of the subaction from the value at the end of the subaction.  $a^4$  represents the mean velocity of the base (forward/backward), and  $a^5$  represents the binary gripper state. Each action is characterized through six manually defined subactions. Therefore, if an action consists of less than six subactions, rows with zeros are added at the end, while the duration of a subaction depends on the teleoperator and is thus not fixed.

### 2.3 Clustering of Percepts

The CSL algorithm (Sect. 2.4) requires percepts to be converted to an abstract representation that can then be used to ground natural language. The abstract representation is obtained through clustering. In this study, DBSCAN is used, which is a density-based clustering algorithm proposed by Ester et al. [6], because it does not require the number of cluster specified in advance<sup>2</sup>. Instead, it determines the number of clusters automatically, while only requiring two parameters, i.e. the radius  $\epsilon$  and threshold  $minSamples$ . All points within the radius  $\epsilon$  of a core point are assigned to the same cluster as the core point. Core points are points that have more than  $minSamples$  points within radius  $\epsilon$  around them [25]. Cluster numbers were calculated prior to grounding so that they could be provided to the CSL algorithm. The obtained object and action feature vector clusters achieved adjusted rand scores [13] of 0.89 and 0.97, respectively.

### 2.4 Cross-Situational Learning

The idea of CSL has lead to the development of a variety of algorithms that realize CSL in different ways, e.g. through the use of probabilistic models [1, 21], for grounding of words through percepts in artificial agents. This section proposes a novel online CSL algorithm for grounding of words, which employs CSL in a way that, to the best of our knowledge, has not been proposed or used before.

---

<sup>2</sup> The used DBSCAN implementation is available in *scikit-learn* [18].

---

**Algorithm 1.** The grounding procedure takes as input all words ( $W$ ) and percepts ( $P$ ) of the current situation as well as the sets of previously obtained word-percept ( $WP$ ) and percept-word ( $PW$ ) pairs and returns sets of grounded words ( $GW$ ) and percepts ( $GP$  ).

---

```

1: procedure GROUNDING( $W, P, WP, PW$ )
2:   Update  $WP$  and  $PW$  using  $W$  and  $P$ 
3:   for  $j = 1$  to word_number do
4:     Save highest  $WP$  to  $GW$ 
5:   end for
6:   for  $j = 1$  to percept_number do
7:     Save highest  $PW$  to  $GP$ 
8:   end for
9:   return  $GW \cup GP$ 
10: end procedure
```

---

Initially, the set of grounded words ( $GW$ ) and percepts ( $GP$ ) is empty. For each situation the algorithm takes the obtained perceptual information and uses it together with the perceptual information of all previous situations to ground the words of all encountered instructions. Before the actual grounding procedure, phrases are detected and auxiliary words are discarded by checking a predefined dictionary<sup>3</sup>. Afterwords, a set of percepts  $WP$  is created for each word, in which each percept is saved with a number that indicates how often it occurred together with that word. The same is also done for percepts, i.e. for each percept a set of words  $PW$  is created. Then, the highest word-percept pair is determined and saved to the set of grounded words  $GW$ . All other word-percept pairs the word is part of will not be considered for the selection of the next highest word-percept pair during the next iterations because it is already grounded. Additionally, the percept that was used to ground the word will not be available to ground any other words. These restrictions are applied until all percepts have been used for grounding once. If there are still ungrounded words left, all percepts will become again available for grounding, until all words have been grounded. This last step is necessary to ground synonyms. After all words have been grounded the same process is repeated for percept-word pairs to assign synonymous percepts to the same word. Finally, the sets of grounded words and percepts are merged. Algorithm 1 summarizes the grounding procedure.

---

<sup>3</sup> The used instructions contain only the article *the* as an auxiliary word, i.e. a word that has no corresponding percept, and eight phrases, e.g. *lord of the ring* or *lift up*. In this study, two manually predefined dictionaries were used to identify them, while we will investigate to create the dictionaries automatically and in an unsupervised manner during grounding, in future work.

### 3 Experimental Setup

A human tutor and HSR robot<sup>4</sup> are interacting in front of a tabletop. One of the five different objects {BOTTLE, CUP, BOX, CAR, and BOOK} is placed on the table (Figs. 1 and 2). Each of the objects can be referred to by five different names as shown in Table 1. During the experiment the robot performs five different actions on each object (Fig. 2), where each action can be described by two different names as illustrated in Table 2.



**Fig. 2.** Illustration of action *lift up* executed by the robot in a tabletop scene.

**Table 1.** Overview of the objects with their corresponding synonyms.

Object	Synonyms				
Bottle	coca cola	soda	pepsi	coke	lemonade
Cup	latte	milk	milk tea	coffee	espresso
Box	candy	chocolate	confection	sweets	dark chocolate
Car	audi	toyota	mercedes	bmw	honda
Book	harry potter	the godfather	narnia	lord of the rings	the hobbit

A total of 125 different sentences are given to the robot by the human tutor in order to allow it to ground object and action names using the recorded perceptual data. Each sentence consists of either two or three words and has one of the following two structures: “*action the object*” or “*action object*”, respectively<sup>5</sup>, where *action* and *object* are substituted by the corresponding names (Tables 1 and 2).

<sup>4</sup> The Human Support Robot from Toyota, which is used for the experiment, can move omnidirectional and has a cylindrical shaped body with one arm and gripper. It has 11 degrees of freedom and is equipped with a variety of different sensors, such as stereo and wide-angle cameras. [[Official Toyota HSR Website](#)].

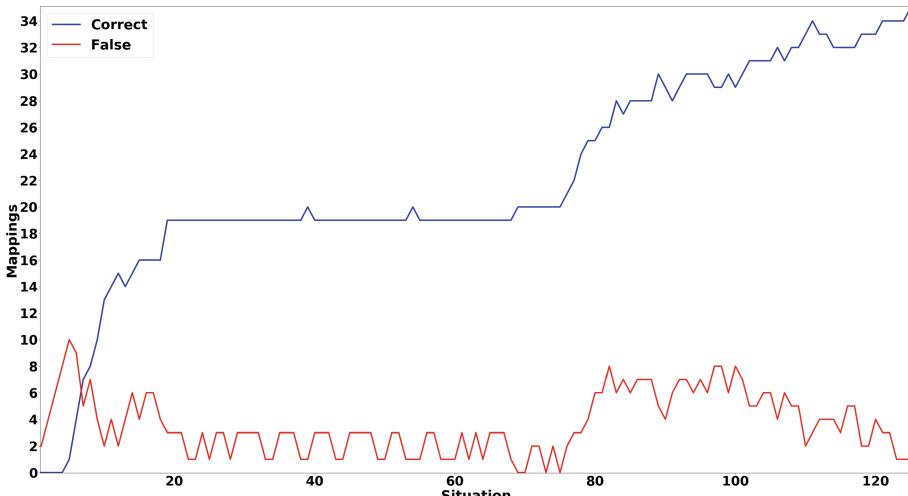
<sup>5</sup> The latter is only used for sentences with the BOOK object. For example: “LIFT UP HARRY POTTER” represents the structure “*action object*”, while “LIFT UP the LEMONADE” represents the structure “*action the object*”.

The experimental procedure consists of three phases as described below:

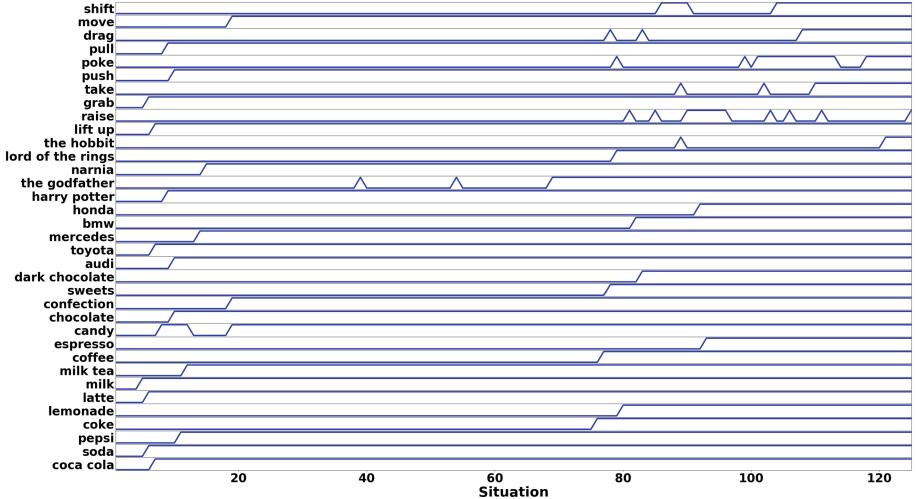
1. Collection of semantic and perceptual information for the different situations.
  - (a) An object is placed on the table and the robot determines its geometric characteristics so as to calculate its feature vector (Sect. 2.1).
  - (b) A sentence is given by the human tutor to the robot and the robot executes the given action through teleoperation while several kinematic characteristics are recorded and converted into an action feature vector (Sect. 2.2).
2. Clustering of perceptual information (Sect. 2.3).
3. The CSL algorithm is used to ground words using the geometric characteristics of objects and the action feature vectors (Sect. 2.4).

**Table 2.** Overview of the used actions.

Synonym 1	Synonym 2	Description
lift up	raise	The object will be lifted up
grab	take	The object will be grabbed, but not displaced
push	poke	The object will be pushed with the closed gripper i.e. it will not be grabbed
pull	drag	The object will be grabbed and moved towards the robot
move	shift	The object will be grabbed and moved



**Fig. 3.** Cross-situational learning results showing the number of correct and false mappings for all 125 situations encountered by the robot.



**Fig. 4.** Illustration of correct word mappings for all words used in this study. The figure shows all 125 situations encountered by the robot.

## 4 Results and Discussion

The employed cross-situational learning algorithm is able to successfully ground all 35 words, used in this study, through their corresponding percepts. Figure 3 shows that during the first situations most created mappings are false because the algorithm has not much data available. After 7 situations the number of correct mappings is for the first time higher than the number of false mappings, which does not change for the rest of the experiment. However, after 19 situations the number of correct mappings plateaus for nearly 50 situations before it finally increases until all words are correctly grounded. This behavior is caused by the fact that no new words are encountered so that the number of correctly grounded words can not increase over 20. Additionally, the number of false mappings varies between one and three. This constant variation in the number of false mappings is due to information from new situations. Figure 4 shows all words and whether there are grounded for each encountered situation. For most of the words, once they are correctly grounded their mapping does not change anymore, while only in a few cases, e.g. for *raise* or *poke*, a previously correctly grounded word is at a later time incorrectly mapped. Overall it takes 125 situations until all words are grounded successfully, while there is still one false mapping for the word *raise* to the percept of BOX.

## 5 Conclusions and Future Work

We investigated a multimodal framework for grounding synonymous object and action names through the robot visual perception and proprioception during its

interaction with a human tutor. Our cross-situational learning model was set up to learn the meaning of object and action names using geometric characteristics of objects obtained from point cloud information and kinematic features of the robot joints recorded during action execution.

The proposed model allowed the grounding of synonyms through real percepts without the use of any syntactic or semantic information. However, it relies on a manually defined dictionary to identify auxiliary words and phrases. Additionally, only a small number of words and very simple sentences have been used.

In future work, we will consider automatic identification of phrases and auxiliary words. Furthermore, we will obtain grounding results for longer and more complex sentences that contain a larger number of words. Finally, we will extend the model so as to include other modalities such as color.

## References

1. Aly, A., Taniguchi, A., Taniguchi, T.: A generative framework for multimodal learning of spatial concepts and object categories: an unsupervised part-of-speech tagging and 3D visual perception based approach. In: IEEE International Conference on Development and Learning and the International Conference on Epigenetic Robotics (ICDL-EpiRob), Lisbon, Portugal, September 2017
2. Blythe, R.A., Smith, K., Smith, A.D.M.: Learning times for large lexicons through cross-situational learning. *Cogn. Sci.* **34**, 620–642 (2010)
3. Clark, E.V.: The principle of contrast: a constraint on language acquisition. In: *Mechanisms of Language Acquisition*, pp. 1–33. Lawrence Erlbaum Associates (1987)
4. Craye, C., Filliat, D., Goudou, J.F.: Environment exploration for object-based visual saliency learning. In: IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, May 2016
5. Dawson, C.R., Wright, J., Rebguns, A., Escárcega, M.V., Fried, D., Cohen, P.R.: A generative probabilistic framework for learning spatial language. In: IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL), Osaka, Japan, August 2013
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD), Portland, Oregon, USA, pp. 226–231, August 1996
7. Filin, S., Pfeifer, N.: Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS J. Photogram. Remote Sens. (P&RS)* **60**, 71–80 (2006)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM (CACM)* **24**(6), 381–395 (1981)
9. Fisher, C., Hall, D.G., Rakowitz, S., Gleitman, L.: When it is better to receive than to give: syntactic and conceptual constraints on vocabulary growth. *Lingua* **92**, 333–375 (1994)
10. Fontanari, J.F., Tikhanoff, V., Cangelosi, A., Ilin, R., Perlovsky, L.I.: Cross-situational learning of object-word mapping using neural modeling fields. *Neural Netw.* **22**(5–6), 579–585 (2009)

11. Fontanari, J.F., Tikhanoff, V., Cangelosi, A., Perlovsky, L.I.: A cross-situational algorithm for learning a lexicon using neural modeling fields. In: International Joint Conference on Neural Networks (IJCNN), Atlanta, GA, USA, June 2009
12. Harnad, S.: The symbol grounding problem. *Physica D* **42**, 335–346 (1990)
13. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
14. International Federation of Robotics: World robotics 2017 - service robots (2017)
15. Kemp, C.C., Edsinger, A., Torres-Jara, E.: Challenges for robot manipulation in human environments. *IEEE Robot. Autom. Mag.* **14**(1), 20–29 (2007)
16. Koster, K., Spann, M.: MIR: an approach to robust clustering-application to range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **22**(5), 430–444 (2000)
17. Nguyen, A., Le, B.: 3D point cloud segmentation: a survey. In: 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM). IEEE, Manila, November 2013
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
19. Pinker, S.: *Learnability and Cognition*. MIT Press, Cambridge (1989)
20. Roesler, O., Aly, A., Taniguchi, T., Hayashi, Y.: A probabilistic framework for comparing syntactic and semantic grounding of synonyms through cross-situational learning. In: ICRA-18 Workshop on Representing a Complex World: Perception, Inference, and Learning for Joint Semantic, Geometric, and Physical Understanding, Brisbane, Australia, May 2018
21. Roesler, O., Aly, A., Taniguchi, T., Hayashi, Y.: Evaluation of word representations in grounding natural language instructions through computational human-robot interaction. In: Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Daegu, South Korea, March 2019
22. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the viewpoint feature histogram. In: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, pp. 2155–2162, October 2010
23. Sappa, A.D., Devy, M.: Fast range image segmentation by an edge detection strategy. In: Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling (3DIM), Quebec City, Quebec, Canada, August 2002
24. Schnabel, R., Wahl, R., Klein, R.: Efficient ransac for point-cloud shape detection. *Comput. Graphics Forum* **26**(2), 214–226 (2007)
25. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Trans. Database Syst. (TODS)* **42**(3), 19 (2017)
26. She, L., Yang, S., Cheng, Y., Jia, Y., Chai, J.Y., Xi, N.: Back to the blocks world: learning new actions through situated human-robot dialogue. In: Proceedings of the SIGDIAL 2014 Conference, Philadelphia, U.S.A., pp. 89–97, June 2014
27. Siskind, J.M.: A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition* **61**, 39–91 (1996)
28. Smith, A.D.M., Smith, K.: Cross-Situational Learning, pp. 864–866. Springer, Boston (2012). [https://doi.org/10.1007/978-1-4419-1428-6\\_1712](https://doi.org/10.1007/978-1-4419-1428-6_1712)
29. Smith, K., Smith, A.D.M., Blythe, R.A.: Cross-situational learning: an experimental study of word-learning mechanisms. *Cogn. Sci.* **35**(3), 480–498 (2011)
30. Steels, L., Loetzsche, M.: The grounded naming game. In: Steels, L. (ed.) Experiments in Cultural Language Evolution, pp. 41–59. John Benjamins, Amsterdam (2012)

31. Strom, J., Richardson, A., Olson, E.: Graph-based segmentation for colored 3D laser point clouds. In: International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan (2010)
32. Taniguchi, A., Taniguchi, T., Cangelosi, A.: Cross-situational learning with Bayesian generative models for multimodal category and word learning in robots. *Front. Neurorobot.* **11**, 66 (2017)
33. Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S., Roy, N.: Approaching the symbol grounding problem with probabilistic graphical models. *AI Mag.* **32**(4), 64–76 (2011)
34. Toyota Motor Corporation: HSR Manual, 2017.4.17 edn., April 2017



# A Novel Model for Emotion Detection from Facial Muscles Activity

Elahe Bagheri<sup>1</sup>(✉), Azam Bagheri<sup>2</sup>, Pablo G. Esteban<sup>1</sup>,  
and Bram Vanderborght<sup>1</sup>

<sup>1</sup> Robotics and Multibody Mechanics Research Group,

Vrije Universiteit Brussel and Flanders Make, Brussels, Belgium

{elahe.bagheri,pablo.gomez.esteban,bram.vanderborght}@vub.be

<sup>2</sup> Electrical Engineering Group, Chalmers University of Technology,

Gothenburg, Sweden

bazam@chalmers.se

**Abstract.** Considering human's emotion in different applications and systems has received substantial attention over the last three decades. The traditional approach for emotion detection is to first extract different features and then apply a classifier, like SVM, to find the true class. However, recently proposed Deep Learning based models outperform traditional machine learning approaches without requirement of a separate feature extraction phase.

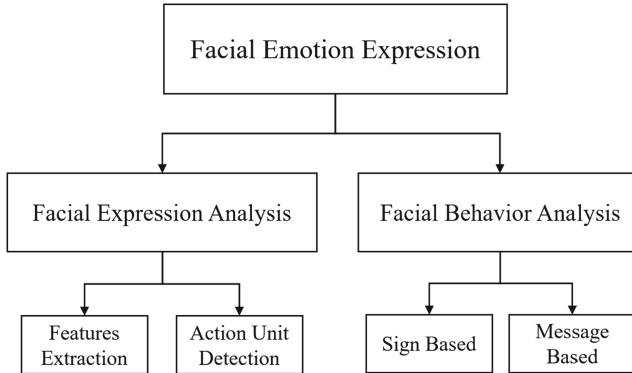
This paper proposes a novel deep learning based facial emotion detection model, which uses facial muscles activities as raw input to recognize the type of the expressed emotion in the real time. To this end, we first use OpenFace to extract the activation values of the facial muscles, which are then presented to a Stacked Auto Encoder (SAE) as feature set. Afterward, the SAE returns the best combination of muscles in describing a particular emotion, these extracted features at the end are applied to a Softmax layer in order to fulfill multi classification task.

The proposed model has been applied to the CK+, MMI and RAD-VESS datasets and achieved respectively average accuracies of 95.63%, 95.58%, and 84.91% for emotion type detection in six classes, which outperforms state-of-the-art algorithms.

**Keywords:** Facial Emotion Recognition · Facial Muscles Activity · Stacked Auto Encoder · Facial Action Units

## 1 Introduction

For the last two decades we have been started to use different smart devices and applications in our daily life, robots are going to be used in our shops [5], schools and hospitals [22]. However, lots of them lose their favor by losing novelty effect. Haag et al. [10] argued the communication between humans and systems can improve by considering emotions as an additional interaction modality. Meanwhile researchers showed the systems which recognize and respond to human's



**Fig. 1.** General Facial Emotion Recognition approaches.

emotions are more caring, likable, supportive and trustworthy [2]. Hence, recognizing human's emotion became an important topic to study.

Emotion detection is the ability to recognize another's affective state, which typically involves the integration and analysis of expressions through different modalities, like facial expression, speech, body movements and gestures [3]. Since 55% of human emotions are conveyed by facial expression [17], Facial Emotion Recognition (FER) is the most investigated method for human emotion recognition task.

FER contains two main parts, facial expression analysis and facial behavior analysis, as shown in Fig. 1. Facial expression analysis carried out via two main approaches, feature extraction and Action Unit (AU) detection. The feature extraction approaches proceed on by detecting face region and facial components, e.g., eyebrows, eyes, nose and mouth from an input image. Then two different types of features are extracted: *geometric* and *appearance* features. Geometric features represent the positions of salient points of the face, e.g., ends of the eyes, end of the nose, mouth and the shape of the facial components, while appearance features represent the text variations of the face, e.g., color, edge density, crinkles, and wrinkles [28]. Finally, the pre-trained machine learning classifier attempts to classify the given face as portraying one emotion [12].

The AU detection methods, however, are independent of facial appearance and analyse facial muscles movements by tracking AUs. Each AU indicates fundamental movements of a single or a group of muscles<sup>1</sup>. Through facial expression of different emotions, different combinations of AUs are activated. Ekman [9] defined the Facial Action Coding System (FACS), which encodes the movements of AUs to describe human facial movements and converts the detected AUs to the corresponding emotion. An important advantage of the AU detection methods is that they remove the need of analysing complex high-dimensional features [24].

<sup>1</sup> <https://imotions.com/blog/facial-action-coding-system/>.

Facial behavior analysis is the other way to perform FER. Cohn et al. [4] proposed two conceptual approaches for studying the facial behavior: “message-based” approach and “sign-based” approach. Message-based approaches categorize facial behaviors as the meaning of expressions and are widely used by psychologists. Message-based methods can be divided into discrete categorical and continuous dimensional methods. Discrete categorical methods assign an expression to one of pre-defined prototypical categories, including six basic emotions proposed by Ekman [8] like anger, disgust, fear, happiness, sadness, and surprise, while continuous emotional methods describe each facial expression by continuous axes, such as arousal and valence [30].

Sign-based approaches, however, describe facial actions regardless of their meaning, and different expressions are classified based on the activated AUs [19]. Indeed, sign-based approaches are similar to AU detection approaches.

Since sign-based algorithms are trained to detect activated AUs in a given image or video to recognize the emotion, the sign-based FER problems can be transformed into the problem of activated AU detection [25]. Hence, applying a proper toolkit, like OpenFace [1] the activation values of facial AUs can be obtained and used for model training for emotion detection. However, as Du et al. [6] showed, determining the exact combination of activated AUs in each emotion is difficult. Thereby, the main contribution of this study is finding the most pivotal activated AUs in each emotion. To this end, we developed a Stacked Auto Encoder (SAE) deep network on the statues of 15 facial AUs to extract the high-order features of the input data that is not possible to obtain by humans. Given automatic extracted features, we added a Softmax layer to full-fill the classification task.

The remain of this paper is structured as follows: Sect. 2 presents a review on previous work. The proposed model is illustrated in Sect. 3. Section 4 demonstrates the experimental results. Finally, Sect. 5 concludes this paper.

## 2 Related Work

Originally classical machine learning algorithms such as Bayesian Networks [18], Gaussian Mixture Models [26], Hidden Markov Models [23], and Neural Networks [29] have been applied to detect expressed facial emotions. The quality of the training data, e.g., image resolution, face view angle and also the way emotions are labeled, strongly influences the results of the training algorithm and is the main obstacle for classical FER algorithms.

In contrast, promising results of neural network methods and deep learning (DL) based approaches in comparison with classical machine learning algorithms, caused to propose numerous DL based FER methods in the research community. Emergence of deep learning as a general end to end learning approach dispels handcraft feature detection problem too [7].

There are two approaches in FER, one which does not use the input’s temporal information so called frame-based, and the other, which uses the temporal information of images and is known as sequence-based. The input in frame-based

approaches is an image without a reference frame, while the input in sequence-based approach is a sequence of one or more frames [13]. Since our proposed model categorizes as frame based, in this section we focus on the state-of-the-art algorithms of the frame-based methods.

Pitaloka et al. [21] used a Convolutional Neural Network (CNN) based method to recognize 6 basic emotions. The proposed method comprises of 5 layers including two sets of convolution layer, two max-pooling layers and a fully connected layer for classification. After pre-processing, the input image is fed to the first convolution layer to extract features like edges, corners and shapes. The output image then is passed to the first max-pooling layer to reduce the image size. The compact image then is sent to the second convolution layer to obtain higher order features and afterward is passed to the second max-pooling layer to reduce the final output size. The fully connected layer at the end, classifies the output image into one of the six basic emotions. However, the performance of the proposed algorithm decreases when the dimension of images is increased regarding to the complexity of the high dimensional images.

Liu et al. [14] proposed a sign-based deep neural network architecture called AU-aware Deep Networks (AUDN) in order to investigate the effect of AUs in emotion recognition. The proposed AUDN includes three sequential modules. In first module a convolution layer stacked by a max-pooling layer generates a complete representation of all expression-specific appearance variations. Then in the second module, an AU-aware receptive field layer searches the subsets of the over-complete representation to find the best simulating of the combination of the AUs. The third module consists of multilayer Restricted Boltzmann Machines (RBM) to learn hierarchical features. Once the features obtain, a linear SVM classifier is applied to recognize the six basic emotions. However, AU-aware layers, in second module, are not able to detect all FACS in images.

Although different state-of-the-art algorithms are proposed in the field of FER, emotion detection has remained a challenging problem in computer vision. In this study, we propose a new SAE-based model to cope with the challenge of the FER in two steps. In the first step, the proposed SAE aims at extracting the most pivotal AUs and in the next step these extracted AUs are applied to the categorical Softmax classifier to detect six basic emotions. Next section details the proposed model.

### 3 Proposed Model

According to the sign-based FER approaches, one way to recognize facial emotion expression is detecting the status of all individual AUs and then analyzing combinations of activated AUs. For example, if a face has been analyzed as having activated AU5, and AU26, a properly trained algorithm should classify it as expressing “surprise”. However, Du et al. [6] showed that encoding the activated AUs into a specific emotion is difficult, if the expressed emotion is a mixture of several emotions. For instance, when some one is surprised by a good news all AUs related to both happiness and surprise can be activated, however, if be

**Table 1.** The list of applied Action Units and related emotions. The pivotal AUs of each emotional state obtained by proposed model are indicated by sign \*, and the pivotal AUs obtained by [6] are indicated by +.

No	AU Description	Happiness	Sadness	Fear	Anger	Surprise	Disgust
1	AU01_Inner Brow Raiser		*	*+		+	
2	AU02_Outer Brow Raiser			*		*+	
3	AU04_Brow Lowered		+	+	*+		
4	AU05_Upper Lid Raiser					*	
5	AU06_Cheek Raiser	*		*			
6	AU07_Lid Tightener				+		
7	AU09_Nose Wrinkle						*+
8	AU10_Upper Lip Raiser						*+
9	AU12_Lip Corner Puller	*+					
10	AU15_Lip Corner Depressor			*+			
11	AU17_Chin Raiser		*				+
12	AU20_Lip Stretcher			*+			
13	AU23_Lip Tightener				*		
14	AU25_Lip Part	+		*+		+	
15	AU26_Jaw Drop			*		*+	

shocked of an online scam, the AUs related to sadness, anger and surprise can be activated at the same time. This ambiguity of emotion expression makes the FER a challenging task.

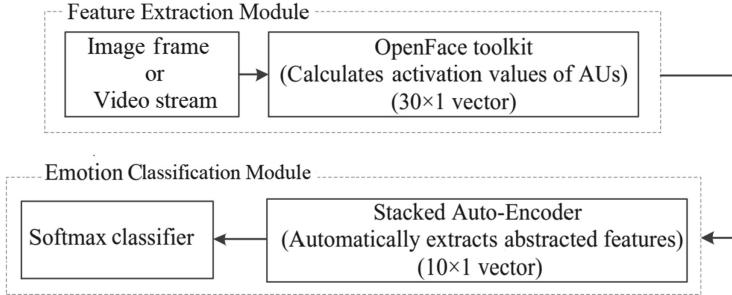
The SAE is able to extract higher order features and detect relations between AUs, which is not possible by human experts or conventional machine learning techniques, therefore, we used a SAE deep network to extract the most effective combinations of AUs in each emotion and used them as the feature set to train our classifier. Figure 2 shows the overall scheme of our proposed model for emotion type detection task. Also the list of the applied AUs is shown in Table 1. The next subsections explain principles of the SAE and the architecture and methodology of the developed deep SAE for emotion type detection.

### 3.1 Principles of the Stacked Auto Encoder

A SAE is a deep neural network consisting of several hidden layers in which the output of each layer is imposed as input to the next layer. By inner layers higher order features, i.e., those are not easily possible for humans to craft, are obtained. Equation 1 gives the encoding step for  $k^{th}$  layer.

$$a^{k+1} = F(\omega^k a^k + b^k), \quad (1)$$

where  $F$  is the activation function, e.g., sigmoid or Rectified Linear Units (ReLU),  $\omega$  and  $b$  are corresponding weight vector and bias value to the units



**Fig. 2.** OpenFace is able to read both images and videos and returns the activation value of different AUs. Passing AU values to SAE, abstracted features are obtained, which by feeding to a Softmax classifier the type of emotion, which AUs are showing is obtained.

**Table 2.** The architecture of the applied SAE neural network for six class classification task.

	Layer	Input size	Output size
Encoder	Dens layer (ReLU)	$30 \times 1$	$70 \times 1$
	Dens layer (ReLU)	$70 \times 1$	$70 \times 1$
	Dens layer (ReLU)	$70 \times 1$	$70 \times 1$
Decoder	Dens layer (ReLU)	$70 \times 1$	$70 \times 1$
	Dens layer (ReLU)	$70 \times 1$	$70 \times 1$
	Dens layer (sigmoid)	$70 \times 1$	$10 \times 1$
Fully connected	Classifier (Softmax)	$10 \times 1$	$6 \times 1$

of  $k^{th}$  layer. The decoding step is given by running the decoding stack of each AE in reverse order as shown in Eq. 2.

$$a^{n+k+1} = F(\omega^{n-k} a^{n+k} + b^{n-k}), \quad (2)$$

where  $a^n$  contains the information of interest and is the activation of the deepest layer of hidden units. Applying the input values as output values, the SAE will learn the high-order, i.e., low dimension features of input values at the layer  $n$ . This vector gives a representation of the input in term of higher order features, which can be used for classification problems by feeding  $a^n$  to a Softmax classifier. After training the SAE, the encoder part of the network is saved and the activation values of the last layer are imposed to the classification layer, which uses a Softmax activation function to include more than two classes.

### 3.2 SAE Architecture for Emotion Type Detection

Table 2 shows the architecture of the proposed SAE for emotion detection. We used OpenFace to extract the AU values of training data. The activation values of

**Table 3.** State-of-the-art algorithms in FER over CK+ and MMI datasets.

Authors	Approach	Emotions	Feature extraction	Model	Datasets	Train and test	Accuracy (%)
Hasani et al. [11]	Sequenced	7,6	AAM	CNN, CRF	CK+, MMI	80% train, 10% test	93.0, 78.6
Zhao et al. [31]	Sequenced	6	LBP, Gabor multiorientation	SVM	CK+, MMI	Leave one out	93.9, 71.9

**Table 4.** Number of samples for each emotion class in two datasets.

	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Total
CK+	45	59	25	69	28	83	309
MMI	32	28	28	42	32	41	203

15 different AUs, in both regression and binary scale, are presented by a  $30 \times 1$  vector and imposed as an input to the developed SAE, where the regression values of AUs are normalized between 0 to 1. The extracted features from SAE are in the shape of a  $10 \times 1$  vector, which are applied to the Softmax layer. Imposing the abstracted features vector into the Softmax layer, an original input data is classified into one of the six different basic emotion classes.

By applying 2D grid search, the hyper parameters of the SAE, e.g., learning rate and dropout are selected optimally in the learning process. The number of epochs and batch size are set as 200 and in the fully connected layer, “Adam” is used as optimizer and “Softmax” is used as supervised categorical classifier. Reconstructing the obtained features from SAE revealed the most pivotal AUs in each emotion as shown in Table 1.

## 4 Verification and Results

To verify the accuracy of the proposed model we applied it to three well-known datasets and compared obtained results with the results from two state-of-the-art methods, which showed convincing performance on these datasets. Table 3 summarizes two baselines. One of the baseline methods used a convolutional neural network, while the other used a SVM method. Since the training and testing approach and the used datasets are different for baselines, we defined different experiments to be in align with compared method. However, as both baselines used confusion matrix to show the accuracy of their model, we also showed the accuracy of our model by confusion matrix. In following we first review the applied datasets, then the comparison between proposed model and baselines are discussed through different experiments. For easiness of read in next subsections happiness, sadness, fear, anger, disgust, and surprise are indicated by H, Sa, F, A, D, and Su respectively.

#### 4.1 Data Bases

The extended Cohn-Kanade database (CK+) [16], contains 593 frontal face poses images of 123 subjects ranging from 18 to 50 years old. However, only 327 sequences from 118 subjects have labels. MMI, contains 203 video sequences, including different head poses and subtle expressions of 19 participants with ages ranging from 19 to 62 years old [20]. Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [15], contains frontal face poses videos from 12 female and 12 male, all north American actor and actress, expressing six basic emotions, calm, and neutral. While the sequences of all datasets start with the neutral state frame and end at the apex of the target emotion [31] removed the beginning frames and [11] labeled them as neutral, thereby we also removed the beginning frames of both datasets. Table 4 shows the number of each expression class in CK+ and MMI datasets in our experiments.

#### 4.2 Experiment A: Recognition Rate on CK+ Dataset for 6 Emotion Classes

The first experiment is conducted on the CK+ dataset. The best accuracy over CK+ for the six emotion recognition presented by Zhao et al. [31], which obtained by leave-one-out cross validation strategy. Hence, we also applied leave-one-out method to verify the accuracy of the proposed model. Table 5 shows the comparison between confusion matrices of proposed model and results reported in [31] over CK+ dataset.

Proposed model outperforms the baseline for 3 classes out of 6 classes, i.e., anger, sadness, and surprise, while baseline has higher accuracy rate for detecting disgust emotion. The model could detect all samples of happiness and surprise, i.e., 100% accuracy. The lowest accuracy is for recognising the fear class samples as 88% with misclassifying one sample as disgust and one sample as surprise. Overall the average accuracy of the proposed model is higher than baseline, i.e., 95.63% compared to 93.9%.

**Table 5.** Experiment A, confusion matrices for six emotion classification over the CK+ dataset, validated by leave-one-out technique.

(a) The result of the baseline [31].

(b) The result of the proposed model.

**Table 6.** Experiment A, six emotions classification over the MMI dataset.

(a) Accuracy comparison with 4 other baselines. (b) Confusion matrix of the proposed model validated by 10 fold cross validation.

Research	Method	Acc (%)	A	D	F	H	Sa	Su
Zhao et al [31]	SVM	71.92	<b>A</b> <b>93.75</b>	3.12	0.0	0.0	3.12	0.0
Hasani et al [11]	CNN CRF	78.67	<b>D</b> 3.22	<b>96.77</b>	0.0	0.0	0.0	0.0
Proposed model	SAE	<b>95.58</b>	<b>F</b> 0.0	0.0	<b>90.00</b>	0.0	3.33	6.66

### 4.3 Experiment B: Recognition Rate on MMI Dataset for 6 Emotion Classes

The second experiment performed on MMI dataset for which [11] obtained the best performance over it. We applied 10-fold cross validation to report our results, because Hasani et al. [11] used 5-fold cross validation and Zhao et al. [31] used 10-fold cross validation for verification. Table 6a shows that the proposed model outperforms both baselines significantly, i.e., 95.6% compared to 78.67% and 71.92%.

Since confusion matrices of baselines are not provided in main references, we compared the overall obtained accuracy. However, the confusion matrix of the proposed model is shown in Table 6b. Analysing Table 6b, the best accuracy is obtained for happiness and sadness with the accuracy of 100% and the lowest accuracy obtained for fear class with accuracy of 90%.

### 4.4 Experiment C: Recognition Rate on RAVDESS Dataset for 6 Emotion Classes

For further validation, we tested the proposed model on RAVDESS dataset [15]. While RAVDESS contains both facial and speech data, it is mostly used for speech emotion recognition and, to our best knowledge, is not used for FER, hence to confirm our results, we used Weka [27] to obtain the accuracy of four well-known classical machine learning models including K-nearest neighbors, e.g., 1NN and 2NN, Multilayer perceptron (MLP) with learning rate of 0.3, and decision tree (M5P). The batch-size for all models set as 200.

We designed a subject-independent experiment, i.e., the dataset is partitioned into two subsets for train and validation such that 18 subjects (9 female and 9 male), i.e., 75% of the total dataset considered as training set and the other 6 subjects (3 female and 3 male), i.e., 25% of the total dataset considered as test set.

Table 7a shows the comparison between proposed model (SAE) with four other classical machine learning approaches. The proposed model outperforms baselines in three classes of anger, fear and sadness out of six classes. The best

**Table 7.** Experiment C, confusion matrices for six emotions over the RAVDESS dataset.

(a) Comparison of different baselines over RAVDESS dataset for six emotion classification.

	<b>A</b>	<b>D</b>	<b>F</b>	<b>H</b>	<b>Sa</b>	<b>Su</b>	<b>Avg%</b>
<b>SAE</b>	<b>86.9</b>	90.2	<b>83.9</b>	90.9	<b>82.6</b>	75.0	<b>84.91</b>
<b>1NN</b>	80.5	83.9	73.6	<b>91.0</b>	75.5	78.2	80.45
<b>2NN</b>	83.2	81.9	72.0	89.5	74.1	78.1	79.8
<b>CART</b>	83.6	73.9	72.4	80.2	63.1	74.5	74.61
<b>MLP</b>	69.7	<b>100</b>	51.6	79.8	57.5	<b>86.7</b>	74.21

(b) Confusion matrix of the proposed model over the RAVDESS dataset.

	<b>A</b>	<b>D</b>	<b>F</b>	<b>H</b>	<b>Sa</b>	<b>Su</b>
<b>A</b>	<b>86.97</b>	3.10	3.19	0.00	2.46	4.26
<b>D</b>	2.33	<b>90.29</b>	2.26	0.00	4.00	0.61
<b>F</b>	3.54	3.30	<b>83.94</b>	0.00	5.02	4.17
<b>H</b>	0.9	0.46	2.4	<b>90.91</b>	0.73	4.58
<b>Sa</b>	3.31	6.40	6.06	0.00	<b>82.67</b>	1.53
<b>Su</b>	10.70	3.07	7.86	0.00	3.33	<b>75.00</b>

**Table 8.** Experiment C, confusion matrices for six emotion classification over the RAVDESS dataset for 4 different baselines.

(a) Confusion matrix of 1NN model.

	<b>A</b>	<b>D</b>	<b>F</b>	<b>H</b>	<b>Sa</b>	<b>Su</b>
<b>A</b>	<b>80.55</b>	4.33	3.71	0.00	4.15	7.24
<b>D</b>	4.49	<b>83.98</b>	2.66	0.00	5.18	3.67
<b>F</b>	4.78	4.38	<b>73.59</b>	0.00	6.55	10.68
<b>H</b>	1.00	0.00	1.23	<b>91.00</b>	1.72	4.15
<b>Sa</b>	5.84	6.44	3.60	0.00	<b>75.54</b>	8.55
<b>Su</b>	6.69	4.31	6.01	0.00	4.79	<b>78.17</b>

(b) Confusion matrix of 2NN model.

	<b>A</b>	<b>D</b>	<b>F</b>	<b>H</b>	<b>Sa</b>	<b>Su</b>
<b>A</b>	<b>83.20</b>	3.10	4.82	0.00	3.50	5.35
<b>D</b>	6.28	<b>81.92</b>	2.66	0.00	4.45	4.66
<b>F</b>	7.16	2.96	<b>71.99</b>	0.00	7.46	10.39
<b>H</b>	1.50	0.00	3.47	<b>89.55</b>	2.06	3.32
<b>Sa</b>	8.01	4.47	4.82	0.00	<b>74.07</b>	8.59
<b>Su</b>	7.09	3.49	7.27	0.00	4.02	<b>78.09</b>

(c) Confusion matrix of decision tree.

	<b>A</b>	<b>D</b>	<b>F</b>	<b>H</b>	<b>Sa</b>	<b>Su</b>
<b>A</b>	<b>83.63</b>	3.09	3.5	0.00	2.66	7.09
<b>D</b>	5.6	<b>73.90</b>	2.8	0.00	1.23	16.49
<b>F</b>	8.78	5.36	<b>72.45</b>	0.00	0.00	12.56
<b>H</b>	5.16	0.22	5.97	<b>80.22</b>	6.18	2.21
<b>Sa</b>	15.45	3.60	10.47	0.00	<b>63.12</b>	7.33
<b>Su</b>	13.50	1.5	8.01	0.00	2.37	<b>74.54</b>

(d) Confusion matrix of MLP.

	<b>A</b>	<b>D</b>	<b>F</b>	<b>H</b>	<b>Sa</b>	<b>Su</b>
<b>A</b>	<b>69.73</b>	5.99	0.0	0.0	0.0	24.28
<b>D</b>	0.0	<b>100</b>	0.0	0.0	0.0	0.0
<b>F</b>	37.25	2.63	<b>51.64</b>	0.0	0.0	8.47
<b>H</b>	5.71	0.26	7.33	<b>79.81</b>	6.34	5.27
<b>Sa</b>	9.29	2.71	28.18	0.0	<b>57.49</b>	2.32
<b>Su</b>	0.0	13.32	0.0	0.0	0.0	<b>86.68</b>

performance for happiness achieved by 1NN and for disgust and surprise by MLP. The overall average accuracy of the proposed SAE based model is 84.91% which outperforms all other baselines. The confusion matrix of the proposed model on the test dataset is shown in Table 7b. Also, the confusion matrices of provided baselines over RAVDESS are shown through Table 8.

## 5 Conclusion

Since one facial expression might have an ambiguity or similarity to some other basic emotions, precise Facial Emotion Recognition is a challenging task. To

find the best features for recognizing different emotions we used Stacked Auto Encoder, which is able to find high order features, which are not possible to craft by humans. The provided raw input data for SAE is the activation value of AUs, the final output, i.e., feature set, is the combination of most pivotal AUs for each basic emotion. The obtained feature set then is imposed to a Softmax classifier layer to find 6 basic emotions.

The proposed method is compared with several key methods and the experiments' results show that it is capable to outperform all rival methods. The proposed method achieves average accuracy of 95.63%, 95.55% and 84.91% for CK+, MMI and RAVDESS datasets respectively. Overall the best accuracy obtained for classifying happiness, while the worst result obtained for classifying fear.

In future work, we will apply the proposed method to classify more emotion classes. Also other features like head pose and gaze direction will be investigated to improve the accuracy of the proposed model. Furthermore, we will apply the proposed SAE to find the intensity of the detected emotion.

**Acknowledgment.** The work leading to these results has received funding from the European Commission 7th Framework Program as a part of the DREAM project, grant no. 611391 and the ICON project ROBO-CURE.

## References

1. Baltrušaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: OpenFace 2.0: facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, pp. 59–66. IEEE (2018)
2. Brave, S., Nass, C., Hutchinson, K.: Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. *Int. J. Hum.-Comput. Stud.* **62**(2), 161–178 (2005)
3. Caridakis, G., Castellano, G., Kessous, L., Raouzaiou, A., Malatesta, L., Asteriadis, S., Karpouzis, K.: Multimodal emotion recognition from expressive faces, body gestures and speech. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 375–388. Springer (2007)
4. Cohn, J.F., Ambadar, Z., Ekman, P.: Observer-based measurement of facial expression with the facial action coding system. In: The Handbook of Emotion Elicitation and Assessment, pp. 203–221 (2007)
5. De Gauquier, L., Cao, H.L., Gomez Esteban, P., De Beir, A., van de Sanden, S., Willems, K., Brengman, M., Vanderborght, B.: Humanoid robot pepper at a Belgian chocolate shop. In: Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pp. 373–373. ACM (2018)
6. Du, S., Tao, Y., Martinez, A.M.: Compound facial expressions of emotion. In: Proceedings of the National Academy of Sciences, p. 201322355 (2014)
7. Ebrahimi Kahou, S., Michalski, V., Konda, K., Memisevic, R., Pal, C.: Recurrent neural networks for emotion recognition in video. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pp. 467–474. ACM (2015)
8. Ekman, P.: Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique (1994)
9. Ekman, P.: Facial Action Coding System (FACS). A human face (2002)

10. Haag, A., Goronzy, S., Schaich, P., Williams, J.: Emotion recognition using bio-sensors: first steps towards an automatic system. In: Tutorial and Research Workshop on Affective Dialogue Systems, pp. 36–48. Springer (2004)
11. Hasani, B., Mahoor, M.H.: Spatio-temporal facial expression recognition using convolutional neural networks and conditional random fields. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2017, pp. 790–795. IEEE (2017)
12. Ko, B.: A brief review of facial emotion recognition based on visual information. *Sensors* **18**(2), 401 (2018)
13. Liliana, D.Y., Basaruddin, T.: Review of automatic emotion recognition through facial expression analysis. In: 2018 International Conference on Electrical Engineering and Computer Science (ICECOS), pp. 231–236. IEEE (2018)
14. Liu, M., Li, S., Shan, S., Chen, X.: AU-aware deep networks for facial expression recognition. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 1–6. IEEE (2013)
15. Livingstone, S.R., Russo, F.A.: The Ryerson audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. *PLoS One* **13**(5), e0196391 (2018)
16. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended Cohn-Kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 94–101. IEEE (2010)
17. Mehrabian, A.: Nonverbal Communication. Routledge, Abingdon (2017)
18. Miyakoshi, Y., Kato, S.: Facial emotion detection considering partial occlusion of face using Bayesian network. In: 2011 IEEE Symposium on Computers & Informatics (ISCI), pp. 96–101. IEEE (2011)
19. Mollahosseini, A., Chan, D., Mahoor, M.H.: Going deeper in facial expression recognition using deep neural networks. In: 2016 IEEE Winter conference on applications of computer vision (WACV), pp. 1–10. IEEE (2016)
20. Pantic, M., Valstar, M., Rademaker, R., Maat, L.: Web-based database for facial expression analysis. In: 2005 IEEE International Conference on Multimedia and Expo, pp. 5–pp. IEEE (2005)
21. Pitaloka, D.A., Wulandari, A., Basaruddin, T., Liliana, D.Y.: Enhancing cnn with preprocessing stage in automatic emotion recognition. *Proc. Comput. Sci.* **116**, 523–529 (2017)
22. Pop, C.A., Simut, R., Pintea, S., Saldien, J., Rusu, A., David, D., Vanderfaillie, J., Lefever, D., Vanderborght, B.: Can the social robot probo help children with autism to identify situation-based emotions? A series of single case experiments. *Int. J. Humanoid Rob.* **10**(03), 1350025 (2013)
23. Schuller, B., Rigoll, G., Lang, M.: Hidden Markov model-based speech emotion recognition. In: Proceedings of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2003, vol. 2, pp. II–1. IEEE (2003)
24. Tong, Y., Liao, W., Ji, Q.: Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(10), 1683–1699 (2007)
25. De la Torre, F., Cohn, J.F.: Facial expression analysis. In: Visual Analysis of Humans, pp. 377–409. Springer (2011)
26. Vydana, H.K., Kumar, P.P., Krishna, K.S.R., Vuppala, A.K.: Improved emotion recognition using GMM-UBMs. In: 2015 International Conference on Signal Processing and Communication Engineering Systems (SPACES), pp. 53–57. IEEE (2015)

27. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
28. Xiaoxi, M., Weisi, L., Dongyan, H., Minghui, D., Li, H.: Facial emotion recognition. In: 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), pp. 77–81. IEEE (2017)
29. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(1), 39–58 (2009)
30. Zhang, L., Verma, B., Tjondronegoro, D., Chandran, V.: Facial expression analysis under partial occlusion: a survey. *ACM Comput. Surv. (CSUR)* **51**(2), 25 (2018)
31. Zhao, L., Wang, Z., Zhang, G.: Facial expression recognition from video sequences based on spatial-temporal motion local binary pattern and Gabor multiorientation fusion histogram. *Math. Probl. Eng.* **2017**, 12 (2017)



# Producing Parameterized Value Functions Through Modulation for Cognitive Developmental Robots

Alejandro Romero, Francisco Bellas<sup>(✉)</sup>, Jose A. Becerra,  
and Richard J. Duro

CITIC Research Center, GII, Universidade da Coruña, A Coruña, Spain  
{alejandro.romero.montero, francisco.bellas,  
jose.antonio.becerra.permuy, richard}@udc.es

**Abstract.** Parameterizing value functions as a representation of robotic tasks in different domains allows for their generalization, and can provide a way to transfer knowledge to new situations. To this end, in this paper we propose a modulation based mechanism embedded within a cognitive architecture for robots. It makes use of the combined operation of the long-term memory and the motivational system in order to select candidate primitive value functions for transfer. These are then adapted to the new situation through the addition of modulatory ANNs to progressively conform new parameterized value functions able to address more complex situations in a developmental manner. The proposed method is tested in a Baxter robot, which must solve different tasks in a cooking setup.

**Keywords:** Developmental robotics · Motivational system · Value functions · Long-term memory · Cognitive architecture

## 1 Introduction

When one is dealing with robots that must operate in open-ended learning settings [1], being able to reduce the number of interactions with the world the robot requires to learn to perform a task in a new domain becomes of paramount importance. The process of reusing knowledge acquired in one task to improve performance in others is called transfer. Transfer learning has been addressed very intensively in the Reinforcement Learning literature [2, 3] and also in the Deep Reinforcement one [4]. In the majority of papers, it is addressed by transferring policies or parts of policies to be reused [5, 6].

On the other hand, as Deisenroth et al. [7] point out, training individual policies for each single potential task, as often carried out in Reinforcement Learning, is not affordable in open-ended learning. This is even more evident when continuous task variations are contemplated. An example could be that of positioning objects at different points on a table. It would seem wasteful to have to develop a policy for each particular position in which the robot needs to place the object. If one is able to generalize at the task or domain level, that is, deduce how to solve a new task from the

interpolation or extrapolation of solutions to previous tasks, the transfer problem becomes even more accessible.

Some approaches resort to learning local policies and achieving generalization by combining them, for instance through a gating mechanism [8], as in the work of Mülling et al. [9] in Reinforcement Learning based robotics, or in that of [10] from the behavior based robotics perspective. These combinations can become quite complex through the creation of compound hierarchies as in the case of Konidaris and Barto [11], who create reusable options with the objective of constructing hierarchies of policies to solve sequences of tasks. However, gating mechanisms base their operation on discrete instances of policies and it is not clear how to generalize them to continuously varying tasks.

To try to address this issue, several authors have resorted to the concept of parameterized policies as a path to a more continuous policy generalization. In other words, the problem is addressed as that of determining a function in the algebraic dual space, that is, the space of functions over policies (or actions) that, through a set of parameters, allows choosing the particular policy to activate. Within this approach, some authors consider the parameterization at the level of actions and address the problem of learning the action selection policy that determines the parameters that must be input to the action to solve the required task. An example is the Q-PAMDP approach used in [12]. Others start from a set of specific problems, which they somehow parametrize, and try to obtain a policy that can solve the whole set of problems and that will adapt to each one of them depending on the parameter value. In other words, they include the task parameter vector in the state descriptor and handle the entire set of tasks as a single one. This presents several drawbacks. Among them, the set of tasks needs to be known *a priori* before generalizing and, in an open-ended learning scenario, this is not always feasible.

Different authors have tried to improve on the aforementioned approach. In the work of [13], the authors sample tasks from a distribution for which they learn policies and use them to estimate the topology of the lower-dimensional piecewise-smooth manifold on which the policies lie, as a way to model how policy parameters are modified as task parameters vary. From a Deep Reinforcement Learning perspective, Zhao et al. [4] try to factor the tensor representing a task into task dependent and task independent components. They use the latter to transfer knowledge from one task to the next providing seeds through a curriculum of progressively more complex tasks.

Many of the approaches assume that a designer decides what policies or knowledge nuggets it is going to use to generate the new policy. However, in open-ended learning processes within the robotics domain, a robot is expected to learn an unbounded sequence of *a priori* unknown tasks, preferably without designer supervision. It is the robot itself who must decide what previous knowledge may be useful to address the new problem. It must make hypotheses on what policies may be useful for the new domain and task, and adequately combine them when necessary. Thus, as mentioned by [13], the question of how to actively select training tasks to improve the overall readiness of a parameterized skill, given a distribution of expected future tasks, needs to be addressed.

In this paper, we present an initial proposal for a developmental approach to the generalization of value functions, expressed here as artificial neural networks, through

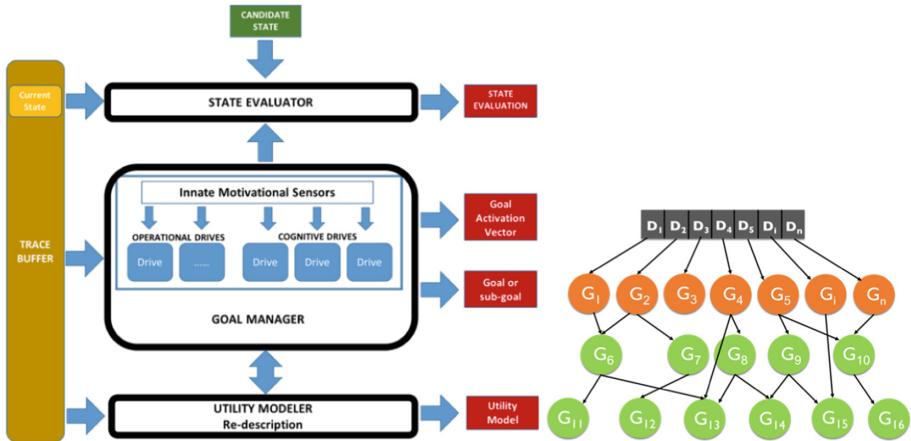
their combination and progressive parameterization by means of modulating networks, as a path to allow for transfer and generalization. The use of value functions as a more general representation of tasks provides a certain independence from the action space, thus allowing the robots to find different policies leading to the same goal depending on their available action repertoire. On the other hand, establishing a progressive developmental approach towards this parameterization allows for the adaptation of the robot to the actual experience it acquires through its particular path of interaction with the world, without having to make any hypotheses on what will be necessary.

In practical implementation terms, the approach revolves around the interaction between a motivational engine and a Long Term Memory (LTM) structure within the Multilevel Darwinist Brain (MDB) cognitive architecture [14, 15] in order to select the appropriate value functions to use for transfer. These seeds are then adapted to the particular domain through a modulation based strategy that allows for the progressive incorporation of parameters to the value functions. These components of the MDB as well as the modulation based approach will be described in the following sections and a series of initial experiments over a real robot will be presented.

## 2 Motivational Engine

The MDB cognitive architecture [14] has been extensively improved in the last three years within the DREAM EU project. It is out of scope here to go into its detailed operation, but it should be pointed out that it was designed to endow an autonomous robot with open-ended learning capabilities. The MDB starts its operation with a set of innate drives, utility models, forward models and policies that are required for the system to start learning autonomously. This innate knowledge allows the robot to explore its environment seeking goals, learning new models, and policies. As time progresses, the MDB architecture obtains and stores more perceptual data from the autonomous interaction of the robot with its environment, which make up the basic information required to learn different knowledge nuggets. These are sent to a Long-Term Memory (LTM) for their management and for decision making. The consequences of the actions that are applied are analyzed by a Motivational Engine and the LTM to improve the relations between innate knowledge and acquired knowledge with the final objective of allowing lifelong learning by means of a continuous fulfillment of the robot drives.

The approach to motivation followed in the MDB architecture starts from the premise that, when in an environment, the robot should try to learn how to fulfill its innate drives, which determine its internal motivations. Therefore, it must seek areas of its perceptual state space for which some drives are fulfilled, which are called goals. Thus, a key element in the architecture is the motivational engine, which must try to find goals, store them by linking them to other goals and ultimately to the innate drives, and learn how to get to them from any point in its perceptual state space so that it can fulfill its drives. The structure of the motivational engine developed for the MDB is displayed in the left part of Fig. 1 and described in depth in [16]. It consists of three main modules.



**Fig. 1.** Motivational Engine operation scheme (left) and graph structure of the goal and drive relations (right).

The function of the Goal Manager (central block) is to find goals and store them. This implies relating them, when possible, as goal graphs that may facilitate future operation (as displayed in Fig. 1 right). These goal graphs are usually kept in the Long-Term Memory, but they are managed by the Goal Manager. Consequently, it is also its function to determine what goals in the graph should be active and how much in order to fulfill the drives. The goal manager makes use of the State Evaluator, a module that determines the expected value of a perceptual state according to the robot's experience.

To determine the expected value of a state space point, and thus provide information on how to reach goals, there is a need to produce utility models. Whatever the representation used, the Utility Modeler component of the Motivational Engine is in charge of extracting information from the robot operation in order to produce utility models leading to the different goals. The utility models could be represented in many different ways [16], but in this paper they will correspond to the concept of value function used in reinforcement learning, that is, a function which provides the expected utility for a given perceptual state.

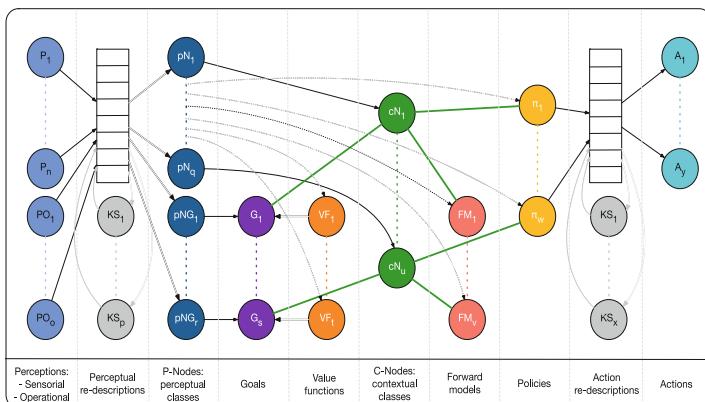
In order to provide a value function that helps the robot reach the goal, we follow a procedure already presented in [17]. As a summary, let's start by stating that the information that is available is the trace of the previous  $n$  states the robot has gone through before reaching the goal. By propagating the reward obtained at the goal back along the trace and its neighborhood, one obtains a training set that consists of state space points with a corresponding expected value. This training set can be used to train an artificial neural network to model the expected value of this area of state space. In addition, an associated certainty function can be generated that represents how confident we are of the utility value assigned to the different points in that area (as we go further away from the goal and the points in the trace, that confidence decreases). As more traces to the goal are obtained, the model progressively becomes better and encompasses more of state space, allowing the robot to consistently reach the goal.

In this paper, we will assume that several value models (in the form of ANNs) have been obtained by the robot under different circumstances in previous interactions with the world. Here we will concentrate on the problem of determining which, if any, of them may be relevant to a new situation faced by the robot, so that they can be used as initial hypotheses in a knowledge transfer process and on how to adapt them to this new situation. These previously obtained goals and value models are stored in the Long-Term Memory of the architecture and, in the next section, we provide a brief description of the LTM used in the MDB and the way it leads to the pre-activation of value functions that might be relevant to a new situation that is detected.

### 3 Long-Term Memory

Long-term memories (LTM) are used in cognitive architectures to store knowledge that has proven useful so that it can be reused whenever it is required in the future. Additionally, if we want to implement a LTM able to support life-long learning, knowledge representation must be as efficient as possible (in order to support fast activation of elements when necessary) and, at the same time, provide support for all the learning and operational procedures required by the agent, taking into account that the knowledge it uses is in the form of unlabeled ANNs. This implies an operational mechanism based on the activation of nodes in an associative structure whose connections are progressively established through experience.

Figure 2 shows the current LTM implementation in the MDB architecture which is described in more detail in [15]. It contains the typical knowledge nuggets present in any cognitive architecture: perceptions (raw or re-descriptions), policies, goals (and the value functions that lead to them) and forward/world models. There are also two components that are particular to the LTM implementation of the MDB: contextual nodes (C-nodes) and perceptual nodes (P-nodes).



**Fig. 2.** LTM implementation diagram for the MDB.

C-nodes represent contexts and are in charge of connecting the perceptions, the goal, the world model and the policy that were active in a context for which value was obtained. Perceptual nodes (P-nodes) are used to provide generalizations over perceptions. P-nodes are connected to C-nodes and each one delimits the sub-space of the perceptual space for which a given C-node is meaningful and must be activated. How these C-nodes and P-nodes are created and improved over time is explained in [15]. Here, it suffices to say that the LTM uses the feedback provided by the motivational engine (regarding whether robot actions are improving value or not) to link knowledge nuggets to C-nodes and tune P-nodes after each policy execution.

The operation of this LTM is activation based. The activation flow starts with the arrival of new perceptions. Those P-nodes that determine that these perceptions are within the perceptual class they represent become active. In parallel, and depending on the previous perceptual flow, the world model that is predicting the operation of the current world more accurately will become active. At the same time, the motivational engine will be doing its thing and, depending on the activation of its innate drives, it will percolate this activation to the goals stored in the LTM that are connected to them (and consequently, to the value functions linked to these goals). These activations will lead to the activation of those C-nodes for which their connected world model, P-node and goal are active.

It is also possible for the robot to find itself in a new situation for which more than one of the C-nodes become active at the same time. If they are connected to different goals, this implies that several of them are desirable according to the motivational engine and are relevant according to the perceptions in this new situation. Now, the new domain/task can be described as a combination of the value functions of the relevant goals. This combination can lead to a combination of the corresponding policies or even to the generation of a completely new policy that is better adapted to this combination of value functions. Whichever the case, the functions need to be combined and their relative importance established. This is the other case where the modulation based mechanism plays a role.

## 4 Value Function Modulation Combination

We are interested on how to adapt and combine value functions that have already been learnt in previous environments and situations and are, thus, stored in Long-Term Memory (LTM), in order to produce initial guesses on the value function of a new situation. Taking inspiration from modulatory approaches in real brains [17] we propose an output modulation based strategy using artificial neural networks that allows for the generation of value functions adapted to new situation that make use of the previous value functions without actually modifying them.

The simplest output modulation based architecture is made up of a modulating ANN and a modulated ANN. Output modulation is achieved by multiplying their outputs in order to generate the final output. Whether an ANN (or node in graph terminology) is modulating or modulated is given by which one is trained (the modulator). It is important to note here that the modulating ANN, when trained, learns to modify the output of the modulated ANN as a function of the modulating inputs. When

operating, depending on its inputs, the modulating ANN applied to the modulated ANN produces a continuous variation of the function of the modulated ANN.

If the modulation based architecture grows, when there is more than one node  $M_i$  ( $i = 0, 1, \dots, n$ ) that modulate one output of another node  $X_j$ , the resulting modulating value will be the product of the individual modulations. Assuming that a node  $M_i$  has to modulate the values of  $n$  outputs, its number of outputs must necessarily be  $n$ . Finally, we will assume that, when there is more than one node that provides values for the same output (more than one value function providing expected utility for the same point in state space), this output receives the sum of these values.

As previously indicated, to produce this type of modulation based value functions (VFs onwards) we will start by assuming that a set of VFs, as well as their certainty areas or P-nodes, have already been obtained in previous situations and are stored in LTM. When a new situation arises where the P-node corresponding to the C-node of a particular VF becomes activated, we may assume that the VF may be relevant for that situation. Consequently, if it does not perform adequately, it will be chosen to be adapted to it through the creation of a new modulating ANN that will use, as input to the modulator, the perceptions available to the system, out of which it will choose the relevant ones during the training process. More than one VF may become activated, and in this case, all of them will be combined and become candidates for joint modulation. The final modulated structure constitutes a new VF but, in this case, parameterized by the variables that constitute the inputs to the modulating ANN. Obviously, any modulated structure can participate in a new modulating structure, thus providing a way to smoothly scale the level of parameterization of the VFs.

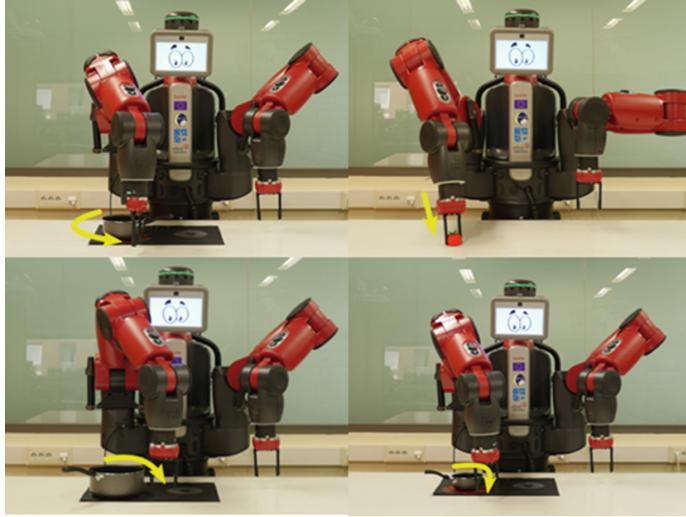
## 5 Real Robot Experiment

To illustrate the approach to producing parameterized VFs through modulation, a real robotic experiment has been designed. The experimental setup includes a Baxter robot, a white table, a small and a big cooker, some pots and a tomato. The stoves can be on or off, and they can have a pot on them or not. Throughout this section, different situations will be presented to the robot, which must solve them by creating appropriate utility functions in a developmental fashion.

When the experiment starts, we assume that the robot has previously learned some VFs, which are stored in LTM. Specifically, two primitive VFs are considered:

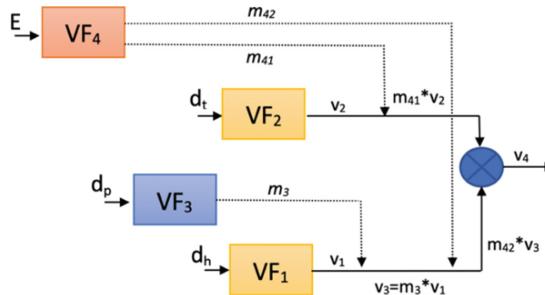
- *Avoiding a heat source (VF1)*: the robot is able to avoid with its hand a hot pot of a predefined size (0.20 cm of diameter), as shown in Fig. 3 (top left). This VF has one input, the distance to the heat source  $dh$ , and one output, the estimated value ( $v1$ ). As a consequence:  $v1 = VF1(dh)$ .
- *Approaching an object (VF2)*: the robot is able to grab a tomato with its gripper, as shown in Fig. 3 (top right). This VF has one input, the distance to the tomato  $dt$ , and one output, the estimated value ( $v2$ ). It provides the estimated value in the whole range of possible distances to the tomato. That is:  $v2 = VF2(dt)$ .

What is relevant here is that these two primitive VFs were obtained separately, which means that the robot never faced a situation with a tomato and a hot pot before.



**Fig. 3.** Top images: the robot learned to avoid a hot pot of a predefined size (left) and to grab a tomato with its gripper in a different environment (right). Bottom images: it learned to avoid a bigger (left) and smaller hot pot (right) simply by modulating the original VF.

In the following subsections, two new scenarios will be presented to the robot that include combinations of the basic elements in the cookers. To solve them, new VFs must be learned, in this case by modulation and parameterization of the primitive VFs in a developmental way, as displayed in Fig. 4.



**Fig. 4.** Schematic representation of the VF combination created in this experiment. Dotted lines correspond to modulation operations (multiplication in this case), while the circle on the right corresponds to a linear combination of modulated signals.

### 5.1 Avoid a Hot Pot Whatever Its Size

In this first setup (bottom images of Fig. 3), the robot must learn to avoid hot objects of different sizes starting from the knowledge it has about avoiding a heat source ( $VF_1$ ) of fixed size. As the new pots are bigger,  $VF_1$  does not work here and the robot receives a

pain signal. To solve this, there are two options: learn a new VF from scratch or learn it as a modulation of  $VF_1$ . In terms of modulation, it simply implies multiplying the output of the primitive VF as a function of the pot size and, in addition, this second approach creates new knowledge by reusing a previous one. Consequently, the second option will be the one used here, as commented above.

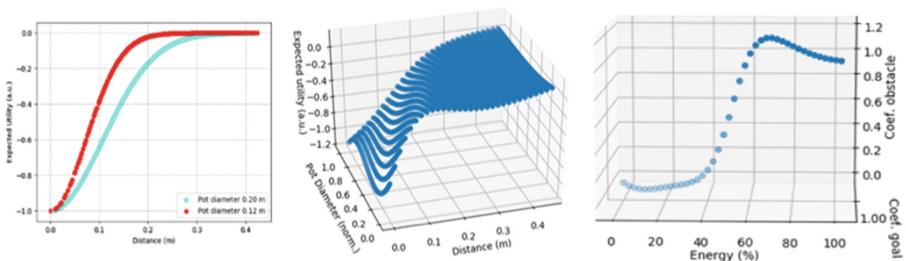
The learning process works as follows: the robot starts operating in this setup and, as the state space is similar to the primitive one, the stored P-node corresponding to  $VF_1$  is activated and  $VF_1$  is imported from LTM. But, as  $VF_1$  does not work properly now, the avoid-pain drive increases. Consequently, the motivational engine selects the active primitive  $VF_1$  for adaptation through the creation of a new modulating ANN, leading to the learning of a new  $VF_3$  for this new situation.

Specifically, the MDB creates a modulator VF,  $VF_3$ , which has one input, the pot diameter ( $dp$ ) and one output ( $m_3$ ) a factor that multiplies the output of  $VF_1$ ,  $v_1$ , to obtain the new predicted value  $v_3$  for the new context (displayed in the bottom part of Fig. 4). Summarizing:

$$VF_3 = M_1(VF_1), \text{ where } M_1 \text{ represents a modulator function}$$

$$m_3 = VF_3(dp), \text{ where } VF_3 \text{ learning is evaluated using } v_3 = m_3 * v_1$$

The learning of  $VF_3$  is carried out online using the Adam algorithm with a MLP neural network following the procedure explained in Sect. 2. Learning was performed during 180 epochs of training, which corresponds to 7 traces and around 50 samples, that is, at a very low computational cost as compared to obtaining the original primitive  $VF_1$ , which required 670 epochs, 11 traces and 940 samples. Figure 5 displays, on its left side, the output of  $VF_1$  (cyan line) compared to the output of the modulated  $VF_3$  (red line) for a specific pot diameter of 0.12 cm, where it can be observed that the response is basically the same in both. The 3D shape of  $VF_3$  is displayed in Fig. 5 (center), which shows how the distance at which the robot begins to perceive negative utility varies depending on pot diameter, as expected.



**Fig. 5.** 2D state space covered by the VFs of the first setup: primitive avoiding  $VF_1$  in cyan and modulated  $VF_3$  in red (left). Parameterized Value Function  $VF_3$  (center) in a 3D representation. Output provided by the modulator ANN in the second setup (right).

This approach allows us to use previous knowledge to learn more efficiently in a simple way. In addition, the parameterization of the VF allows learning all the intermediate cases, which, if learned individually from scratch, would imply a long and tedious process. When this learning stage finishes, the robot is able to avoid a hot pot whatever its size, and the new parametrized  $VF_3$  is stored in LTM.



**Fig. 6.** Top (small pot): the Baxter robot has high energy value, so it avoids the hot pot to reach the tomato. Bottom (big pot): the energy level is low, so it passes the hand over the hot pot to obtain energy fast.

## 5.2 Grasp a Tomato Avoiding a Hot Pot When Hungry

In this second setup, in addition to the hot pot, we introduce a pan with a tomato (see Fig. 6 for an overall view of the new setup). The robot must learn to grab the tomato without touching the hot pot. The modulation strategy now implies the combination of the primitive VF corresponding to *approaching* ( $VF_2$ ), and the one obtained in the previous stage, *parametrized avoiding* ( $VF_3$ ). As a consequence, in this new situation, the robot receives reward as a combination of positive ( $VF_2$ ) and negative ( $VF_3$ ) feedbacks. The best VF in the combined case should guide the robot hand towards the tomato maximizing positive utility, that is:

$$VF_4 = M_2(VF_2, VF_3) = M_2(VF_2, M_1(VF_1))$$

In this setup, we introduce an internal sensor of energy ( $E$ ) that increases when it grabs the tomato and decreases if it touches the hot pot, so the robot can modulate its behavior depending on it. The new modulator ANN,  $VF_4$ , has only one input, the internal energy ( $E$ ), and two outputs ( $m_{41}$  and  $m_{42}$ ) that multiply the output of each

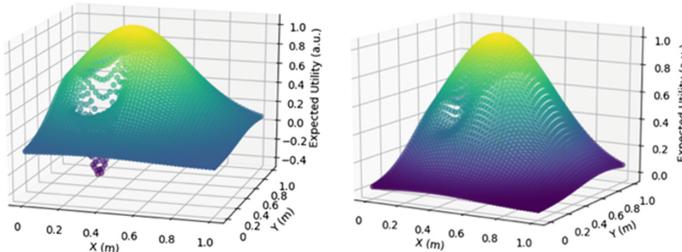
primitive VF (*VF*<sub>2</sub> and *VF*<sub>3</sub>), as shown in the diagram of Fig. 4. The new expected value (*v*<sub>4</sub>) is obtained as the linear combination of these two modulated outputs:

$$\begin{aligned}[m_{41}, m_{42}] &= \text{VF}_4(E), \text{ where } \text{VF}_4\text{learning is evaluated using } v_4 \\ &= m_{41} * v_2 + m_{42} * v_3 \end{aligned}$$

Although this setup is more complex, in terms of the modulator inputs and outputs, it is simple. In fact, the learning process performed with the Adam algorithm required 125 epochs of training, corresponding to 6 traces, which again was carried out in real time with no relevant delay that affected the performance of the robot.

The resulting modulator ANN *VF*<sub>4</sub> provides the output shown in the right image of Fig. 5, which displays how the coefficient modulating *VF*<sub>4</sub> becomes smaller as the energy of the robot decreases. This means that if the robot “feels hungry”, it will prioritize reaching the tomato in the most direct way, minimizing the importance of the hot pot. Whereas, if the energy value is high, its priority will be to avoid pain, and it will try to avoid the hot pot on its way to the goal.

The response produced by the robot is displayed in the sequences of Fig. 6. Top images correspond to a small pot while bottom ones correspond to a big pot. When the energy value is high, it is able to grasp the tomato with its hand avoiding the hot pot (top pictures), since the important thing is not to feel pain. However, when the robot “feels hungry” (low energy value) the priority is to grab the tomato in the most direct way (bottom pictures). Additionally, the resulting modulated VFs associated to the situations shown in Fig. 6 can be seen in Fig. 7. As expected, the new VF is a combination of the primitive ones with the operation ranges adapted according to the energy value.



**Fig. 7.** 3D representation of the state space covered by *VF*<sub>4</sub>. Left: when Energy value is high. Right: when energy value is low.

With these final results, the potential of the modulation approach for producing parameterized value functions becomes evident. It allows solving complex situations by reusing previously learned knowledge in a very efficient way, being this the base of cognitive developmental robotics.

## 6 Conclusions

In this paper we propose a modulation based approach to the developmental acquisition of parameterized value functions (VF) as a way to generalize them and to provide for transfer learning. This modulation structure allows for the adaptation and combination of primitive VF so that the cognitive architecture can progressively construct and adapt compound parameterized VFs for ever more complex situations in a developmental manner. This is achieved by leveraging the operation of the LTM and the Motivational Engine of the cognitive architecture in order to select candidate primitive VFs that are then adapted and generalized to the new situations through modulation. The operation has been illustrated on a simple real robot example, where a complex response has been obtained in a developmental way. Thus, the robot was able to produce, in real time, new parameterized VFs as modulation of other previously learned VFs.

**Acknowledgments.** This work has been partially funded by the EU's H2020 research programme (grant No 640891 DREAM), Ministerio de Ciencia, Innovación y Universidades of Spain/FEDER (grant RTI2018-101114-B-I00), Xunta de Galicia and FEDER (grant ED431C 2017/12), and by the Spanish Ministry of Education, Culture and Sports for the FPU grant of Alejandro Romero.

## References

1. Doncieux, S., et al.: Open-ended learning: a conceptual framework based on representational redescription. *Front. Neurorobotics* **12**, 59 (2018)
2. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: a survey. *J. Mach. Learn. Res.* **10**, 1633–1685 (2009)
3. Lazaric, A.: Transfer in reinforcement learning: a framework and a survey, adaptation, learning, and optimization (2012)
4. Zhao, C., Hospedales, T. M., Stulp, F., Sigaud, O.: Tensor based knowledge transfer across skill categories for robot control. In: Proceedings of IJCAI, pp. 3462–3468 (2017)
5. Devin, C., Gupta, A., Darrell, T., Abbeel, P., Levine, S.: Learning modular neural network policies for multi-task and multi-robot transfer. In: Proceedings IEEE ICRA (2017)
6. Fernández, F., García, J., Veloso, M.: Probabilistic policy reuse for inter-task transfer learning. *Rob. Auton. Syst.* **58**, 866–871 (2010)
7. Deisenroth, M.P., Englert, P., Peters, J., Fox, D.: Multi-task policy search for robotics. In: Proceedings IEEE ICRA (2014)
8. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. *Neural Comput.* **3**, 79–87 (2008)
9. Mülling, K., Kober, J., Kroemer, O., Peters, J.: Learning to select and generalize striking movements in robot table tennis. *Int. J. Robot. Res.* **32**, 263–279 (2013)
10. Duro, R.J., Santos, J., Becerra, J.A.: Evolving ANN controllers for smart mobile robots. In: Future Directions for Intelligent Systems and Information Sciences, pp. 34–64. Springer, Heidelberg (2000)
11. Konidaris, G., Barto, A.: Building portable options: skill transfer in reinforcement learning. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 895–900 (2007)
12. Masson, W., Ranchod, P., Konidaris, G.: Reinforcement learning with parameterized actions. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)

13. Da Silva, B., Konidaris, G., Barto, A.: Learning parameterized skills. In: Proceedings ICML 2012, pp. 1443–1450 (2012)
14. Bellas, F., Duro, R.J., Faiña, A., Souto, D.: Multilevel Darwinist Brain (MDB): artificial evolution in a cognitive architecture for real robots. *IEEE Trans. Auton. Mental Dev.* **2**(4), 340–354 (2010)
15. Duro, R.J., Becerra, J.A., Monroy, J., Bellas, F.: Perceptual generalization and context in a network memory inspired long term memory for artificial cognition. *Int. J. Neural Syst.* **29** (06), 1850053 (2019)
16. Prieto, A., Romero, A., Bellas, F., Salgado, R., Duro, R.J.: Introducing separable utility regions in a motivational engine for cognitive developmental robotics. *Integr. Comput. Aided Eng.* **26**(1), 3–20 (2019)
17. Harris-Warrick, R.M.: Modulation of neural networks for behaviour. *Ann. Rev. Neurosci.* **14**, 39–57 (1991)



# Benchmarking Deep and Non-deep Reinforcement Learning Algorithms for Discrete Environments

Fernando F. Duarte<sup>1</sup>(✉), Nuno Lau<sup>1</sup>, Artur Pereira<sup>1</sup>, and Luís P. Reis<sup>2</sup>

<sup>1</sup> IEETA, University of Aveiro, Aveiro, Portugal

{fjosefradique, nunolau, artur}@ua.pt

<sup>2</sup> LIACC, University of Porto, Porto, Portugal

lpreis@fe.up.pt

**Abstract.** Given the plethora of Reinforcement Learning algorithms available in the literature, it can prove challenging to decide on the most appropriate one to use in order to solve a given Reinforcement Learning task. This work presents a benchmark study on the performance of several Reinforcement Learning algorithms for discrete learning environments. The study includes several deep as well as non-deep learning algorithms, with special focus on the Deep Q-Network algorithm and its variants. Neural Fitted Q-Iteration, the predecessor of Deep Q-Network as well as Vanilla Policy Gradient and a planner were also included in this assessment in order to provide a wider range of comparison between different approaches and paradigms. Three learning environments were used in order to carry out the tests, including a 2D maze and two OpenAI Gym environments, namely a custom-built Foraging/Tagging environment and the CartPole environment.

**Keywords:** Reinforcement Learning · Planning · Deep Q-Network · Q-Learning · Value Iteration · Neural Fitted Q-Iteration · Policy gradient optimization

## 1 Introduction

The introduction of the Deep Q-Network (DQN) [1, 2] algorithm showed the viability of Deep Learning (DL) in the domain of Reinforcement Learning (RL) tasks. Since then, many new improvements as well as new approaches to this algorithm have been proposed. As an example of this, Double Deep Q-Network (DDQN) [3] addresses the problem of the overestimation of the action values that both Q-Learning as well as DQN seem to suffer from, by decoupling the selection of the action from its evaluation. As another example, the Dueling Network Architecture (DNA) [4] proposes a new neural network architecture to help generalize learning across actions by explicitly separating the representation of state values and action advantages. Prioritized Experience Replay (PER) [5] proposes a framework to prioritize experience to help improve the efficiency of the learning process. As yet another example, in [6] the authors propose a categorical algorithm that learns the distribution of the discounted returns as opposed to using the expectation of this return. Adding to this plethora of algorithms,

the policy gradient approach [7], of which the Trust Region Policy Optimization (TRPO) [8, 9] method is an example of, is also a viable alternative.

Regarding the discrete case, there are also some non-deep RL algorithms such as Q-Learning (QL) and Value Iteration (VI) that can be used. This plethora of algorithms may however pose some challenges, particularly when it comes to choose from one of them to solve a given RL task. In the case of choosing from one of the DQN variants, the choice of algorithm may be even more challenging as each of these algorithms enables substantial performance improvements on its own, making it difficult to decide on which one to use. Furthermore, a planner may also be used to solve some RL tasks as opposed to using a RL algorithm.

The objective of this paper is therefore to perform a performance benchmark of some of these RL algorithms for the discrete control case, both from the deep as well as the non-deep learning fields, with a special focus on DQN and its variants. While this study does not intend to be a comprehensive benchmark of all the RL algorithms proposed so far, it intends nevertheless to provide a range of comparison, as wide as possible, between different methods, approaches and paradigms. Neural Fitted Q-Iteration (NFQ), Vanilla Policy Gradient (PGO) and a planner were also included on the benchmark assessment in order to achieve this. A 2D maze and two OpenAI Gym environments, namely a custom-built Foraging/Tagging environment and the CartPole environment were used to carry out the tests. The remainder of this paper is structured as follows: Sect. 2 presents the related work. Section 3 presents the testing framework used to perform the benchmark. The results obtained are presented and discussed in Sect. 4. Finally, Sect. 5 presents the conclusions and future work.

## 2 Related Work

In [10] the authors propose the Multiple Deep Q-Network (MDQN) algorithm, a generalization of the Double Q-Learning (DQL) [11] algorithm that allows the use of any number of decoupled functions using the Bellman equations. The authors then benchmarked this algorithm with DQN and DDQN. The PyGame Learning Environment (PLE) was used to perform the tests. This benchmark is however limited to only DQN and two of its variants. In [12] the authors benchmark several RL algorithms over four categories of continuous control tasks, specifically: basic, locomotion, partially observable and hierarchical tasks. All these tasks were implemented using physics simulators, such as MuJoCo. This benchmark does not, however, consider the discrete case. Similarly, in [13] the authors present a survey of Deep RL (DRL). This survey starts by introducing the RL formalism as well as its challenges. Next the authors present the different classes of RL algorithms. Special focus is given to value function and policy search based DRL methods. To conclude, the study presents ongoing research and discusses open challenges. The several RL algorithms presented are however not benchmarked. Finally, the work proposed in [14] investigates the challenges posed by reproducibility, proper experimental techniques and reporting

procedures. For this purpose, several model-free policy gradient methods for continuous control were used. The objective of this benchmark was not to compare the algorithms performance-wise but to assess the influence of extrinsic factors (e.g. hyperparameters) and intrinsic factors (e.g. environment properties) on reproducibility.

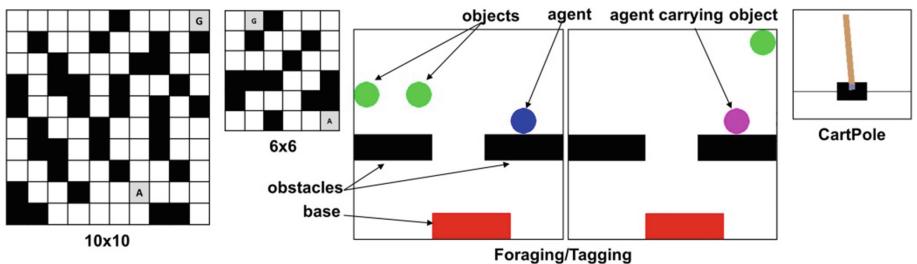
### 3 Testing Framework

The full list of algorithms tested is presented in the list below.

- Value Iteration;
- Q-Learning;
- Neural Fitted Q-Iteration;
- Deep Q-Network;
- Double Deep Q-Network;
- Dueling Network Architecture;
- Prioritized Experience Replay;
- Double DNA (DDNA), a combination of DDQN with DNA;
- (PERDDNA), a combination of PER with DDQN and DNA;
- Vanilla Policy Gradient;
- POPF [15], a mature open-source forward-chaining temporal planner (PL).

The learning environments used to perform the benchmark, see Fig. 1, include a 2D maze (left) with 2 configurations (6 × 6 and 10 × 10) and two OpenAI Gym learning environments, namely, a custom-built Foraging/Tagging environment (middle) and the CartPole environment (right). All the environments feature a discrete action space.

The Maze learning environment consists of a 2D maze. The agent starts from a predefined starting point (*A*) and must find its way to the goal position (*G*) through the maze in the least number of steps possible. The agent can perform 4 different actions (move left, right, up or down) to achieve its goal.



**Fig. 1.** All environments benchmarked. The 6 × 6 and 10 × 10 Maze environment configurations (left), the Foraging/Tagging environment (middle) and the CartPole environment (right).

In the Foraging environment the agent must collect the objects placed randomly in the world and take then back to the base in the least number of steps possible. The task is considered complete when all the objects have been retrieved to the base. Two different versions of this environment were used: in version 4a, the agent can perform 4 actions (move left, right, up and down) and the grab/release actions are performed automatically. In version 6a the grab and release actions were added to the set of possible actions and the agent must explicitly grab and release the object. In the Tagging environment (reuses the Foraging environment) the goal is to tag the positions of the objects. The agent should tag all the objects only once in the least number of steps possible. The 5 possible actions are (move left, right, up, down and tag). All environments feature an  $8 \times 8$  grid like environment (i.e. grid world) with 1 agent and 2 objects placed randomly every time the game is reset.

In the CartPole environment a pole is attached by an un-actuated joint to a cart which moves along a track. The pendulum starts upright and the goal is to prevent it from falling over. The system is controlled by moving the cart to the left or to the right (2 actions). The episode ends when the pole is more than  $15^\circ$  from vertical or the cart moves more than 2.4 units from the center. The version used for this environment (CartPole-v1) limits the number of steps per episode to 500.

The performance metrics used, see Table 1, allow the assessment of the algorithms throughout the training phase (marked with +) as well as the assessment of the quality of the final controllers produced (marked with /). Metrics marked with \* are specific to the CartPole environment (MPR stands for the maximum possible reward). The definition of successful policy varies for each of the environments considered and the corresponding successful policy checks  $\Gamma$  are presented in Table 2.

**Table 1.** Performance metrics.

Metric	Description
+NSP	Number of successful policies obtained during training
+TFSP	Training time elapsed until the first successful policy was obtained
+FSPE	Episode where the first successful policy was obtained
+BPR	Best return obtained by the policies
+BPE	Episode where the best return was obtained
'NSTG	Number of games successfully completed by the final controller
**NSP75	Number of policies that obtained at least 75% of the MPR
**NSP100	Number of policies that obtained 100% of the MPR
/*NSTP50	Number of games where the final controller obtained at least 50% of the MPR
/*NSTP75	Number of games where the final controller obtained at least 75% of the MPR
/*NSTP100	Number of games where the final controller obtained 100% of the MPR
+RT	Total training time (or time needed to solve the task in the planner's case)

**Table 2.** Successful policy checks  $\Gamma$  used for each of the environments ( $M$  represents the number of game setups and  $P$  is the maximum number of steps allowed per game).

Env.	Successful policy check $\Gamma$
Maze	Policy played all test games successfully ( $M = 1$ and $P = 100$ )
Foraging	Policy played at least 50% of the test games successfully ( $M = 10$ and $P = 250$ )
Tagging	Policy played at least 50% of the test games successfully ( $M = 10$ and $P = 250$ )
CartPole	Policy obtained on average at least 50% (250) of the MPR ( $M = 10$ and $P = 500$ )

The pseudocode of the testing framework used to assess the algorithms is presented in Listing 1 (the values used to parameterize the testing framework are presented in Table 3). Each algorithm is assessed over a given number of trials  $T$ , where a trial consists of several episodes  $E$ . In this setup an episode is composed of  $\gamma$  game setups  $G$ . A game setup  $G$  is the result of resetting a game to a random new initial state (except for the Maze and the CartPole environments which always start on the same state). Each controller  $C$  to be trained is initialized at the beginning of each trial and may be trained during the episode or at the end of it. After the completion of each episode, the controller  $C_t$  trained so far is assessed. This assessment is done by making the controller play a given number  $M$  of game setups. It is also at this point that the policy is verified as being successful or not using the successful policy check  $\Gamma$ .

---

### Pseudocode

---

```

For  $t$  in  $T$ 
    Initialize controller  $C_t$ 
    For  $e$  in  $E$ 
        For  $y$  in  $\gamma$ 
            Initialize game setup  $G_y$ 
            While not terminal or  $p \leq P$  # play until completion or for a maximum of  $P$  steps
                terminal = play  $G_y$       # perform one step
                Train  $C_t$       # if applicable
            End While
            Train  $C_t$       # if applicable«
        End For
        Test  $C_t$  using  $M$  game setups      # each game is played until completion or for  $N$  steps
        Verify successful policy check  $\Gamma$  for  $C_t$ 
        Collect training statistics for  $e$ 
    End For
    Test  $C_t$  using  $M'$  game setups      # each game is played until completion or for  $N'$  steps
    Collect training statistics for  $t$ 
End For
Average all the collected statistics over the  $T$  trials      # assessment complete

```

---

**Listing 1.** Pseudocode of the testing framework used to assess the algorithms.

**Table 3.** Values used to parameterize the testing framework.

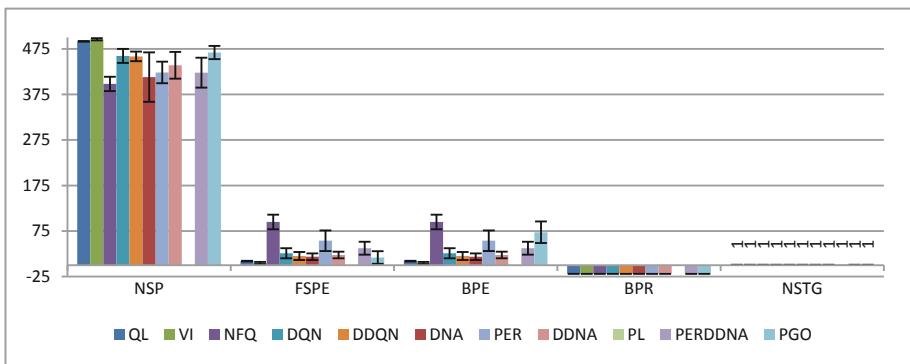
Environment	T	E	$\gamma$	P	M	N	$M'$	$N'$
Maze	10	500	1	1000 ( $6 \times 6$ ), 2000 ( $10 \times 10$ )	1	100	1	100
Foraging	10	1000	10		2000	10	250	250
Tagging	10	1000	10		2000	10	250	250
CartPole	10	500	10		500	10	500	250

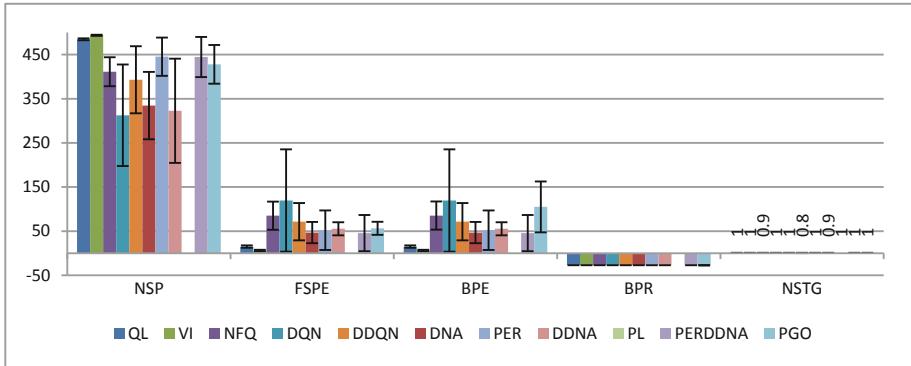
At the end of each trial the quality of the final trained controller is assessed by making the controller play a given number  $M'$  of game setups. At the end of the assessment process all the statistics collected during the training and testing steps are averaged over the number of trials performed T in order to obtain the final statistics presented in this work (the whiskers on the bar graphs represent the standard deviation).

## 4 Results

### 4.1 Maze Learning Environment

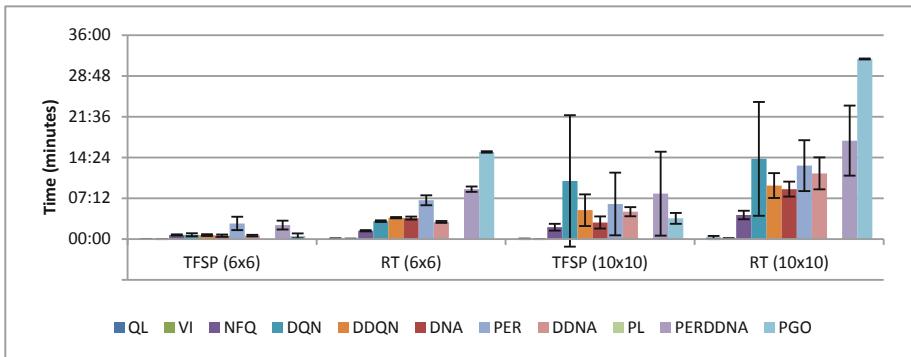
Overall, VI obtained the best NSP (495.9), TFSP (00:00) and FSPE (4.8) results. Amongst the DL methods, PGO obtained the best NSP (467), TFSP (00:33) and FSPE (16.8) results. DQN (NSP = 459.4), DNA (FSPE = 18.7) and DDNA (TFSP = 00:36) were the best amongst the DQN variants for these indexes. PGO was the method with the worst RT (15:23). PER (02:45) and PERDDNA (02:28) performed the worst for the TFSP index. All the algorithms were able to train policies that obtained the maximum possible returns for this maze configuration (BPR = -19) as well as final controllers that were able to successfully solve the maze (NSTG = 1). See Figs. 2 and 4.

**Fig. 2.** Results for the  $6 \times 6$  Maze environment configuration (performance metrics).



**Fig. 3.** Results for the  $10 \times 10$  Maze environment configuration (performance metrics).

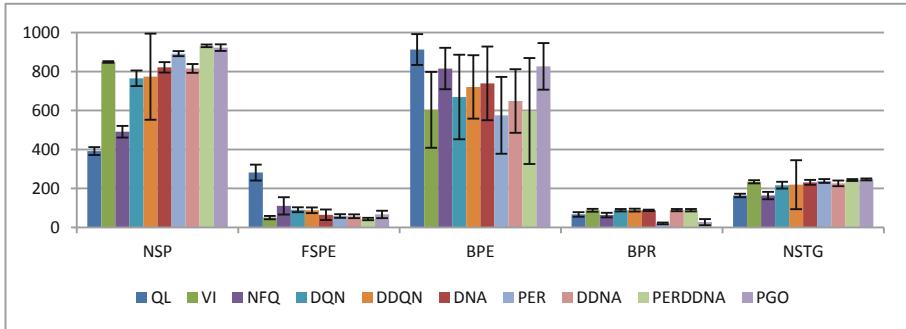
Concerning the  $10 \times 10$  configuration, see Figs. 3 and 4, VI performed best overall with  $\text{NSP} = 494$ ,  $\text{TFSP} = 00:00$  and  $\text{FSPE} = 6.3$ . PER (NSP = 445.4), DNA (TFSP = 02:55) and PERDDNA (FSPE = 45.7) were the best amongst the DL variants. All the algorithms were able to train policies that obtained the maximum possible returns (BPR = -27), however NFQ, DNA and DDNA (NSTG of 0.9, 0.8 and 0.9 respectively) did not always produce a final controller able to successfully solve the maze. PGO was again the method with the worst RT (31:49). DQN (10:16) performed the worst for the TFSP index. The planner solved both mazes easily (RT under 1 s).



**Fig. 4.** Results for both Maze environment configurations (time metrics).

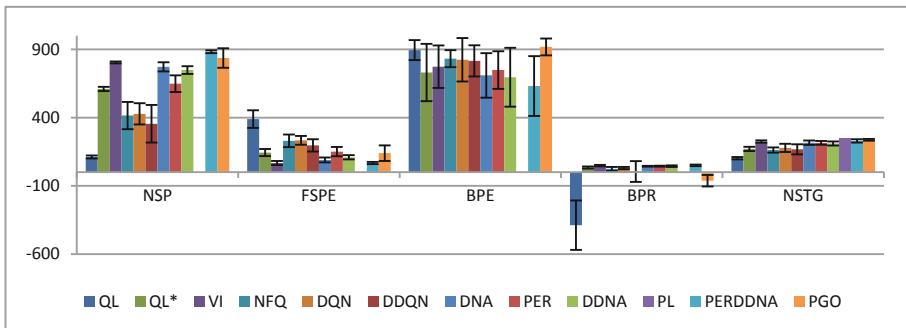
## 4.2 Foraging Learning Environment

For version 4a of the environment, PERDDNA (NSP = 932 and FSPE = 43.1), VI (TFSP = 01:45) and DDQN (BPR = 88.9) obtained the best results for the respective indexes. PER (BPR = 20.4) and PGO (BPR = 27.3) produced the worst policies in terms of quality. Regarding the final controllers obtained, PGO performed the best (NSTG = 246.1). Concerning the RT index and contrary to the Maze environment, PGO (41:59) performed the best amongst the DL variants. See Figs. 5 and 7.

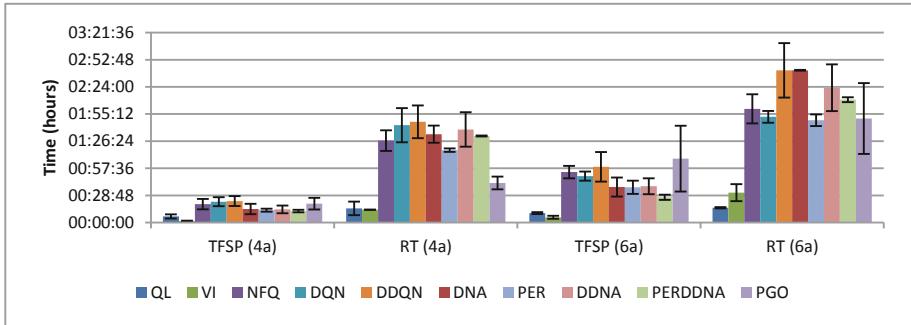


**Fig. 5.** Results for the 4a Foraging environment version (performance metrics).

Regarding version 6a, see Figs. 6 and 7, PGO (NSTG = 238), VI (TFSP = 05:35 and FSPE = 67.1) and PERDDNA (NSP = 882.6 and BPR = 50.1) obtained the best results. QL obtained the best result regarding the RT index (15:42) but performed clearly the worst on both environments and would benefit from more interaction with the environment as shown by the results in QL\* ( $\gamma = 30$ ). Again, PGO performed better than most DQN variants (RT = 01:50:24) being only surpassed by PER (RT = 01:48:37). Amongst the DQN variants PERDDNA performed better overall (NSP = 882.6, TFSP = 26:53, FSPE = 67.6, BPE = 631.2, BPR = 50.1 and NSTG = 229.5). The planner solved all test games ( $M' = 250$ ) in 11 s.



**Fig. 6.** Results for the 6a Foraging environment version (performance metrics).

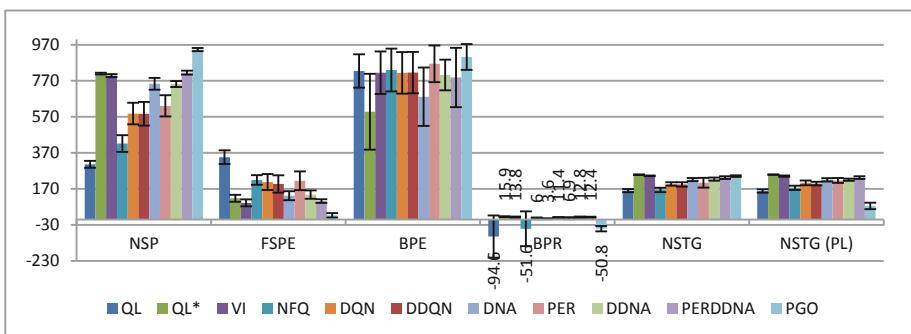


**Fig. 7.** Results for the 4a and 6a Foraging environment versions (time metrics).

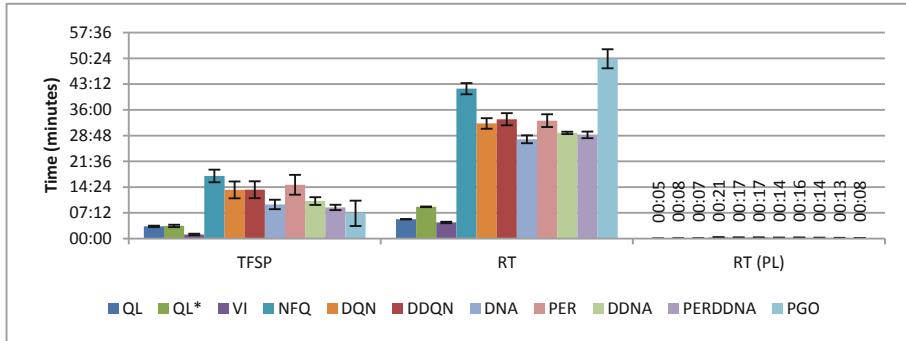
### 4.3 Tagging Learning Environment

This environment allowed mixing planning and learning in order to solve the Foraging task presented earlier. In this scenario, the controller trained to solve the Tagging task was used to find the positions of the objects (i.e. tag the objects). These positions were used to automatically generate the model for the planner, which was then used to solve the Foraging task (results presented by the NSTG (PL) and RT (PL) indexes).

Overall, see Figs. 8 and 9, PGO performed the best during the training and testing phases (but clearly the worst regarding RT = 50:17). However, when using the final controller together with the planner to solve the Foraging task, PGO performed the worst (NSTG (PL) = 75.4). Regarding the DQN variants, PERDDNA obtained the best NSP (815.3) and TFSP (08:41) results, PER was the worst regarding the TFSP index (15:02), DDNA obtained the best BPR results (12.8) whereas DDQN obtained the worst (3.6). Concerning the results for the mixed planning/learning Foraging task (NSTG (PL)), VI performed the best (240.8), followed by PERDDNA (232.4) and DDNA (223). DDQN performed the worst amongst the DQN variants (199). Overall, QL performed the worst and as shown by the QL\* results ( $\gamma = 30$ ), would clearly benefit from more interaction with the learning environment.



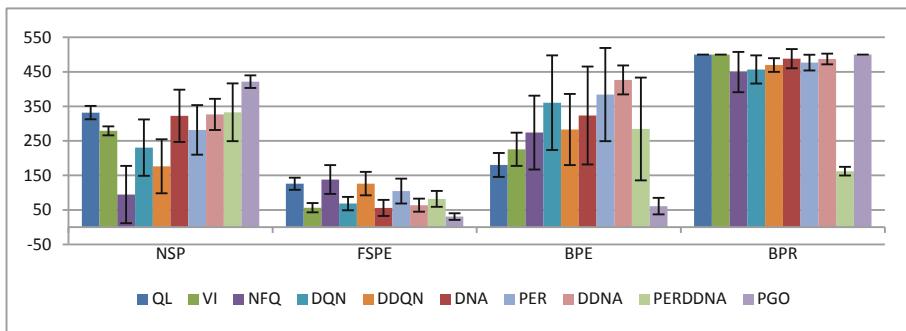
**Fig. 8.** Results for the Tagging learning environment (performance metrics).



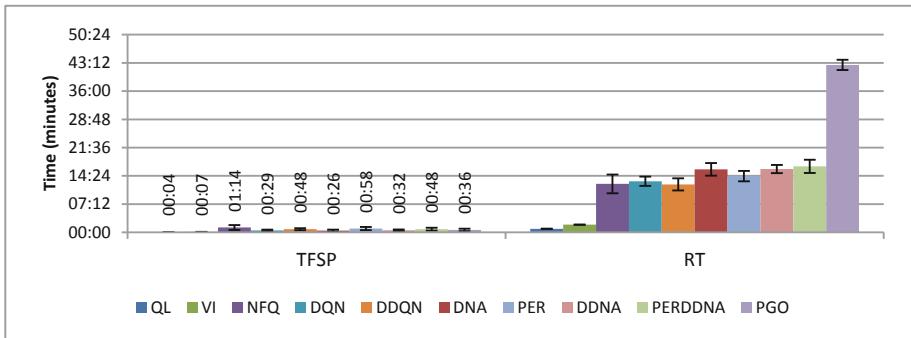
**Fig. 9.** Results for the Tagging learning environment (time metrics).

#### 4.4 CartPole Learning Environment

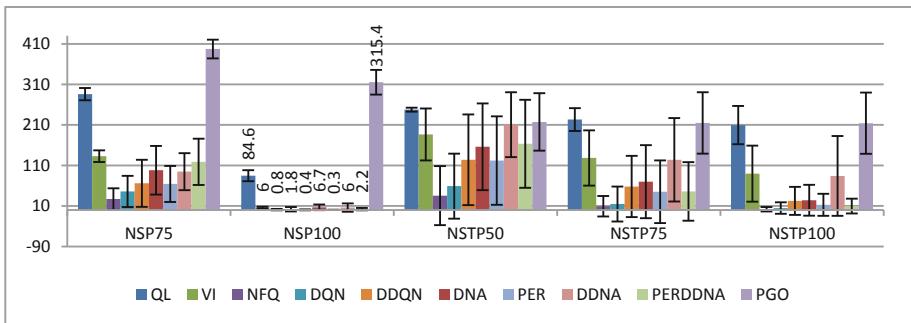
Overall, PGO (better during training, although the worst in terms of RT (42:41)) and QL (better during testing) obtained the best results, see Figs. 10, 11 and 12. Regarding the DQN variants during the training phase, DDQN obtained the worst NSP index (176.2), PERDDNA was the best in this regard (NSP = 332.8), PER needed the most interaction time to obtain its first successful policy (TFSP = 58 s), while PERDDNA obtained the lowest BPR (162.2). Concerning the test results overall, DQN was the worst of the DQN variants. DDNA (210.6, 124 and 84 for the NSTP50, NSTP75 and NSTP100 indexes respectively) was the best in this regard. NFQ obtained the worst training and testing results overall. The NSTP50 index for this environment is analogous to the NSTG index.



**Fig. 10.** Results for the CartPole learning environment (performance metrics).



**Fig. 11.** Results for the CartPole learning environment (time metrics).



**Fig. 12.** Training and testing results for the CartPole learning environment.

## 4.5 Discussion

Overall, VI and QL performed quite well in all the environments (QL would benefit from more interaction with the environment in some of the tasks) and seem to be good choices for this kind of tasks (small state spaces). Concerning the DQN variants, overall PERDDNA, PER, DNA and DDNA obtained better results. DDQN and DQN seem more difficult to assess comparatively as they obtained very similar results in some of the environments. In general, NFQ performed worse than the DQN variants. PGO performed quite well (particularly in the CartPole environment) and outperformed the DQN variants in some of the tasks. However, it would benefit from more interaction with the environments as shown by the poor BPR results obtained, particularly in the Foraging and the Tagging environments (i.e. the controller still grabs, releases or tags objects improperly). The planner was able to solve all the tasks successfully in the least amount of time and is also a good choice (for small action spaces). Table 4 presents the best overall as well as the best DL algorithms (between parenthesis) for each of the environments. Consult Fig. 3 for a list of the algorithms that were not able to train a controller that could solve the  $10 \times 10$  Maze (All\*).

**Table 4.** List of the most performant algorithms per environment.

Env.	NSP	TFSP	BPR	NSTG	RT
Maze $6 \times 6$	VI (PGO)	VI (PGO)	All	All	VI (NFQ)
Maze10 $\times 10$	VI (PER)	VI (PERDDNA)	All	All*	VI (NFQ)
Foraging 4a	PGO	VI (PERDDNA)	DDQN	PGO	VI (PGO)
Foraging 6a	PERDDNA	VI (PERDDNA)	VI (PERDDNA)	PGO	QL (PGO)
Tagging	PGO	VI (PGO)	VI (DDNA)	VI (PGO)	VI (DNA)
CartPole	PGO	QL (DNA)	QL, VI (PGO)	QL (PGO)	QL (DDQN)

## 5 Conclusion

This work presented a benchmark on the performance of various RL algorithms for discrete domains. Several non-deep as well as deep RL algorithms and a planner were included in order to provide a wider range of comparison between these different approaches and paradigms. For the purposes of these tests, a 2D maze and two OpenAI Gym environments were used. As future work, this benchmark can be extended to include more challenging learning environments as well as more RL algorithms and planners in order to better refine the results and the conclusions obtained.

**Acknowledgements.** This work was supported by National Funds through the FCT - Foundation for Science and Technology in the context of the project UID/CEC/00127/2019 and also by FCT PhD scholarship SFRH/BD/145723/2019.

## References

1. Mnih, V., et al.: Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015)
2. Mnih, V., et al.: Playing atari with deep reinforcement learning. *CoRR* abs/1312.5602 (2013)
3. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 2094–2100. AAAI Press, Phoenix (2016)
4. Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., de Freitas, N.: Dueling network architectures for deep reinforcement learning. In: 33rd International Conference on Machine Learning (ICML 2016), pp. 1995–2003. JMLR, New York (2016)
5. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. In: International Conference on Learning Representations, San Juan, Puerto Rico (2016)
6. Bellemare, M.G., Dabney, W., Munos, R.: A distributional perspective on reinforcement learning. In: 34th International Conference on Machine Learning, pp. 449–458. JMLR, Sydney (2017)
7. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: 12th International Conference on Neural Information Processing Systems, pp. 1057–1063. MIT Press, Cambridge (1999)
8. Schulman, J., Levine, S., Moritz, P., Jordan, M., Abbeel, P.: Trust region policy optimization. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, pp. 1889–1897. JMLR, Lille (2015)

9. Wu, Y., Mansimov, E., Liao, S., Grosse, R., Ba, J.: Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5280–5289. Curran Associates, California, USA (2017)
10. Urtans, E., Nikitenko, A.: Survey of deep Q-network variants in PyGame learning environment. In: 2nd International Conference on Deep Learning Technologies, pp. 27–36. ACM, Chongqing (2018)
11. Van Hasselt, H.: Double Q-learning. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems, pp. 2613–2621. Curran Associates, Vancouver (2010)
12. Duan, Y., Chen, X., Houthooft, R., Schulman, J., Abbeel, P.: Benchmarking deep reinforcement learning for continuous control. In: 33rd International Conference on Machine Learning, pp. 1329–1338. JMLR, New York (2016)
13. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: a brief survey. IEEE Sig. Process. Mag. **34**(6), 26–38 (2017)
14. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D.: Deep reinforcement learning that matters. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 3207–3214. AAAI Press, Louisiana (2018)
15. POPF Homepage. <https://nms.kcl.ac.uk/planning/software/popf.html>. Accessed 13 June 2019

# **Mobile Robots for Industrial Environments**



# Versatile and Massive Experimentation of Robot Swarms in Industrial Scenarios

Vivian Cremer Kalempa<sup>1,2(✉)</sup>, Marco Antonio Simoes Teixeira<sup>1</sup>,  
and André Schneider de Oliveira<sup>1</sup>

<sup>1</sup> Universidade Tecnológica Federal do Paraná (UTFPR),  
Av. Sete de Setembro, Curitiba, PR 80230-901, Brazil  
[marcoteixeira@alunos.utfpr.edu.br](mailto:marcoteixeira@alunos.utfpr.edu.br), [andreoliveira@utfpr.edu.br](mailto:andreoliveira@utfpr.edu.br)

<sup>2</sup> Universidade do Estado de Santa Catarina (UDESC),  
Luiz Fernando Hastreiter St., 180, São Bento do Sul, SC, Brazil  
[vivian.kalempa@udesc.br](mailto:vivian.kalempa@udesc.br)

**Abstract.** This paper aims to present an automated experimentation platform designed to work with multiple mobile robots into industrial scenarios using Robot Operating System (ROS). Robot swarm is a system with complexity proportional of group size, expensive and with arduous setup. Virtual environments can be used to expedite the testing, but also are very difficult due to a hard-work to configure each robot. The proposed platform is a tool set to easily configure the experimentation environment, aiming the swarm tasks, with most popular perception systems, absolute or relative localization reference, and position controllers. The user specifies in tool only the amount of robots required, sensor and function, without having to configure each robot individually for the simulation. This paper presents examples to multi-robot setup in industrial scenarios, through a simple parameterization script. The memory consumption is demystified to given to allow the estimation of computational resources necessary to perform the practical experimentation.

**Keywords:** Robot · Swarm · Experimentation · ROS

## 1 Introduction

Robot swarm is a concept bioinspired by organizations of natural swarms such as ants, bees, birds, fish, wolf, and even humans [15], aiming to reproduce collective behaviors. For example, insects provide one of the best-known examples of self-organized biological behavior, through local and limited communication, they can perform impressive behavioral features: maintaining the health of the colony, caring for the young, responding to the invasion, and so on [12].

Another example is the ants, which show social behaviors. These insects live in family units, and their life cycle is organized around this unit known as ant colony [5]. The ant colony as a whole is capable of performing complex tasks such as feeding and breeding large nests, although a single ant that shows no specific intelligence.

Following this analogy, [3] verifies that a set of robots is not just a group. It has some unique characteristics, which found in insect swarms, for example, decentralized control, lack of synchronization, simple and (almost) identical members. Reinforcing the definition given by [3], [13] defines a swarm of robots as an approach for coordinating a large number of relatively simple, autonomous, non-centrally controlled physical robots capable of establishing social communication and operate by some sense of biological inspiration.

Robot swarm is commonly classified in six great research areas [6], as swarm size, communication range, communication topology, communication bandwidth, swarm reconfiguration, and swarm units processing capacity. These areas can be rigorously subdivided into main research topics, as discussed in [7, 15], resulting in: *biological inspiration* [16]; *communication* [2]; *control approach* [5]; *mapping and location* [9]; *transport and manipulation of objects* [1]; *reconfigurable robotics* [8]; *motion coordination* [11]; *learning* [10]; *task allocation* [4].

Researches about robot swarm aim the development of cooperative or coordinate behaviors to real robotic swarm (in low scale) or virtual swarms (in high scale). Thus, the main characteristics identified in recent works of robot swarm indicated some tendencies, which will be used as a setup requirement of virtual experimentation platform, as

- robots must support wireless communication to exchange information with other robots or maybe with a central node;
- robots should also be programmable in parallel through a wireless communication channel since control algorithms are commonly the same for all robots;
- robots must be able to interact physically with each other and the environment;
- robots should have a long autonomy (concerning battery capacity), because the swarm may need to operate for a long period to execute a specific task;
- robots should be as cheap as possible;
- the interference between the multiple sensor systems and the effect of environmental factors should be minimal;
- robots should be small enough not to increase a lot the environment size, but large enough not to limit the robot's expandability or increase the cost of swarm robots due to component miniaturization.

The virtual experimentation with multiple robots can be an exhaustive task due to the setup need to be made individually to each robot. In a large-scale swarm (a hundred or thousand of robots), the virtual configuration can be extremely wearing. The computational effort also needs to be considered because the introduction of each new robot requires new computational resources. The purpose of this paper is to present a new experimentation platform, that automatically configures a swarm robot simulator, aiming the simulation of large-scale, with homogeneous and heterogeneous robots, different perceptions systems, to the main tasks of a robot swarm.

This work is organized as follows: Sect. 2 presents the open-source experimentation platform focused on robot swarms to facilitate the development of applications in this area. Section 3 shows an example application using the Swarm Stage

package. Section 4 presents an analysis of RAM consumption of some scenarios and Sect. 5 presents conclusions and future work.

## 2 Experimentation Platform - *Swarm Stage*

The proposed experimentation platform is developed as an automated configuration script that executes a complete setup of Stage Simulator [14], to achieve the versatile and massive simulation of robot swarms, due to fast reconfiguration and low computational load, allowing the high scale of the swarm compared to other simulators like VREP<sup>1</sup>, Gazebo<sup>2</sup> and ARGoS<sup>3</sup>.

The required structure is a simple parametrization script, where is a specified number of robots, perception systems, controller type (position or velocity), reference kind (odometry - local - or localization - global), scenario and initial formation. The main focus is to evaluate the introducing of multiple mobile robots into industrial tasks, in accord to Industry 4.0 concepts.

This experimentation platform is based in ROS been full open-source, called as package *swarm\_stage\_ros*<sup>4</sup>. The codes are available in GitHub<sup>5</sup> and the videos developed with the experiments can be seen on YouTube<sup>6</sup>.

### 2.1 Configuration

The configuration of the desired simulation scenario is done through the *swarm.yaml* parameterization file, as presented in Listing 1.1.

**Listing 1.1.** Example of a *swarm.yaml* parameter file

```

1 swarm:
2   formation: 'square' #no, wedge, square or diamond
3   robots: 0 #used to "no" formation
4   random_colors: 'yes' #yes or no
5   side: 3 #number of robots per formation side
6   completed: 'no' #when used formation
7   leader: 'no' #yes or no
8   leader_sonar_sensor: 3 #0, 1 or 3
9   leader_laser_sensor: 'no' #yes or no
10  other_robots_sonar_sensor: 0 #0, 1 or 3
11  other_robots_laser_sensor: 'no' #yes or no
12  scenario: 'stadium' #stadium, forest or hospital
13  position_controller: 'yes' #yes or no
14  reference: 'relative' #relative(dom) or global
15  publish_tf: 'yes' #yes or no, when reference = 'relative'
16  run: 'yes'
```

The *formation* parameter allows one of four possible values to be chosen: *no*, *wedge*, *square* or *diamond*. If *no* is chosen, it is understood that no specific formation will be used. However, the desired number of robots in the *robots* parameter will be required.

<sup>1</sup> <http://www.coppeliarobotics.com/>.

<sup>2</sup> <http://gazebosim.org/>.

<sup>3</sup> <https://www.argos-sim.info/>.

<sup>4</sup> [http://wiki.ros.org/swarm\\_stage\\_ros](http://wiki.ros.org/swarm_stage_ros).

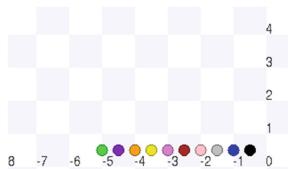
<sup>5</sup> [https://github.com/VivianCremerKalempa/swarm\\_stage\\_ros](https://github.com/VivianCremerKalempa/swarm_stage_ros).

<sup>6</sup> <https://www.youtube.com/channel/UCztalFc6fapGIhQN2Q0jCuw>.

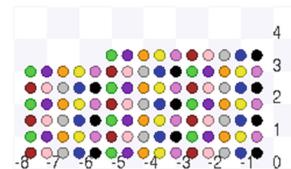
Consider the following parameterization in Listing 1.2. The generated scenario with the number of robots equal 10 is presented in Fig. 1. If the numbers of robots are 100, the generated scenario is presented in Fig. 2.

**Listing 1.2.** Example of *no* formation and 10 robots

```
1 swarm:
2   formation: 'no' #no, wedge, square or diamond
3   robots: 10 #used to "no" formation
```

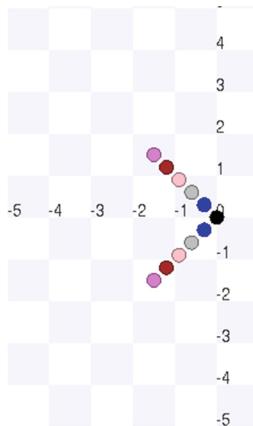


**Fig. 1.** Generated scenario with the parameter *formation* = ‘no’ and *robots* = 10.

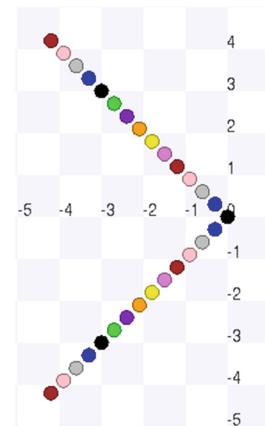


**Fig. 2.** Generated scenario with the parameter *formation* = ‘no’ and *robots* = 100.

Now consider *formation* = ‘wedge’. In this case, the value for the *side* parameter must be entered. The *side* parameter is the number of robots per side. Figures 3 and 4 present two *wedge* formations, with *side* = 6 and *side* = 15.

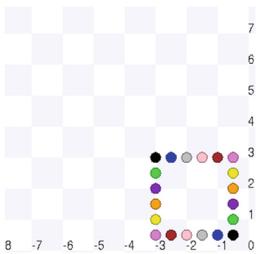


**Fig. 3.** Generated scenario with the parameter *formation* = ‘wedge’ and *side* = 6.

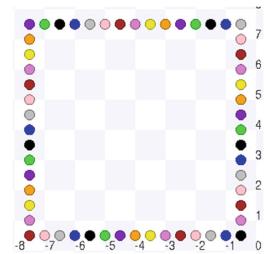


**Fig. 4.** Generated scenario with the parameter *formation* = ‘wedge’ and *side* = 15.

For the *square* formation, the *side* parameter also needs to be informed. Figures 5 and 6 present two situations, where the *side* is 6 and 15, respectively.

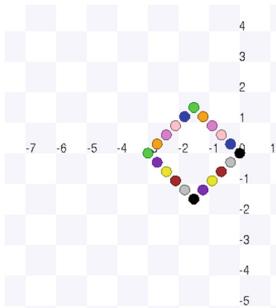


**Fig. 5.** Generated scenario with the parameter formation = ‘square’ and side = 6.

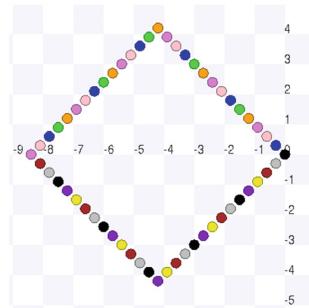


**Fig. 6.** Generated scenario with the parameter formation = ‘square’ and side = 15.

Lastly, there is the *diamond* formation. For this formation it is also necessary to inform the number of robots per side. Figures 7 and 8 present the *diamond* formation with *side* = 6 and *side* = 15.



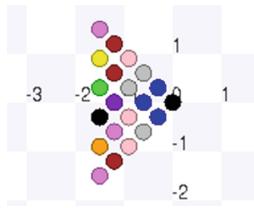
**Fig. 7.** Generated scenario with the parameter formation = ‘diamond’ and side = 6.



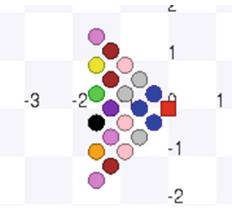
**Fig. 8.** Generated scenario with the parameter formation = ‘diamond’ and side = 15.

Like the *side* parameter, the *completed* parameter is used when the *formation* parameter is one of the options: *wedge*, *square* and *diamond*. Figure 9 exemplify the *wedge* formation with parameter *completed* = ‘yes’. Figure 3 exemplify *wedge* formation with the parameter *completed* = ‘no’. Figure 11 shows an example when using *side* = 45 and *completed* = ‘yes’. In this case the total of 1035 robots is obtained.

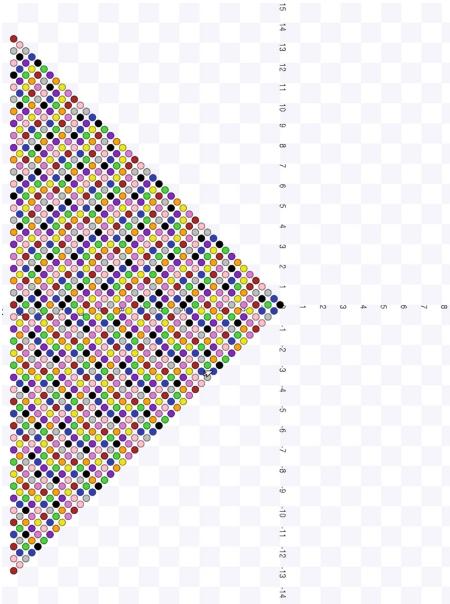
In addition, when the *formation* parameter is one of the options: *wedge*, *square* and *diamond*, a leader may be necessary. Figure 10 present an example for *wedge* formation, with the *leader* = ‘yes’ parameter. The leader has a format different from the other robots and always has the color red. The leader can be configured with 1 sonar sensor or 3 sonar sensors. The *leader\_sonar\_sensor* parameter allows the value 0 to be entered, when no sonar sensor will be used,



**Fig. 9.** Generated scenario with the parameter completed = ‘yes’ and formation = ‘wedge’.



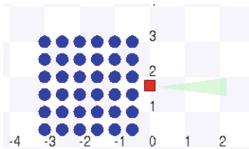
**Fig. 10.** Generated scenario with the parameter leader = ‘yes’ and formation = ‘wedge’.



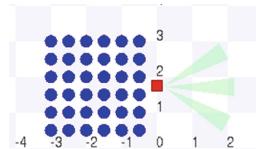
**Fig. 11.** Generated scenario with the parameter formation = ‘wedge’, side = 45, completed = ‘yes’ (1035 robots).

1 when 1 sonar sensor is used, and 3 when 3 sonar sensors are used for the leader robot. Figures 12 and 13 present scenarios where the leader has 1 sonar sensor and 3 sonar sensors, respectively.

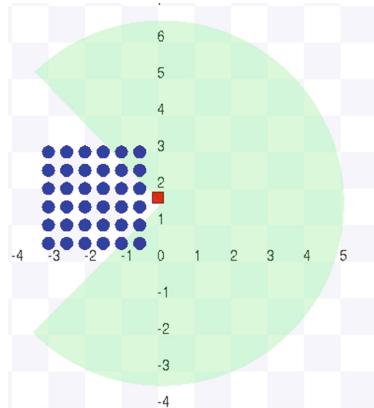
It is also possible to choose to use a laser sensor to the leader. In this case, the *leader\_laser\_sensor* parameter only allows the value *no* to indicate that a laser sensor will not be used or *yes* to indicate that a laser sensor will be used. It is not possible to use a sonar and laser sensor at the same time, and it is only necessary to choose one of the two types of sensors. Figure 14 presents a scenario where the leader has a laser sensor.



**Fig. 12.** Generated scenario with the parameter leader = ‘yes’ and leader.sonar\_sensor = 1.



**Fig. 13.** Generated scenario with the parameter leader = ‘yes’ and leader.sonar\_sensor = 3.

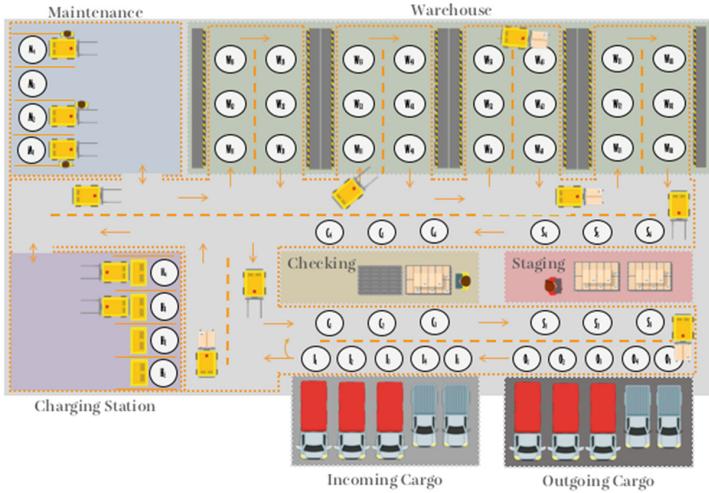


**Fig. 14.** Generated scenario with the parameter leader = ‘yes’ and leader.laser\_sensor = 1.

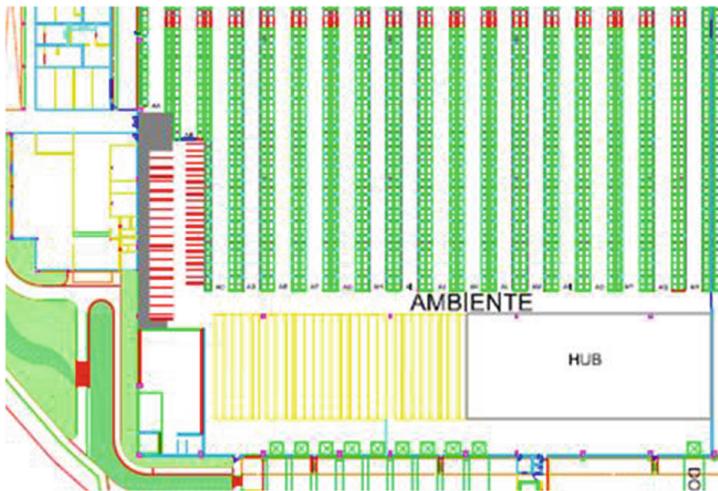
### 3 Application Example of *Swarm Stage*

The purpose of this demonstration is to present how the *swarm\_stage\_ros* package makes working with swarms of robots simpler and focuses on the application and not on the required configurations. More complex examples can be realized from that basis, such as the factory scenario shown in Fig. 15 which has its inspiration in Fig. 16.

In this scenario, when some cargo arrives in *Incoming cargo*, it is made available in the *Checking* sector to be inspected and then stored. If the merchandise goes through the inspection, it can be stored in *Warehouse*. *Warehouse* has eight racks and four corridors between them, where the robots can move in one direction. To fulfill a customer’s request, the order is taken from the *Warehouse* rack and available in *Staging*. Upon withdrawal of the *Staging*, the cargo is loaded into the *Outgoing cargo*. If in any of these situations the robot should go to *Charging Station* if its battery is not enough or it is taken to the *Maintenance* sector for repairs when necessary.

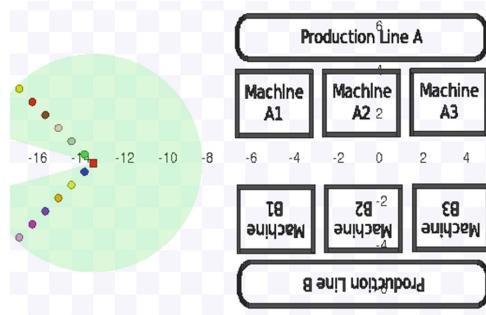


**Fig. 15.** Virtual experimentation scenario for the factory experiment.



**Fig. 16.** Diagram - real warehouse.

Another example of a factory with two production lines, as shown in Fig. 17. The approach evaluation is performed with a group of mobile robots to execute the cleaning services in an industrial plant with specific group requirements, aiming to maintain the agents' formation to expand the cleaning area. The group is composed of thirteen vacuum cleaner robots and a leader equipped with a 2D laser scanner, and followers do not have any sensors.



**Fig. 17.** Virtual experimentation scenario with  $\wedge$  formation.

### 3.1 Formation

This example application addresses the position control of 20 robots forming the square formation. For this, the 20 robots need to form the design of a cross. The *swarm.yaml* file for this scenario is presented in Listing 1.3.

**Listing 1.3.** Parameterization for the position control environment

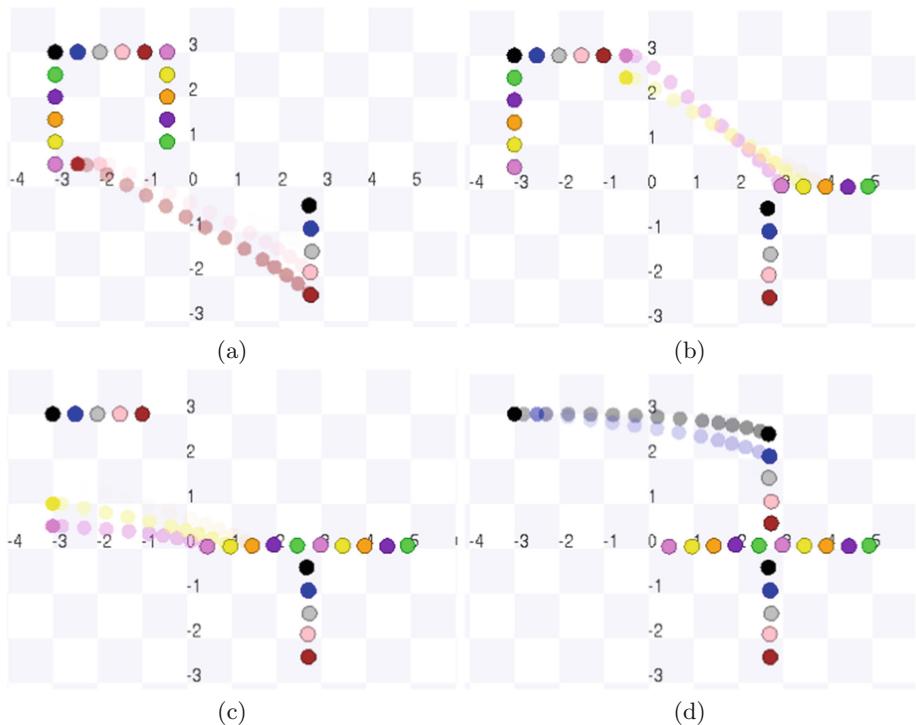
```

1 swarm:
2   formation: 'square' #no, wedge, square or diamond
3   robots: 0 #used to "no" formation
4   random_colors: 'yes' #yes or no
5   side: 6 #number of robots per side
6   completed: 'no' #when used formation
7   leader: 'no' #yes or no
8   leader_sonar_sensor: 0 #0, 1 or 3
9   leader_laser_sensor: 'no' #yes or no
10  other_robots_sonar_sensor: 0 #0, 1 or 3, use this field for formation = 'no'
11  other_robots_laser_sensor: 'no' #yes or no
12  scenario: 'stadium' #stadium, forest or hospital
13  position_controller: 'yes' #yes or no
14  reference: 'relative' #relative(odom) or global
15  publish_tf: 'yes' #yes or no, use this field when reference = "relative"
16
17 run: yes

```

As the *position\_controller* parameter is *yes*, at the end of processing the *controller.launch* file is generated with a node calling the *posctrl* program for each robot and a node calling the *controller* program. This program calculate the position of each robot in order to form the cross, so that there are no collisions between the robots. Thus, each side of the square formation forms an edge of the resulting cross. It is also important to note that the position of each robot is published in a topic with the name of the robot, for example: /robot\_0.

On the other hand, the *posctrl* program subscribes to the topic with the name of the robot, for example, /robot\_0 and publishes in its *cmd\_vel* topic. Upon receiving a message, the *subCallback()* method is invoked. This method is responsible for receiving the desired position to which the robot should go. For this, it calculates the linear and angular distance from the point where the robot is currently on the map. Thus, it is possible to calculate the linear and angular velocities that the robot needs to execute to reach its destination. The result obtained is shown in Fig. 18. The time took by the simulation was three minutes and 27 s.



**Fig. 18.** Formation square becoming a cross.

## 4 Analysis

The Table 1 presents the analysis of the use of RAM in kilobytes, where is evaluated a robot swarm with square formation without a leader. The first column of the table shows the number of robots used in the experiment. The second column presents the use of RAM using a scenario without the use of sensors. The third column presents the result using sonar sensor, the fourth column with three sonar sensors and the fifth column with laser sensor only. Finally, the last column displays the result using the *position\_controller* parameter with a value of *yes*. In this case reference was used as *relative* and *publish\_tf* as *yes*. RAM consumption information was obtained using the *smem*<sup>7</sup> tool from Linux.

It is noticed that the increase in RAM is more significant when the increment of sensors occurs for the number of robots from 240. In this case, the consumption of RAM increases around 9% when added to the scenario one sonar sensor; 14% with three sonars and 10% with one laser sensor.

In relation to the use of the position controller, we can see that the increase in the use of RAM is already significant from 12 robots, where there is an increase of 28% in relation to the scenario without any control and sensor; this adds up to

<sup>7</sup> <https://www.selenic.com/smem/>.

**Table 1.** Computational analysis (kilobytes)

# robots	Without sensor	With sonar	With 3 sonars	With LIDAR	With pose controller
1	127.043	127.427	127.418	127.286	132.049
12	129.635	130.082	130.004	130.328	166.381
120	152.797	156.964	160.457	158.293	668.327
240	177.930	193.840	202.133	195.538	1.100.053

337% when the number of robots is 120 and 518% when the number of robots is 240.

Already the memory consumption, for the same scenario, just now watching the number of robots suffers a significant increase from 120 robots. In the sensorless situation, when using 120 robots, there is a 20% increase of memory concerning the scenario with one robot. Even for the scene of 240 robots, the growth is 40%. For the sensor scenarios, what is a more remarkable increase is the scenario with three sonars, where for 120 robots, in relation to the scenario with one robot, there is an increase of 26% of memory consumption and for 240 robots that percentage goes to 59%.

Finally, analyzing the position controller about the number of robots, considering the scenario with one robot for the situation with 12 robots, we already have a 26% increase in memory consumption. This increase goes to 406% when the number of robots is 120 and 733% when the number of robots is 240.

## 5 Conclusions

The experimentation with robot swarm can be a time-consuming and tedious task if there is no automated help to set up the simulation environment and to start its execution. This paper aimed to show a novel ROS package to solve this issue. This package performs easy environment setup through a parametrization script to determine the number of robots, perception systems, position or velocity controller, local or global reference, industrial scenario, and initial formation.

An estimative of memory consumption is discussed to allow the users to predict the required resources to specific experimentation. However, robot quantity is the dependent of the computer capacity to execute individual resources, since each robot will require more computational it. For example, to simulate a swarm with 120 robots, all with sonar sensors and position control, the usage of RAM needed is 668.327 Kb.

The *swarm\_stage\_ros* package can support researches and educational activities, even in complex situations. Future works will demonstrate the development of more scenarios and the possibility of subgroups of robots, with the potential of a leader for each subset and also the integration of other types of sensors besides the laser and sonar are possible.

## References

1. Alkilabi, M.H.M., Narayan, A., Tuci, E.: Cooperative object transport with a swarm of e-puck robots: robustness and scalability of evolved collective strategies. *Swarm Intell.* **11**(3), 185–209 (2017)
2. Ben-Ari, M., Mondada, F.: *Swarm Robotics*, pp. 251–265. Springer, Cham (2018)
3. Beni, G.: From swarm intelligence to swarm robotics. In: *Swarm Robotics*, pp. 1–9 (2005)
4. Dai, W., Lu, H., Xiao, J., Zheng, Z.: Task allocation without communication based on incomplete information game theory for multi-robot systems. *J. Intell. Robot. Syst.* **94**, 841–856 (2018)
5. Deshpande, A.M.: Robot swarm based on ant foraging hypothesis with adaptive Lévy flights. Master's thesis, University of Cincinnati (2017)
6. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: A taxonomy for swarm robots. In: *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1993)*, vol. 1, pp. 441–447, July 1993
7. Parker, L.E.: Current research in multirobot systems. *Artif. Life Robot.* **7**, 1–5 (2006)
8. Parrott, C., Dodd, T.J., Groß, R.: HyMod: a 3-DOF hybrid mobile and self-reconfigurable modular robot and its extensions. In: *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems (DARS 2016)*. Springer Tracts in Advanced Robotics. Springer, Berlin (2016, in press)
9. Ramachandran, R.K., Wilson, S., Berman, S.: A probabilistic approach to automated construction of topological maps using a stochastic robotic swarm. *IEEE Robot. Autom. Lett.* **2**(2), 616–623 (2017)
10. Ramanathan, P., Mangla, K.K., Satpathy, S., Ramanathan, P., Mangla, K.K., Satpathy, S.: Measurement. Smart controller for conical tank system using reinforcement learning algorithm **116**, 422–428 (2018)
11. Seda, M.: Roadmap method vs. cell decomposition in robot motion planning. In: *International Conference on Signal Processing, Robotics and Automation, Greeks*, pp. 127–132 (2007)
12. Sharkey, A.J.C.: Robots, insects and swarm intelligence. *Artif. Intell. Rev.* **26**(4), 255–268 (2006)
13. Sharkey, A.J.C., Sharkey, N.: The application of swarm intelligence to collective robots. In: *Advances in Applied Artificial Intelligence* (2006)
14. Vaughan, R.: Massively multi-robot simulation in stage. *Swarm Intell.* **2**(2), 189–208 (2008)
15. Yogeswaran, M., Ponnambalam, S.G.: Swarm robotics: an extensive research review. In: Fuerstner, I. (ed.) *Advanced Knowledge Application in Practice*. InTech (2010)
16. Zelenka, J., Kasanický, T., Budinská, I.: A self-adapting method for 3D environment exploration inspired by swarm behaviour. In: Ferraresi, C., Quaglia, G. (eds.) *Advances in Service and Industrial Robotics*, pp. 493–502. Springer, Cham (2018)



# Edge-Enabled Autonomous Navigation and Computer Vision as a Service: A Study on Mobile Robot's Onboard Energy Consumption and Computing Requirements

Jens Lambrecht<sup>1(✉)</sup> and Eugen Funk<sup>2</sup>

<sup>1</sup> Chair Industry Grade Networks and Clouds, Technische Universität Berlin,  
Berlin, Germany  
[lambrecht@tu-berlin.de](mailto:lambrecht@tu-berlin.de)

<sup>2</sup> Gestalt Robotics GmbH, Columbiadamm 31, 10965 Berlin, Germany  
[funk@gestalt-robotics.com](mailto:funk@gestalt-robotics.com)  
<http://www.ignc.tu-berlin.de>

**Abstract.** Autonomous mobile transport systems are a crucial part of flexible factory logistics enabling a highly-adaptable industrial production organization. Edge computing bares the potential to support commissioning and operation of these mobile robots through offloading computation in a very flexible and scalable manner. Among these benefits, less onboard computation means also less energy consumption and longer operation durations. In order to evaluate the specific effects of offloading computing following the Software as a Service principle, we consider different software service distribution scenarios in terms of real-time critical navigation as well as additional computer vision functions. We present an empirically study using and assessing 4G radio networks in order to realize distributed control scenarios within industrial environments. The software services are offloaded to the edge following a modern microservice approach. Our results show that onboard computing can be covered by a low-cost single-board computer and energy consumption can be reduced by 4.9% to 18.4% through offloading computation depending on the payload and average velocity. Regarding to additional computer vision algorithms, the energy savings are even higher.

**Keywords:** Distributed control · Autonomous mobile robots · Cloud robotics · Edge computing · Wireless communication networks

## 1 Introduction

Highly-flexible production systems are characterized by frequent reconfiguration and adaptation towards changing production requirements. This is mainly driven by shorter product life cycles and ongoing trends of product customization, resulting in smaller lot sizes and rearrangement of production processes and

resources [1]. This trends deeply affect the overall organization of intralogistics. Static interlinking of particular production stations following a flow principle turns out to be a crucial bottleneck. Autonomous transport systems (ATSSs) take a key role to close this gap interlinking production stations by flexible sensor-driven navigation solutions. Technologies regarding path planning, simultaneous localization and mapping (SLAM) are enabling dynamic collision avoidance, versatile and location-flexible usage as well as further path optimization [2]. Within the age of Industry 4.0, ATSSs have to be considered as cyber-physical systems with a massive need for communication with other machines, other ATSSs and (superordinate) control systems [3]. Moreover, there is a potential to provide added value through mobile cognitive services, e.g. visual monitoring, layout and asset tracking and tracing or automated stock taking. This is facilitated by a recent performance boost in computer vision (CV) and machine learning technologies, i.e. especially Deep Learning for visual detection and recognition tasks [4]. Besides benefits, this technology has a high demand for computation power [5]. However, local computing power of ATSSs is restricted and eating up battery power. Offloading computation through service-based distributed software architectures bear additional potential towards a more flexible software deployment and maintenance through the whole life cycles. Moreover, hardware costs and energy consumption of the ATSSs can also be potentially reduced through offloading computation. However, the specific empirical effects of offloading computation have only scarcely been investigated so far. Furthermore, reliable and robust connectivity is a basic prerequisite in the industrial context. Whereas Wifi typically has practical restrictions in industries, we intend to investigate the usage of 4G radio networks in this context.

Within this publication, we are introducing an empirical study on the effects of using edge computing for offloading computation in a 4G radio network enabling navigation and machine intelligence as a service for ATSSs. The study investigates industrial applicability of this approach addressing the following research questions:

- What is the expected range on enhancing on-premises energy consumption and operation time?
- What are remaining requirements for on-premises computing on the ATSSs?
- Can 4G meet the demands for industrial connectivity in this context?

The paper is organized as follows: in Sect. 2 we summarize current research in cloud robotics; in Sect. 3 we introduce our software architecture and explain the specific software functions; in Sect. 4 we describe the experiments and its results; finally, we present our conclusion and outline future work in Sect. 5.

## 2 State of the Art on Virtualized Robot Control

As ATSSs can be considered as mobile robots, we provide an overview of cloud robotics research, i.e. offloading local computation to virtualized infrastructures.

Cloud robotics is about connecting physical robots with internet-accessible virtualized resources (cloud and edge) in order to support their operation through external computing and storage. Hence, the robot operation relies on or is supported by external data, services or programming [6]. In addition, the usage of external software services in robotics bears the potential for more enhancements:

- Interoperability and connected value-added services (e.g. big data or machine learning) through unified interfaces;
- Software life cycle: automated deployment and continuous integration (CI);
- Horizontal and vertical scalability as well as redundancies of control architectures;
- Less onboard hardware costs.

As an initial initiative in this field, the DAvinCi framework and its related project [7] showed the advantages of cloud computing in robotics by parallelizing and scaling a SLAM algorithm using a Hadoop cluster following a Software as a Service (SaaS) approach. Later, Kehoe et al. [8] supported a 3D grasping task for a mobile robot through a robust object detection module based on Google’s cloud services. In turn, the GostaiNet project [9] enabled to execute robotic vision and speech algorithms through compatible services in the cloud; it allowed seamless control of any robots using a web browser and hosted the services on the GostaiNet robotics cloud.

Previously, the project RoboEarth [10] provided an open-source cloud robotics framework with shared databases and a cloud engine that enabled robots to share knowledge and access powerful cloud services. Consequently, the sharing of information between robots, e.g. maps, was focused using the Robot Operating System (ROS) as communication middleware suitable for distributed control and synchronous as well as asynchronous communication. The communication between robots and a single ROS environment in the cloud was achieved by using the rosbridge library [11].

More recently, in the context of industry 4.0, the objective of the project pICASSO [12] was the development of efficiency gains through the flexible provision of control technology for cyber-physical systems in industrial production. The existing, monolithic control technology of robot and machine tool controllers was broken up, modularized and expanded with mechanisms of cloud computing, such as central data processing and service-oriented software architectures. This project mainly focused a factory cloud scenario, i.e. edge computing. The project investigated the related requirements, e.g. latency and package loss, and available communication principles towards the suitability of common industrial control services [13]. Further implementations and testing activities aimed for virtualized motion planning of a robotic manipulator with regards to real-time requirements. Effects on control performance, availability and scalability were investigated and documented [14].

In contrast to the existing research, we intend to empirically investigate the effects of offloading typical navigation and computer vision services for mobile robots and ATS towards the edge. Whereas 4G and 5G radio networks provide additional Quality of Service (QoS) and low latencies, we see recent poten-

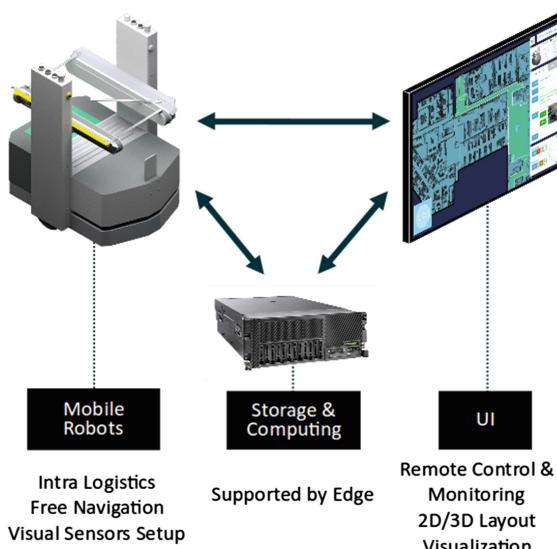
tial towards a reliable offloading of real-time critical functions like navigation as well as extending the function range of mobile robots through the usage of edge-enabled computer vision services. As a novelty, we aspire to quantitatively identify standard values for latency requirements as well as revealing demands towards onboard computation as well as positive effects on the operating durations through offloading the navigation stack and additional functions.

### 3 Concept

Within this section, we introduce the overall concept enabling flexible offloading of specific software functions from running on-premises on the ATSs towards scalable usage on a virtualized edge platform. As main software functions, we consider autonomous navigation as well as an optional Deep-Learning-based environmental analysis. As radio network technology a 4G campus network is utilized. In the following, we will present architecture and communication as well as the scope of the considered software functions.

#### 3.1 Distributed Architecture

This section describes the edge-enabled communication structure and architecture for the ATSs. Hardware, such as the ATS, UI devices (tablet, etc.) as well as the edge server are interconnected following a distributed computation principle



**Fig. 1.** Connected devices within our distributed control architecture: autonomous transport systems/mobile robots, edge computing devices for storage and computing as well as user interfaces (UI)

(see Fig. 1). At the software architecture level, we are following the microservice paradigm [15] providing modularized software functions that:

- run independently,
- are covering a small set of functions with a specific focus and
- are supporting decentralization through easy and automatic orchestration, monitoring and transfer between different infrastructures.

Our microservices are encapsulated in Docker containers that are integrated in a DevOps pipeline orchestrated through the Kubernetes framework.

### 3.2 Communication Middleware

We are using ROS as basic communication middleware supporting distributed synchronous (ROS services) and asynchronous communication (ROS topics) as well as data storage. The ROS framework was enhanced by security functions in order to secure communication on the network and transport layer. The usage of ROS simplifies the declaration and implementation of transmitted messages between units and manages them using a ROS master node. The transmission of sensor data to another device is done through reading raw data in binary from the sensor and converting and transferring it to a ROS topic.

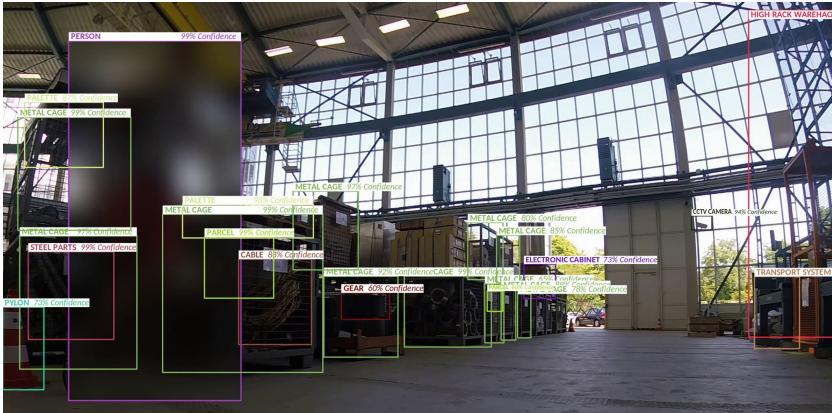
### 3.3 Service Distribution

For safety reasons, basic functions for collision avoidance and people safety are onboard services on the ATS. These highly time-sensitive functions are realized by a Programmable Logic Controller (PLC) which is directly connected to the motor control and the emergency stop.

**Autonomous Navigation as a Service.** Regarding autonomous navigation, the only robotic software modules left onboard (on-premises) are ROS nodes for publishing sensor data, i.e. a 2D Lidar sensor and a camera, and subscribing Cartesian velocity values for motion control. Hereby, we shortly introduce our main edge microservices enabling autonomous navigation:

- SLAM: providing simultaneous self-localization and mapping in 2D through an adaptive Monte Carlo localization (amcl);
- Local Planner: local trajectory roll-out and collision avoidance on the 2D plane, produces velocity commands to send to the motor nodes of the mobile base;
- Global Planner: a grid-based high-level path planner that uses the map to compute an overall path for a robot. Typically, its high level path are passed to the global planner for execution.

Further edge services include managing the map, planning recovery behaviors and calculating costmaps.



**Fig. 2.** Mobile edge-enabled environmental analysis and stock taking through CNN-based object detection visualized through colored bounding boxes

**Deep-Learning-Based Environment Analysis.** Further machine learning services enabling object detection are located on the edge. Augmenting ATS with extended machine learning functions can enhance the application scope as well as optimized logistic operations, e.g. through context-aware behavior. We are considering a mobile environment analysis and stock taking scenario through robust object detection based on Convolutional Neural Networks (CNN). The implemented object detection is based on a Faster R-CNN [16] implemented in TensorFlow with an Inception-ResNet CNN backbone [17]. This CNN topology is known for high precision. It was trained towards a customized object dataset. Figure 2 illustrates an exemplary object detection result including a privacy-aware anonymization of detected persons following [18].

## 4 Experiments and Empirical Evaluation

Following the architecture concept in Sect. 3, we implement a test setup involving an ATS and an edge cloud within a public 4G network environment. The radio network connectivity was implemented through a 4G USB dongle and a VPN tunnel. The edge cloud is represented by a factory server equipped with an OpenStack environment in order to scale and manage virtual machines. The empirical evaluation took place for a realistic industrial setup in a machine hall. Initially, we intend to verify if offloading the full navigation stack is feasible with regard to the resulting latencies. Furthermore, we evaluate the effects on offloading computation towards on-premises demands for computation as well as on energy consumption through scenario-based test series. The implementation of different on-premises and offloaded computing scenarios is supported by the flexible microservice architecture and service orchestration using Docker containers.

#### 4.1 Network Latencies

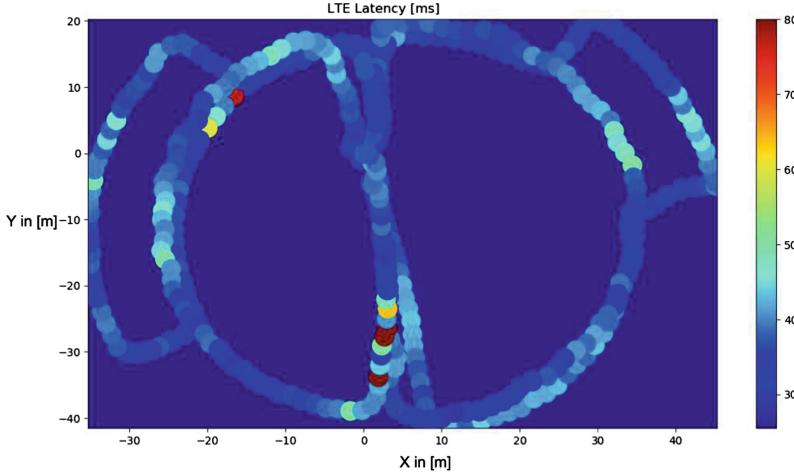
SLAM, global planning as well as local planning nodes are offloaded towards the edge server. Whereas the global planner is a non-real-time critical function, local planning and SLAM are typically real-time critical functions affecting directly the robot's mobility. In case of an increased latency, the feedback loop results in a delayed localization and provision of motion goals. At worst, this leads to a complete immobility for high latencies. In order to identify specific latency requirements and the suitability of the edge as computing infrastructure, we implement a QoS monitoring to enable testing and empirical evaluation of different network and distribution scenarios. Figure 3 is showing the resulting 4G, map-related latencies for a roundtrip cycle between ATS and edge. The average latency was  $\bar{x} = 43.5$  ms with a standard deviation of  $s = 7.6$  ms.

The practical effect of latency regarding to offloaded navigation is a slower reaction, i.e. dead time, within the closed control loop. For average latencies below  $l_{thresh} = 50$  ms there is no recognizable effect on the practical motion of the ATS in comparison to on-premises navigation. We examined further practical effects by adding artificial latencies to the setup. In terms of latencies that are slightly greater than approximately  $l_{thresh}$ , there is a visual effect when turning corners, i.e. less efficiency in motion due to delayed motion vector adaption. For latencies above  $l_{critical} = 150$  ms this effect results in an overshoot that prevents the overall operation. The magnitude of this effect depends on the operational velocity of the ATS. Within our results, we considered an average velocity of  $\bar{v} = 0.4$  m/s, which tends to be a common average operation speed in industrial practice. With regard to slower velocities, an overall operation is still possible even with greater latencies without any greater restrictions. Ongoing tests confirmed these assessments. If one may assume a fixed latency, control commands may even be optimized to avoid the resulting deviations. As in practical use, the ATS is relocating itself and tries to dynamically compensate motion deviation within a feedback loop the motion trajectory may tend to oscillate depending on the control parameters. Based on additional test results yet to be published, we do not recommend to offload the navigation stack at all even for small velocities if the average latencies greater than  $l_{critical} = 150$  ms.

#### 4.2 On-Premises vs. Offloaded Computation

Within this subsection, we empirically examine the effects of offloading control functions on the requirements for on-premises computing and energy consumption. However, the total energy consumption might be higher in scenarios offloading software services towards edge, due to additional energy consumption of the edge server and even of the 4G communication infrastructure. But in this specific case, we are interested in positive effects on the operation time through less on-premises energy consumption of the ATSs.

**Setup and Experiment Design.** The considered ATS is a Festo Robotino, i.e. an omnidirectional mobile platform controlled by an industrial PC (PC/104)



**Fig. 3.** Exemplary latency map (roundtrip) for an industrial environment: result of a combination of SLAM and 4G latency measurements between ATS and edge

with an i5 2.4 GHz processor and 16 GB RAM. In addition, the Robotino ATS is equipped with a Hokuyo URG-04LX 2D Lidar sensor for navigation purposes and a webcam for CV tasks. In terms of simplification, the usage of the PLC has been waved within the experiments. The implemented modularized navigation services are derived from the standard ROS navigation stack including the following crucial microservices: local path planning, global path planning and SLAM (see Sect. 3.3). Optionally, we are considering a CNN-based Computer Vision (CV) function for object detection (see Sect. 3.3).

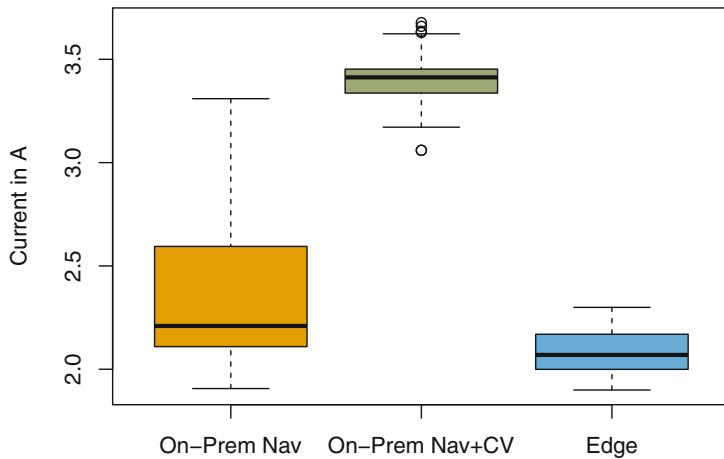
Within our testing environment the Robotino drove autonomously in a long-term test between random locations on its map. Within different scenarios, we implemented navigation and computer vision services on-premises as well as on the edge, comparing CPU and RAM utilization and energy consumption. Altogether, a minimum 1000 respective measurements have been recorded over a period of several hours per scenario.

**CPU Load and RAM Utilization.** Table 1 is showing the mean average values of CPU and RAM load for navigation on-premises, navigation and CV on-premises as well as the offloaded scenario. The measurements were done through automatic logging using the Unix system monitor tool *htop*. Implementing both navigation and computer vision on-premises leads to a constantly full capacity load for the CPU with obvious negative effects on the performance of the navigation through internal latencies. Whereas offloading the software services leads to a constant low demand for CPU and RAM, solely navigation on-premises has a broad scattering of CPU load with a standard deviation of  $s = 12.3\%$ . Consequently, the onboard industrial PC could be downgraded within an edge-controlled scenario.

**Table 1.** Onboard CPU and RAM usage for different distribution scenarios of navigation and computer vision (CV) services

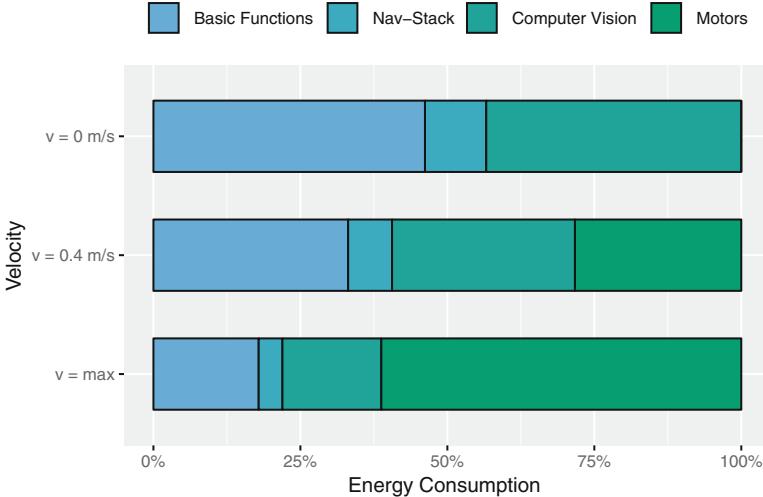
	On-premises navigation	On-premises navigation & CV	Edge offloading
CPU	41.4%	98.7%	12.9%
RAM	1.33 GB	4.11 GB	0.738 GB

**Energy Consumption.** The energy consumption is measured through the overall system current that can be accessed through the Robotino API. The voltage level is constant. Initially, we considered a scenario with an averages ATS velocity of  $\bar{v} = 0.4\text{ m/s}$ . Figure 4 shows the current distribution for the three earlier mentioned distribution scenarios. It can be observed that the scattering concerning to a solely on-premises navigation also applies for the current. Considering the mean average values, we can state that through offloading the navigation stack the energy consumption could be reduced by 10.85% and in terms of offloading both navigation and computer vision the energy consumption could be reduced by 38.59%. Certainly, these values only apply to the specific velocity and payload.

**Fig. 4.** Boxplot of robot's system current for on-premises navigation (orange), on-premises navigation and computer vision (green) as well as edge-offloaded navigation and computer vision (blue) for an average velocity of  $\bar{v} = 0.4\text{ m/s}$  without additional payload

Therefore, we investigated the overall range of energy savings by adding experiments with minimum ( $v = 0\text{ m/s}$ , no payload) and maximum motor energy consumption ( $v = 0\text{ m/s}$ , maximum payload). As the maximum velocity is software-restricted to  $v_{max} = 2.78\text{ m/s}$ , we rather simulated the motor load

through adding maximum payload with regard to driving statically against an obstacle that was not visible for the Lidar sensor. As the velocity is given the PID motor controllers provide maximum energy to attain the set values. Furthermore, we identify the amount of current that is needed to feed the basic functions (sensors, industrial PC) at  $v = 0 \text{ m/s}$  without any additional computing. Moreover, we distinguish between the current needed for computing and the motor current. Figure 5 illustrates the resulting energy shares for the full velocity/payload range.



**Fig. 5.** Average percentaged energy shares for basic ATS functions, additional navigation & computer vision services as well as motors, compared for different average velocities without additional payload and maximum velocity/payload

**Discussion.** Hence, the potential energy saving for offloading control functions for navigation ranges between 4.98% and 18.41% and for offloading additional complex CV it ranges between 20.86% and 53.8% (see Table 2). These findings can only hardly be transformed into general statements because our considered scenario is simplified, e.g. we neglect the usage of a on-board safety PLC, and depends on a specific hardware setup. However, our results proof that offloading computation towards cloud and edge can extend the operation time of ATS significantly, especially if additional CV functions are used. For this purpose, our results represent a rough indication about numeric effects. Our results have been finally verified through additional test comparing the overall operating duration within the different scenarios.

With regard to the CNN-based CV, there is another performance-related advantage through offloading from a CPU computation towards an edge-based

**Table 2.** Range of percentaged energy savings (depending on velocity/payload) through offloading navigation services as well as navigation & CV software services to the edge

Velocity/payload	Max	Min
Navigation	4.89%	18.41%
Navigation & CV	20.86%	53.8%

GPU-supported computation. Consequently, there is an overall benefit regarding the computing time that compensates the additional network latency. This is because the computation of CNNs is highly benefiting from GPU-enabled parallelization. In our scenarios, we implement the object detection service on the edge supported by a Nvidia Tesla K80 GPU. Compared with a local implementation on the ATS, we measured an acceleration by a factor of more than 10 including the network latencies. However, it is possible to compute CNNs efficiently onboard through dedicated embedded devices, e.g. Nvidia Jetson AGX Xavier, that rises the overall onboard complexity and power consumption, i.e. minimum 50 W for the Xavier in normal use cases.

## 5 Conclusion

We presented a microservice-based architecture for autonomous mobile transport systems and mobile robots supporting flexible service distribution between usage on-premises and on an edge server. Different distribution scenarios have been tested in a real production environment using a public 4G radio network. We showed that it is feasible to offload the full navigation stack towards the edge in terms of latencies and resulting delays for the overall control cycle.

Our empirical tests quantitatively prove the advantages of offloading computation towards the edge in terms of energy consumption and resulting operation time as well as onboard computing requirements. Despite the qualitative values only apply for a simplified hardware setup, we can clearly indicate the potential energy savings for different scenarios and can state that offloading the navigation computation leads to a longer operation time. This effect is even greater if the mobile systems use additional functions towards environment analysis, e.g. CV for object detection. In most cases, especially regarding CNNs the edge is a basic enabler for the usage of complex CV functions because most of the embedded PCs on the mobile systems would be overstrained or would provide a bad performance, e.g. concerning to the frame rate. Consequently, in terms of offloading the on-premises computing requirements could be covered through a low-cost single-board computer instead of a common industrial PC. That may have a positive effect on the cost of production of ATSs as well as on the commissioning duration for software installation and more software related works along the life cycle.

## References

1. Zhong, R.Y., Xu, X., Klotz, E., Newman, S.T.: Intelligent manufacturing in the context of industry 4.0: a review. *Engineering* **3**(5), 616–630 (2017)
2. Cadenaet, C., et al.: Past, present, and future of simultaneous localization and mapping: towards the robust-perception age. *IEEE Trans. Robot.* **32**(6), 1309–1332 (2016)
3. Lu, Y.: Industry 4.0: a survey on technologies, applications and open research issues. *J. Ind. Inf. Integr.* **6**, 1–10 (2017)
4. Wang, S., Wan, J., Li, D., Zhang, C.: Implementing smart factory of industrie 4.0: an outlook. *Int. J. Distrib. Sens. Netw.* **12**(1), 3159805 (2016)
5. Wan, J., Zhang, D., Zhao, S., Yang, L.T., Mauri, J.L.: Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Commun. Mag.* **52**, 106–113 (2014)
6. Kehoe, B., Patil, S., Abbeel, P., Goldberg, K.: A survey of research on cloud robotics and automation. *IEEE Trans. Autom. Sci. Eng.* **12**(2), 398–409 (2015)
7. Arumugam, R., et al.: DAvinCi: a cloud computing framework for service robots. In: *IEEE International Conference on Robotics and Automation*, pp. 3084–3089 (2010)
8. Kehoe, B., Matsukawa, A., Candido, S., Kuffner, J., Goldberg, K.: Cloud-based robot grasping with the Google object recognition engine. In: *IEEE International Conference on Robotics and Automation*, Karlsruhe, pp. 4263–4270 (2013)
9. Mester, G.: Cloud Robotics Model, Interdisciplinary Description of Complex Systems, vol. 13, pp. 1–8 (2015)
10. Waibel, M., et al.: RoboEarth. *IEEE Robot. Autom. Mag.* **18**(2), 69–82 (2011)
11. Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: ROS for non-ROS users. In: *Robotics Research: The 15th International Symposium ISRR*, pp. 493–504. Springer (2017)
12. Vick, A., Horn, C., Rudorfer, M., Küger, J.: Control of robots and machine tools with an extended factory cloud. In: *IEEE World Conference on Factory Communication Systems (WFCS)*, Palma de Mallorca, pp. 1–4 (2015)
13. Schlechtendahl, J., Kretschmer, F., Sang, Z., Lechler, A., Xu, X.: Extended study of network capability for cloud based control systems. *Robot. Comput.-Integr. Manuf.* **43**, 89–95 (2017)
14. Vick, A., Vonasek, V., Penicka, R., Krüger, J.: Robot control as a service - towards cloud-based motion planning and control for industrial robots. In: *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, Poznan, pp. 33–39 (2015)
15. Namiot, D., Sneps-Sneppe, M.: On micro-services architecture. *Int. J. Open Inf. Technol.* **2**(9), 24–27 (2014)
16. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, vol. abs/1506.01497 (2015)
17. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, Inception-ResNet and the impact of residual connections on learning. *CoRR*, vol. abs/1602.07261 (2016)
18. Kittmann, T., Lambrecht, J., Horn, C.: A privacy-aware distributed software architecture for automation services in compliance with GDPR. In: *IEEE International Conference on Emerging Technologies and Factory Automation* (2018)



# A Bio-Inspired Approach for Robot Swarm in Smart Factories

Ronnier Frates Rohrich<sup>1(✉)</sup>, Marco Antonio Simoes Teixeira<sup>1</sup>, Luis Piardi<sup>2</sup>,  
and André Schneider de Oliveira<sup>1</sup>

<sup>1</sup> Universidade Tecnológica Federal do Paraná (UTFPR), Av. Sete de Setembro,  
Curitiba, PR 80230-901, Brazil

{rohrich, andreoliveira}@utfpr.edu.br, marcoteixeira@alunos.utfpr.edu.br

<sup>2</sup> Research Center in Digitalization and Intelligent Robotics (CeDRI),  
Instituto Politécnico de Bragança, Campus de Santa Apolónia,  
5300-253 Bragança, Portugal  
piardi@ipb.pt

**Abstract.** Recent theoretical developments have revealed that the influences and efficiency that mobile robots have brought to society in the last years are incredibly revealing and should be explored in applications for the benefit of the community and the corporate world. Access to this technology enables the development of innovative research for increasingly active industrial environments. This research constitutes a relatively new area which has emerged from the problems of the industry that aims to automate activities considered costly efficiently. A common strategy used to study mobile robots, in production, is to automate work routines through robots, but specific tasks improve specific works. This paper proposes a new approach to use a SWARM of mobile robots to solve problems in the industry based on the bio-inspired solution. The bacteria can have actions that guarantee the survival of their colony; for this purpose, a series of measures can be adopted by the bacteria constituting the colony. This approach has been widely adopted in the field of SWARM of mobile robots with technical and sensory restrictions, to realize a plausible application in the industrial environment. The results of the experiment found clear support for the methodology created, and the bio-inspired SWARM proved to be potentially useful for applications in real industrial robot solutions conforme artigo [11].

**Keywords:** Swarm of mobile robot · Cognition mechanism · Bacterial colony

## 1 Introduction

A common strategy used to study SWARM of robots is to search inspiration at nature [10]. Some examples such, ants and bees were a few years ago. This approach was a started point to think about other organisms that could be related better with the proposed of this work. The *level of simple complexity* was

the first characteristic sought in these organisms because one of the objectives of this work is to be based on organisms that have simple cognitive systems or behaviors to have been implemented in simple mobile robots. The bacterium is an organism that has the particular features intended for this work. Although they do not have a formal cognitive system, cognitive behaviors are verified, for example, searching for source energy, communication for social behavior, hierarchical organization, collective memory and learning from experience [3,4].

Similarly to living beings, robots can have different levels of complexity. The ability to “feel” the world is related to the number of available sensors in a robot, and consequently, the cost of this system is linked to this feature [17]. Eventually, the appropriate solution involving robotic systems in the industrial environment is not to increase the complexity of a robot by the number of sensors [12], but increase the number of robots working collaboratively as in a colony of bacteria, which can be called SWARM of robots [15]. In such applications, robots must manage their tasks, coordination, and energy like in a colony of bacteria.

One of the essential improvements in the smart industry is employed artificial intelligence (AI) to solve problems like communication and auto-formation from swarm robotics system [2]. The issues like delay in warehousing to delivery, quality control can benefit itself with this technology, and these problems are vital for the companies. The industry has heavy demands on adaptive with very different orders. For example, in the delivery field, each order is unique in terms of sorting, packing, and delivering. The system to control this situation must be intelligent to realize the personal requirements and deliver in a correct address, efficiently and safely. The prediction with cost reduction when uses robotic in the industry is 40% [13]. A cheaper and intelligent methodology for swarm mobile robots that could be implemented for a *Smart Industry*.

A strategy that is becoming increasingly common in industrial environments is the use of multi-robots approach to do multiple tasks [1]. The mobile robots used in this work are automatic guided vehicles (AGVs). But, the proposed solution in this work increases the results based on implementation a methodology inspired by a bacteria colony, because they are a physiologically simple individual, very successful in survival for decades and high performance in a large number of collaborative activities. The approach that is SWARM of mobile robots inspired in bacteria had never been used in industry solutions. Many similarities have been observed in this context, for example, many mobile robots looking for an area to recharge your battery [14] is a similar process that a bacteria colony search by energy source in nature [16]. The main objective of this work is to offer a optimize and an efficient solution for the problems in an industry with a SWARM of robots. A simulation for multiple robots in an industrial environment inspired in the limitation of an individual and bacteria colony was done and will be presented. SWARM of mobile robots brings a different approach to solve problems. The benefits are expected in efficiency (less battery consumption with the correct time to perform a task), auto formation to execute a different task, replace a robot is easy and cheap.

An outline of this paper is as follows. In Sect. 2, a brief state of the art is presented. Section 3 describes the architecture of the system, namely, bio-inspired and bacteria cognition. Section 4 is shown a Bio-Inspired System used at this work. Section 5 describes the application development. Experimental results are presented in Sect. 5, and finally, Sect. 6 concludes the paper and present the future works.

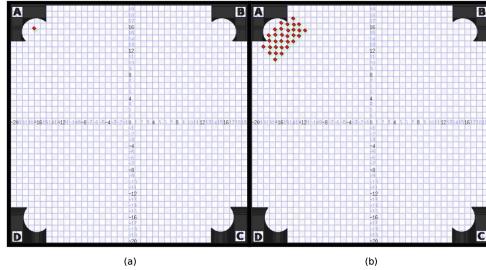
## 2 Related Work

A series of recent studies have indicated that SWARM of robots will be a great solution for several problems in the industry over the next few years. Recently applications present efficient and satisfactory results, such as [5], this work described a modern optimization technique for coordination in the multi-robot system such as the max-sum algorithm, that has the aim to transport an item from loading to unloading bays in the warehouse. Here, the crucial component is to avoid interference when moving in the environment. The first problem in this work could be observed because just one task is enabled for a robot, if something new happens, the task of the robot must be re-planned, and the empirical methodology was slightly explained. The literature review shows that a human-swarm interaction (HSI) is closely related to the development of a specific solution for a successful implementation of multiple robots in logistics and industrial environments, as in [8] that a survey was developed to show a several points that should be considered when a SWARM of robots is used in industry environment. This work identifies core concepts needs to design a human swarm-system and gives an explanation about works that have the same basis of this paper, such as *Bioinspired* models, *Control Theory*, *Amorphous Computing* and *Physical-Inspired*. SWARM tasks and behaviors were explained and guided the experiments of this work, such as the spatial organization, navigation, and collective decision-making behaviors. An important feature explored in [8] is communication. Two approaches are presented in this work: first, when the operator is outside or inside the swarm, second, the communication between the robots concerning the topology of the swarm network. On the first situation, it is common to find an operator behind the computer terminal receiving all information from the swarm. However, some authors have also suggested that the HSI will also be present in the industry to bring smart solutions [9].

## 3 System Architecture

The SWARM system was designed considering the evolution of different needs and configurations that different problems can present. However, the evolutionary studies show that the success to this prokaryotic organism is surviving millions of years on planet earth, is due to the fact of working in collaboration with other individuals of the same species. A colony of bacteria is the standard format to find this organism in the nature. One of main reasons bacteria work collaborative is the difficulty finding energy sources, because the locomotion of

the bacteria is extremely slow and covers small areas for the perception. In the Fig. 1(a) has been shown a single robot simulating a bacteria in the huge environment to explore alone.



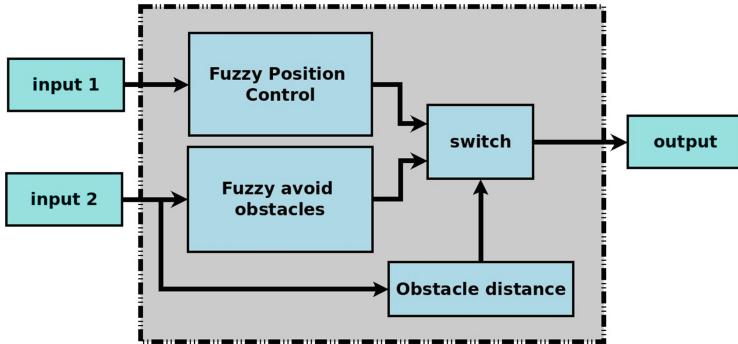
**Fig. 1.** (a) A single mobile robot equipped with laser scan inserted in a large environment. This robot is able to map a small area due to the short range of its laser sensor. (b) The thirty one robots are enabled to start collective tasks using a strategy bio-inspired in bacterial colonies.

The bacterium has cognitive mechanisms, as such the process of differentiation (perceiving the presence of food and locomotion up to it) and epigenetic memory (stored in the genetic code). The bacteria learn from past experiences.,when formed colonies of bacteria, new cognitive mechanisms are observed, as presented in [3]:

- Chemotaxis - feel the “taste” of the environment. E.g. Bacteria are attracted by food and repel from regions where the presence of antibiotics is detected;
- Quorum sensins - transmission of messages between the bacteria of the colony through chemical messages. This mechanism controls the growth pattern of the colony, and the messages may be attractive or repulsive;
- Sporulation - when the environment in which the bacterium were found is extremely without energy sources, thus, the colony takes on a spore and stem format, initiating the process of reproduction. This state is irreversible, a collective decision is made, whose main objective is the colony’s survival;
- Conjugation - transfer of information extracted from the environment. Type of sexual behavior (horizontal gene transfer), exchange of knowledge by genes.

The first advantage to use a swarm of robots in industry and in anyone application is the covered area [15]. Consequently, tasks such as exploration the environment are fastest than a exploration with a single robot. In the Fig. 1 (b) thirty one mobile robots show a great area that is covered by a swarm of robots.

The control system implemented to move the robots inside the industrial plant uses fuzzy logic to define the relations of angular velocity and linear velocity that must be set. In the Fig. 2 is showed this architecture. A fuzzy controller is a usually technique to control the movements and direction of robots [6].



**Fig. 2.** The control strategy of the leading SWARM robots consists of two Fuzzy systems (e.g., Fig. 2), where one is responsible for the robot's position control, while the other is responsible for the obstacle deviation control.

This fuzzy controller was used in every third one robots of SWARM, whether obstacles are dynamic (other robots) or fixed. Fuzzy position control kicks in and begins sending to the robots constantly the speeds linear and angular when setting the desired point for the robots of SWARM. These inputs act as a key, where they proceed to the robots according to a task rule.

## 4 Bacteria-Inspired Smart Behaviour

The behaviors that bacteria have carried out over billions of years have guaranteed survival on the earth. The formation of colonies containing several bacteria is the main characteristic to be highlighted during the evolution of this species. A fundamental and straightforward cognitive mechanism that was present in Sect. 3 of bacteria is the “*Chemotaxis*”, it is responsible to “feel” the environment and inspired the behavior of the robots in this work. Because the same activity that a bacteria colony has when it is immersed in a rich food environment is applied for robots, and when a harmful substance is detected, the colony must avoid this specific area. In table 1 is possible to verify a similarity between some features that the robots and bacteria have in common. The planning of this work is to start to explore some of these behaviors to solve problems founded in industry, as presented in [7].

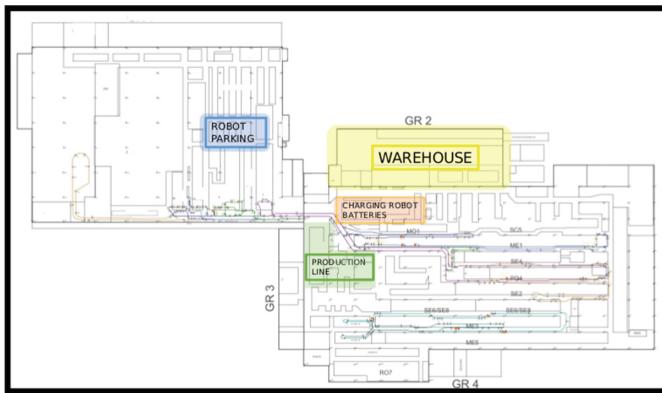
## 5 Application Development

The Bio-inspired SWARM of robots is composed by thirty one mobile robots, that should be simulated some behaviours from a colony of bacteria. Some cognitive mechanisms had been involved to present the approach of this work that must be related a swarm of robots and colony bacterial, because the focus is optimize routine tasks in the industry. However, it is imperative that activities

**Table 1.** Analogy between the behaviours of bacteria and robots that was used to inspired the action of robots

Behaviour	Bacteria	Robots
Stay alive	<i>Find food</i>	<i>Find power source</i>
Feel environment	<i>Chemical receptors</i>	<i>Sensor monitoring</i>
Decision making	<i>Quorum sensing</i>	<i>Fuzzy logic</i>
Cluster	<i>Colony</i>	<i>Swarm</i>
Regions to avoid	<i>With antibiotic</i>	<i>With obstacles</i>
Communication	<i>Chemical messages</i>	<i>ROS messages</i>

such as loading and unloading are performed with the minimization of errors and failures. Because, the minor problem can generate relevant amounts of injury. It was defined some bio principles for this approach. Based on the survival of a bacterial colony is related to the meeting of energy sources, some premises were established as the process of differentiation. It was associated with a specific work that the swarm should be perform. In normal situation the robots are performing a specific task, in this paper, the SWARM is responsible for picking up at the warehouse and taking them to the production line.

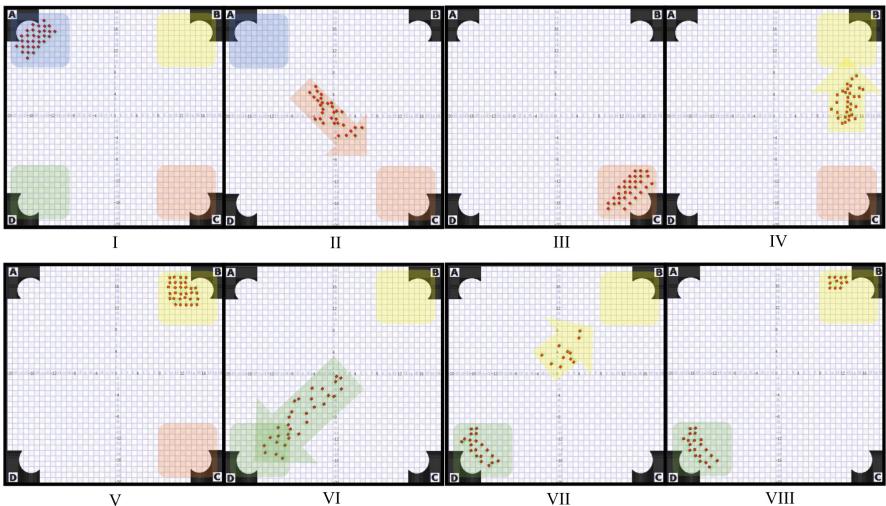


**Fig. 3.** Industrial plant.

The use of SWARM of robots in industries involving several processes that present different layouts. The industrial plant presented in Fig. 3 is an inspiration for this work, and the control of these robots is prepared for eventual layout changes.

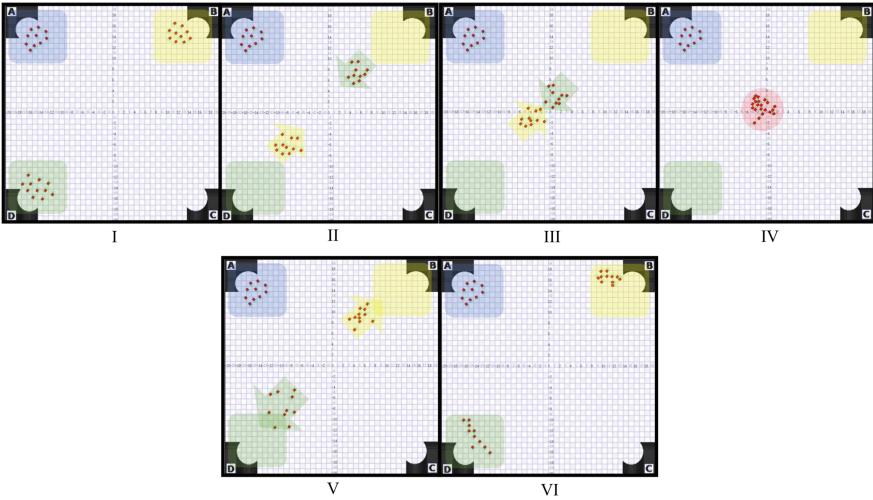
Using this plant as inspiration was created a simulation environment containing the four main areas in which SWARM perform their tasks: area (A) robot parking, area (B) warehouse, (C) charging robot batteries and (D) production

line. When the robots have not task, they parked in (A). The complete development of the task routine performed by SWARM are presented in the Fig. 4. In the step (I) the SWARM are waiting for instructions to start the activities. In (II), the robots are moving to charge batteries in area (C). After all robots have arrived in the zone to recharge the batteries (III), they are able to move on to the next logistical stage to pick up the spare parts in warehouse, represented by (IV). After the robots collect the parts (stage (V)), it begins the process of transporting the material to the production area (D), showed by stage (VI), representing the last step of the process. When the cycle begin in this moment and every steps were satisfactorily completed, the system is ready to enter in the cyclical process. If the robot's batteries are appropriate level to perform the tasks, they will continue collecting parts in area (B) warehouse and leading to area (D) process line. In the sequence, while some robots are still leaving the spare parts on the production line (stage (VII)), and the remaining robots go to warehouse to collect more spare parts (stage (VIII)).



**Fig. 4.** Simulation environment representing: (A) robot parking, (B) warehouse, (C) charging station and (D) production line area.

Some specific situations are observed during the tasks performed by the swarm of robots. A commonly seen case occurs during the process in which spare parts are removed from the warehouse and taken to the production line one more time. Robots must be able to bypass dynamic obstacles (others robots), because each robot has a different time for the task between picking a spare part and taking it to the production line. This situation is observed in the Figs. 5. A crowded area is expected when different groups of robots (e.g. two SWARM robots) is moving on opposite directions (stage (I) and (II)) allowing accidents between the robots.

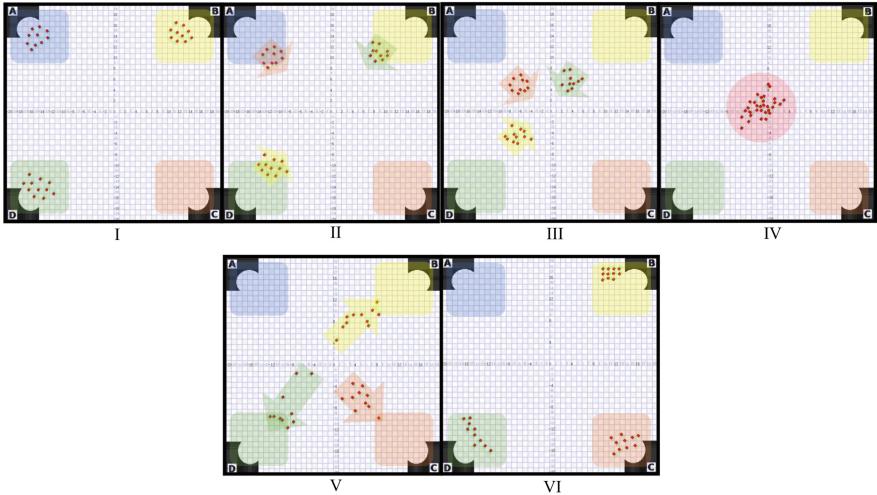


**Fig. 5.** Environment state representation where different groups of SWARMS robots can collide.

Throughout the operation time, the fuzzy control system is constantly checking for the need for trajectory changes. When an obstacle is detected, the controller inserted in each of the robots is responsible for ensuring the correct displacement of the robots. In Fig. 5 stage (III) and (IV), the robots start to detect obstacles and avoid collisions. As the robots perceive each other the obstacle deviation control is triggered. A crowded area is identified in red area. The movements must be performed slowly to avoid possible accidents. In (V) and (VI) stage, the robots have already surpassed the risk zone, avoiding the collisions, so that each robot is able to carry out its task safely.

The same fuzzy control was implemented for the robots move inside the plant and realize the necessary deviations during their journey. A third swarm of robots may be coming out of the parking (A) and driving to charge the batteries (C), so there may be the finding of robots in three different directions (I). Subsequently, they start to move (II) to the objective regions. In figure (III) is observed the approximation between each group of swarm. In stage (IV) it is observed a crowded region where there is difficulty for locomotion of the robots, due to possibility of collisions between the swarm robots. In (V) each robot autonomously avoid the collision driving without interference to its destination. Finally in step (VI) each robot returns to perform its activity, according to the group in which it integrates.

The last case study observed in this work refers to fixed obstacles found in industrial environments. In addition to the robots having to develop their routine activities, they must be prepared to avoid layout changes in an industrial plant. In the Fig. 7 is presented the result of the simulation. After the swarm

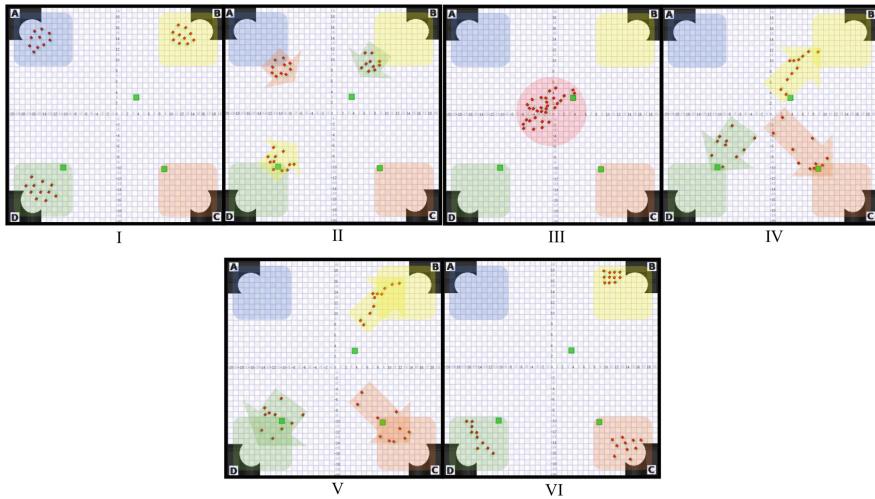


**Fig. 6.** Case study with three swarm groups are located in different areas and subsequently, they start to move to complete the group objective.

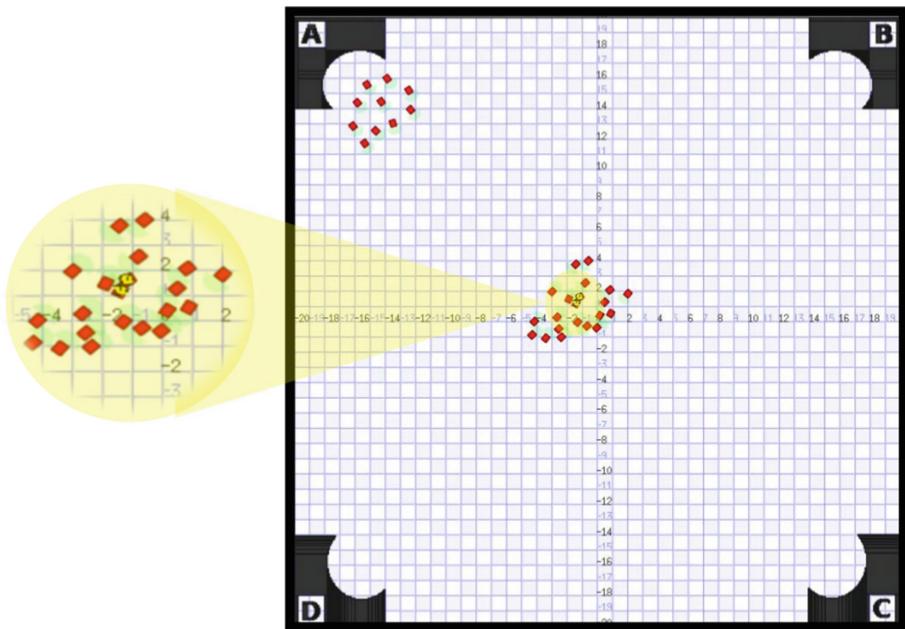
of robots are stopped (I), they move already needing to realize the deviation of fixed obstacles found in the plant (II). A large crowded region is observed (III), because robots need to divert from other robots and fixed obstacles. In step (IV) the robots find some more fixed obstacles before reaching their destination. Finally, the robots approach the target area (V) and perform their tasks (VI).

## 6 Results

The main objective of the work was to develop a behavior of a bio-inspired SWARM robots on the actions and cognitive mechanisms that bacterial colonies have developed over thousands of years. The application in the industrial environment brought as innovation the possibility of implementing this system considering that the behavior of bacteria, which are one of the main examples of success in the species survival, making a parallel with the activities that robots need to perform in smart factories. The SWARM of robots application, bio-inspired in bacterial colony behavior proved to be effective, since the cognitive mechanisms (differentiation) observed by SWARM assisted the execution of simulated tasks in the virtual environment. The group displacement of the robots, among all planned areas (parking, loading, warehouse and production line) was precise and without any failure, the time for execution of each task was appropriate with each activity. It is necessary to optimize the implemented fuzzy controller, because in a situation where the robots detected the frontal collision with other robots in opposite directions it was noticed some flaws in the process, causing small collisions, as can be observed in Fig. 8. However all the tasks were completed. The obstacles used in this work to simulate area to avoid by the the robots, as the behavior bacteria colony present when the presence of antibiotics



**Fig. 7.** Experiment with fixed obstacles in the industrial environment.



**Fig. 8.** In the Fig. 8 is presented a type of collision that happen. This events was detected in crossing areas that have robots in different directions.

in environment, proved to be a good solution, but new experiments suggest that a larger area of inflation should be used in order to reduce the time to divert obstacles.

## 7 Conclusion and Future Work

Algorithms inspired by systems of nature, have been part of decades of the daily life of researchers around the world. However, a new bio-inspired approach to bacterial colony to be implemented with SWARM of robots in industrial environments has a high degree of innovation in its concepts and methodologies. The cognitive mechanisms of bacterial colonies evidenced throughout this work suggest an intimate insight of a bio-inspired approach to the solution of problems found in the industry and that allow the use of SWARM of mobile robots. Therefore, the current scenario of the development of technologies for robotic systems favors and motivates more and more research in this context, therefore, the conclusions obtained in this work suggest the continuation of this line of research to obtain even more promising results. Although the robots used for the SWARM simulation fulfill their final objectives, the tasks could be revised, making SWARMS of robots specialized for each activity, because depending on the task a *LASER* sensor can result in a better performance than a *POINT-CLOUD*. The system controller should be improved because the fuzzy controller proved to be extremely effective in helping to displace the robots in the same direction, but when placed in opposite directions, the robots, the fuzzy controller was not able to correct the trajectory and avoid small collisions between robots. Testing with real robots in laboratories has already begun for a better observation of the behavior and results achieved, however more experiments are still lacking for more reliable conclusions. It is expected that in the next work tests in laboratories and in the industrial environment will be realized.

## References

1. Arvin, F., Krajník, T., Turgut, A.E., Yue, S.: CosΦ: artificial pheromone system for robotic swarms research. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 407–412, September 2015
2. Bayindir, L.: A review of swarm robotics tasks. Neurocomputing **172**, 292–321 (2016). <http://www.sciencedirect.com/science/article/pii/S0925231215010486>
3. Ben-Jacob, E.: Social behavior of bacteria: from physics to complex organization. Eur. Phys. J. B **65**(3), 315–322 (2008). <https://doi.org/10.1140/epjb/e2008-0022-x>
4. Eshel, B.J., Herbert, L.: Self-engineering capabilities of bacteria. J. R. Soc. Interface **3**(6), 197–214 (2006). <https://doi.org/10.1098/rsif.2005.0089>
5. Farinelli, A., Boscolo, N., Zanotto, E., Pagello, E.: Advanced approaches for multi-robot coordination in logistic scenarios. Robot. Auton. Syst. **textbf{90}**, 34 – 44 (2017). <http://www.sciencedirect.com/science/article/pii/S092188901630447X>, special Issue on New Research Frontiers for Intelligent Autonomous Systems
6. Johnson, J., Jesu Godwin, D.: Indoor navigation of mobile robot using fuzzy logic controller. In: 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1–7, March 2015

7. Junior, L.S., Nedjah, N.: Efficient strategy for collective navigation control in swarm robotics. *Procedia Comput. Sci.* **80**, 814–823 (2016). <http://www.sciencedirect.com/science/article/pii/S1877050916308468>, International Conference on Computational Science, ICCS 2016, 6–8 June 2016, San Diego, California, USA (2016)
8. Kolling, A., Walker, P., Chakraborty, N., Sycara, K., Lewis, M.: Human interaction with robot swarms: a survey. *IEEE Trans. Hum.-Mach. Syst.* **46**(1), 9–26 (2016)
9. Lee, C.: Development of an industrial Internet of Things (IIoT) based smart robotic warehouse management system. In: CONF-IRM 2018 (2018). <https://aisel.aisnet.org/confirm2018/43>
10. Li, S., Batra, R., Brown, D., Chang, H.D., Ranganathan, N., Hoberman, C., Rus, D., Lipson, H.: Particle robotics based on statistical mechanics of loosely coupled components. *Nature* **567**, 361–365 (2019)
11. Mariethoz, S., Almer, S., Baja, M., Beccuti, A.G., Patino, D., Wernrud, A., Buisson, J., Cormerais, H., Geyer, T., Fujioka, H., Jonsson, U.T., Kao, C., Morari, M., Papafotiou, G., Rantzer, A., Riedinger, P.: Comparison of hybrid control techniques for buck and boost DC-DC converters. *IEEE Trans. Control Syst. Technol.* **18**(5), 1126–1145 (2010)
12. de Sá, A.O., Nedjah, N., de Macedo Mourelle, L., dos Santos Coelho, L.: Multi-hop localization method based on tribes algorithm. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A., Torre, C.M., Taniar, D., Apduhan, B.O., Stankova, E., Wang, S. (eds.) *Computational Science and Its Applications - ICCSA 2016*, pp. 156–170. Springer, Cham (2016)
13. Wen, J., He, L., Zhu, F.: Swarm robotics control and communications: imminent challenges for next generation smart logistics. *IEEE Commun. Mag.* **56**(7), 102–107 (2018)
14. Wu, Y., Teng, M., Tsai, Y.: Robot docking station for automatic battery exchanging and charging. In: 2008 IEEE International Conference on Robotics and Biomimetics, pp. 1043–1046, February 2009
15. Yang, B., Ding, Y., Jin, Y., Hao, K.: Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis. *Robot. Auton. Syst.* **72**, 83–92 (2015). <http://www.sciencedirect.com/science/article/pii/S0921889015000937>
16. Zhong, Y., Lin, J., Ning, J., Lin, X.: Hybrid artificial bee colony algorithm with chemotaxis behavior of bacterial foraging optimization algorithm. In: 2011 Seventh International Conference on Natural Computation, vol. 2, pp. 1171–1174, July 2011
17. Durović, P., Grbić, R., Cupec, R., Filko, D.: Low cost robot arm with visual guided positioning. In: 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1120–1125, May 2017



# Positioning System for UAV Precision Tasks Near Walls in GPS Denied and Metallic Environments

Felix Orjales<sup>1</sup> , Javier Losada-Pita<sup>2</sup> , Alejandro Paz-Lopez<sup>3</sup> , and Alvaro Deibe<sup>1</sup>

<sup>1</sup> Integrated Group for Engineering Research, Universidade da Coruña,  
A Coruña, Spain

[felix.orjales@udc.es](mailto:felix.orjales@udc.es)

<sup>2</sup> Navantia-UDC Joint Research Unit, Universidade da Coruña, A Coruña, Spain

<sup>3</sup> CITIC Research Center, Universidade da Coruña, A Coruña, Spain

**Abstract.** A positioning system based on the combination of various measurements is being developed in order to allow precise UAV navigation on GPS denied environments. In the case of study, the task is developed near walls in a closed fully metallic environment with nearly homogeneous floor and walls. This led up to some interesting challenges. The first one regarding RF signal transmission on a metallic environment. Secondly, the difficulty of using SLAM or other computer vision navigation solutions due to the homogeneity of the walls and floor. Finally, a working place located near walls, where positioning systems tend to offer worst accuracy. To overcome these challenges, the proposed positioning method combines an Ultra Wide Band (UWB) based system, for global location, with unidirectional laser range finders, for very precise near-wall measurements. Moreover, RF transmission on various frequencies and results of the proposed positioning precision experiments performed are shown and results analyzed.

**Keywords:** UAV · GPS denied environments · Positioning · IPS · Ultra Wide Band · LIDAR

## 1 Introduction

UAV have become widespread in recent years due to their versatility and the improvements and new developments regarding batteries, control systems and computing power. This increase in popularity has favored an important improvement of their capabilities and subsequently their usage.

Positioning systems have also been improving at a similar pace. Nowadays, outdoors applications can benefit from the availability of satellite based positioning systems (e.g. GPS, GALILEO, GLONASS or BEIDOU) using new lightweight and improved refresh rate receivers. This kind of positioning systems offer a variable accuracy depending on the number of visible satellites but a few meters of error are expected [1] when enough satellites are on view. This error can be reduced combining various satellite constellations [2], or using enhanced systems like Differential GPS

(DGPS), that employs a network of fixed stations with known positions that broadcast local corrections. Other methods such as visual, inertial or radio-frequency based solutions [3], as well as barometers to measure the height can be used to improve positioning accuracy even further.

Indoor applications, on the other hand, have to rely on other positioning methods, as the satellite signal cannot reach there. Added to this, the confined and limited workspace imply, in general, tighter tolerances for flight plans and the need for higher accuracy in order to avoid obstacle collision. This leads to the need for better positioning systems and more precise navigation control for the UAV to operate with the desired level of safety.

There are several wireless technologies that can be employed to measure the position of a UAV flying indoors. However, despite the technology they are based on, there are at least two hardware components involved: a signal transmitter and a measuring unit, that is responsible for the position calculation. In general, one of the two parts remain on a fixed and known position and the other is attached to the vehicle. The main characteristics that define Indoor Positioning Systems (IPS) are two: the method they use to estimate the position and the physical layer employed.

Accuracy and refresh rate requirements, among others, should be used to select an appropriate IPS for a particular task. For the selection phase, the IPS can be considered from two perspectives, the measurement approach and the positioning algorithm. There are multiple approaches to measure distances, generally divided into time-based, power-based and angle-based [4]. The latter is not commonly used in IPS, as it requires additional specialized hardware. Power-based estimations such as Received Signal Strength Indicator (RSSI), that measure signal strength on the receiver side of an RF signal, attenuate with the distance travelled. Moreover, these measurements tend to be very sensitive to multipath propagation, and translation into distance tend to be inaccurate and environment dependent [5]. Within those based in time, Time of Arrival (ToA) is one of the more well-known, and it is based on the measurement of the time elapsed by the signal to reach the receiver from the transmitter. There are numerous variants based on ToA that differ on the way to synchronize transmitter and receiver and others that combine it with other techniques to improve accuracy [6].

The algorithm accuracy tends to be in line with the time employed to calculate the position, which implies that higher accuracy leads to longer times or more requirements for computing power. This usually translates into a decrement of the refresh rate of position information or an increase of computing power and, subsequently, weight and power consumption, of the measurement unit.

Regarding the physical layer used for location, IPS can be based among others on light (visible, or invisible like infrared), sound (audible and ultrasonic), magnetic field (natural and artificially generated) and RF (Bluetooth, Wi-Fi, UWB, etc.). In [7] and more recently [8] a survey of the most relevant IPS available have been compiled with a classification and a comparison of their main characteristics. The weight that a UAV can carry also need to be considered, but the environment and task requirements play the most significant role in the IPS technology selection.

## 1.1 Environment Requirements

One of the defining factors for the IPS is the working environment, as it has significant impact over the method performance and UAV requirements. In this regard, the presented work will be carried out within a ship. This introduces some constraints for the UAV itself such as size limitation caused by the hatch openings to access the operating spaces.

However, one constraint that stands out among the others is the building material. Ships are mainly build using steel, so every compartment within them is the equivalent of a giant metallic cage. This results in distorted magnetic earth field, which leads to errors with the interpretation of signals from the magnetic part of the IMU, inducing position and orientation measurement errors. It can also generate rebounds, multipath problems, etc. affecting the wireless signal commonly used with UAV and also the IPS wireless signal for measurement or communication.

## 1.2 Task Requirements

The IPS developed should adapt to the working environment as well as to the desired task. The presented method focuses on UAV works near walls, like inspection or thickness measurement, that requires very good accuracy on the IPS, a suitable and specifically designed UAV and a control system capable of maintaining the position dealing with the disturbances generated by the UAV near the wall.

As been said, IPS generally tend to offer better accuracy on the central part of the room and worsen approaching the walls [9, 10]. This is usually caused by signal loss, rebounds or other distortions on the wireless transmitted signal. All these effects, coupled with high required accuracy to avoid collisions while flying the UAV close to the walls, generates a challenge for the IPS and the UAV that the presented approach will try to overcome.

## 1.3 Related Work

A lot of work has been done developing IPS in the recent years, with proliferation of wireless network infrastructure on buildings. This ease the implementation of the positioning systems and reduces cost. However, most of them are focused on positioning people or objects [11]. These approaches do not offer the required accuracy, refresh rate and low delay required to perform an UAV flight relying on them.

Works focused on UAV IPS are mainly based on UWB positioning systems [12], since it does not affect other wireless links. UWB uses short pulses over a wide spectrum and it can be fast enough for UAV locating purposes. Vision-based solutions employing LIDAR [13] or laser beams [14] are also popular and generally combined with inertial systems [15] or UWB [16] to increase accuracy and reliability of the positions obtained.

Regarding the UAV platform there are some UAV specifically designed or adapted for inspection [17], and other wall contact specific tasks [18]. Some of them are even capable of wall climbing [19] or rotor tilting to access difficult to reach spots [20].

## 2 Proposed Solution

In this section a detailed description of our proposed solution will be made. Most relevant characteristics of the positioning system and UAV will be explained. Moreover, the foundations behind the component selection and the development of the positioning system are explained.

### 2.1 UAV Platform

In order to test the proposed positioning system, and to be able to develop real tests of precision tasks near walls, a UAV was selected, instrumented and improved for our purposes. The platform selected is shown in Fig. 1 and the reasons behind its main characteristics selection will be explained below.



**Fig. 1.** Octocopter selected fully extended and folded for transportation

With safety in mind we opted for an octocopter, since with this configuration, adequate motor sizing and a well-tuned controller, the UAV is able to overcome a single rotor failure and land safely. To be able access and transport the UAV to the working places in the ships innards, we have selected a foldable structure that can easily be carried through the hatches.

The total dimensions of the folded UAV are 1070 mm long, 455 mm wide and 145 mm tall and it offers a standard 155 mm rail for attaching the required extra equipment. The propellers are rotated by brushless electric motors, and are foldable too in order to further reduce total size of folded UAV. The propellers being foldable also help with the safety of the platform in case of an incident. In the current configuration this platform can carry up to 5 kg of additional load for the intended task.

**On Board Equipment.** A Pixhawk autopilot was selected for low level control of the UAV since it is open source and this allows the extension of their functionalities to better suit the proposed task needs. Radio link for manual control between RC controller and vehicle is based on digital 2.4 GHz commercial equipment. For inspection and control purposes a camera and a real time video transmitter are used, with a working radio-frequency that will be selected based on results from tests on the real working environment performed with commercially available alternatives.

High level control of the UAV for more complex autonomous behaviors and positioning system calculations are performed in an Intel NUC i5 mini pc, attached to the underside of the vehicle. To power the propulsion system, the mini PC and other platform subsystems, a battery made up of two 6 cell, 8 Ah lithium polymer batteries, connected in parallel is used.

The rest of the equipment to be carried on board will be determined by the specific requirements for the intended task. In our case, for thickness measurement, purposes a retractable carbon fiber stick is mounted with a custom 3D printed support to the 155 mm rails. This is made in order to allow the stick to touch the wall despite small movements of the UAV and it demonstrates the possibility of carrying a tool like an ultrasonic thickness measurement probe.

## 2.2 Positioning System

In order to fulfill all the requirements previously explained and to perform autonomously precision tasks near walls, a high accuracy positioning system is needed. Due to the early mentioned task and working environment constraints, the proposed solution is based on the combination of UWB and punctual LIDAR sensors, as will be explained next.

**UWB Based IPS.** There are several commercially available UWB indoor positioning systems. After a careful study of alternatives, the Pozyx system was selected. This IPS consists on various anchors that are located on known predefined positions and serve as base for the positioning of the moving tags.

Employing a Two Way Ranging (TWR) technique, distances between tag and anchors are calculated, and tag location determined by triangulation with them. TWR offers an advantage over TDoA, as it does not need synchronization between tag and

anchor. On the other hand, the time consumed is higher and this leads to a limitation on the number of tags that can be used on a system simultaneously.

At least three anchors are required to obtain a position from this UWB system, and four is the minimum number recommended by the manufacturer. A large number of anchors can offer better results, especially when some of them are shaded out due to interference with other objects (UAV frame, people, etc.) but this also increases the complexity of the triangulation. Up to eight anchors are allowed in this system. The anchors should be in line of sight with the tag for optimal performance. The recommended setup is with anchors around and the tag in the middle. Expected accuracy is between 50–150 mm, with less accurate measurements near walls.

**LIDAR Based IPS.** Because of UWB technology limitations near walls, a complementary positioning system is required. Taking into account that the critical measurement to guarantee a safe operation is the distance to the wall, a punctual LIDAR based system was developed. Three LIDAR sensors were arranged, one of them horizontal and facing forward on the UAV. The other two are symmetrical, with a horizontal angle of 30° from the central one and pointing down with an angle of 21°.

**Calculating Distance and Angle to the Wall.** To be able to perform precision tasks near the wall, angle and distance information between the UAV and the wall need to be determined with very good accuracy and low latency.

Distance information can be obtained from both positioning systems but with different performances. UWB offers worse position estimations when the UAV is close to walls, LIDAR based one works better near walls, and has higher refresh rate. UWB accuracy can be challenged by anchor shading, and LIDAR only works in a constrained orientation between UAV and wall. These redundant but complementary behaviors can be merged to increase the accuracy over the whole working area.

Regarding the angular heading information, this can generally be calculated outdoors using the magnetometer of an IMU. Metallic walls of the ship generate distortions on the earth magnetic field, causing orientation errors on such IMU-based estimation. The proposed LIDAR array offer a good alternative for this estimation.

*Distance Estimation Using UWB.* UWB reference system is used for all the calculations as it serves as the base for the developed enhanced IPS. Walls on interior ship locations are generally flat surfaces. Assuming a wall as a plane that cuts the coordinate axis on points  $a$  on  $x$  axis,  $b$  on  $y$  and  $c$  on  $z$ , it can be defined by the expression:

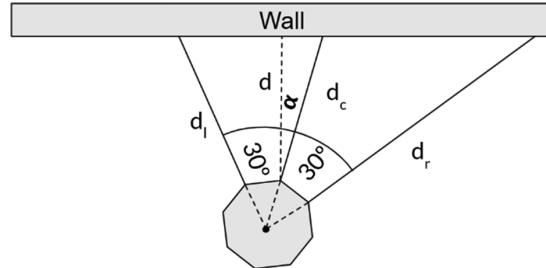
$$\frac{1}{a} x + \frac{1}{b} y + \frac{1}{c} z = 1 \quad (1)$$

Then the distance  $d$  between wall and UAV can be calculated as:

$$d = \vec{P} \cdot \vec{n} - D \quad (2)$$

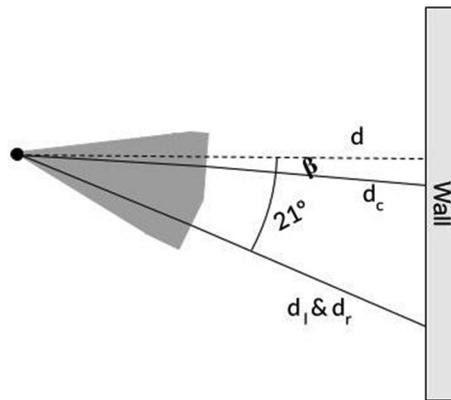
being  $P$  the UWB based, UAV position vector,  $n$  the unitary normal vector to the wall, and  $D$  the distance from the wall plane to the origin.

*Distance Estimation Using LIDAR.* The LIDAR array is designed so all the directions converge in a point, exactly where the antenna of the UWB is located, as can be seen in Fig. 2.



**Fig. 2.** Top view of the LIDAR and horizontal angle  $\alpha$  between UAV and wall

In order to simplify calculations, some reasoned assumptions are made. The first one is related to the wall: we defined it as generic plane previously and from this point on we will assume that it is vertical, as it is the case in most scenarios. The second assumption will be that pitch angle between the wall and the UAV ( $\beta$  in Fig. 3) is zero: during normal operation the pitch angle of the UAV should be very small as every variation of orientation from the horizontal is translated into a displacement; separation between the wall and UAV will be small too.



**Fig. 3.** Lateral view of the LIDAR and vertical angle  $\beta$  between UAV and wall

We can formulate an equation system to obtain distance  $d$  and angle  $\alpha$  between UAV and wall, using measurements obtained from the LIDAR sensors ( $d_c$ ,  $d_l$  and  $d_r$ ):

$$d = d_c \cos \alpha \quad (3)$$

$$d = d_l \cos (30 - \alpha) \quad (4)$$

$$d = d_r \cos(30 + \alpha) \quad (5)$$

This is a nonlinear, overdetermined equation system. An iterative approach can be used to solve this system, in order to minimize a previously defined error vector. In the proposed method, however, an approximate solution to this equation system is obtained, directly solving  $\alpha$  from the last two equations:

$$\alpha = \tan^{-1} \left( \frac{d_l - d_r}{d_l + d_r} \right) \quad (6)$$

and calculating  $d$  from  $d_c$  and  $\alpha$  on the first equation.

Using these equations, the measurements from both positioning systems can be compared and yaw angle between wall and UAV obtained. This angle cannot be calculated using the IMU due to the magnetic distortions of the working environment.

### 3 Testing and Results

In this section a series of test will be described and the results analyzed in order to evaluate the performance of the proposed solution.

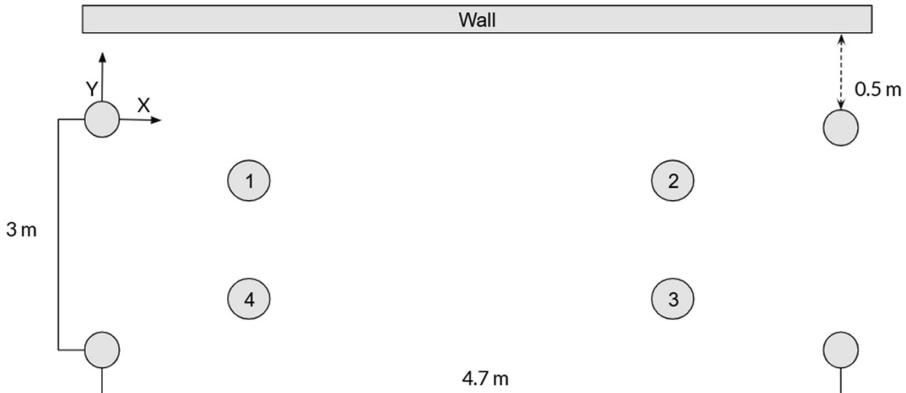
#### 3.1 Test Description

Three tests were designed to evaluate possible problems or interferences between the multiple wireless links used, to analyze the accuracy of the UWB system, and to compare UWB and LIDAR based positioning systems working on near wall tasks.

**RF Test.** The first of the tests involved the three wireless links usually used in a UAV: control, telemetry and video. A measurement of signal strength will be performed in various points of the working environment: a fully closed ship compartment. A collection of commercially available frequencies has been chosen to determine the best working ones on this environment.

This test was performed prior to any flight to detect possible problems with the wireless links, and to guarantee a safe operation where the pilot can take manual control at any time.

**Static Positioning Accuracy Test.** The second test was performed to evaluate the real accuracy obtained using the UWB positioning system on the real working environment. This test consisted of four independent measurements took in the working area as shown in Fig. 4. Four anchors were used on the corners of the room.



**Fig. 4.** Anchor position and measurement points for static positioning accuracy test

**Wall Distance Approaching Test.** This test was performed to evaluate the differences between both positioning systems and the accuracy obtained on the measurements of the distance between the wall and the UAV.

It consists of an approximation movement approaching the wall at a constant height with the LIDAR array parallel to the ground, a waiting period and a return to the original position. This reproduces the sequence of movements to perform an ultrasonic thickness measurement.

### 3.2 Results

**RF Test.** In the first place, the RC control link was tested on different points of the working environment. The RSSI of the signal was good, over 80% during all the test process including the corners and the control signals transmitted without problems.

After that, the telemetry link to the ground station was tested in the same fashion, there were some dropped packets on the more distant positions but the connection was stable and reliable.

Finally, the real time video link test was performed using three different RF bands and various antennae types, including circularly polarized ones, that help rejecting multipath signals. While RSSI seemed right, video quality was poor with noise and interruptions. Even using the less powerful transmitter with the higher frequency the results were far from good due to the rebounds affecting the analog video signal.

In future tests, both a digital video link and an analog transmitter with programmable output power will be tested to evaluate their performance on this environment.

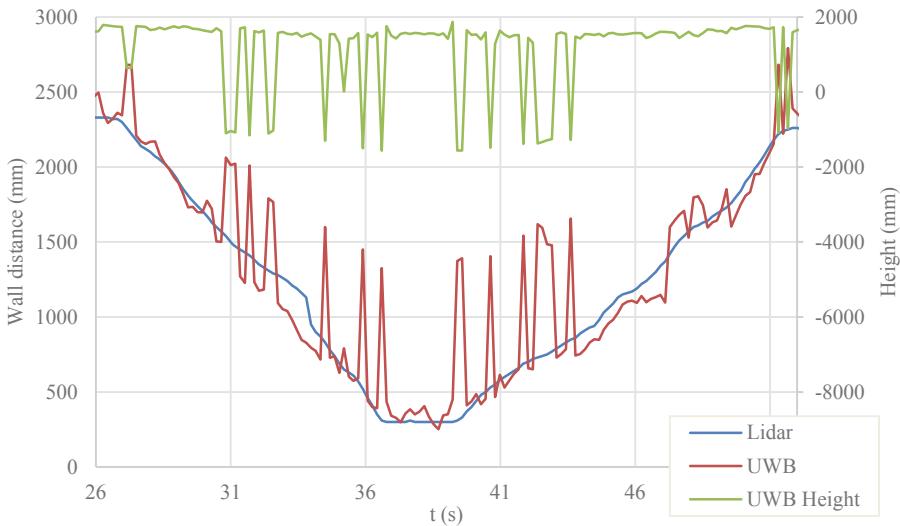
**Static Positioning Test.** The anchors were situated precisely on the designed position. After powering up all of them, the tag was connected and situated on the defined position for at least thirty seconds. Position estimation results are shown in Table 1.

**Table 1.** Static positioning accuracy results (measurements in mm)

Number	Real position	Estimated position	Mean error	Standard deviation
1	(1000, -1000)	(1004.83, -981.93)	45.76	29.18
2	(3000, -1000)	(3108.70, -778.32)	304.82	130.43
3	(3000, -2000)	(3018.41, -1982.75)	39.35	33.75
4	(1000, -2000)	(840.55, -1899.88)	195.35	54.28

The estimated positions have sometimes an error bigger than 150 mm, which is the error margin specified by the manufacturer. This error could be caused by shading one or more anchors. If that is the case it could be compensated by adding more anchors in the surroundings. Despite this, in points 1 and 3 although with good and stable position measure, the accuracy is not enough to perform an approximation maneuver to a wall.

**Wall Distance Approaching Test.** This last test was performed just after the static position, using the same anchor configuration and the results are shown on Fig. 5.

**Fig. 5.** Distance to the wall measured by UWB and LIDAR

Three signals are presented: the wall distance calculated using the described method for the UWB and the LIDAR and the UWB height estimation. At first glance, it can be seen that the UWB measurements present some errors with points where the height signal goes to negative values instantly.

This could be caused by triangulation problems when one of the anchors is obstructed by the UAV frame or due to rebounds or artifacts caused by the walls. These

errors could be filtered out, however finding the true cause of the problem could improve the end results for the future work with this positioning system.

## 4 Conclusions

In this paper a positioning system for the use of UAV near walls in GPS denied and metallic environments is presented. The system proposed relies on the combination of an UWB for global localization and a LIDAR array for precision near-wall location.

Real tests were performed on the working environment, a metal compartment inside a ship, in order to evaluate the performance of the wireless links, the accuracy of the positioning system and its robustness. Some problems were detected with real time analog video that potentially could be solved using an analog transmitter with less output power to reduce the rebounds or a digital video link. The rest of the wireless communications offered good results.

Regarding the positioning system, the static tests of the UWB were less accurate than expected and the moving tests near the wall suffered from errors that need to be solved before attempting autonomous missions. More tests will be performed with a higher number of anchors to eliminate the possibility of a triangulation error if something, like the UAV frame, obscure one of them.

## References

1. Wing, M.G., Eklund, A., Kellogg, L.D.: Consumer-grade global positioning system (GPS) accuracy and reliability. *J. For.* **103**, 169–173 (2018). <https://doi.org/10.1093/jof/103.4.169>
2. Li, X., Ge, M., Dai, X., Ren, X., Fritzsche, M., Wickert, J., Schuh, H.: Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. *J. Geod.* **89**, 607–635 (2015). <https://doi.org/10.1007/s00190-015-0802-8>
3. Gross, J.N., Gu, Y., Rhudy, M.B.: Robust UAV relative navigation with DGPS, INS, and peer-to-peer radio ranging. *IEEE Trans. Autom. Sci. Eng.* **12**, 935–944 (2015). <https://doi.org/10.1109/TASE.2014.2383357>
4. Dwiyasa, F., Lim, M.H.: A survey of problems and approaches in wireless-based indoor positioning. In: 2016 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2016, pp. 4–7 (2016). <https://doi.org/10.1109/IPIN.2016.7743591>
5. Wu, Z.H., Han, Y., Chen, Y., Liu, K.J.R.: A time-reversal paradigm for indoor positioning system. *IEEE Trans. Veh. Technol.* **64**, 1331–1339 (2015). <https://doi.org/10.1109/TVT.2015.2397437>
6. Gueuning, F.E., Varlan, M., Eugène, C.E., Dupuis, P.: Accurate distance measurement by an autonomous ultrasonic system combining time-of-flight and phase-shift methods. *IEEE Trans. Instrum. Meas.* **46**, 1236–1240 (1997). <https://doi.org/10.1109/19.668260>
7. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **37**, 1067–1080 (2007). <https://doi.org/10.1109/TSMCC.2007.905750>
8. Brena, R.F., García-Vázquez, J.P., Galván-Tejada, C.E., Muñoz-Rodríguez, D., Vargas-Rosales, C., Fangmeyer, J.: Evolution of indoor positioning technologies: a survey. *J. Sens.* **2017** (2017). <https://doi.org/10.1155/2017/2630413>

9. Retscher, G., Moser, E., Vredeveld, D., Heberling, D.: Performance and accuracy test of the WLAN indoor positioning system “ipos” principle of WLAN positioning. In: Presenting 3rd Workshop on Positioning, Navigation and Communication, WPNC 2006, University Hannover, Germany, pp. 7–16 (2006)
10. Faragher, R., Faragher, R., Harle, R.: An analysis of the accuracy of bluetooth low energy for indoor positioning applications (2009)
11. Zhang, D., Xia, F., Yang, Z., Yao, L., Zhao, W.: Localization technologies for indoor human tracking. In: Proceedings of the 2010 5th International Conference on Future Information Technology, Future 2010, pp. 1–6 (2010). <https://doi.org/10.1109/FUTURETECH.2010.5482731>
12. Shi, G., Ming, Y.: Survey of indoor positioning systems based on ultra-wideband (UWB) technology, vol. 348, pp. 1269–1278 (2016). <https://doi.org/10.1007/978-81-322-2580-5>
13. Li, R., Liu, J., Zhang, L., Hang, Y.: LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments. In: Proceedings of the 2014 DGON Inertial Sensors and Systems, ISS 2014, pp. 1–15 (2014). <https://doi.org/10.1109/InertialSensors.2014.7049479>
14. Kara Mohamed, M., Patra, S., Lanzon, A.: Designing simple indoor navigation system for UAVs. In: 2011 19th Mediterranean Conference on Control & Automation, MED 2011, pp. 1223–1228 (2011). <https://doi.org/10.1109/MED.2011.5983054>
15. García Carrillo, L.R., Dzul López, A.E., Lozano, R., Pégard, C.: Combining stereo vision and inertial navigation system for a quad-rotor UAV. *J. Intell. Robot. Syst. Theory Appl.* **65**, 373–387 (2012). <https://doi.org/10.1007/s10846-011-9571-7>
16. Benini, A., Mancini, A., Longhi, S.: An IMU/UWB/vision-based extended kalman filter for mini-UAV localization in indoor environment using 802.15.4a wireless sensor network. *J. Intell. Robot. Syst. Theory Appl.* **70**, 461–476 (2013). <https://doi.org/10.1007/s10846-012-9742-1>
17. Bodie, K., Brunner, M., Pantic, M., Walser, S., Pfändler, P., Angst, U., Siegwart, R., Nieto, J.: An omnidirectional aerial manipulation platform for contact-based inspection (2019)
18. Ikeda, T., Yasui, S., Fujihara, M., Ohara, K., Ashizawa, S., Ichikawa, A., Okino, A., Oomichi, T., Fukuda, T.: Wall contact by octo-rotor UAV with one DoF manipulator for bridge inspection. In: IEEE International Conference on Intelligent Robots and Systems, September 2017, pp. 5122–5127 (2017). <https://doi.org/10.1109/IROS.2017.8206398>
19. Myeong, W., Myung, H.: Development of a wall-climbing drone capable of vertical soft landing using a tilt-rotor mechanism. *IEEE Access* **7**, 4868–4879 (2019). <https://doi.org/10.1109/ACCESS.2018.2889686>
20. Kamel, M., Verling, S., Elkhatib, O., Sprecher, C., Wulkop, P., Taylor, Z., Siegwart, R., Gilitschenski, I.: Voliro: an omnidirectional hexacopter with tilttable rotors (2018). <https://doi.org/10.1109/MRA.2018.2866758>



# CRAS (Climbing Robot for Autonomous InSpection): The Challenges of a High-Temperature Tank

Nicolas Dalmedico<sup>1</sup>(✉), Higor Barbosa Santos<sup>1,2</sup>, Juliano Scholz Slongo<sup>1,2</sup>,  
Marco Antônio Simões Teixeira<sup>1,2</sup>, Piatan Sfair Palar<sup>1,2</sup>,  
Vinicius Vargas Terres<sup>1,2</sup>, André Schneider de Oliveira<sup>1,2</sup>,  
Lúcia Valéria Ramos de Arruda<sup>1,2</sup>, Flávio Neves Júnior<sup>1,2</sup>,  
and Julio Endress Ramos<sup>2</sup>

<sup>1</sup> Universidade Tecnológica Federal do Paraná (UTFPR),  
Curitiba, PR 80230-901, Brazil  
[ndalmedico@alunos.utfpr.edu.br](mailto:ndalmedico@alunos.utfpr.edu.br)

<sup>2</sup> CENPES/Petrobras, Rio de Janeiro, RJ 21941-915, Brazil  
<http://portal.utfpr.edu.br/>

**Abstract.** This paper introduces the mobile robot CRAS (Climbing Robot for Autonomous inSpection) for autonomous NDT inspection of weld beads from industrial super-duplex steel vessels. The surfaces to be inspected are under high temperature (80 °C–135 °C) and the inspection is based on ultrasound. CRAS presents magnetic wheels as adhesion method and a perception system able to identify and follow weld beads. This paper approaches some current challenges for such inspection mainly due high temperatures and adopted solutions as well as future steps of CRAS development.

**Keywords:** Autonomous mobile robot · High temperature · Magnetic adhesion · Weld inspection

## 1 Introduction

The integrity of oil storage tanks is an important concerning in the petrochemical industry. Failure to do regular maintenance procedures can result in disasters carrying environmental and material damage, as well as presenting risks for the workers.

Storage tanks must regularly undergo inner and outer inspection, regarding structural faults like cracks, corrosion, weld defects, and other types of material degradation need to be identified as quickly as possible to avoid catastrophic failures. The traditional method for internal inspection requires an empty and clean tank. This imply to temporarily transfer all the product volume inside the tank to be inspected to another tank. Residues must also be removed. Water is used to wash the tanks for better sanitation. Then, exhausts are used to remove

gases that may remain inside the tanks. All these precautions are needed to protect the people inspecting the tank so that they do not breathe chemicals that may bring harm, as can be seen in [14].

Inspection in storage tanks is usually done through ultrasonic Non-Destructive Test (NDT). Ultrasonic inspection means emitting sound waves in high frequency through a material to detect the echo coming back from changes of medium. Concerning weld inspection, the echos may come from flaws in the weld bead like cracks, porosity or lack of fusion. The ultrasound monitoring equipment can be calibrated assuring all defects can be pinpointed and dimensioned. The inspection using phased-array takes a step further using more than one source of ultrasound arranged in a line or matrix. The emitted waves compose a wavefront that can be directed towards a specific angle, and the resulting echo allows the generation of an image representing the section of weld bead inspected. For mobile inspection robots, the ultrasonic hardware is usually attached at the bottom of the robot, lowering the wedge and transducer to the weld when examining. More about NDT inspection can be found at [1]. Ultrasound signals are highly dependant on the medium due to the acoustic impedance. Changes in temperature, material, density, and wave frequency may affect the velocity of sound. A temperature gradient in the material may even direct the sound wave away from a flaw. Thus, ultrasound inspection under high temperature conditions impose several challenges to autonomous robotic inspection. In this paper, high-temperature is defined by the range of temperature between 80 °C and 135 °C. In the literature, this range may be defined as up to 300 °C.

The use and relevance of mobile robots have been growing in areas where there is human labor in dangerous or unhealthy environments. In the inspection of spherical tanks of LPG (Liquefied Petroleum Gas), the inspector is exposed to toxic substances in a highly explosive environment. Therefore, in addition to improving the quality of the inspection process, inspection robots make the inspectors' work a lot safer. The use of robots for inspection also makes work more efficient and reduces operational costs, since it minimizes the exposure of workers in hazardous or difficult to reach environments. Inspection robots, according to [10], can be classified by their mechanism of adhesion and locomotion. As for the adhesion mechanism, we can mention: magnetic, pneumatic, mechanical, electrostatic, and chemical. As for the type of movement: arms/legs, wheels, guides, and tracks.

Several inspection robots have been addressed in academy and industry, as in [2], where a magnetic climber robot is used to inspect marine vessels. Due to the size and carrying capacity of the robot, the inspection is carried out by a camera that sends information wirelessly to a control station. In addition to performing image processing, the station control commands the entire movement of the inspection robot. This type of solution that presents only the video inspection is an useless solution for oil tanks.

In order to improve the maneuverability and mechanical stability of the inspection robots, an omnidirectional climber robot, the omniclimber, for inspection of flat or convex ferromagnetic structures is proposed in [12]. The omni-

climber has excellent maneuverability and adaptability for various structures with different curvature and thickness, even with omnidirectional magnetic wheels. The only problem is the load capacity that the robot can undergo. In a test made with the omniclimber, in a structure of thickness 3 mm, it is capable of carrying only 370 g. This load capacity must be shared with the inspection hardware and other possible sensors for localization and navigation.

Inspection robots need to drive over the structure of focus. How good it performs this job depends on its degree of mobility. According to [11], the degree of mobility is the freedom of movement of a robot. Therefore, mobility is characterized by two factors: the type of wheels and their arrangement on a mechanical topology. The degree of drivability is the number of steerable center wheels that can be moved independently to move a robot. In short, the degree of maneuverability represents the degrees of freedom of a robot, that is, the ability to freely position the Instantaneous Center of Rotation (ICR). The ICR is the intersection of the wheel's axes of a robot and represents the radius of the curvature of the movement to be performed according to the arrangement of the wheels.

LASCA (Laboratory of Automation and Advanced Control Systems), from Federal University of Technology - Paraná (UTFPR), has developed a robot for ultrasonic inspections in oil storage tanks [5,9,13,15]. The first version of the project was called AIR (Autonomous Inspection Robot) back when the high-temperature inspection was not a requirement, and the robot's job was only to perform room temperature weld-bead inspection. Older papers are describing it refers to the project by this name. Its topology is aimed at improving maneuverability and magnetic adhesion.

The robot is currently on its second version, and another new and final version of the robot is envisioned. Now, the development received a new focus on the ultrasonic inspection of high-temperature vessels. This new project is the focus of this paper, and the new robot explicitly made for high-temperature inspection is called Climbing Robot for Autonomous inSpection (CRAS). High temperature changes the magnetic permeability of the metal, affecting the wheels' force of adhesion, which was countered by developing new wheels with stronger magnets. The heat also changes how the ultrasonic inspection behaves by refracting, attenuating and slowing the acoustic front wave. The inspection problem will be approached in the last section.

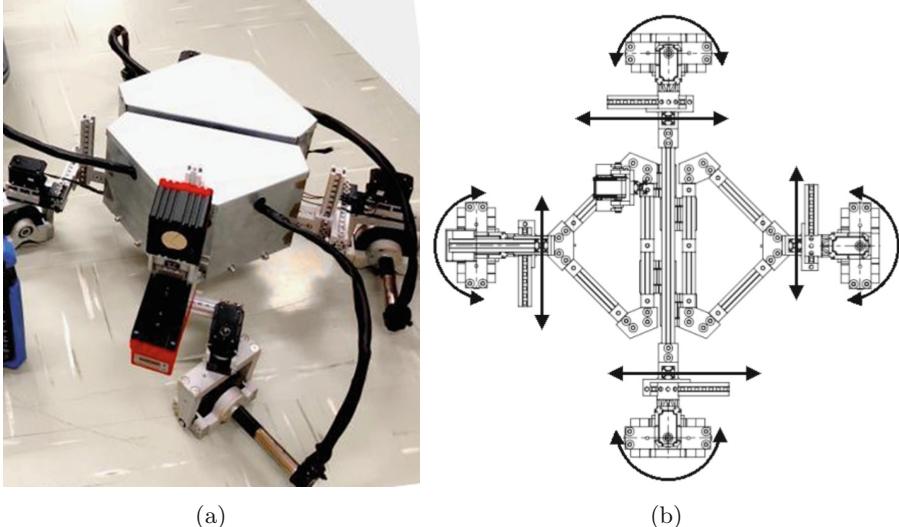
## 2 Robot Description

In this section, the Climbing Robot for Autonomous inSpection (CRAS), shown in Fig. 1a, will be presented. The CRAS is being developed at LASCA, which is a research laboratory in the areas of automation, control and system optimization located at UTFPR - Curitiba, Brazil.

### 2.1 Topology

CRAS is designed to have high maneuverability over metal surfaces, according to [7]. It is a climbing robot with four directional independent magnetic wheels.

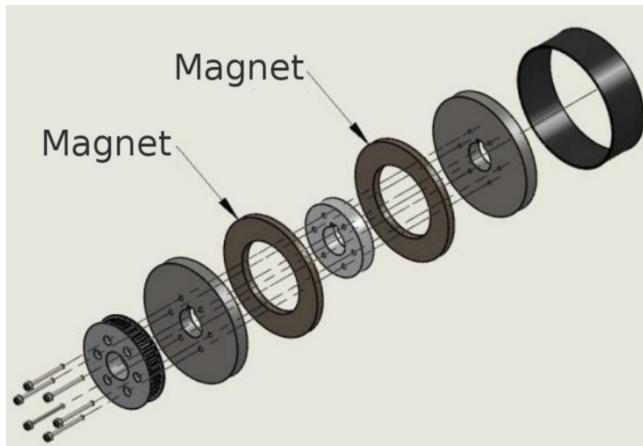
The CRAS has four wheels due to the safety standards required for tank inspection since there must be guarantees that the robot will not detach during its navigation. If the robot had only three wheels, there would be a reduction of the total adhesion force of the robot, which could cause a fall due to the possibility of some wheel detaching (for example, because of the decrease in the magnetic field of the wheel when going over a weld bead). The design of the magnetic wheels was proposed by [8]. The chassis in the form a bipartite diamond is responsible to provide suspension to the wheels by bending and following the curvature of the surface the robot is navigating. The CRAS topology is shown in Fig. 1b. Each robot's wheel has a brushless motor and a servo motor. The brushless motors are responsible for the movement of the robot, while the servo motors define the direction of movement. The robot's maximum velocity is 1 m/s, though it can be set as higher at the expense of torque.



**Fig. 1.** CRAS: a: robot and b: topology.

## 2.2 Adhesion System

The magnetic wheels are composed of two permanent annular magnets made of an alloy of neodymium, iron and boron (NdFeB). The magnets are positioned between two SAE 1020 steel discs and attached by screws (Fig. 2), where both the steel discs and the screws were chosen by their low magnetic permeability. The low permeability allows the magnetic flux to better travel the metal surface, ensuring a higher adhesion strength to the surface, reaching 40 kgf by wheel [7,8].



**Fig. 2.** Parts and assembly of the robot's magnetic wheel.

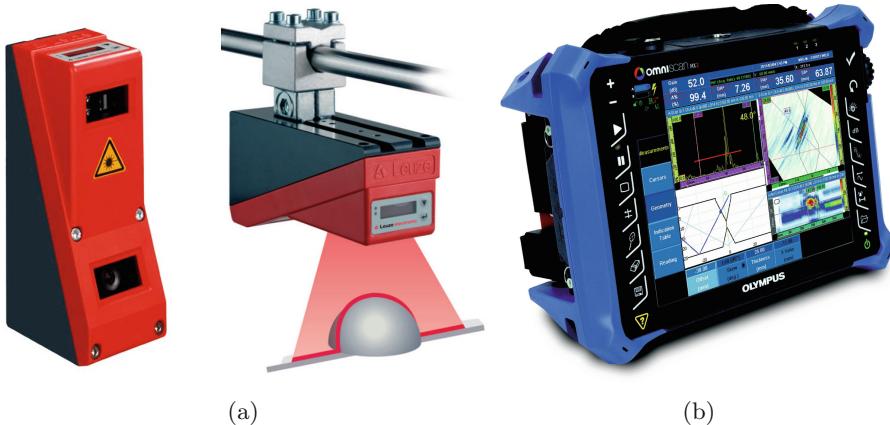
### 2.3 Perception System

The robot has several perception sources that are necessary to perform tasks such as mapping, path planning and odometry. The robot was planned as a full autonomous robot, but the operator can assume direct or assisted control as well. Direct control, or full control, means the robot will not assist the operator in any way. Assisted control means the robot will align itself with the weld bead but the operator can control velocity, direction of movement over the weld bead and decide which bead to follow when over an intersection. Autonomous mode means the robot will inspect and follow all the weld beads that are identified in the tank with no intervention required by the operator.

The robot can identify the weld bead as well as the center of the weld filament, taking into account details such as corrosion and wear of the weld. The purpose of identifying the weld bead is to make the robot, even when remotely operated, to be able to fully carry out the weld inspection, always driving in line with the center of the weld. The perception source capable of performing the identification of the weld bead is the LRS36/6 sensor, which is shown in Fig. 3a, according to [3].

This source of perception operates with a laser scan, performing several measurements and generating a cloud of 2D points (range and angle). From the collected data, it is possible to identify the weld bead, taking into account the difference in height between the weld and the rest of the tank.

The OmniScan MX2, shown in Fig. 3b, was the NDT (non-destructive testing) equipment chosen for inspection. It is widely used in many industrial sectors for composites inspection or welded plates in pressurized vessels and pipes of various diameters. Moreover, it can be combined with other components to form a complete inspection system.



**Fig. 3.** a: Laser scan sensor used to identify weld beads. b: Omniscan MX2 console.

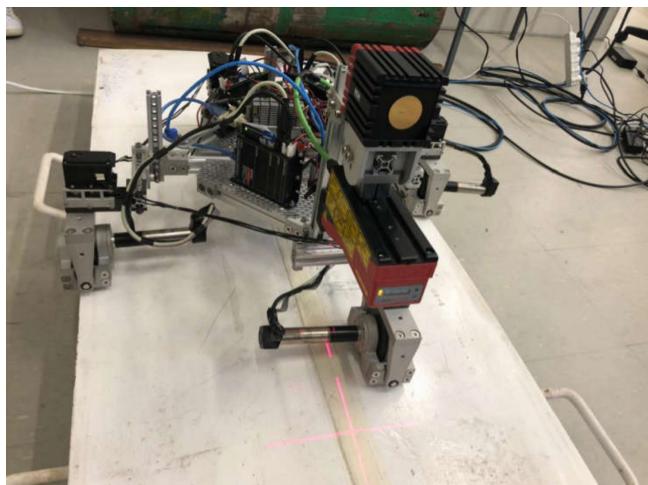
The Omniscan MX2 is a modular ultrasonic inspection platform with a simplified setup and calibration process for phased array ultrasonic testing [6]. The Olympus 5L64-A32 probe is a linear phased array transducer with a design optimized for weld inspection. It is typically applied in manual or automated inspections of welds up to 60 mm thick. Its transducer has 64 active elements, with 5 MHz frequency, arranged linearly. The wedge used in the tests (Olympus SA32C-ULT-N60L-IHC wedge) is made of a resin known as ULTEM, it is designed for testing at temperatures up to 150 °C and has a nominal 60° refractive angle on steel with longitudinal waves.

The final version of the robot will also have a 3D point cloud camera, such as the one fixed in front of the robot in Fig. 1a: a Mesa SR4000 time-of-flight 3D camera, as there are prediction and navigation algorithms explicitly developed for the robot, allowing it to predict the structure of the tank, it is inspecting by identifying weld bead junctions and consulting the tanks construction standards, as can be seen in [13]. 3D cameras also allow autonomous navigation and mapping algorithms to be applied.

### 3 Robot Validation

Experiments were carried out on a metal plate to validate the topology of the inspection robot, as can be seen in Fig. 4. A guide laser that was fixed in front of the robot is used to help the visual localization of the weld bead by the operator. Due to the robot and plate dimensions, the movement was restricted to moving forward or backward. Although the limited room, it can be observed that the magnetic adhesion was very effective, and the robot moved effortlessly despite the high adherence.

Based on the large load capacity of the magnetic adhesion system and the positive experiences by team members in the application of this type of adhesion



**Fig. 4.** Inspection and adhesion experiment on test plate.

system in past projects [4], it was decided its use on the next version of the robot. Another experiment was done on the lateral of an empty storage tank to assess the mobility and adhesion better when not on optimal position. The Fig. 5 shows the robot on the lateral surface as another validation experiment.



**Fig. 5.** Adhesion experiment on the lateral of an empty storage tank.

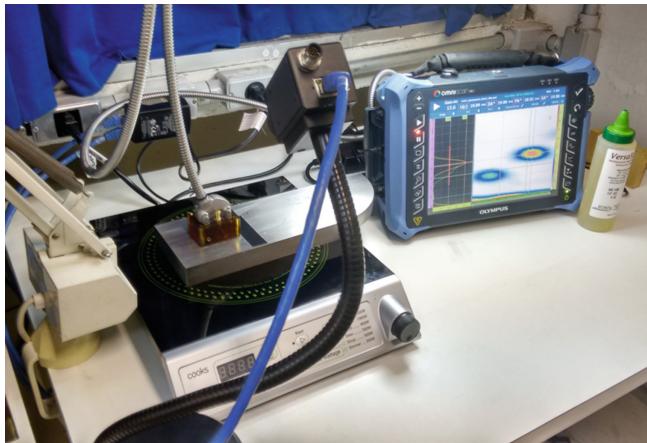
## 4 High Temperature Ultrasonic Inspection

The robot was initially designed to perform the ultrasonic inspection on general purpose steel LPG storage tanks. The new requirement of also attending the super-duplex steel tanks, which are constantly working at a temperature of 90 °C, brought many new challenges. One of them is finding out how the temperature of the super-duplex plates affects the ultrasonic readings. In this section, we will use 0 as a subscript for variables before heating,  $f$  as a subscript for values after heating, and an apostrophe for incorrect values. The equipment needs to know the propagation velocity of sound in the material to identify the echo and perform the correct sizing of the flaw. The velocity depends on acoustic impedance, frequency of the sound emitted, and temperature. The following experiment has the objective of finding the variation of propagation velocity inside a super-duplex steel plate (the material of the study object, a high-temperature super-duplex oil tank). Its acoustic impedance is already known in room temperature (27 °C), being it  $v_0 = 5773.7$  m/s. Therefore, variations on temperature for pulses of ultrasound of same frequency allow the calculation of the new propagation velocity on the 90 °C mark (the vessel's operating temperature), represented by the equipment by dilation of the measured plate thickness.

The experiment consisted of using a cooktop induction source to heat a super duplex steel plate made for ultrasonic equipment testing and calibration. The cooktop has 1200 W and a power regulator. The steel testing plate has a thickness of 25 mm at 27 °C. The experiment had monitored a plate hole with 5 mm deep, using a transducer attached to a wedge-shaped for high temperatures on the opposite side of the plate. The hole depth monitoring was carried out by an Olympus OmniScan, and the data was processed after the experiment using the Olympus OmniPC software version 4.4R4. A phased-array transducer was used attached to a flat wedge for perpendicular incidence so that the ultrasonic distortion is minimized. During the experiment, the wedge was fixed on top of the plate so that the movement would not affect the results. The coupling agent used in the interface between the wedge and plate was the VersaSonic, a coupling of general purpose made for high-temperatures (up to 398 °C).

A thermographic camera and a multimeter with a thermocouple were used to measure temperature during the experiment. The thermographic camera records the infrared radiation emitted from objects, which corresponds to their temperature and thermal emissivity. The camera used was a FLIR A35, capable of capturing images and recording videos in 60 frames per second with a resolution of  $320 \times 256$  pixels. The camera captures the infrared spectrum between 7.5 and 13  $\mu\text{m}$  and has reliable data when capturing objects with the temperature between -25 °C and 135 °C. Acquisitions made with the camera require a known emissivity to correct inform temperature. Like most metals, the super-duplex plate has a low thermal emissivity, approximately 0.35, a 3M scotch tape made for high temperature (up to 135 °C) was used, whose thermal emissivity is informed on its datasheet, being it 0.95. The tape was applied on a section of the plate to allow monitoring temperature on a surface with known emissiv-

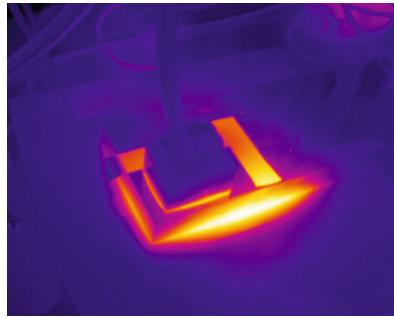
ity and low reflectivity. The software used for acquisition and analysis of data collected by the thermographic camera was the FLIR ReasearchIR. This software shows the image captured by the camera, defines regions of interest to be monitored, uses filters on the obtained data, performs camera calibration, and records videos. The thermocouple was kept in contact with the superior side of the test plate, close to the wedge. Its margin of error is 1° or 1% of the reading, up to 400 °C. An image of the experimental setup can be seen in Fig. 6.



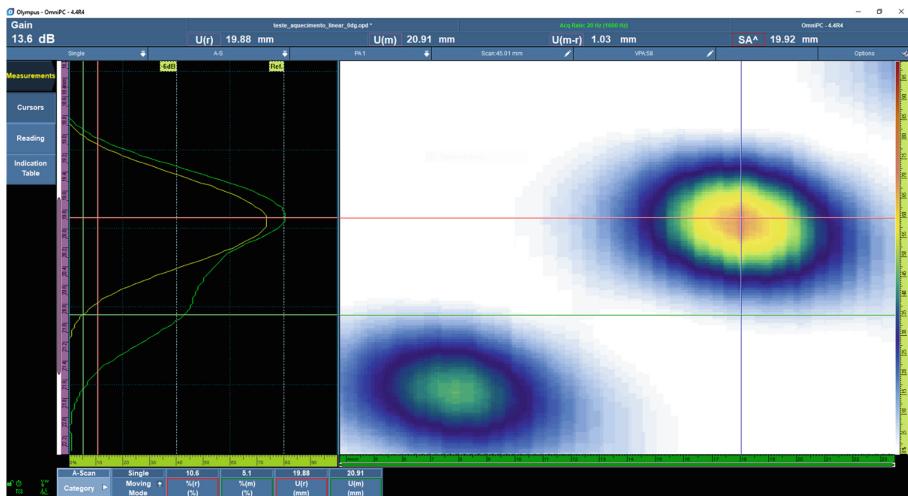
**Fig. 6.** Experiment with induction heater, super-duplex test plate, wedge and transducer, thermographic camera and Omniscan console.

Two experiments were done. The first was done as reference for the temperature showed by the thermal camera, where no ultrasonic acquisition was made. In this experiment, the plate was heated from 27 °C to 115 °C while its temperature was monitored by the thermal camera and the thermocouple, to find out how the data relates. This occurs because the thermal camera has several configurations to be done before a reading (reflective temperature, relative humidity, distance, emissivity) that the thermocouple does not. The second experiment was done heating the plate from 27 °C to 115 °C while performing an ultrasonic inspection, without moving the wedge, with the wedge initially in 27 °C. While the plate was heated, the values of depth for the hole and the total attenuation was observed. Figure 7 shows an image of the setup while being monitored by the thermographic camera.

After the tests, the ultrasonic data recorded was analyzed using the software Olympus Omnipc. In Fig. 8, it's shown the results of the second test in A-scans (left) and S-scans (right). In the A-scan, we can observe the envelopment (in green), representing the 58th shot (using Visual Probe Analysis) envelopment throughout the test. The reference cursor is positioned at the point of maximum amplitude of the envelopment at the initial sweeping time, while the measuring cursor is positioned at the point of maximum amplitude at the end of the experiment.



**Fig. 7.** One of the frames captured by the thermographic camera. Points close to yellow represents higher temperatures.



**Fig. 8.** Data analysis using the software OmniPC. Monitoring the position of flaw measured for temperature range 27 °C and 90 °C.

By verifying the information highlighted down at the reading bar in Fig. 8, it is noted that the flaw was initially positioned at  $p_0 = 19.88$  mm from the superior side of the plate, and by the end of the experiment, it is positioned at  $p'_f = 20.91$  mm. This represents an alteration in the depth perception of the flaw of  $\Delta p = 1.03$  mm. This alteration in the depth perception of the fault has two factors: the dilation of the material (super-duplex steel) and reduced acoustic velocity, both happening due to the temperature. To erase the dilation factor from the equation so that the acoustic speed can be calculated, the test plate's thickness was measured before and after heating using a pachymeter. The thickness before was  $th_0 = 25$  mm, rising to  $th = 25.15$  mm when the temperature was increased to 90 °C. The value of dilation allows finding the

shift of fault position, being it  $\Delta Fth = 0.1193$  mm, meaning the real position of the flaw is at  $p_{r0} = 19.9993$  mm from the superior face of the plate.

Knowing the real shift of the flaw by the dilation, it's now possible to find the erroneous assessment of position by the ultrasound, which relates to the difference in sound velocity. The portion of error corresponding to the difference in velocity is  $e = 0.9107$  mm. Through calibration, it was measured that the acoustic velocity in the super duplex plate, at  $27^\circ\text{C}$ , is  $v_0 = 5773.7$  m/s. To find the propagation time inside the plate after heating, we find the time of flight of sound inside the plate using  $v_0$  and  $p_f$  resulting in  $toff = 3.6216$   $\mu\text{s}$ . Knowing the real time of flight allow us to know the real velocity using the real depth based on thermal dilation  $p_{r0}$ . The resulting velocity found, at  $90^\circ\text{C}$ , was  $v_f = 5522.2$  m/s.

## 5 Conclusion

The robot has successfully accomplished the tasks concerning adhesion and locomotion. The team is very optimistic as the critical challenges of high-temperature inspection and magnetic adhesion are being overcome.

As a prospective evaluation, it is necessary to perform tests with the umbilical cord of the robot, composed of a pipe with water for ultrasonic coupling, power and transmission cables, and a safety rope. The weight of the umbilical should be evaluated to ensure the stability and adhesion of the robot on the super-duplex tank, possibly requiring new changes to the composition of the magnetic wheel. Regarding the topology of the robot, a proposal that meets the safety requirements defined in standards and also gives the robot an even better degree of maneuverability on metallic surfaces is in development. Such topology will ensure the magnetic adhesion during movement of the robot along the surface to be inspected. Mechanical adjustments are predicted for the robot to perform well both when driving on surfaces with large or small curvature radii, and may, through misalignment of the wheels, overcome small obstacles without compromising its adhesion. In regards of the ultrasonic equipment, a full curve relating the acoustic velocity and surface temperature shall be calculated, as well as analyzing the influence of a wedge with a temperature gradient (hot bottom surface, cold transducer connection).

Among the next activities remain the integration of the robot to the systems of perception and location, and the final tests of maneuverability and navigability of the complete system. Future works about ultrasonic inspection, point to finishing the phased-array model that the robot will carry and its full adaptation to inspection procedures on hot surfaces. A test body will also be constructed with a bigger super-duplex steel plate and a heating system that replicate the inspection environment. With this, tests may be proved in situations similar to those found in the field, allowing full validation and finishing the project.

## References

1. Carvalho, A., Rebello, J., Souza, M., Sagrilo, L., Soares, S.: Reliability of non-destructive test techniques in the inspection of pipelines used in the oil industry. *Int. J. Press. Vessels Pip.* **85**(11), 745–751 (2008)
2. Eich, M., Vögele, T.: Design and control of a lightweight magnetic climbing robot for vessel inspection. In: 2011 19th Mediterranean Conference on Control and Automation (MED), pp. 1200–1205. IEEE (2011)
3. Electronic, L.: LRS - light section sensor for object detection, June 2018. [https://leuze.com/en/deutschland/produkte/messende\\_sensoren/3d\\_sensoren\\_1/lichtschmittsensoren\\_1/lrs\\_7/selector.php?supplier\\_aid=50111330&grp\\_id=1331722677208&lang=eng](https://leuze.com/en/deutschland/produkte/messende_sensoren/3d_sensoren_1/lichtschmittsensoren_1/lrs_7/selector.php?supplier_aid=50111330&grp_id=1331722677208&lang=eng)
4. Espinoza, R.V., de Oliveira, A.S., de Arruda, L.V.R., Junior, F.N.: Adhesion loss prediction of a climbing robot through magnetic field analysis by artificial neural networks. In: 22nd International Congress of Mechanical Engineering, pp. 3–7 (2013)
5. Espinoza, R.V., de Oliveira, A.S., de Arruda, L.V.R., Junior, F.N.: Navigation stabilization system of a magnetic adherence-based climbing robot. *J. Intell. Robot. Syst.* **78**(1), 65–81 (2015)
6. Olympus: High temperature ultrasonic testing, February 2018. <https://www.olympus-ims.com/en/applications/high-temperature-ultrasonic-testing/>
7. Rosa, A.B., Gnoatto, R.: Reprojeto e construção de protótipo de um robô de inspeção de cordões de solda em superfícies metálicas verticais e esféricas (segunda geração). B.S. thesis, Universidade Tecnológica Federal do Paraná (2015)
8. Rovani, A.: Desenvolvimento do protótipo de um robô para inspeção de cordões de solda em superfícies metálicas verticais. Industrial mechanical engineering - monograph, Federal University of Technology - Paraná (2013)
9. Santos, H.B., Teixeira, M.A.S., de Oliveira, A.S., de Arruda, L.V.R., Neves-Jr, F.: Quasi-omnidirectional fuzzy control of a climbing robot for inspection tasks. *J. Intell. Robot. Syst.* **91**(2), 333–347 (2018)
10. Schmidt, D., Berns, K.: Climbing robots for maintenance and inspections of vertical structures - a survey of design aspects and technologies. *Robot. Auton. Syst.* **61**(12), 1288–1305 (2013)
11. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots. MIT Press, Cambridge (2011)
12. Tavakoli, M., Viegas, C., Marques, L., Pires, J.N., De Almeida, A.T.: OmniClimbers: omni-directional magnetic wheeled climbing robots for inspection of ferromagnetic structures. *Robot. Auton. Syst.* **61**(9), 997–1007 (2013)
13. Teixeira, M.A.S., Santos, H.B., Dalmedico, N., de Arruda, L.V.R., Neves-Jr, F., de Oliveira, A.S.: Intelligent environment recognition and prediction for NDT inspection through autonomous climbing robot. *J. Intell. Robot. Syst.* **92**, 1–20 (2018)
14. Toolboxes, T.: AST inspection savings using in-service robotics, March 2016. <http://www.ttoolboxes.com/training/CourseDocuments/37/ASTInspectionSavingsUsingInServiceRobotics.pdf>
15. da Veiga, R.S., de Oliveira, A.S., de Arruda, L.V.R., Junior, F.N.: Localization and navigation of a climbing robot inside a LPG spherical tank based on dual-LIDAR scanning of weld beads. In: Robot Operating System (ROS), pp. 161–184. Springer (2016)



# Indoor Environment Monitoring in Search of Gas Leakage by Mobile Robot

João Braun<sup>1</sup>, Luis Piardi<sup>1,2</sup>✉, Thadeu Brito<sup>2</sup>, José Lima<sup>2,4</sup>, Ana Pereira<sup>2</sup>,  
Paulo Costa<sup>3,4</sup>, and Alberto Nakano<sup>1</sup>

<sup>1</sup> Federal University of Technology - Paraná (UTFPR), Toledo, Brazil  
jneto.1996@alunos.utfpr.edu.br, nakano@utfpr.edu.br

<sup>2</sup> Research Centre in Digitalization and Intelligent Robotics (CeDRI) and IPB,  
Bragança, Portugal

{fpiardi,brito,jllima,apereira}@ipb.pt

<sup>3</sup> Faculty of Engineering of University of Porto (FEUP), Porto, Portugal  
paco@fe.up.pt

<sup>4</sup> Centre for Robotics in Industry and Intelligent Systems - INESC TEC,  
Porto, Portugal

**Abstract.** Inspection based on mobile autonomous robots can assume an important role in many industries. Instead of having fixed sensors, the concept of assembling the sensors on a mobile robot that performs the scanning and inspection through a defined path is cheaper, configurable and adaptable. This paper describes a mobile robot, equipped with several gas sensors and a LIDAR device, that scans an established area by following a trajectory based on way-points searching for gas leakage and simultaneously avoid obstacles in the map. In other words, the robot follows the trajectory while the gas concentration is under a defined value and surrounding the obstacles. Otherwise, the autonomous robot starts the leakage search based on a search algorithm that allows to find the leakage position. The proposed methodology is verified in simulation based on a model of the real robot. The search test performed in a simulation environment allows to validate the proposed methodology.

**Keywords:** Algorithm gas search · Simulation · Autonomous mobile robot · LIDAR

## 1 Introduction

Nowadays robots can be used in several contexts such as inspection of industrial plants, search for environmental pollutant sources, explosives and drugs at airports and harbors. In the context of gas leakage, robots can be used for inspections in several structures, mainly in industrial plants, to prevent harm to humans and nature, and avoiding high financial losses [1]. The need of finding

the leak source without exposing humans to hazardous environments and accessing remote locations is what drives many researches in robotized leak detection [2].

In this context, robots can map any area while searching for predetermined levels of gases. In order to do this, the robot must be equipped with sensors with the ability to measure the gas emission, to locate itself, to plan the route to an objective point and to deviate from known and unknown obstacles. Developing a method that accomplishes these tasks can help industrial sectors avoid potential risks to workers. Therefore, the purpose of this work is to implement a method that meets these requirements through a simulation approach, to avoid possible failures in the real system. With the simulated environment, it is possible to test and validate the proposed gas search algorithms to perform the mentioned tasks, simulating adverse situations in the environment for virtual robot decision making. The SimTwo simulator arises a friendly way to simulate a 3D environment, the mobile robot and the gas emission distribution from a source [3].

This work is structured as follows. After the introduction, a brief state of art is presented in Sect. 2. Then, in Sect. 3, the robot model is stated showing the similarities and differences between the real robot and the simulated one. The algorithms and the gas distribution model used in this work are explained in Sect. 4. The project results are shown in Sect. 5. The conclusion and future work are discussed in Sect. 6.

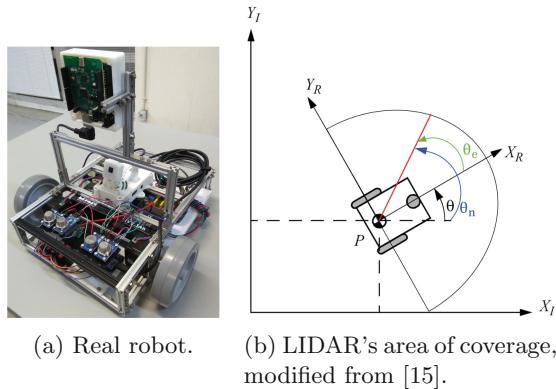
## 2 State of Art

Studies and development in mobile robotics have steadily increased over the years. Of all the possible tasks a mobile robot can do, having gas sensors is a great way to detect gas leakage, that is, robots with a “sense of smell” [4]. Detecting gas leakage can also be performed by more than one robot, as [5] demonstrates, by applying the Kalman filter it is possible to command five robots to find a certain concentration of gas.

The gas detection response is directly influenced by the discrimination of the gases to be detected, then [6] demonstrates an approach that inserts an array of sensors into the mobile robots. In this sense, each sensor is responsible for identifying a specific type of gas, speeding up the identification process. Another approach of multiple sensors to discriminate gases can be seen in [7], that develop a system with sensors that together distinguish different concentrations of propane, acetone, and ethanol. In order to detect large scale gas, [8] applies the decentralized Gas Distribution Map (GDM) method. Generating a Hilbert map through probabilistic representations, it addresses the task of finding gas concentration in the multiple classes. In this way, it points to an alternative way to GDM to map an environment with measurements learned from the place. As the mobile robot advances into the environment, it is possible to automatically perform unsupervised learning through the system developed by [9]. In an on-the-fly way the system discriminates the gases present in the site, which can serve as an extra tool for rescue teams.

### 3 Robot Model

In order to validate the approach and algorithms for the inspection and gas leakage detection, the structure of a real mobile robot already developed, will be simulated (seen in Fig. 1a). For further information about the robot, such as its localization system, the reader is referred to [10–13]. The Hokuyo Laser Ranger Finder (LRF) was modeled in SimTwo and validated in [14]. This model, applied in this project, presents noise similar to the real device. In this way, it is possible to obtain results near to the real sensor. This transducer detects obstacles in two dimensions via Time of Flight technology. The real configuration setup was applied in simulation: number of laser beams, the laser position in the free space  $[x, y, z]$ , its angle in the free space, the area coverage ( $180^\circ$  semi circle with 3–400 cm radius), sample period and the noise. In this way, the sensor area of coverage is illustrated in Fig. 1b. After the validation in simulation presented in this work, the robot will be tested in a real controlled scenario.



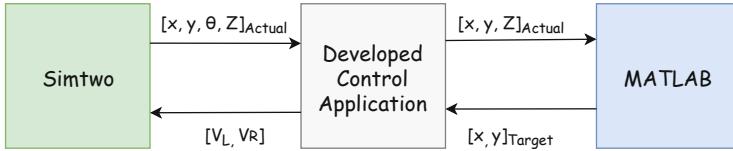
**Fig. 1.** Real structure of the mobile robot.

The software used to create the simulation environment is the SimTwo [3]. This software presents a realistic model of the 3D environment, which represents the dynamic constraints that exist in the real environment. In this sense, with the environment created through the simulator, is possible to validate the approach for the detection and search of gas-emitting sources, and then, in a future stage of the work, to carry out the tests in real environment.

### 4 Path Planning and Gas Search Algorithm

To search for the gas leakage avoiding obstacles not only the gradient search algorithm is needed but also a path planning. The path planning is used to reach the points that the gas search algorithm computed in an optimal way, without colliding with the obstacles. Binary Heap A\* is utilized for the path planning,

based on [16]. First, the user elaborates an inspection trajectory for the robot by selecting way-points in the developed control application. After, the application will interpolate a spline function with these points, as it can be seen in the next Section. The SimTwo simulator sends the robot pose and the gas concentration,  $[x, y, \theta, z]_{Actual}$ , to the application. The variables of communication between these processes are presented in Fig. 2.



**Fig. 2.** System communication structure.

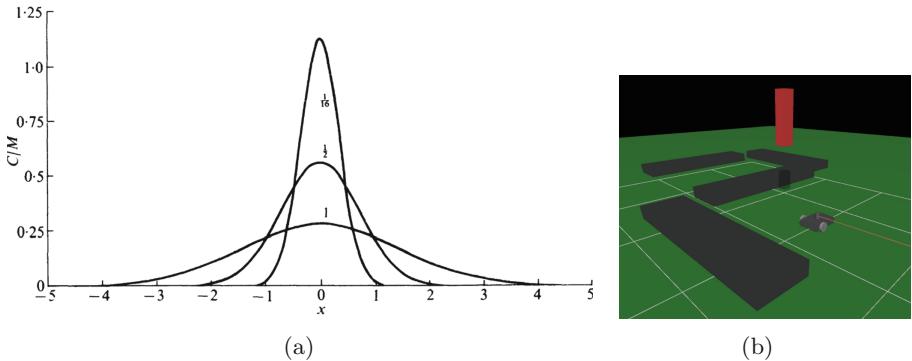
With this information, and sampling the spline function, an algorithm (already presented in [17]), makes the robot follow the trajectory computed by the interpolation. The sampling criterion is performed by the Euclidean distance between points that belong to the curve. During all the process, the robot keeps measuring gas substances. In simulation, the system simulates the leak through a gas model. The gas concentration ( $Z$ ) model for a constant diffusion can be seen in [18]

$$Z(x, y) = \frac{M}{2 \cdot \sqrt{(\pi \cdot D \cdot t)}} \cdot e^{-\frac{(h)^2}{4 \cdot D \cdot t}} \quad (1)$$

given by Eq. (1) for the position of the robot  $(x, y)$ . Where  $M$  represents the amount of substance deposited at time  $t = 0$  at  $[x_0, y_0]$ ;  $D$  is the diffusion constant;  $t$  the time and  $h$  is  $\sqrt{(x - x_0)^2 + (y - y_0)^2}$ , i.e, the Euclidean distance between the measure position  $[x, y]$ , and  $[x_0, y_0]$  the gas peak position.

As one can see, this model is time dependent, at each time the model represents another concentration value. However, the robot converges rapidly to the gas model peak position. Moreover, the diffusion is slow configured by parameter  $D$ . Thus the time influence was discarded by choosing a constant value for  $t$ . Figure 3a denotes the concentration values in one dimension, for several values of  $D \cdot t$ .

It is assumed an indoor environment so the influence of the wind is discarded. In future work, it will be addressed the wind disturbance as well as noise in gas acquisition. The simulated environment can be seen in Fig. 3b where the red cylinder is used to represent the gas leak. Hereupon, the SimTwo will simulate the gas sensors measurements by calculating the gas concentration at the actual robot position. Note that, the gas leak source is simulated assuming a depressurized leakage, in this way, nothing will influence the shape of the distribution. If the concentration pass a certain threshold (defined by the user with range from 0 to 100), the system will discard the user defined trajectory, and will start to seek for the gas source.



**Fig. 3.** Figure (a) represents the gas model in one dimension for several  $D \times t$  and  $x$  values. Figure (b) displays the simulation environment.

The proposed algorithm for gas search is running on MATLAB and the communication variables can be seen in Fig. 2 that uses UDP Ethernet datagrams. The application will send the robot pose and the concentration  $[x, y, \theta, z]_{Actual}$  to MATLAB that process the search algorithm for the leak localization. By this way, MATLAB sends the next position  $[x, y]_{Target}$  with high probability of a higher concentration to probe. MATLAB was chosen to run the search tool, as it is easier to implement more complex algorithms.

The MATLAB implemented search algorithm is described as follows. Let  $[x, y]_{Actual}$  be the robot actual position and  $[x, y]_{Previous}$  the robot previous position. Consider  $d$  the previous direction took by the robot, defined by  $d = [x, y]_{Actual} - [x, y]_{Previous}$  and consider  $\bar{d}$  the direction between the robot actual position and the position where was identified the higher gas concentration ( $[x, y]_{Higher}$ ) and can be defined as  $\bar{d} = [x, y]_{Higher} - [x, y]_{Actual}$ . Define  $Z_{Previous}$ ,  $Z_{Actual}$  and  $Z_{Higher}$  the gases concentrations at  $[x, y]_{Previous}$ ,  $[x, y]_{Actual}$  and  $[x, y]_{Higher}$ , respectively.

If  $Z_{Previous} \leq Z_{Actual}$ , the robot target position will be defined as

$$[x, y]_{Target} = \begin{cases} [x, y]_{Actual} + d, & \text{if } Z_{Higher} \leq Z_{Actual}; \\ [x, y]_{Actual} + d + \bar{d}, & \text{otherwise.} \end{cases} \quad (2)$$

If  $Z_{Previous} > Z_{Actual}$ , the robot target position will be in the opposite direction, that means, if  $d$  is a horizontal movement then the new direction will be the opposite diagonal direction. In the same way, if  $d$  is a diagonal direction then the new direction will be the opposite horizontal direction. When the procedure take an opposite direction the choice to take right/left depends of the condition of  $Z_{Higher}$ , that is, if  $Z_{Higher}$  is greater (or lower) than the  $Z_{Actual}$ . In this way, soon after the robot quits the inspection trajectory, the robot will search for a direction with a greater gas concentration and will keep moving in the same direction until the gas concentration decreases. When this happens, it will search for the next direction where the gas concentration is higher, and so on.

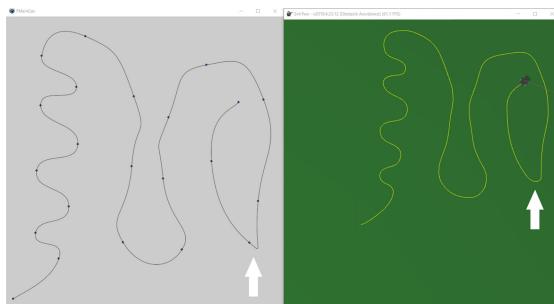
The described algorithm takes the robot to a near location of the source. This algorithm makes no assumption of the exact gas model 1 and will generate a set of target points and gas concentration measures with increasing concentration values.

## 5 Results

Several tests were evaluated to validate the inspection route, the path planning, the obstacle avoidance system and the gas search algorithm (with and without the obstacles). In this sense, they will be presented in this Section in the following order: Inspection route, Obstacle avoidance, Gas search algorithm without obstacles and Gas search combined with the first and second items.

### 5.1 Inspection Route

The user can select control way-points so that the control application can interpolate spline functions that pass through all the control points. Soon after, the control application samples the route, as the robot has a point following controller. The sampling happens every 20 cm. Thus, the path followed by the robot has a smooth performance and by this way a faithful representation to the original spline. The sampling has a low pass characteristic as can be seen in Fig. 4, especially by the white arrows indicators.

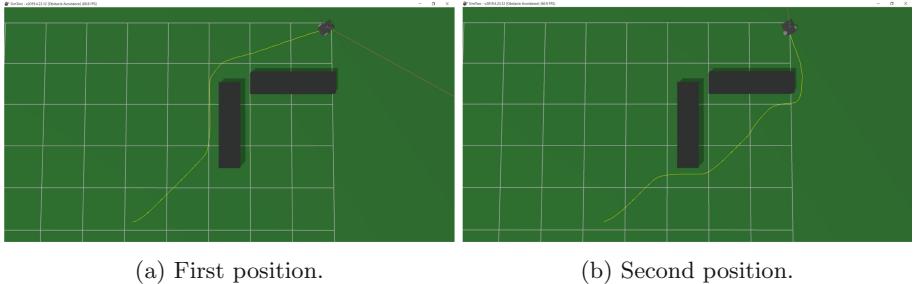


**Fig. 4.** Application screenshot of the planning trajectory inserted by way-points from the user.

The robot's controller is developed in such a way that whenever the robot gets near the actual point, the control application sends another point. It is repeated until the last point, where the robot follows a deceleration function to stop to the last point. As the controller is not on this scope of this work, the developed controller details can be seen in [11].

## 5.2 Obstacle Avoidance

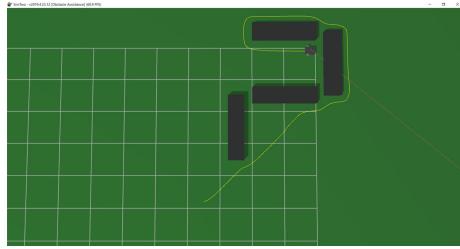
Two scenarios were tested to prove the path planning algorithm developed for this work. In the first scenario, the robot is inserted near the origin  $[x, y] = [0.2, 0.2]$  and set to go to the point  $[x, y] = [5, 5]$ . Then, is added two obstacles blocking the direct path to the target. Note that, the path planning algorithm developed is based in Binary Heap A\* method, that uses a proportionality constant to generate sub optimal paths in less time [19]. Thus, it is difficult to predict if the robot will choose the shortest path. Figure 5a displays the first scenario.



**Fig. 5.** Path planning test scenario using a Binary Heap A\*. The grid does not represent the map limitations.

In another test, the robot is inserted a little farther to the right  $[x, y] = [0.5, 0.2]$ , to force the robot to choose the right path, as the heuristic cost to reach the target point is lower. Bear in mind that the LIDAR model works from 3 cm to 400 cm. In this concept, the robot can not see obstacles beyond this range. Thus, the robot continues the right path until it sees the rightmost obstacle. Although, when it sees this obstacle, the heuristic cost to go back to the left path is still higher. Thus, the robot continues with the first path choice. Figure 5b presents this behaviour. Consider that the grid is just illustrative, it does not mean the maps limitations.

Finally, in the second scenario, two more obstacles are placed to jam the path planning. As can be seen in Fig. 6, the robot is placed in  $[x, y] = [1.5, 0.2]$  to decrease even further the heuristic cost to the robot perform the right path. However, as the robot can not see beyond 400 cm, the robot does not know the obstacles until they reach its range. In this test, the robot keeps going to the farthest path even though the left one was the shortest. Making the robot circle all the obstacles.

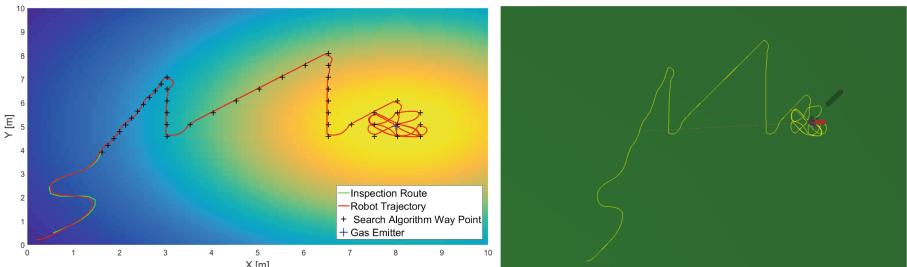


**Fig. 6.** Path planning test second scenario using a Binary Heap A\*. The grid does not represent the map limitations.

### 5.3 Gas Search Algorithm Without Obstacles

As explained in Sect. 4, the algorithm does not takes assumption on the gas model and its maximum concentration value, as in the real scenario this can not be predicted. In this way, the algorithm is tested without obstacles in just one scenario, because the main objective of this work is to validate the algorithm with the obstacle avoidance feature. The gas position model in this test can be seen in Fig. 10.

Moreover, Fig. 7a illustrates the comparison between the path performed by the robot and the route made by the user in the application as well as the gas search algorithm points.



(a) Comparison of the inspection route and (b) Path planning created in SimTwo. gas search algorithm.

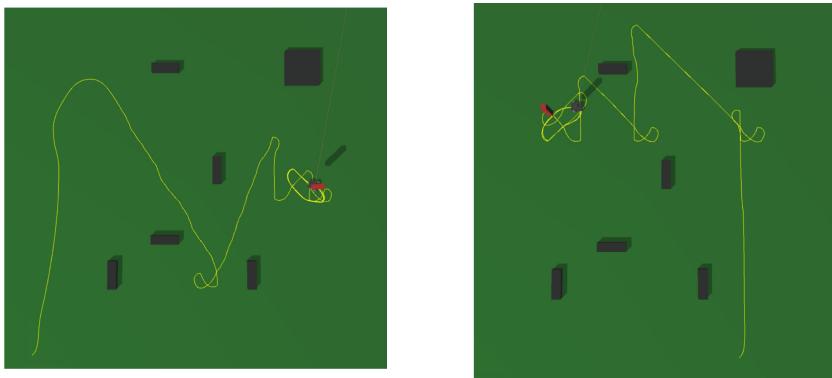
**Fig. 7.** Path planning test along with gas search algorithm without obstacles.

Figure 7b presents the simulation result that was represented by Fig. 7a. Note that, when the robot abandons the inspection route and starts to search for the gas leak, the black plus symbols in Fig. 7a starts to appear in the plot (which are the points computed by the gas search algorithm), i.e., the robot follows the points sent by the gas search algorithm. Moreover, it is important to note the similarity between the path performed by the robot and the path generated by

the simulation. Little differences can be seen because of the spline sampling and the controller. In addition, as explained, the search algorithm behaviour makes the robot keep going until the gradient becomes negative, then it changes its direction. Finally, when it reaches the maximum point, it keeps circling the peak value. It was decided to not implement a stop point as the idea behind this work is to find the gas leak. Therefore, the robot keeps circling it.

#### 5.4 Gas Search with Obstacle Avoidance

Finally, two challenging scenarios were tested with the gas algorithm combined with the path planning. The first analysis is applied to obstacles in the same gas model position in the previous verification. Figure 8a displays the robot and the algorithms behaviour. In this concept, the inspection route is drawn (green line in Fig. 9) to the robot start the test.



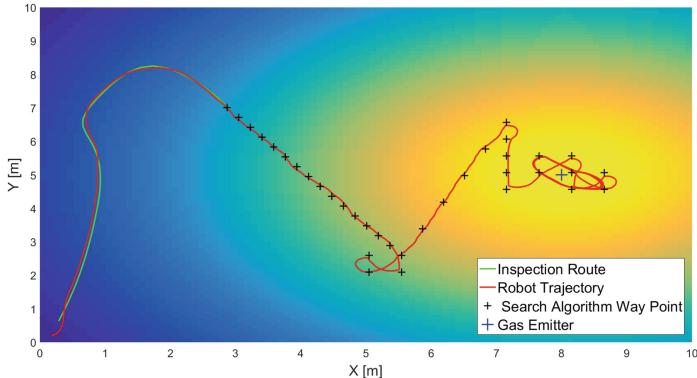
(a) Gas search algorithm with obstacles in the first scenario test. (b) Gas search algorithm with obstacles in the second scenario test.

**Fig. 8.** Gas search tests with obstacles.

As can be seen in Figs. 8a and 9, the robot could perform the search with several obstacles placed in the scenario. Moreover, note that the robot followed the gas search points smoothly (black plus symbols in Fig. 9). Additionally, whenever the gas search algorithm sent a point that the robot could not reach, the control application with the path planning, redirected the robot to the nearest point accessible. The robot with the algorithms converged to the peak without any problems. Figure 10 displays the gas model in the first test.

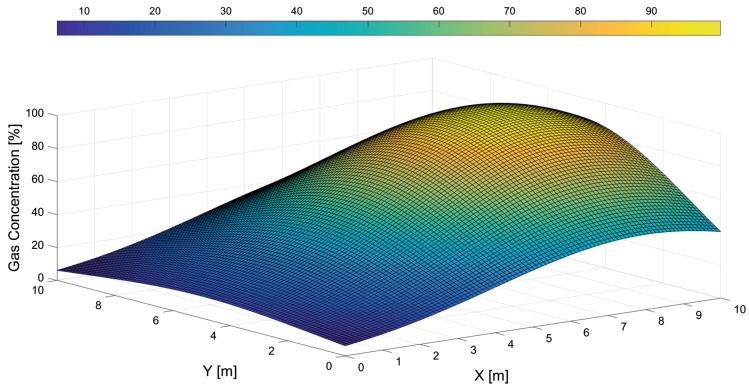
In the second scenario the gas model position is changed to  $[x, y] = [2.5, 7]$  and the robot to  $[x, y] = [7.5, 0.5]$ . Figures 8b and 11 displays the robot's path behaviour in the second scenario with the obstacles.

As can be seen, during the second test, the robot could perform the search without colliding with any obstacles. The gas search algorithm behaviour is well



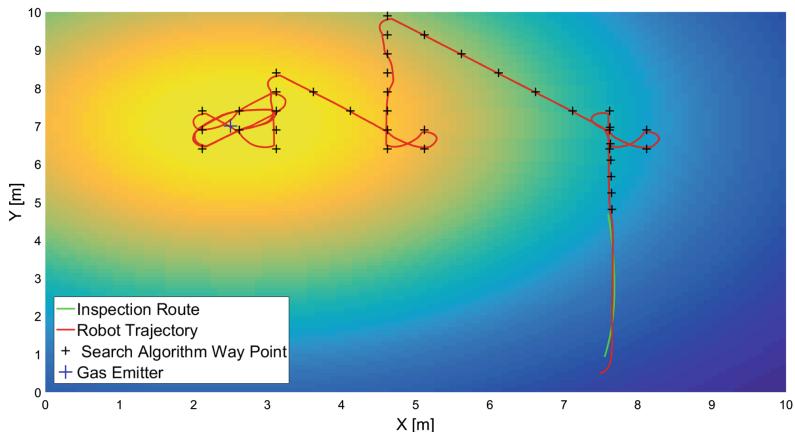
**Fig. 9.** Inspection route, path performed by the robot and the algorithm search points in a flat plot during the first scenario.

seen in the change of directions in the robot's path, represented by the yellow line in Fig. 8b. As explained in Sect. 4, the robot keeps moving in the direction that has a higher positive gradient and, when it becomes negative, it changes direction. Therefore, soon after the robot quits the inspection route (when the black plus symbols starts to appear), the robot keeps going north until the gradient becomes negative and then changes its direction, and this happens two more times, until the robot finds the gas leak and keeps circling it.



**Fig. 10.** Gas model in the first scenario in the combined test.

Again, the green line, which is the inspection route and the red line, which is the path performed by the robot, are similar (seen in Fig. 11). This displays that the robot's controller is well adjusted.



**Fig. 11.** Inspection route, path performed by the robot and the algorithm search points in a flat plot during the second scenario.

## 6 Conclusion and Future Works

This work presented and validated through realistic simulation an autonomous mobile robot that could perform an inspection route selected by the user measuring for gas substances. Moreover, if a certain gas threshold value passed, the robot would abandon the trajectory and search autonomously for the gas source, avoiding obstacles simultaneously. In addition, the gas model, the obstacle avoidance system, path planning and data gathering techniques are presented. The gas search algorithm proved to be well suited for the indoor simulation environment, and combined with the obstacle avoidance algorithm, the tests showed promising results. Finally, the path similarity between the inspection route and the path performed by the robot are very similar, proving the controller is well adjusted in simulation. In future work, the idea is to test this in a real environment.

**Acknowledgements.** This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project (POCI-01-0145-FEDER-006961), and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

## References

1. Baetz, W., Kroll, A., Bonow, G.: Mobile robots with active IR-optical sensing for remote gas detection and source localization. In: 2009 IEEE International Conference on Robotics and Automation, pp. 2773–2778. IEEE (2009)
2. Kowadlo, G., Russell, R.A.: Robot odor localization: a taxonomy and survey. Int. J. Robot. Res. **27**(8), 869–894 (2008)

3. Costa, P., Gonçalves, J., Lima, J., Malheiros, P.: SimTwo realistic simulator: a tool for the development and validation of robot software. *Theory Appl. Math. Comput. Sci.* **1**(1), 17–33 (2011)
4. Wandel, M., Lilienthal, A.J., Duckett, T., Weimar, U., Zell, A.: Gas distribution in unventilated indoor environments inspected by a mobile robot. In: IEEE International Conference on Advanced Robotics, ICAR 2003, Coimbra, Portugal, 30 June - 3 July 2003, vol. 1, pp. 507–512. University of Coimbra (2003)
5. Marques, L., Martins, A., de Almeida, A.T.: Environmental monitoring with mobile robots. In: 2005 IEEE/RSJ International Conference on Intelligent Robotics and Systems, pp. 3624–3629. IEEE (2005)
6. Trincavelli, M.: Gas discrimination for mobile robots. *KI-Künstliche Intell.* **25**(4), 351–354 (2011)
7. Xing, Y., Vincent, T.A., Cole, M., Gardner, J.W., Fan, H., Bennetts, V.H., Schaffernicht, E., Lilienthal, A.J.: Mobile robot multi-sensor unit for unsupervised gas discrimination in uncontrolled environments. In: 2017 IEEE SENSORS (2017)
8. Zhu, P., Ferrari, S., Morelli, J., Linares, R., Doerr, B.: Scalable Gas sensing, mapping, and path planning via decentralized hilbert maps. *Sensors* **19**(7), 1524 (2019)
9. Fan, H., Hernandez Bennetts, V., Schaffernicht, E., Lilienthal, A.J.: Towards gas discrimination and mapping in emergency response scenarios using a mobile robot with an electronic nose. *Sensors* **19**(3), 685 (2019)
10. Lima, J., Costa, P.: Ultra-wideband time of flight based localization system and odometry fusion for a scanning 3 DoF magnetic field autonomous robot. In: Iberian Robotic Conference, pp. 879–890. Springer, Cham (2017)
11. Piardi, L.: Application of a mobile robot to spatial mapping of radioactive substances in indoor environment. Master's thesis. IPB (2018)
12. Piardi, L., Lima, J., Costa, P.: Development of a ground truth localization system for wheeled mobile robots in indoor environments based on laser range-finder for low-cost systems. In: ICINCO, no. 2, pp. 351–358 (2018)
13. Braun, J., Brito, T., Lima, J., Costa, P., Costa, P., Nakano, A.: A Comparison of A\* and RRT\* algorithms with dynamic and real time constraint scenarios for mobile robots. In: Proceedings of the 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, vol. 1, pp. 398–405. SIMULTECH (2019). ISBN 978-989-758-381-0. <https://doi.org/10.5220/0008118803980405>
14. Lima, J., Gonçalves, J., Costa, P.J.: Modeling of a low cost laser scanner sensor. In: CONTROLO'2014—Proceedings of the 11th Portuguese Conference on Automatic Control, pp. 697–705. Springer, Cham (2015)
15. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots. MIT Press, Cambridge (2011)
16. Sheikh, T.S., Afanasyev, I.M.: Stereo vision-based optimal path planning with stochastic maps for mobile robot navigation. In: International Conference on Intelligent Autonomous Systems, pp. 40–55. Springer, Cham (2018)
17. Piardi, L., Lima, J., Costa, P., Brito, T.: Development of a dynamic path for a toxic substances mapping mobile robot in industry environment. In: Iberian Robotic Conference, pp. 655–667. Springer, Cham (2017)
18. Crank, J.: The Mathematics of Diffusion. Oxford University Press, Oxford (1979)
19. Lima, J., Costa, P., Costa, P., Eckert, L., Piardi, L., Moreira, A., Nakano, A.: A\* Search Algorithm Optimization Path Planning in Mobile Robots Scenarios. In: International Conference of Numerical Analysis and Applied Mathematics (2018)



# Performance of New Global Appearance Description Methods in Localization of Mobile Robots

Vicente Román<sup>(✉)</sup>, Luis Payá<sup>(✉)</sup>, María Flores<sup>(✉)</sup>, Sergio Cebollada<sup>(✉)</sup>, and Óscar Reinoso<sup>(✉)</sup>

Engineering Systems and Automation Department, Miguel Hernandez University,  
Elche, Alicante, Spain

{vicente.roman,lpaya,m.flores,sergio.cebollada,o.reinoso}@goumh.umh.es

**Abstract.** Autonomous robots should be able to perform localization and map creation robustly. In order to solve these problems many studies and techniques have been evaluated over the past few years. This work focuses on the use of an omnidirectional vision sensor and global appearance techniques to describe each image. Global-appearance techniques consist in obtaining a unique vector that describes globally the panoramic image. Once the images have been described the mobile robot can use these descriptors both to create a map of the environment or to estimate its position and orientation in the environment. The main objective of this work is to propose and test new alternatives to describe scenes globally. The results will be used to propose new robust methods to estimate the position and orientation of the robot, from the combination of several measurements of similitude of visual information. Therefore, the present work is an initial study towards a new localization method. In this initial study a comparative the previous and the new methods is performed. The experiments will be carried out with real images that have been taken in an heterogeneous scenario where simultaneously humans and robots work together. For this reason, variations of the lighting conditions, people who occlude the scene and changes on the furniture may appear.

**Keywords:** Localization · Mobile robots · Global appearance descriptors · Omnidirectional images

## 1 Introduction

Over the years, a revolution on the robotics world has occurred. The presence of mobile robots in different environments has increased, and their perception and interpretation abilities have had to improve to cope with more challenging situations. To be truly autonomous in large, changing and heterogeneous environments, the task of a mobile robot is twofold. Not only it has to be able to navigate around heterogeneous places creating a map (mapping task) but it also has to estimate its position (localization task).

In the scientific literature, a variety of methods can be found. These methods have tried to solve the mapping and localization problems using different sensors and approaches. About the sensors, encoders, sonars or lasers are some proposed alternatives, but these sensors need maintenance and they use relatively quite a lot energy. Another alternative is GPS, but this one is only useful in outdoor places. A third option is to use visual information. Visual sensors offer a big amount of data with a relatively low price. Omnidirectional cameras are a possible configuration among vision sensors. They constitute an advantageous option due to the fact that, they offer images with a field of view of 360° around the camera axis [17]. These systems have been extensively studied in recent years since they provide the robot with complete information about its surrounding, which is specially useful for navigation tasks. Works like [15] and [12] show datasets containing images captured by such omnidirectional camera systems.

Images contain a big amount of data, and relevant information must be extracted from them. This information must be useful to describe the environment as seen from a specific point of view. Nowadays local appearance descriptors are extensively used. These methods try to describe specific points, corners or local zones. SIFT [5] and SURF [2] are well-known local appearance descriptors. Relatively good solutions in mapping and localization tasks have been obtained using these descriptors. For example, Angeli et al. [1] used these descriptors to carry out topological localization and Murillo et al. [7] develop a local appearance descriptor for mobile robots navigation in indoor environments. Global appearance descriptors are an alternative method to obtain relevant information from the scenes. These methods describe globally each image summarising it in an unique vector, which is expected to be more invariant against global changes. In addition, as they describe the image with an unique vector, the mapping and localization task will be simplified to a pairwise comparison between vectors. Some description methods have been studied, among them the Histogram of Oriented Gradients (HOG) can be highlighted. HOG is used in [11] to solve localization tasks. Other option is the Gist descriptor, which was introduced by Oliva and Torralba [8] and used in outdoors environments [16]. Other important methods to obtain global appearance descriptors are the Fourier Transform [6] or the Radon Transform [13]. These alternatives have been used in works as Paya et al. [10] to build maps or in Berenguer et al. [3] where the Radon Transform is used to perform localization as an image retrieval problem. Finally, Román et al. [14] used these global appearance methods in real environments to solve the localization task under different lighting conditions.

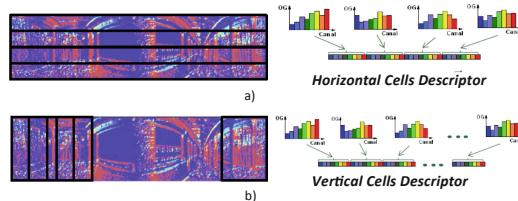
As shown in the next section, global appearance descriptors can be defined to be invariant against robot rotations in the ground plane when omnidirectional images are used. Traditional global-appearance methods have grouped the information from the panoramic images in horizontal blocks or cells, which allows to obtain the same information independently on the robot orientation. This work compares this classic formulation with a new one in which the cells are vertical and partially overlapped. To sum up, a new method is proposed to describe and compare images globally, and tested in a localization framework. Also, a comparative evaluation with traditional methods is included.

## 2 Global Appearance Descriptors

In this section a review of the global appearance descriptors used in the experiments is done. The goal of these descriptors is to extract global information from panoramic images trying to keep the relevant data with the minimum memory. To achieve this aim, HOG and Gist descriptors are used. In both cases the starting point is a panoramic image  $i(x, y) \in \mathbb{R}^{N_x \times N_y}$  and after these methods each image is reduced to a vector  $\mathbf{d} \in \mathbb{R}^{l \times 1}$ .

In order to obtain this descriptor, the image has to be divided in different cells. Different descriptors can be obtained, depending on the shape of these cells. The first option is the classic way, used in [14] where the vector is built with uniformly distributed and non-overlapped horizontal cells. The image is divided in a set of horizontal cells. Since the images are panoramic the result in each row will be the same independently of robot's orientation. The second option consists in defining a set of vertical cells. In broad lines, two steps are needed. Firstly the descriptor is built putting together the information obtained from each vertical cell; then to compare two descriptors the algorithm first detects if they have different orientation. In this case, the algorithm will rotate one of them until they have the same relative orientation. Once the relative orientation is the same the descriptors can be compared. In this way, although the descriptor depends on the robot orientation, the algorithm can compare descriptors thanks to the step that normalizes this orientation. Figure 1 shows both methods to build the descriptor: using horizontal cells or vertical ones.

As stated before, although the methods are similar, when the vertical cells are used, some overlapping between consecutive cells must exist. With these overlapping zones it will be easier to the algorithm to detect the relative different orientation and to rotate descriptor. The overlapping zones can be seen on Fig. 1.



**Fig. 1.** Approaches to build a global-appearance descriptor from a panoramic image: (a) with horizontal and (b) with overlapped vertical cells.

### 2.1 Histogram of Oriented Gradients, HOG

The method was described by Dalal [4]. It mainly consists in calculating the image gradient, obtaining module and orientation of the gradient in each pixel. After that, these module and orientations are evaluated and stored by areas, the descriptor is build by putting together the information collected in each area.

The classic method divides the areas using horizontal cells whereas the second method uses vertical cells in order to define the areas. Both will be invariant to rotations so they can be used in the localization tasks.

The descriptor size will depend on diverse parameters. This way, the vector length depends on the number of bins of the orientations histogram  $b$  and the number of cells in which the image is divided. To define the size cells, this work will use  $k_{1v}$  to refer to the number of vertical cells and  $k_{1h}$  to refer to horizontal cells. Throughout this work the parameters are not constant and we study how these changes affect to the result. The HOG descriptor reduces a panoramic image into a vector whose size is  $\mathbf{d} \in \mathbb{R}^{b \cdot k_{1h} \times 1}$  when the cells are horizontal. On the other hand, as started before, when the cells are vertical, some overlapping between them must exist. For that reason, a new parameter named  $dist$  appears.  $dist$  refers to the distance between the beginning of consecutive cells. During the experiments this parameter is constant  $dist = 2$ . Therefore,  $k_{1v}$  does not refer to the number of cells but it is related to the cells width. The cells width is determined by  $\frac{N_y}{k_{1v}}$ ,  $N_y$  is the number of pixels in the width of the panoramic image. Finally descriptor size when the cells are vertical is  $\mathbf{d} \in \mathbb{R}^{N_y \cdot \frac{b}{dist} \times 1}$ .

## 2.2 Gist

Gist descriptor was initially proposed by Oliva et al. [9] and it was developed by Siagian et. al. [16] where it was tested in three outdoor environments. This method exposes the image to some Gabor filters with different orientations in several resolution levels. After that, the images are reduced evaluating their mean intensity in different areas. Like in HOG, the image can be divided with horizontal and vertical cells.

In this case, the descriptor size will depend on the number of orientations of Gabor filters  $m$ , the number of cells in which the images are split  $k_2$  and the number of different resolution models used  $r$ . During the experiments this latter parameter  $r$  will be constant,  $r = 2$  when using horizontal cells and  $r = 1$  with vertical ones. For that reasons the descriptor is a vector  $\mathbf{d} \in \mathbb{R}^{1 \cdot m \cdot k_{2v} \times 1}$  using vertical cells and  $\mathbf{d} \in \mathbb{R}^{2 \cdot m \cdot k_{2h} \times 1}$  using horizontal ones. As in HOG, this work will use  $k_{2v}$  if we are talking about vertical cells and  $k_{2h}$  if we are talking about horizontal cells.

## 3 Database

To test the proposed global appearance method in a localization framework, an heterogeneous and dynamic environment is needed. COLD database [12] provides some indoor routes that fulfill these specifications. In COLD database the robot travels along some laboratories where people are normally working. In addition, the routes were taken under very different lighting conditions: a cloudy day, a sunny one and at night. Taken these facts into account, the robot moves in a changing environment and it has to deal with occlusions, lighting changes and changes introduced by human activity in the environment. Therefore, these



**Fig. 2.** Real-life changes in a heterogeneous environment.

situations offer an ideal test bench as far as the proposed descriptors are concerned. Using COLD database it is possible to evaluate the robustness of global appearance descriptors using them in localization and mapping task because it offers variations that might happen in real scenarios. In Fig. 2 is possible to see some snapshots of the heterogeneous environment.

The images offered by COLD were taken using a mobile robot equipped with an omnidirectional camera. This equipment was built using an ordinary camera mounted with a hyperbolic mirror, which takes omnidirectional images. Once the images have been transformed into panoramic, it is possible to apply the proposed global appearance descriptors. Moreover, the dataset includes the ground truth of the images, which will be used to check the correctness of the localization algorithm. These data will never be used in the localization task, the localization task will be achieved with pure visual information.

As it was mentioned before, the COLD dataset was taken in different buildings and days. The routes inside each building are similar and they were performed in diverse days with different lighting conditions. To carry out the experiments the longest route has been chosen; (Freiburg Part A, Path 2, size 3) [12]. This route was chosen because its size and high complexity due to the fact that many glass walls and windows appear so the external lighting conditions affect substantially in the localization task.

## 4 Experiments

In this section the experiments are presented. They validate both global appearance methods, the new and the classic one. The first subsection shows how the model has been created. Subsection 4.2 tests the localization process and shows the performance of each method. Finally, Subsect. 4.3 studies the computational cost of the processes.

## 4.1 Model of the Environment

In order to carry out a model or map of the environment must be previously available. As started in Sect. 3, COLD database offers diverse routes inside different buildings and lighting conditions. To create a preliminary a set of 556 images has been selected with a distance of 0.30 m between their consecutive capture points. This model has been created with cloudy database images.

Once the model has been built with known images, to solve the localization task a new test image will be compared with the descriptors in this model. The algorithm calculates the nearest neighbor using *Cityblock* distance,  $d_1(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^l |a_i - b_i|$ .

Once the nearest neighbor has been calculated using global appearance descriptors, the geometric distance between the capture point of the image and the nearest neighbor is calculated. This real distance is calculated because COLD database offers the coordinates where each image had been taken. These coordinates have been used only as a ground truth to check the error. The localization task is carried out with pure visual information.

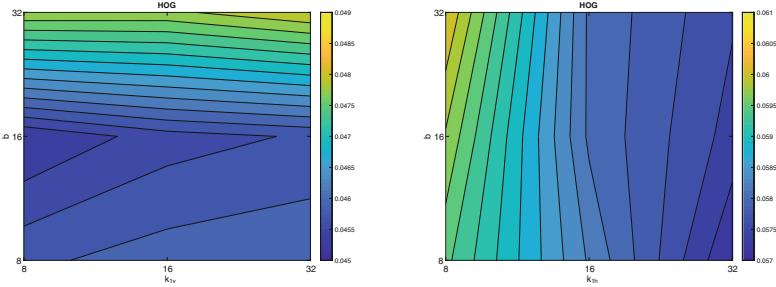
The experiments have been done using different descriptors and varying the parameters mentioned in Sect. 2, in order to evaluate each parameter influence in the task. There parameters are shown in Table 1. It is important to highlight the fact that whereas  $b$ ,  $k_{1v}$ ,  $k_{1h}$ ,  $m$ ,  $k_{2v}$  and  $k_{2h}$  have been varied,  $dist$  and  $r$  are constant,  $dist = 2$ ,  $r = 2$  when the cells are horizontal and  $r = 1$  when the cells are vertical. As explained in Sect. 2, these parameters will define the descriptor size. The higher the parameters are the longer the descriptor and the slower the process will be.

**Table 1.** Parameters that impact on the location process

Descriptor	Parameters
HOG	$b \Rightarrow$ number of bins per histogram
	$dist \Rightarrow$ distance between the beginning of consecutive vertical cells
	$k_{1v} \Rightarrow \frac{N_y}{k_{1v}}$ width of the cells using vertical cells
	$k_{1h} \Rightarrow$ number of horizontal cells
Gist	$m \Rightarrow$ number of Gabor filters
	$r \Rightarrow$ different resolution models
	$k_{2v} \Rightarrow$ number of vertical cells
	$k_{2h} \Rightarrow$ number of horizontal cells

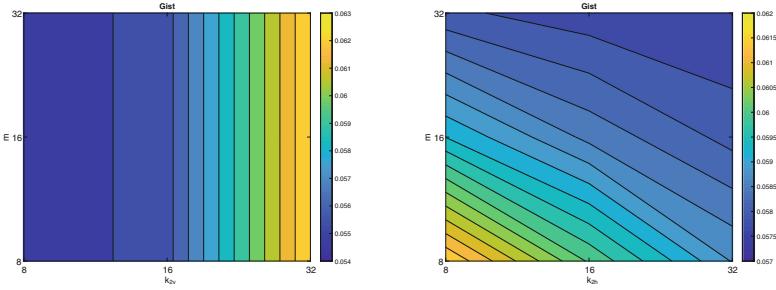
## 4.2 Position Estimation

As started before, cloudy images had been used to create the reference model. Then, to study the robustness of the global appearance descriptors in a localization framework, some real images from 3 different paths have been chosen.



(a) Error Using Vertical Cells

(b) Error Using Horizontal Cells

**Fig. 3.** HOG. Localization error (m) using test images from the **cloudy** dataset versus  $k_1$  and  $b$ .

(a) Error Using Vertical Cells

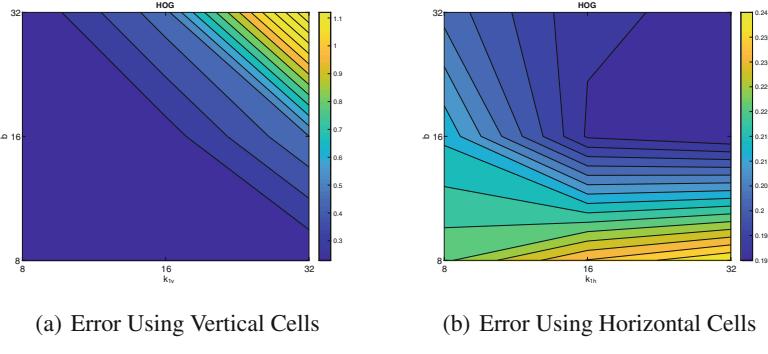
(b) Error Using Horizontal Cells

**Fig. 4.** Gist. Localization error (m) using test images from the **cloudy** dataset versus  $k_2$  and  $m$ .

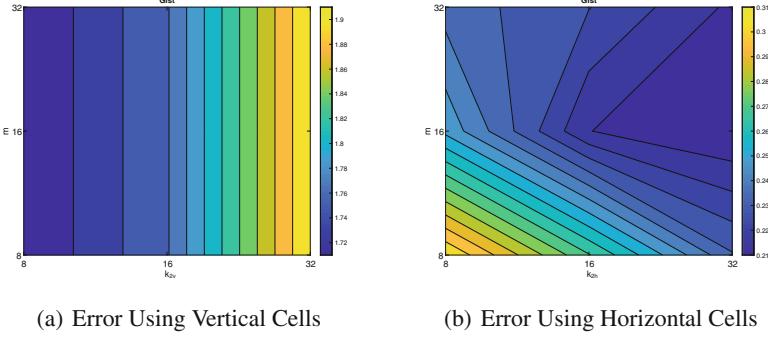
These paths were taken in the same building but with cloudy, night and sunny conditions. These test databases are respectively composed of 2778, 2896 and 2231 images and they cover a 105 m path. Those are the new test images and the algorithm evaluates which image from the reference model is the most similar.

The error is calculated as the geometrical distance between the captured point of every test image and its corresponding nearest neighbor. Once the algorithm has run with all cloudy images, Figs. 3 and 4 show HOG and Gist mean geometric error. The figures show the error in meters. In both cases, it is possible to see the error using the descriptor with vertical cells and the classic descriptors use, with horizontal cells.

The result using HOG depends on  $b$  more than on the number of vertical cells  $k_{1v}$ . On the other hand, when using horizontal cells, the error depends on these cells  $k_{1h}$  more than the number of bins  $b$ . The error increases when  $k_b$  increases and vertical cells are used, whereas the higher  $k_{1h}$  is, the worse the error is. The minimum error is 0.0453 m using vertical cells and  $k_{1v} = 8$   $b = 16$  and 0.0572 m using horizontal cells and  $k_{1v} = 32$   $b = 8$ .



**Fig. 5.** HOG. Localization error (m) using test images from the **night** dataset versus  $k_1$  and  $b$ .

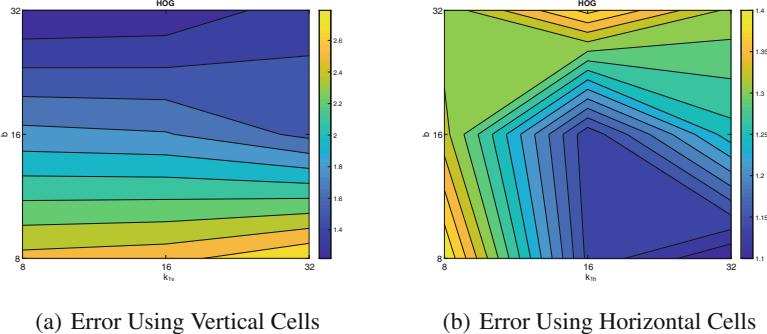


**Fig. 6.** Gist. Localization error (m) using test images from the **night** dataset versus  $k_2$  and  $m$ .

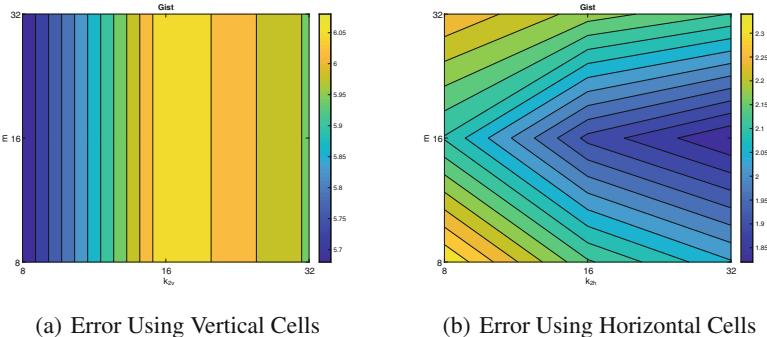
When using Gist we can observe that number of Gabor filters  $m$  has a lower influence in the result and the higher  $k_{2v}$  the higher is the error using vertical cells. By contrast, using horizontal cells the higher  $k_{2h}$  and  $m$  are the better is the result. The lowest error is 0.0564 m using  $k_{2h} = 8$   $m = 8$  and vertical cells and 0.0577 m using  $k_{2h} = 32$   $m = 32$  and horizontal cells.

After these experiments, the algorithm has been run with all the test night images. Figures 5 and 6 show HOG and Gist mean error in meters.

The behaviour of HOG descriptors using vertical or horizontal cells is completely different. Whereas using vertical cells the lower both,  $k_{1v}$  and  $b$ , are the lower the error is; using horizontal cells the higher  $k_{1h}$  and  $b$  are the lower the error is. In addition, we can see that using vertical cells both parameters have the same influence. In contrast, using horizontal cells the error depends more on the number of horizontal cells,  $k_{1h}$ . Apart from  $k_{1v} = 32$  and  $b = 32$  when localization error is extreme, errors using vertical cells is similar to errors using horizontal cells. The most accurate results are obtained using  $k_{1v} = 8$ ,  $b = 8$  and



**Fig. 7.** HOG. Localization error (m) using test images from the **sunny** dataset versus  $k_1$  and  $b$ .



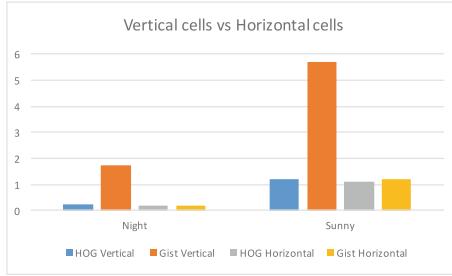
**Fig. 8.** Gist. Localization error (m) using test images from the **sunny** dataset versus  $k_2$  and  $m$ .

vertical cells when the error is 0.227 m whereas using  $k_{1v} = 32$  and  $b = 16$  and horizontal cells the error is 0.189 m.

About Gist, first of all it is seen that  $m$  has a poor influence when using vertical cells and the lowest  $k_{2v}$ , the lowest the error is. On the other hand, when using horizontal cells, the higher  $k_{2h}$  and  $m$  are the lowest the error is. Using Gist method the error is higher using vertical cells. The best solution is obtained with  $k_{2v} = 8$  and  $m = 8$  and vertical cells when the error is 1.711 m and using  $k_{2h} = 32$  and  $m = 16$  and horizontal cells the error is 0.210 m.

Finally, the localization task is repeated considering the test sunny database. The reference maps is the same as in the previous cases, the one that was made with cloudy images. Figures 7 and 8 show the results presenting the error in meters.

The results using HOG descriptors are different when vertical or horizontal cells are used. Whereas using vertical cells the higher  $b$  is the better is the



**Fig. 9.** Minimum error using HOG and Gist descriptor in night and sunny datasets.

result, using horizontal cells the lowest errors are obtained with high number of horizontal cells  $k_{1h}$  and low number of bins  $b$ . The best solutions are 1.2255 m detected with  $k_{1v} = 8$ ,  $b = 32$  and vertical cells and 1.096 m with  $k_{1h} = 32$ ,  $b = 8$  and horizontal cells.

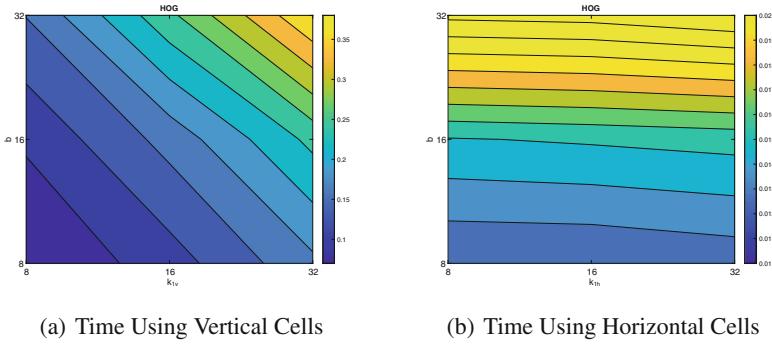
Finally, about the localization error using Gist,  $k_{2v}$  has little influence using vertical cells and using horizontal cells the best results are obtained with a high number of  $k_{2h}$  and medium number of mask  $m$ . The lowest errors are 5.682 m using vertical cells,  $k_{2v} = 8$  and  $m = 8$  and 1.824 m using horizontal cells,  $k_{2h} = 32$  and  $m = 16$ .

Analysing these figures, we consider that HOG descriptor is able to work correctly with both, vertical and horizontal cell. By contrast, Gist descriptor has remarkably bad results using vertical cells whereas using horizontal cells it has suitable results. Figure 9 summarises the minimum error obtained with each kind of cells and descriptors.

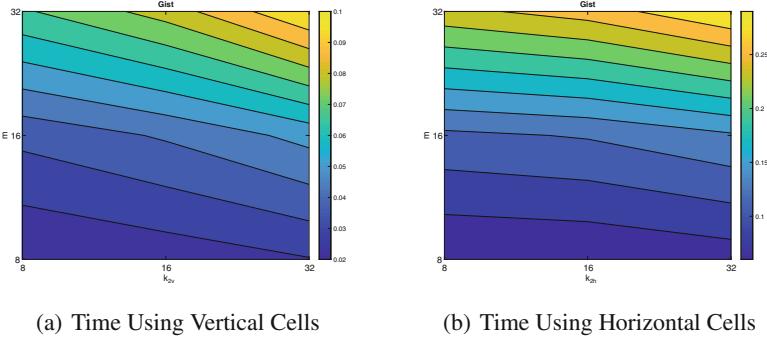
### 4.3 Computational Cost

A low error is an important issue to consider a descriptor as a good solution. But it is also important to consider the computational cost. For that reason, the necessary time to run the algorithms has also been studied. Figures 10 and 11 show the time the machine uses to describe two images and calculate the distance between them. The data are given in seconds. As we explain in Sect. 2, the results depend on descriptors' size and this one depends on the parameters. Taking that into account, HOG descriptor size is  $\mathbf{d} \in \mathbb{R}^{b \cdot k_1 \times 1}$ , in the same way Gist descriptor size is  $\mathbf{d} \in \mathbb{R}^{2 \cdot m \cdot k_2 \times 1}$ .

The experiments has been carried out with a CPU 8-Core Intel Xeon E5® at 3 GHz and using the mathematical tool Matlab®. These time results are not absolute, they depend of the computer which runs the process. But they are comparable because all the calculations have been done with the same machine.



**Fig. 10.** HOG. Localization time (s) per image versus  $k_1$  and  $b$ .



**Fig. 11.** Gist. Localization time (s) per image versus  $k_2$  and  $m$ .

The lower the parameters are, the shorter the descriptor is and the lower the runtime will be. Generally, it is quicker to solve the problem with horizontal cells because it has to transform the images into a vector and to compare both with *cityblock* distance, while using vertical cells the images are described, then the algorithm calculates the relative orientation and rotates one of the images in order to obtain the same relative orientation. Once both images are in the same relative orientation the algorithm calculates the distance. This rotational process increases the computational cost with vertical cells.

Comparing the figures it can be observed that the number of cells,  $k_1$  and  $k_2$ , has less influence in the runtime. Results with HOG and horizontal cells are the quickest, it carries out the process in 0.0125 s, whereas with vertical cells it costs 0.0736 s. Meanwhile using Gist descriptor the process is done in 0.063 s using horizontal cells and in 0.0218 s using vertical cells.

## 5 Conclusion

The current work explains two ways to describe images and it compares both in a localization task. The study was made in an environment where lighting conditions and human activity have negative effects on the localization task. Using only visual information, global appearance descriptors have been compared using horizontal cells and vertical ones. Once the images are described the performance of these descriptors in a localization framework is studied. Both the geometric localization error and the computational cost of the process are studied, and the parameters of the descriptors are optimised.

On the one hand, Gist and HOG offer a relatively good results in cloudy and night environments. The sunny conditions result more challenging, and HOG presents comparatively better results than gist in this case. Talking about vertical or horizontal cells, we consider that with HOG the results do not present substantial differences, so it is possible to use both. Finally, when Gist and vertical cells are used, the error increases prominently so it is not suitable. About times, when vertical cells are used the computational cost increases because the algorithm should describe the images, rotate one of them to have the same relative orientation and then compare both descriptors.

This work can be the first step to design a more suitable solution where both description alternatives can complement themselves in order to achieve a robust localization task. Vertical cells should not be considered as a unique solution but they can contribute towards obtaining a robust solution, in combination with other measurements. In this direction, in a real localization, the classical description method can be combined with other methods and measurements to optimize the localization error.

**Acknowledgements.** This work has been supported by the Generalitat Valenciana through grants ACIF/2018/224 and ACIF/2017/146 and through the project AICO/2019/031: “Creación de modelos jerárquicos y localización robusta de robots móviles en entornos sociales”.

## References

1. Angeli, A., Doncieux, S., Meyer, J.A., Filliat, D.: Visual topological slam and global localization. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 4300–4305. IEEE (2009)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
3. Berenguer, Y., Payá, L., Peidró, A., Gil, A., Reinoso, O.: Nearest position estimation using omnidirectional images and global appearance descriptors. In: Robot 2015: Second Iberian Robotics Conference, pp. 517–529. Springer (2016)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893, June 2005
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)

6. Menegatti, E., Maeda, T., Ishiguro, H.: Image-based memory for robot navigation using properties of omnidirectional images. *Robot. Auton. Syst.* **47**(4), 251–267 (2004). <http://www.sciencedirect.com/science/article/pii/S0921889004000582>
7. Murillo, A.C., Guerrero, J.J., Sagües, C.: SURF features for efficient robot localization with omnidirectional images. In: 2007 IEEE International Conference on Robotics and Automation, pp. 3901–3907. IEEE (2007)
8. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Comput. Vis.* **42**(3), 145–175 (2001)
9. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. *Prog. Brain Res.* **155**, 23–36 (2006)
10. Payá, L., Fernández, L., Reinoso, Ó., Gil, A., Úbeda, D.: Appearance-based dense maps creation-comparison of compression techniques with panoramic images. In: ICINCO-RA, pp. 250–255 (2009)
11. Payá, L., Reinoso, O., Berenguer, Y., Úbeda, D.: Using omnidirectional vision to create a model of the environment: a comparative evaluation of global-appearance descriptors. *J. Sens.* (2016)
12. Pronobis, A., Caputo, B.: COLD: COsy localization database. *Int. J. Robot. Res. (IJRR)* **28**(5), 588–594 (2009). [http://www.pronobis.pro/publications/pronobis\\_2009ijrr](http://www.pronobis.pro/publications/pronobis_2009ijrr)
13. Radon, J.: 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Classic Papers Mod. Diagn. Radiol.* **5**, 21 (2005)
14. Román, V., Payá, L., Reinoso, Ó.: Evaluating the robustness of global appearance descriptors in a visual localization task, under changing lighting conditions. In: ICINCO-RA, pp. 258–265 (2018)
15. Saito, M., Kitaguchi, K.: Appearance based robot localization using regression models. *IFAC Proc. Vol.* **39**(16), 584–589 (2006)
16. Siagian, C., Itti, L.: Biologically inspired mobile robot vision localization. *IEEE Trans. Rob.* **25**(4), 861–873 (2009)
17. Sturm, P., Ramalingam, S., Tardif, J.P., Gasparini, S., Barreto, J., et al.: Camera models and fundamental concepts used in geometric computer vision. *Found. Trends® Comput. Graph. Vis.* **6**(1–2), 1–183 (2011)

# **Radar-Based Applications for Robotics**



# Asynchronous mmWave Radar Interference for Indoor Intrusion Detection

Edgar S. Gonçalves<sup>1</sup>, Francisco C. Teixeira<sup>1</sup>, Daniel F. Albuquerque<sup>1,2(✉)</sup>,  
and Eurico F. Pedrosa<sup>1</sup>

<sup>1</sup> IEETA, DETI, University of Aveiro, 3810-193 Aveiro, Portugal

<sup>2</sup> CI&DETS, ESTGV, Polytechnic Institute of Viseu, 3504-510 Viseu, Portugal

[dfa@estgv.ipv.pt](mailto:dfa@estgv.ipv.pt)

**Abstract.** This paper describes the development of a system for indoor intrusion detection that takes advantage of interference between asynchronous millimetre-wave radars. The approach exploits the information embedded in the interference pattern observed in the Doppler domain when two or more radars operate in a common environment and share the same frequency spectrum. By continuously monitoring the interference, it is possible to detect the corresponding energy variations. A sharp decrease in the interference energy is thus interpreted as an intrusion of an object or a person. Within this approach the source of the interference can be identified taking advantage of beam-forming of MIMO radars. Compared with the standard configuration, which exploits the reflection of radar signals, the proposed setup has the advantage of maximizing the energy available for intrusion detection and an increased capacity of obstacles and walls penetration. When combined with the capacity of mobile robots to dynamically position the radars, this scheme permits the implementation of highly versatile intrusion detection solutions.

**Keywords:** mmWave radar · FMCW radar · Indoor intrusion · Radar interference

## 1 Introduction

An automatic indoor intrusion detection system should detect any unforeseen indoor movement without human intervention. Most of the current available intrusion detection systems are vision-based and such systems only operate well with good light conditions. To overcome this problem, artificial infrared lights or video enhancement methods could be used [8]. However, such systems, besides consuming a considerable amount of power, cannot detect movements behind walls or even small objects. An alternative to vision-based systems are sound-based detection schemes [9]. But this type of system fails to detect sound bellow a given threshold and they might produce false alarms due to ambient noise.

In this paper, we propose an indoor intrusion detection system that is based on radar, more precisely, based in the interference produced by multiple millimetre wave (mmWave) radars operating in the same indoor environment. The approach takes advantage of radars that could be already deployed for other purposes, such as obstacle detection in mobile robotics. For such, a collection of compact stand-alone radar kits was assembled, permitting the deployment in strategical static places or in mobile robots. Equipped with these radars, a scenario can be imagined were a set of mobile robots using these radars could adjust themselves inside the building in order to detect the intrusion in a predetermined area or even through a random patrolling path maximising the resources to cover most of the intrusion events. The use of radar for indoor intrusion detection appears as an interesting alternative to other technologies by overcoming most of their drawbacks, namely visibility limitations, false alarms and miss detection [4, 7, 10].

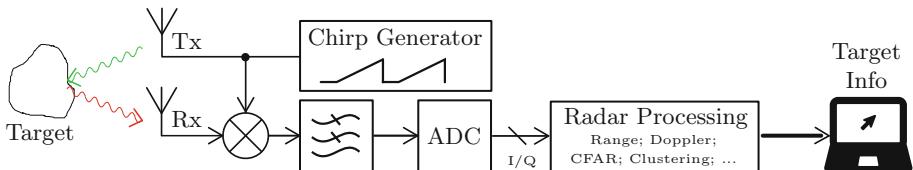
Millimetre wave radar systems are finding wide acceptance in automotive and industrial applications. More recently they are also being adopted in the robotics domain thanks to their low power consumption, compact hardware and high range resolution capabilities [1]. These set of characteristics are accomplished by using high frequency and wideband signals combined with standard ADCs operating at low sampling rates [11]. Currently there are some concerns related to the use of radars in vehicles on public road which demands a solution to detect and mitigate the interference between independent radars sharing a common frequency spectrum [2, 6]. With growing popularity in the robotics area, it is foreseen that problems associated with interference between independent radars will also increase in indoor environments. In this paper the interference phenomenon between mmWave radars will be studied and a method that takes advantage of these interferences is used for indoor intrusion detection without requiring extra hardware.

The remaining of this document is divided into four sections: Sect. 2 describes the general operation of a FMCW radar, introducing it in the context of this work. Section 3 is focused on the interference caused by multiple FMCW radars. Some simulations are presented and also a method to extract information from the interference phenomenon. Section 4 describes the used setup including the system model, hardware and software as well as the results that were achieved. Section 5 synthesizes the work developed and presents its main conclusions.

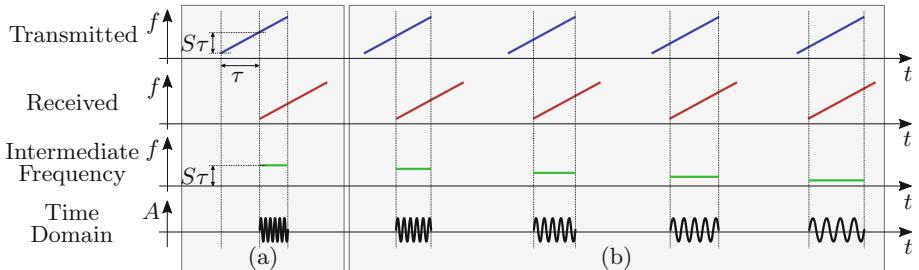
## 2 Basic Principles of FMCW Millimetre Wave Radar

Frequency Modulated Continuous Wave (FMCW) radars are based on the homodyne Radio Frequency (RF) front-end principle as described by the simplified diagram in Fig. 1 [3]. In a FMCW radar, a train (also referred to as a frame) of RF generated chirps (signals in which frequency increases linearly with time) are sent in the direction of a target through a transmitter antenna. Afterwards, the echo produced by the target is captured by the receiver antenna  $\tau$  seconds later and it is mixed with the transmitting signal. Due to the time-varying

frequency of the signals the delay between transmitted and received signal produces a beat signal at an intermediate frequency (IF) which is proportional to the target distance. Figure 2 illustrates an analysis, in frequency and time, of the method used to estimate the range from the Intermediate Frequency (IF) that results from mixing the sent chirp and the received reflection of that same chirp for a single target. This signal is converted to digital, using an ADC, and the radar data of interest is obtained through digital signal processing using an integrated microprocessor (or any other signal processing tool); a first FFT is applied to determine the target distance, a second FFT across the sequence of chirps is implemented to measure the associated target speed (from Doppler); other thresholding/clustering algorithms could also be applied [13]. At the end, a group of points with the target information (range and radial velocity) is provided to an external component, represented by the laptop in Fig. 1.



**Fig. 1.** FMCW radar working principle.



**Fig. 2.** Example of a transmitted and received chirp frame. (a) illustrates a single chirp transmission and (b) extends the representation for a frame of chirps where the reflected chirp has successively shorter delay compared to the emitted chirp (due to target motion).

## 2.1 Non-interference Radar Operation

Ideally, in a typical radar operation, the received RF signal main characteristics are well-known since it was produced by a target reflection of the signal sent by the radar itself. Apart from the target main reflection, multipath phenomenon could occur which implies that there are multiple detected reflections

for the same target. Nevertheless, such multipath reflections will necessarily travel a larger distance and will be more attenuated than those received through direct path [12]. Therefore, the strongest reflections should be associated to a larger and closer target, which are the ones supposed to be detected in first place.

Considering that the round-trip delay is represented by  $\tau$  and assuming that the signal travelling speed is close to the speed of light  $c$ , the resultant travel distance  $d$  can be expressed as:

$$d = c \times \frac{\tau}{2}. \quad (1)$$

Additionally, the tone frequency of the resulting signal in Intermediate Frequency is given by:

$$f_{IF} = \frac{f_1 - f_0}{T_c} \times \tau = S \times \tau, \quad (2)$$

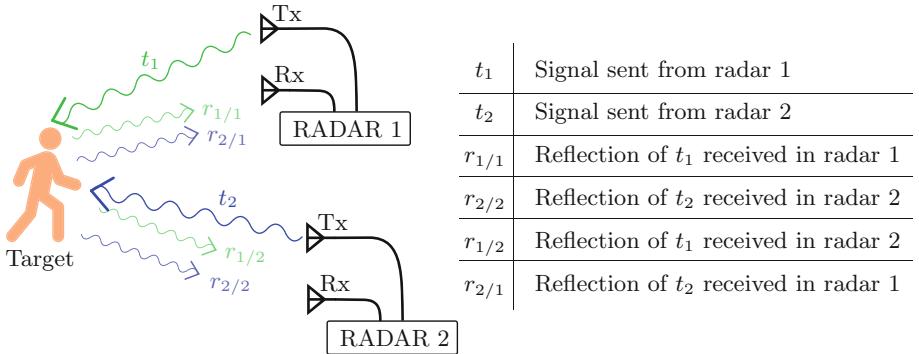
where  $f_0$  is the starting frequency,  $f_1$  is the final frequency,  $T_c$  is the time that the chirp signal takes to sweep from  $f_0$  to  $f_1$  and  $S$  is the chirp slope. Assuming that the tone frequency is obtained by measuring the peak of a FFT performed on the sampled IF signal, combining Eqs. 1 and 2, the distance to the target could be computed from this measured frequency as:

$$d = \frac{c}{2 \times S} \times f_{IF}. \quad (3)$$

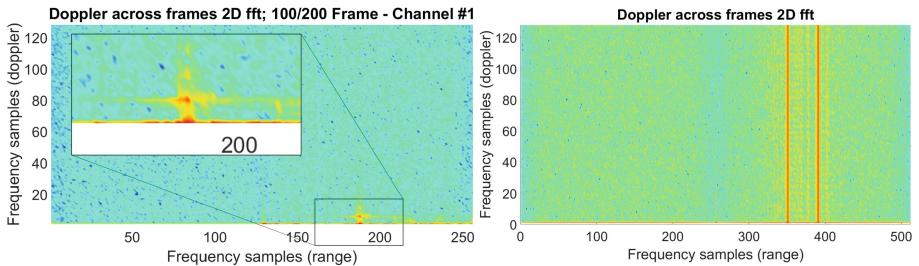
### 3 Multiple Radar Interference

Figure 3 illustrates an environment shared by two spatially separated and independent radars, where  $t_1$  and  $t_2$  are the signals sent from each radar, that will reflect on the target represented by the human shape and bounce back to both radars. Therefore the signals represented by  $r_{1/2}$  and  $r_{2/1}$  are causing interference in both radars.

In a real scenario where multiple radars are emitting simultaneously, despite being possible to mitigate the interference it could lead to confusing them with a target when observing only the range information (first FFT). However, analysing the Doppler information (second FFT) it will reveal a Doppler spreading that cannot be caused by a real target. This effect can be observed in Fig. 4 where is presented a comparison between the Doppler information of a regular radar detection of a moving target and the Doppler information of interference. From Fig. 4 one can notice that a real target produces a sharp Doppler point whereas the interference produces a spreading in the Doppler information. This phenomenon is easily explained due to the use of different clocks in each radar that produce a random phase shift between the signals sent by each radar. Due to the fact that it is not possible to have a target, at a certain range, that is moving at every speed in the Doppler domain, it is clear that it is possible to detect an interference based on the Doppler spreading effect.



**Fig. 3.** Two radars sharing the same environment and target.

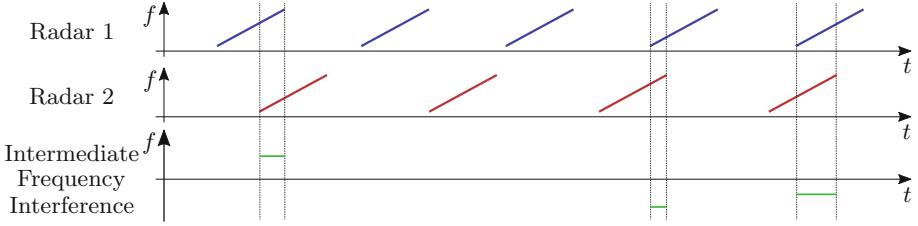


**Fig. 4.** Range and Doppler FFT of a real moving target (left) and interference (right).

### 3.1 Radar Setup for Interference Detection

To study the interference between multiple radars, two independent radars configured with the same parameters can be operated in the same environment. In order to ensure that only the interference is observed, the emitter antenna of the receiver radar was disabled. With this setup, both radars are independent and their signals are uncorrelated in time. Actually, with the available hardware, even if a common trigger signal was employed, synchronising the radars would not be feasible in practice due to the inherent clock drifts and the high frequency signals employed. Considering that the frames of both radars are not synchronised, the chirp parameters must be adjusted in order to ensure one or more interferences in the frame. One method to potentiate the occurrence of interferences is by setting a different chirp idle time (time interval that separates consecutive chirps in the frame) for each radar module, in order to increase the overlapping of frames from both radars as illustrated in Fig. 5. As result of this difference, the number of interference occurrences ( $N_I$ ) is given by:

$$N_I = \frac{N_C \times T_{C_{RX}}}{lcm(T_{C_{TX}}, T_{C_{RX}})} \quad (4)$$



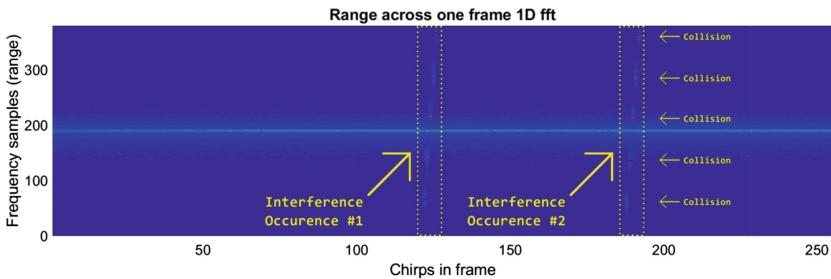
**Fig. 5.** Interference analysis between two radars with the same chirp parameters but with different idle time between chirps.

where  $N_C$  is the number of chirps in a frame, the  $\text{lcm}(\cdot, \cdot)$  function is the least common multiple between the two values and  $T_{C_{RX}}$  and  $T_{C_{TX}}$  are the chirps total duration that takes into account not only the chirp duration but also the idle time for the receiver and transmitter radar respectively.

In order to study the interference phenomenon, a simulation was performed using realistic radar parameters in accordance to the known specifications of the hardware available for practical tests. It was observed that the phenomenon is not fully characterised only by the number of interferences in the frame. In some cases each interference occurrence was composed by several collisions, i.e. in one interference sequenced chirps collide in different areas of the spectrum resulting in different peaks in the intermediate frequency. The number of collisions per interference ( $N_{CI}$ ) is calculated as follows:

$$N_{CI} = \frac{F_S}{S \times \Delta T_{idle}} \quad (5)$$

This interference and collisions phenomenon was later confirmed to occur in practical results as one can observe in Fig. 6, where it is illustrated what is considered an interference and what is considered a collision in this context.



**Fig. 6.** Acquired data with 1st FFT performed for each chirp in one frame represented. Interference occurrence is displayed between the yellow columns and each occurrence has a number of collisions pointed by the horizontal arrows.

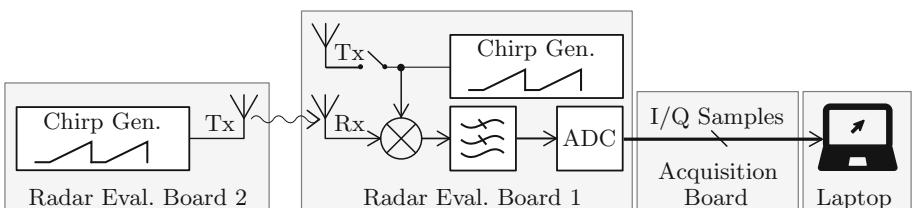
### 3.2 Intrusion Detection

The described radar setup is parametrised in order to generate interferences in each frame. In the majority of applications this is undesirable and can be seen as a worst case scenario where there are multiple radars sharing the spectrum. However, if this interferences are periodical we can exploit them to monitor an area of interest according to the placement of the radar elements. In such scenario, radar signals travel one-way from the emitter to the receiver, so in principle it is received with more power than in regular radar reflections. The increased power of signals available for interference detection becomes an advantage because it allows for coverage of larger areas and the penetration of several types of obstacles. By exploiting these methods, it is possible to configure a scenario where two radars are creating a channel with constant interference. If this interference signal is sampled at an adequate rate (the frequency of the frame) and the channel is crossed by a moving obstacle (person or robot), a variation on the received power of the interference signal is detected. Filtering this signal in order to detect variations on the same scale of people or robots movements speed, it is possible not only to detect intrusion on the area but also to characterise the movement as fast or slow [5].

## 4 Experiments and Results

### 4.1 Experimental Setup

The experimental setup employed in our tests is composed by two radar evaluation boards AWR1642-BOOST, see Fig. 7. The receiver (Radar 1) is connected to the expansion board DCA1000EVM that enables the acquisition of the raw signal sampled directly from the ADC of each receiving channel. Note that the radar board AWR1642 has four receiving antennas that could be used for beam-forming.



**Fig. 7.** Experimental setup block diagram.

The emitter radar (Radar 2) is responsible for producing interference on the receiver radar (Radar 1) by send chirp frames triggered by a Raspberry Pi (Model 3B+). Since there is no additional signal processing required for Radar 2, the Raspberry Pi is a viable solution to implement the configuration with the

radar evaluation board. Moreover, the factory-configured firmware implemented in the emitter radar was configured in order to maximise the frame duty-cycle. The duty-cycle is understood as the ratio between the time that the radar is transmitting a chirp and the overall time of the frame. On the other hand, the receiver radar (Radar 1) is responsible for generating chirp frames, correlating them with the received frames, and sending the IF sampled data to the acquisition board for further signal processing in the laptop. Additionally, TX antenna was temporary disabled to ensure that it is only capture interference signals. A photograph of the used equipments is shown in Fig. 8.



**Fig. 8.** Receiver and emitter hardware. Receiver composed by one AWR1642BOOST board, DCA1000EVM acquisition board and laptop. Emitter in bottom right corner composed by one AWR1642BOOST board and Raspberry Pi3B+.

## 4.2 Signal Processing Setup

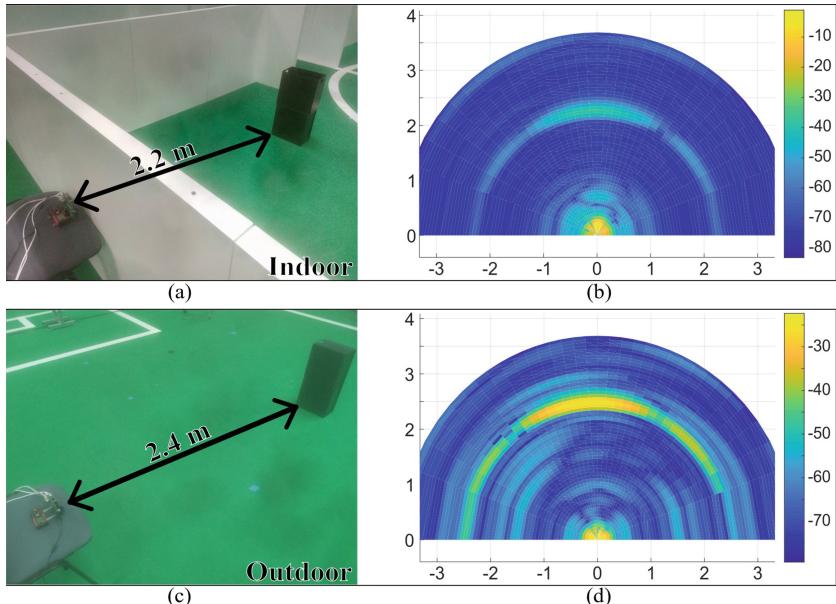
The signal processing is performed by the laptop running a software package from Texas Instruments: mmWave Studio<sup>1</sup>. This software is responsible to interact with the radar board, the acquisition board, and save the acquired samples or display the range and Doppler information. Despite this software having the capability to display the results of these post processing features it was considered impractical to fully analyse the data. Therefore, MATLAB was used to manipulate the data with more versatility, reading the raw data and processing it.

In order to obtain the range information of one frame, the first FFT is performed across the ADC samples dimension, resulting in a graph as shown in Fig. 6. Afterwards, the Doppler information is obtained by performing a FFT on the range data across the chirps dimension and it results in a graph similar to the ones shown in Fig. 4. With the multiple antennas receiving signals, each with its own ADC, digital beam-forming can be performed with post processing in MATLAB to estimate the interference angle-of-arrival.

<sup>1</sup> <http://www.ti.com/tool/MMWAVE-STUDIO>.

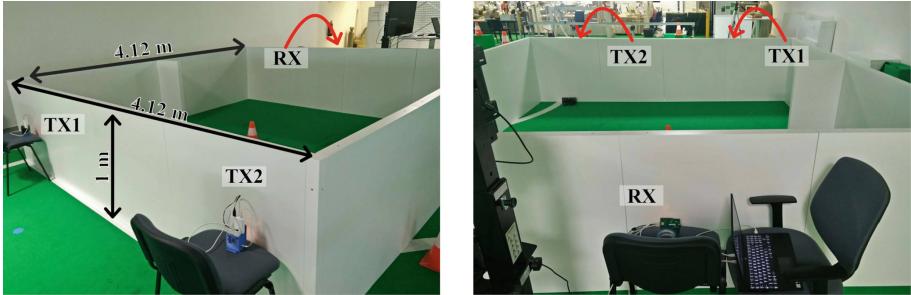
### 4.3 Experimental Data Acquisition and Processing

Initially, tests were performed with the objective to characterise the signal penetration of the radar operating in monostatic configuration. The tests were conducted on the indoor open space of our robotics laboratory, Figs. 9 and 10, and their characteristics are presented in Table 1. The walls used in the compartments represented in the indoor scenarios of are 60 mm thick, made of medium-density fibreboard, with semi hollowed infill and with 12 mm of shell.



**Fig. 9.** Target detection by a radar with (a and b) and without (c and d) wall obstacle in the middle radar.

In the **First test**, depicted in Fig. 9a, a radar was placed near a wall, facing the interior of the room, with a metallic object placed inside approximately 2.2 m apart. Figure 9b is a plot that depicts the target location in meters in relation to the radar, being the brighter colours associated to higher intensity of reflections. The **Second test**, presenting a normal usage of the radar, was performed with the same object at approximately the same distance but without any obstacle in the middle. The corresponding result of this experiment can be observed in Fig. 9c and d, where it is seen that the object clearly corresponds to the spot with higher intensity in the plot, around the coordinate (0,2) meters. In the **First test** the radar signal reflected from the target has to penetrate the wall twice (when travelling in both directions) with a consequent attenuation of signal



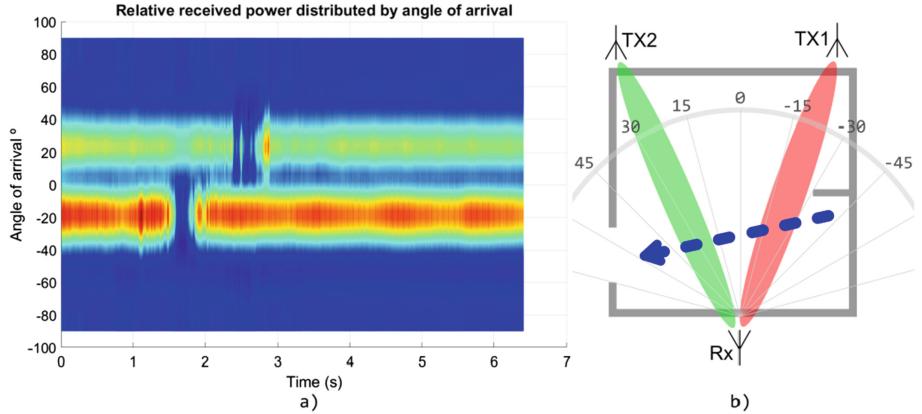
**Fig. 10.** Photography of the indoor scenario used for third test (different point of views represented in the left and the right pictures).

power. Additionally, this way it loses significantly more power when compared to the alternative that is presented in this work where the signal that penetrates the wall and reaches the receiver is the one sent directly from the emitter.

**Table 1.** Characteristics of each experimental test.

	No of radars	No of walls	Radar configuration	Type of target
<b>First test</b>	1	1	Reflection	Static
<b>Second test</b>	1	0	Reflection	Static
<b>Third test</b>	3	2	Interference	Mobile

Finally, the **Third test**, see Fig. 10, involved a person walking in the middle of a square compartment, from one side of the compartment to the other, with a path parallel to radars TX1 and TX2 according to Fig. 11b. Moreover, radars TX1 and TX2 were placed outside of the compartment and the radar RX was placed in the opposite side of the compartment. As can be observed in Fig. 11, the two horizontal stripes correspond to the signals received at Radar RX, sent from the two radars TX1 and TX2, each with its own angle of arrival in relation to the receiver. The interruption caused by the person walking in the indoor environment is represented in Fig. 11 by the received power being considerably reduced in two occasions (one close to 1.8 s from radar TX1 and the other close to 2.5 s from TX2). This result illustrates the feasibility of using asynchronous radars to create a channel with a steady state of interfering patterns that can be monitored along time. Assuming this steady state to correspond to a non-intruded environment, variations of the interference power above a given threshold can be assumed to be caused by an intruder in the interior of the environment that crosses the channel between a pair of radars.



**Fig. 11.** Peak power received by radar RX (a) and top view of the scenario (b) in third test (Higher received power is depicted with warmer colours).

## 5 Conclusion

This work proposes a method to exploit the availability of inexpensive mmWave radar systems to detect intrusions in an indoor environment. A study on the interference of FMCW radars was performed in simulations and further validated by experimental trials demonstrating that with this setup it is possible to acquire enough information to observe an intrusion based on radar interference detection mechanisms. The suggested method can be implemented without being noticed in the indoor environment because the used signals are able to penetrate the wall materials of common indoor scenarios allowing the radars to be placed on outside the compartments of interest. Moreover, the received signal can be resolved in terms of angle-of-arrival creating the possibility to expand the system to use more than two radars, which allows the receiver to detect the intrusion direction. The compact configuration of the radar kits used in the proposed scheme enable deploying the radars in mobile robots and dynamically positioning them in order to configure an optimal sensor arrangement. This versatility permits the implementation of highly effective intrusion detection solutions.

**Acknowledgment.** Project RETIOT, “Reflectometry Technologies to Enhance the Future Internet of Things and Cyber-Physical Systems”, funded by FEEI though COMPETE 2020 and Fundação para a Ciência e Tecnologia (FCT) under contract POCI-01-0145-FEDER-016432. This work is also financed by national funds through FCT under the project UID/CEC/00127/2019 and the project UID/Multi/04016/2016. Furthermore we would like to thank the University of Aveiro (IEETA) and Polytechnic Institute of Viseu (CI&DETS) for their support.

## References

1. Adams, M., Adams, M.D., Jose, E.: Robotic Navigation and Mapping with Radar. Artech House, Norwood (2012)
2. Beise, H.P., Stifter, T., Schröder, U.: Virtual interference study for FMCW and PMCW radar. In: 2018 11th German Microwave Conference (GeMiC), pp. 351–354 (2018)
3. Brooker, G.M.: Understanding millimetre wave FMCW radars. In: 1st international Conference on Sensing Technology, pp. 152–157 (2005)
4. Galati, G., Piracci, E.G., Ferri, M.: High resolution, millimeter-wave radar applications to airport safety. In: 2016 8th International Conference on Ultrawideband and Ultrashort Impulse Signals (UWBUSIS), pp. 21–26. IEEE (2016)
5. Gonçalves, E.: Radar through wall system. Master's thesis, University of Aveiro (2018)
6. Heuel, S.: Automotive radar interference test. In: 2017 18th International Radar Symposium (IRS), pp. 1–7. IEEE (2017)
7. Kiuru, T., Metso, M., Jardak, S., Pursula, P., Häkli, J., Hirvonen, M., Sepponen, R.: Movement and respiration detection using statistical properties of the FMCW radar signal. In: 2016 Global Symposium on Millimeter Waves (GSMM) and ESA Workshop on Millimetre-Wave Technology and Applications, pp. 1–4. IEEE (2016)
8. Ko, S., Yu, S., Kang, W., Park, C., Lee, S., Paik, J.: Artifact-free low-light video enhancement using temporal similarity and guide map. IEEE Trans. Industr. Electron. **64**(8), 6392–6401 (2017)
9. Lee, S.Q., Park, K.H., Kim, K., Ryu, H., Wang, S.: Practical implementation of intrusion detection method based on the sound field variation. In: Proceedings of the 20th International Congress on Sound and Vibration, Bangkok, Thailand, pp. 7–11 (2013)
10. Morinaga, M., Nagasaku, T., Shinoda, H., Kondoh, H.: 24GHz intruder detection radar with beam-switched area coverage. In: 2007 IEEE/MTT-S International Microwave Symposium, pp. 389–392. IEEE (2007)
11. Nowok, S., Kueppers, S., Cetinkaya, H., Schroeder, M., Herschel, R.: Millimeter wave radar for high resolution 3D near field imaging for robotics and security scans. In: 2017 18th International Radar Symposium (IRS), pp. 1–10. IEEE (2017)
12. Wagner, J., Strobel, A., Joram, N., Eickhoff, R., Ellinger, F.: FMCW system aspects for multipath environments. In: 2011 8th Workshop on Positioning, Navigation and Communication, pp. 89–93. IEEE (2011)
13. Winkler, V.: Range Doppler detection for automotive FMCW radars. In: European Microwave Conference, pp. 1445–1448, October 2007



# Machine Learning Methods for Radar-Based People Detection and Tracking by Mobile Robots

José Castanheira<sup>(✉)</sup>, Francisco Curado<sup>ID</sup>, Eurico Pedrosa<sup>ID</sup>, Edgar Gonçalves,  
and Ana Tomé<sup>ID</sup>

Department of Electronics, Telecommunications and Informatics Engineering,  
University of Aveiro, 3810-193 Aveiro, Portugal  
[{jp.castanheira,fcurado,epg,easg,ana}@ua.pt](mailto:{jp.castanheira,fcurado,epg,easg,ana}@ua.pt)

**Abstract.** This paper reports a machine learning approach for people detection and tracking in indoor environments using a compact radar system deployed by a mobile robot. The set-up described in the paper includes a series of experiments carried out in an indoor scenario involving walking people and dummies representative of other moving objects. In these experiments, distinct learning models (a neural network and a random forest) were explored with different combinations of radar features to achieve person versus non-person classification.

**Keywords:** RADAR · Mobile robotics · Machine learning · People detection

## 1 Introduction

### 1.1 Motivation

The utilization of standard sensors such as cameras, laser or infrared in order to detect the presence of people and other objects in a given scenario of interest is often hampered by the diversity of environmental effects such as dust, fog, rain, strong heat or fire. In contrast, radar signals are immune to these disturbances. Additionally, radar systems can be easily parameterized to deal with a diversity of operational conditions. In recent years, the number and diversity of applications of radar sensors have been increasing dramatically motivated by the emergence of inexpensive, self-contained development kits. These devices permit rapid prototyping and testing of radar-based solutions and their reduced cost and dimension make them affordable for academic research and permit their integration in small, inexpensive robots.

Despite the aforementioned advantages, robust feature extraction from radar data is still an open problem because the radar physics is prone to generate a diversity of artifacts that pose significant challenges to conventional signal processing techniques. This problem is especially dramatic in indoor environments due to the ubiquity of obstacles (clutter) normally present in these scenarios.

The present paper introduces a novel machine learning approach integrated with mobile robotic platforms to address the problem of radar-based detection of moving people in an indoor scenario. To the best of our knowledge, none of the alternative works address the specific problem that we tackle in this paper: the detection of people (walking and on wheelchair) in indoor environments based on statistical measures of radar Doppler data acquired by mobile robots. The envisioned applications include elderly monitoring and rescue operations in vision-denied environments. Other related works have focused on the use of micro-Doppler data to detect human motions. Micro-Doppler (effect) is defined as frequency modulation on the returned radar signal, that is a result of the target's micro-motion dynamics, such as mechanical vibrations. Despite the high potential of the approach, exploitation of this effect with the radar kits on hand is not suitable for the current application given the demanding signal processing power required to resolve the micro-Doppler components induced by the targets while simultaneously dealing with the envisaged sensing requirements: (1) maximum detection range, and maximum unambiguous velocity of different moving objects, and (2) the inherent vibrations of mobile robots (carrying the radar sensors) that induce micro-Doppler effects susceptible of masking those of the targets. The paper does not discuss other sensor alternatives (e.g. vision) because the radar is intended to work as a complementary instrument or as a replacement sensor in scenarios where the alternative sensors are prone to fail.

## 1.2 Summary of Prior Work

According to the main data types and the signal processing techniques applied, the approaches proposed in the literature to detect and track moving persons can be divided into the following categories:

**Doppler-Based Detection and Range/Doppler/Azimuth Based Tracking.** This approach is applied in demonstrations and technical documents provided by the Texas Instruments manufacturer of the xWR1642 and xWR1443 families of FMCW radars; see, e.g., [11] and [8]. It is also adopted in the works of [12] and [7]. These methods distinguish moving people from clutter based essentially on Doppler data (representative of the velocity of the target relatively to the radar); static objects can be easily classified as clutter due to their zero-Doppler attribute while non-zero Doppler measurements associated to people motion are used to initiate and propagate target tracks in time; data association performed for each target relies on additional information regarding the target position in 2D, including its range and bearing provided by a phased array radar. The approach has been applied successfully to track multiple people with a single, low-cost MIMO radar but it fails in two relevant aspects: (i) it cannot distinguish moving people from other moving targets of approximately the same dimensions (or radar cross section) and (ii) it cannot detect or track a non-walking person even if he/she is performing some task (seated or standing).

**Range/Doppler-Based Detection and Range/Doppler/Azimuth Based Tracking.** This approach exploits the extended velocity profile (or velocity dis-

persion) characteristic of human motion that stems from the fact that humans do not move as rigid objects but rather present typical oscillating movements of their members and body; for example, there is a typical spreading and contraction pattern of the velocity profile, with a sinusoidal character, in the case of movement of arms and legs of a walking pedestrian. This approach is obviously more robust and holds a larger potential of application in most scenarios where humans are required to interact with other mobile agents (including mobile robots, small vehicles, and other machines with moving parts in general). An interesting implementation of Range/Doppler-based detection is described in [6]. The authors apply a Support Vector Machine to classify pedestrians, vehicles, and other objects in an urban area, using an automotive 24 GHz radar sensor with a bandwidth of 150 MHz as the measuring device. Another potential application of the Range-Doppler detection approach consists of exploiting the patterns of hands movements of a person in order to detect her presence and even to interpret his gestures (for example as remote commands for a robot) [2].

**Returned Signal Strength-Based Detection.** This approach exploits measurements of signal intensity returned from the target that may be used directly, as in the case of classification schemes based on the computed absolute radar cross section (RCS) of targets, or indirectly by exploiting the relative intensity of the echos corresponding to distinct scatterers of the same target. RCS is a property of the target's reflectivity, commonly used to detect and classify airplanes in a wide variation of ranges. In principle, the technique can be applied to distinguish terrestrial targets, because cars, motorcycles, bicycles, and pedestrians have distinct RCS values. In practice, the method may be difficult to apply due to the large dependence of RCS measurements on the orientation of the target relatively to the radar; see, e.g., [9]. To the best of our knowledge RCS has not been applied before to detection and tracking of people.

A comprehensive revision of the use of radar systems together with the application of machine learning for people detection tasks can be found in [5]. The operation of radars installed on mobile robots, particularly in indoor environments, is still poorly covered in the literature. Some notable exceptions are [1, 2, 10], and [3]. However, none of those works tackles the problem of people detection and tracking by moving robots that is addressed in this work.

### 1.3 Main Contributions of the Paper

The main contribution of this paper is the introduction of a new, robust approach for classification of moving objects based on radar measurements acquired by mobile robots in indoor scenarios. In terms of formulation and implementation, the main novelties of the work are: (i) the exploitation of the Doppler histograms representative of the velocity distribution of the points that constitute the point cloud acquired by the radar, originated from the multiple radar scatterers of a given object as a feature that permits a robust distinction between moving rigid bodies and people, and (ii) the utilization of the radar cross section of targets as a classification feature; we introduce here a practical method for its computation

and propose its application in order to implement more robust classifiers and trackers of people.

## 2 Terminology, Problem Formulation and Solution

### 2.1 Basic Terminology and Problem Formulation

**Basic Terminology.** In radar terminology, any object that can be detected by the radar is often designate as *target*. The scalar measure representative of the radial velocity of the target, i.e. its velocity projected onto the central axis of the radar transmitter, is designated as *Doppler* since it is computed based on the frequency shift (Doppler effect) incurred by the radar wave due to the target velocity. A *frame* is a fixed-size packet of reflectivity data (chirp reflections) that characterizes the state of the target in terms of its distance (range) and radial velocity (Doppler) relatively to the radar. The multiple-input multiple-output (MIMO) radar technology considered in this work permits the discrimination of simultaneous reflections from different points in the plane of the radar wave by exploiting the beam-forming capability of the MIMO system. In the present context, a *point cloud* is a collection of points corresponding to individual reflections from a scene acquired at a given instant of time and represented in different positions of the radar sensor grid; these points may correspond to different reflections from a single object or from multiple targets.

**Problem Formulation.** The problem addressed here can be formulate as follows: *Given an indoor scenario characterized by the presence of a diversity of static and mobile objects, implement a method to unambiguously detect the presence of a walking person and estimate its kinematic properties using range and Doppler data acquired with a low-cost radar installed on a mobile robot.*

### 2.2 Proposed Solution

Among other characteristics, the motion of people is characterized by the pendular movements of arms and legs. Other objects, such as robots, usually move as a block. The pattern of motion of people is, therefore, different from the pattern of motion of other mobile agents, and thus, capturing this pattern over time and feeding it to a learning model can provide a mean to dynamically detect people and distinguish them from other objects.

Another issue is that the RCS of a person (even when seated) can be distinguished from that of other objects. Hence, the RCS may be used as a feature in order to make the classification more robust.

**Radar System.** The RADAR device used was the AWR1642-BOOST EVM from Texas Instruments. Details of the radar signal can be found in Table 1.

**Radar Data Processing.** Although the discussion of this topic is beyond the scope of this work, the RCS, represented by the Greek letter  $\sigma$ , can be computed as

$$\sigma = \frac{4\pi^3 d^4 P_r}{\lambda G P_t}, \quad (1)$$

where  $d$  denotes the distance to the object,  $P_t$  and  $P_r$  represent the power emitted by the radar and reflected by the target, respectively,  $\lambda$  is the wave length of the radar signal, and  $G$  is the ratio between the transmission and reception gain.

The velocity of the points belonging to the point cloud of a frame are represented by a histogram. The histogram is calculated between  $-6$  and  $6$  m/s with 150 bins. Therefore the bin size (resolution) is 0,08 m/s. Based on this, the target kinematics is represented as an image by the concatenation of the Doppler histograms of the successive point cloud of all frames. For a better visualization, the values of the bins are color coded; see e.g. Figs. 3 and 4.

Based on this configuration, the data acquired in each test is transformed into a single dataset constituted by 151 elements (150 velocity bins plus one value RCS) and with a number of data samples equal to the number of frames.

**Data Clustering.** When performing measurements using the RADAR, clutter may coexist with the point cloud of an object of interest. It is then important not only to detect this clutter, but to eliminate it, ensuring that these points will not be included in the creation of the dataset. The obvious way to segment the point cloud corresponding to a target is by using an appropriate clustering method. The method chosen in the current implementation was DBSCAN (Density-based spatial clustering of applications with noise), [4]. DBSCAN is a clustering method that relies solely on two base parameters: minimum number of points and minimum distance between them for a set of points to be considered a cluster. Henceforth, minimum number of points will be designated as *MinPts* and the minimum distance will be designated by *Eps*. The parameter values of *MinPts* = 2 and *Eps* = 0.6 were chosen based on practical experiments. Although several clusters can be detected with this method, in the tests performed, we only consider one object in front of the device at a time. Therefore in all our experimental set-ups for each data frame, the largest cluster is considered the cluster of interest.

**Table 1.** Chirp parameters and respective values.

Chirp parameter (units)	Value	Chirp parameter (units)	Value
Start Frequency (GHz)	77	Maximum unambiguous range (m)	5
Slope (MHz/us)	60	Maximum radial velocity (m/s)	5.2936
Samples per chirp	128	Azimuth resolution (degrees)	14.5
Chirps per frame	256	Velocity resolution (m/s)	0.0827
Frame duration (ms)	50	Number of transmission antennas	2
Sampling rate (Msps)	2.5000	Range resolution (m)	0.0488
Bandwidth (GHz)	3.0720	Number of reception antennas	4

**Table 2.** ANN hyperparameters. *hl* stands for hidden layers.

Activation function	Relu	Num. of hl	4
Num. of neurons (1st to last layer)	100, 70, 50, 20	Learning rate	0.001
Maximum iterations	10000	Random state	42

## 2.3 Description of Learning Models Applied

The learning models used in this paper are implemented in the well-known Python machine learning library named **scikit-learn**. Each model is dependent on a set of parameters. The parameters for which the respective value is omitted are considered to use the default values defined in the aforementioned library. The models applied were an Artificial Neural Network (ANN) and a Random Forest. The hyperparameters for the ANN are defined in Table 2. The Random Forest was designed as having a number of estimators of 500 and a maximum depth of 600, and the random state is set to 0.

The ANN was chosen due to its traditional applicability to image processing and classification, and the good results it has achieved in this field. Because the statistical Doppler patterns are represented as 2D images, ANN is an adequate model to apply using this type of data. The Random Forest was chosen because it is a model different from the ANN, and allows us to investigate how this model works in this type of classification task.

## 3 Experiments

The reported experiments consist of a series of tests performed with a radar installed on a small robot, the TurtleBot. In order to address the problem on hand with varying degrees of complexity, two main configurations have been contemplated: one where the radar is installed on a static robot, and another with the radar deployed by a moving robot. In both cases, the moving targets (people and objects) are made to travel along the same set of pre-defined paths shown in Fig. 1.

### 3.1 Scenario Configuration

The tests executed for radar data acquisition were performed in an indoor wide open area, free of obstacles. This simplified scenario is proposed, similarly to other works described in the literature, in order to permit a proof of concept involving the type of classification methods proposed here. The host computer responsible for recording and storing the measurements communicates with the sensor via a Raspberry Pi connected directly to the sensor. The radar device is placed on top of the TurtleBot.

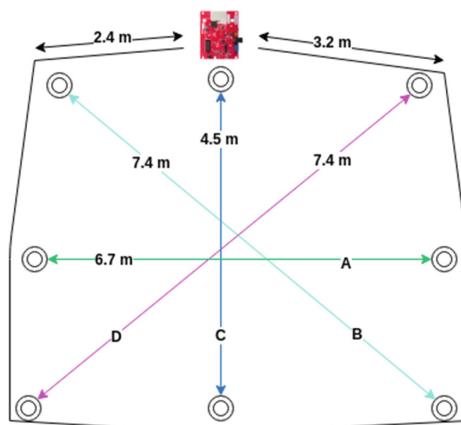
A set of blue marks was placed on the ground in order to serve as a visual guide for the targets and to locate the corners of the field of view (see Fig. 1).

### 3.2 Types of Tests Carried Out

**Tests with Radar Installed on Static Robot.** In order to enable a learning model to distinguish people from other objects, representative data sets of both classes of targets had to be provided. Therefore, tests were made with people but also with a mannequin with a dimension close to that of an adult. Furthermore, with the intent of making this application as inclusive as possible, people with limited mobility were also taken into account by performing tests with a person driving an automated wheelchair.

Each elementary test in this experiment consisted of an object moving in a specific direction within the field of view of the radar. In the test field, four types of directions were considered: from one corner to the opposite one (left to right and right to left); from the top centre of the field to the bottom centre; from the centre side of the field to the other centre side. For each direction, the two ways were considered, yielding therefore a total of 8 different trajectories, as shown in Fig. 1. For each trajectory, 10 different tests were performed, comprising a total of 80 tests for each specific object. Three different types of objects were considered: person walking, person in a wheelchair, and a mannequin. A picture of the mannequin is shown in Fig. 2, as well as the wheelchair used, and the radar positioned on the TurtleBot.

The typical pattern of velocity distribution acquired by the radar when a person is walking in front and towards the device (trajectory C) is depicted in Fig. 3; the plot clearly shows the dispersion of velocity values associated to the walking person. For other directions, such as diagonal or transversely to the orientation of the radar, the pattern is similar. Figure 4 shows the velocity pattern of the mannequin and of a person in a wheelchair, moving in front and



**Fig. 1.** Schematic of the test field with the trajectories considered represented by bidirectional non-black arrows, and the limits of the field of view represented by circumferences. The length of each trajectory is represented in meters, and each one has also a letter identifier.

towards the radar. As can be seen, the velocity pattern of the locomotion of the mannequin is clearly different from that of a person. Most of the points of a person-related point cloud are located within a defined region of the Doppler domain, and this can be explained by the fact that most of the points represent reflections from the torso and thus move with the same velocity. However, as a result of the motion of the arms and legs, there is a clear spread of points along the Doppler dimension. This spread is symmetrical (both in the positive and negative side) and approximately periodic. This is in accordance with the movement of the limbs, which have a velocity superior to the rest of the body and travel in opposite directions. On the other hand, the motion pattern of the mannequin is rather different. As the mannequin moves as a block, the majority of the points have approximately the same Doppler value.

**Tests with Radar Deployed by a Moving Robot.** The tests performed with the radar deployed by the moving robot involved the same targets (people, mannequin, and wheelchair) moving along the same paths as used in the tests executed with the static robot. In the case of the tests performed with the moving robot, the robot travelled along a straight trajectory that crossed or converged with the path of the other targets; see Fig. 1.

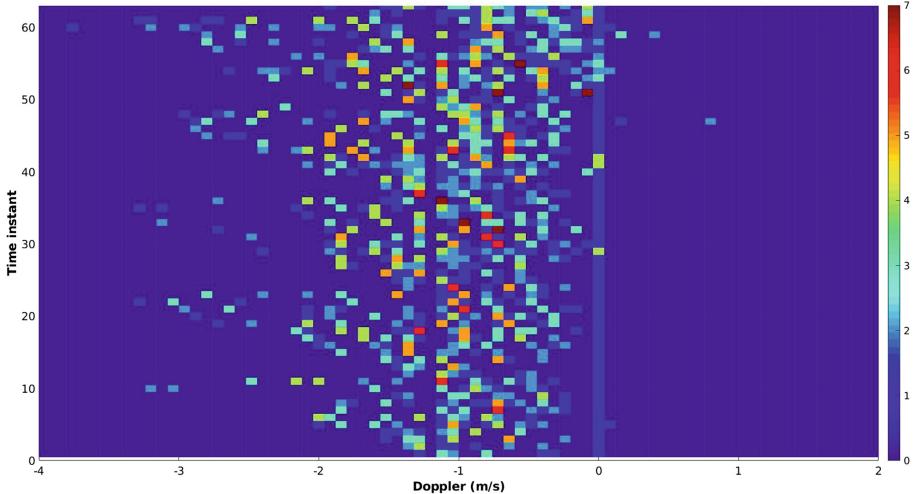
## 4 Results with Radar Installed on the Static Robot

The dataset acquired in this experiment consists of a total of 40430 frames. Out of these 40430 frames, 21872 correspond to people moving (walking or moving in a wheelchair), and the remaining 18558 correspond to other moving objects. For classification purposes, 25% of the dataset was used for test, while the remaining was applied in the training stage. The dataset was previously shuffled to ensure an equal representation of both classes (people moving and other moving objects) in both the training and testing set.

The classification based on the ANN described in Subsect. 2.3, and using as features the normalized histogram bins as well as the RCS, achieved an accuracy



**Fig. 2.** Wheelchair (left) and mannequin (center) used in the tests. TurtleBot with radar installed on its top (right), with an upward inclination of 10°.

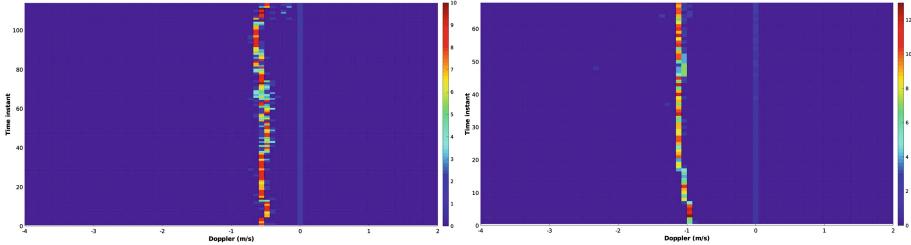


**Fig. 3.** Histogramic representation of Doppler measures of a person walking in front and to the device (trajectory C), with the TurtleBot immobilized. The Doppler axis only ranges from  $-4 \text{ m/s}$  to  $2 \text{ m/s}$  because in this case there are no points beyond those limits. The colour bar on the right represents the absolute number of points.

of 96.1%. Following the standard notation of this field, accuracy is defined as  $\frac{TP+TN}{N}$ , where  $TP$  and  $TN$  stands for true positives and true negatives (respectively), and  $N$  represents the total number of examples. Using the same ANN with similar hyperparameters but excluding the feature RCS, the performance was 92.0%. Using the Random Forest, the accuracy attained was 95.3%. As in the case of the ANN, when the RCS is not used as a feature the accuracy decreases, in this case for a value of 91.9%. The confusion matrices of the ANN and Random Forest are represented in Tables 3 and 4, respectively. Note that, although we only reference the accuracy, all the other metrics (such as precision, recall, etc) can be derived from the confusion matrices.

**Table 3.** ANN confusion matrices in the case of an immobilized TurtleBot.

	Predicted Not Person (with RCS)	Predicted Person (with RCS)
Not Person	4487	175
Person	217	5229
	Predicted Not Person (without RCS)	Predicted Person (without RCS)
Not Person	4357	305
Person	506	4940



**Fig. 4.** Histogramic representation of Doppler measures of a mannequin (left) and person in wheelchair (right) moving in front and to the device, with the TurtleBot immobilized. The Doppler scale is the same as the one in Fig. 3. Notice the dispersion of Doppler of the mannequin, which is caused by small oscillatory movements of its pending arms combined with a certain component of rotation of the body during towing of the dummy. In contrast, the person in the automated wheelchair is practically still. The colour bar on the right of each histogram represents the absolute number of points.

## 5 Results with Radar Installed on the Moving Robot

### 5.1 Constraints

When the radar is installed on a static platform, the objects that contribute with points to the point cloud generated by the radar are only those who have an apparent movement relatively to the sensor. However, in this experiment, as the sensor is moving, all objects exhibit apparent movement relatively to the radar, and thus, all objects contribute with points to the point cloud generated. Hence, in this experiment the point cloud generated is larger than when the radar was immobilized. The first tests revealed that this increase of information (points) is enough to make the radar application provided by the manufacturer to crash. To solve this problem the radar configuration was modified, reducing the number of ADC samples collected per chirp, from 128 to 64.

One of the issues observed in this experiment was that the clustering method wasn't robust enough to detect and retrieve only the cluster of interest. Due to the above mentioned reasons, when the sensor is moving, more clusters are generated than when the robot is immobilized. Thus, choosing the largest cluster as the cluster of interest does not guarantee that the cluster corresponds to the

**Table 4.** Random Forest confusion matrices in the case of an immobilized TurtleBot.

	Predicted Not Person (with RCS)	Predicted Person (with RCS)
Not Person	4450	212
Person	263	5183
	Predicted Not Person (without RCS)	Predicted Person (without RCS)
Not Person	4316	346
Person	468	4978

target of interest; it may be generated by any sufficiently large object nearby, with an apparent motion. To resolve this, an heuristic-based tracking algorithm was developed in order to detect a cluster and to follow it along a sequence of frames, and to detect if the cluster chosen to initialize the track is the right cluster originated with data from the target of interest; see Algorithm 1. This algorithm considers that from frame to frame, the same cluster can not travel more than 1.6 m, because otherwise, it would have a velocity higher than what is expected from objects moving in the environment under study. On the other hand, the total distance traversed by a cluster along the experiment must be equal or greater than 1.5 m. Otherwise, it would mean the cluster barely moved, which can not be the case for the cluster of interest.

## 5.2 Results

In the current experiment, the complete dataset consists of a total of 25431 frames. Out of these 25431 frames, 12720 correspond to people moving, and the remaining 12711 correspond to other moving objects. 25% of the dataset was used for test, while the remaining was applied in the training stage.

Using the ANN described in Subsect. 2.3 and considering the RCS as a feature, the accuracy attained was 97.7%, while ignoring the RCS the performance was 96.2%. For the Random Forest, the accuracy were 97.8% using the RCS as a feature, and 96.5% without RCS. The resultant confusion matrices of the ANN and Random Forest are represented in Tables 5 and 6, respectively.

---

### Algorithm 1. Heuristic tracking algorithm

---

```

Iteration = 1; TotDistance = 0;
while there are data frames do
    if Iteration == 1 then
        Compute and store centroid of cluster;
    else
        Calculate closest centroid to previous one;
        Store centroid of cluster;
        Displacement = distance between previous and closest centroid;
        if Displacement > 1.6 then
            Iteration = 1; TotDistance = 0;
            Start on different cluster;
            Continue;
        end
        TotDistance = TotDistance + Displacement;
    end
    if Iteration == total number of frames then
        if TotDistance < 1.5 then
            Iteration = 1; TotDistance = 0;
            Start on different cluster;
            Continue;
        end
    end
    Iteration = Iteration + 1;
end

```

---

**Table 5.** ANN confusion matrices in the case of a moving robot.

	Predicted Not Person (with RCS)	Predicted Person (with RCS)
Not Person	3135	66
Person	83	3074
	Predicted Not Person (without RCS)	Predicted Person (without RCS)
Not Person	3076	125
Person	115	3042

**Table 6.** Random Forest confusion matrices in the case of a moving robot.

	Predicted Not Person (with RCS)	Predicted Person (with RCS)
Not Person	3133	68
Person	74	3083
	Predicted Not Person (without RCS)	Predicted Person (without RCS)
Not Person	3070	131
Person	91	3066

## 6 Discussion of Results and Conclusions

As can be seen from the results presented, the performance of the classification models tested in this study clearly evidence the high potential of the methods to detect people in indoor environments and to support the application envisaged in our work. Although the tests with the moving TurtleBot hampered the tracking and distinction of the cluster of interest, the methods had a better performance in this case than when the robot was immobilized. In all cases, the classification models had their performance deteriorated when the RCS was not used as a feature. In the experiments where the TurtleBot was immobilized, the ANN had a better performance than the Random Forests. In the experiments where the TurtleBot is moving, the Random Forests had a better performance than the ANN. The RCS had a greater impact on performance in the experiments where the TurtleBot is static, comparing to the experiments where it is moving.

We showed that the approach is effective in two distinct scenarios (robot immobilized and robot moving) which can lead to a wide range of applications in robotics. The presence of clutter and its deleterious effect, which can be typical in indoor environments, was also solved with a simple heuristic tracking algorithm.

In future work we intend to test the same methods in more problematic scenarios (e.g. with furniture and other obstacles) in order to assess their performance and the potential adequacy of different models to the different environments. Moreover, different parameters of the classification models will be also tested, and their results evaluated. The configuration of the radar device can also be altered with the purpose of detecting velocities and distances with a higher resolution or with larger ranges. The output rate can also be increased. By modifying the configuration of the radar, we can study how the different

parameters affect the quality of the data and the performance of the model, and ultimately determine the best radar parameters. Another interesting application is, besides detecting people, to detect other features such as the different trajectories travelled by each object.

**Acknowledgement.** This work is funded by Research Project RETIOT PT2020-03/SAICT/2015 - Fundação para a Ciência e Tecnologia.

## References

1. Barrett, D., Alvarez, A.: mmWave radar sensors in robotics applications. Technical report, Texas Instruments (2017)
2. Berkius, C., Buck, M., Gustafsson, J., Kauppinen, M.: Human control of mobile robots using hand gestures. Bachelor thesis in electrical engineering, Chalmers University of Technology, Gothenburg, Sweden (2018)
3. Dogru, S., Marques, L.: Using radar for grid-based indoor mapping. In: Proceedings of the 19th IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2019, Gondomar, Porto, Portugal, 24–25 April 2019
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD 1996, pp. 226–231. AAAI Press (1996)
5. Gurbuz, S.Z., Amin, M.G.: Radar-based human-motion recognition with deep learning: promising applications for indoor monitoring. *IEEE Signal Process. Mag.* **36**(4), 16–28 (2019). <https://doi.org/10.1109/MSP.2018.2890128>
6. Heuel, S., Rohling, H.: Pedestrian recognition based on 24 GHz radar sensors. Ultra-Wideband Radio Technologies for Communications, Localization and Sensor Applications, chap. 10, pp. 241–256. InTech (2013)
7. Knudde, N., Vandersmissen, B., Parashar, K., Couckuyt, I., Jalalvand, A., Bourdoux, A., Neve, W.D., Dhaene, T.: Indoor tracking of multiple persons with a 77 GHz MIMO FMCW radar. In: 2017 European Radar Conference (EURAD), pp. 61–64 (2017)
8. Livshitz, M.: Tracking radar targets with multiple reflection points (2018). [https://e2e.ti.com/cfs-file/\\_key/communityserver-discussions-components-files/1023/Tracking-radar-targets-with-multiple-reflection-points.pdf](https://e2e.ti.com/cfs-file/_key/communityserver-discussions-components-files/1023/Tracking-radar-targets-with-multiple-reflection-points.pdf). Accessed 13 June 2019
9. Machado, S., Mancheno, S.: Automotive FMCW radar development and verification methods. Master's thesis, Department of Computer Science and Engineering. Chalmers University of Technology, University of Gothenburg, Sweden (2018)
10. Takeuchi, E., Elfes, A., Roberts, J.: Localization and Place Recognition Using an Ultra-Wide Band (UWB) Radar. Springer Tracts in Advanced Robotics, vol. 105. Springer (2015)
11. Texas-Instruments: People Tracking and Counting Reference Design Using mmWave Radar Sensor. TI Designs: TIDEP-01000, March 2018
12. Yamada, H., Wakamatsu, Y., Sato, K., Yamaguchi, Y.: Indoor human detection by using Quasi-MIMO doppler radar. In: 2015 International Workshop on Antenna Technology (iWAT), pp. 35–38 (2015)



# Tracking Drones with Drones Using Millimeter Wave Radar

Sedat Dogru<sup>(✉)</sup>, Rui Baptista, and Lino Marques

Department of Electrical and Computer Engineering,  
Institute of Systems and Robotics, University of Coimbra,  
3030-290 Coimbra, Portugal  
[{sedat,rui.baptista,lino}@isr.uc.pt](mailto:{sedat,rui.baptista,lino}@isr.uc.pt)

**Abstract.** The use of drones has seen a surge in the last few years with their employment in a large variety of applications. However, this popularity has also made improper drone use a threat to privacy, economy and human lives, requiring the development of methods to detect and track drones. In this work, we present and experimentally validate an airborne drone detection system that utilizes a millimeter wave radar system to detect and follow target drones.

**Keywords:** MMW radar · Drone · Detection · Tracking · Unmanned Aerial Vehicle · UAV

## 1 Introduction

The latest improvements in technology have made drones more available, cheaper, and easier to use, increasing their utilization both, for professional and for hobby uses. Although drone use is being regulated and common sense would prevent harm from use of drones, reckless behavior of hobby users is not uncommon, even harming themselves at times<sup>1</sup>. Drones flying around airports present a danger due to a possible collision course with a plane during landing or take-off, and such incidents are becoming wide spread in recent times<sup>2,3,4</sup>. The damage

<sup>1</sup> <https://www.irishtimes.com/news/ireland/irish-news/man-suffers-serious-burns-while-attempting-to-recover-drone-1.3940450>.

<sup>2</sup> <https://www.theguardian.com/technology/2019/jun/17/flights-delayed-as-drones-fly-near-east-midlands-airport-download-festival>.

<sup>3</sup> <https://www.todayonline.com/commentary/what-can-singapore-do-counter-rogue-drones-and-airport-incursions>.

<sup>4</sup> <https://www.todayonline.com/unauthorised-drone-flying-over-changi-airport-causes-37-flight-delays>.

---

This work has been supported by OE - national funds of FCT/MCTES (PIDDAC) under project UID/EEA/00048/2019.

caused by drones is not restricted to airports only. They may hit over infrastructure<sup>5</sup>, invade privacy<sup>6</sup>, or they may even be used for terrorist attacks<sup>7</sup>. These concerns have raised the need to spot, and at some times destroy, intruding drones.

Drones have several characteristics that allow them to be detected using different approaches, such as visual [1, 2, 7, 11, 12], audio [2, 12], Radio Frequency (RF) [4, 9, 10, 12], infrared [2, 8], and radar [1, 3, 9]. Each approach has some strengths, and inevitably some weaknesses. Audio detection relies on detecting and identifying the sound of propellers on the drones and requires discrimination of background noise. Visual inspection has been used both using high resolution daylight cameras and low resolution infrared cameras. Visual inspection still requires favorable atmospheric conditions and a meaningful distance between the cameras and the drone. A problem with fixed visual detection systems is that they are not able to easily estimate the full trajectory of the drone. Infrared cameras rely on detecting heat sources of the drones, such as batteries, motors and motor driver boards. Visual systems need to classify between birds and drones, which at a distance may not be so trivial. Radar is less affected by environmental conditions, and it has been the most commonly used method to detect flying vehicles by military units. However, conventional military radars have been optimized to detect large objects and have difficulty detecting small drones. Additionally, target discrimination may not be trivial. Better radar detection is achieved when the object is several times the size of the wavelength. Hence, for small drones radars operating at smaller wavelengths may produce better results.

In this work, we propose using a radar fixed on a drone, called follower drone, to actively detect and track target drones. This approach allows keeping a target drone always in the detection range of the follower drone. Additionally, actively tracking a drone allows localization of the target drone in low line of sight environments and even when it lands, helping security forces capture the drone and identify the owner.

The next section summarizes the related work, and then in Sect. 3 the proposed tracking algorithm is described. In the Sect. 4, the tracking algorithm is validated with experiments. Finally, Sect. 5 draws a set of conclusions from this work.

## 2 Related Work

Different groups have used cameras to detect drones. Birch and Woo [1] evaluate manual visual detection of drones at various wavelengths such as visual and various infrared regions during daylight, varying the focal lengths of the

---

<sup>5</sup> <https://www.ft.com/content/76728798-7636-11e9-be7d-6d846537acab>.

<sup>6</sup> <https://eu.usatoday.com/story/tech/columnist/2018/09/03/drone-gripes-mount-ho-meowners-complain-breached-privacy-annoyance/1117085002/>.

<sup>7</sup> <https://www.nytimes.com/2018/08/04/world/americas/venezuelan-president-targeted-in-attack-attempt-minister-says.html>.

lenses as well as the sensor size in the visible spectrum. They run experiments to detect quad-copters, octa-copters and fixed wing Unmanned Aerial Vehicles (UAVs), concluding that visual band inspection works best against a uniform background. However, Birch and Woo also estimate that detection from over 1 km would require cameras with giga-pixel resolution. Rozantsev et al. [11] work on detecting a drone from a mobile ground platform and for this purpose they use a camera mounted on the platform. The camera has a partial view of the sky, and the authors are able to detect whether the UAV and the ground platform are on a collision course. Hu et al. [7] similarly work in the visual optic region using a camera mounted at a fixed spot. They first detect the skyline, and mask the image, leaving out only the sky itself and then search in this region using change detection followed by blob analysis to detect a moving UAV. Muller [8] uses static cameras consisting of a high resolution visual band optical system for daylight operation and a low resolution Short-Wave Infra-Red (SWIR) system for detections at night. Muller uses change detection to detect candidates and removes noisy backgrounds due to both motions of surrounding objects like leaves and cars, and also noise due to increased sensitivity of SWIR, using a background model based on density calculations of pixel alarms. Exact source localization using only video images is not trivial, and acoustic detection has been used to compensate this. Christnacher et al. [2] use an acoustic antenna array with a range-gated active imaging system to successfully detect drones. The active imaging system consists of a pan-tilt camera operating at SWIR frequencies and relies on active illumination of the target using an infrared laser.

Radar has also been used in various forms to detect drones, despite the fact that drones may have a very small Radar Cross Section (RCS) making their detection with a radar, especially at long distances, very difficult [5]. Hoffman et al. [6] use a ground based multi-static radar using time domain and micro-Doppler signatures to detect small UAVs. The multi-static radar system consists of three identical nodes operating in the S-Band (2.4 GHz). Quevedo et al. [3] use a static ubiquitous radar operating in the X-Band (8.75 GHz) to detect a flying drone in a range of 2 km with 0.878 m resolution and a probability of detection (PoD) greater than 0.7. Vinogradov et al. [13], instead of using an active radar, rely on a passive radar and simulate a UAV-mounted passive radar to detect intruding drones. The simulation assumes that Long-Term Evolution (LTE) band mobile base stations operate as Illuminators of Opportunity (IOO).

A possibly less trivial drone detection method is based on the utilization of the RF signatures of the drone communication. This approach is especially successful to detect drones for hobby use since these drones depend heavily on RF communication with a ground station both for control and to relay collected video data. Drone hardware is known to use RF communication in a special way, allowing discrimination of drone communication from other RF communication present in the environment. Nguyen et al. [9] investigate different RF-based methods to detect drones like detecting RF reflections from the propellers, detection by eavesdropping the communication between the drone and the base station, and analyzing movement patterns. It has been even shown that the

body motion of the drones causes changes in the RF signals emitted by them, and this has been used by Nguyen et al. [10] to detect and classify different types of drones. Trusting on the power of data fusion, Shi et al. [12] use a stationary drone detection system fusing audio, video and RF detection. However, data fusion is done using a simple or operation on the detections of different subsystems, which independently classify the detections using a Support Vector Machine (SVM).

### 3 Method

The radar used in this work is a 2D radar reporting bearing and range of detected objects. These detected objects are not always valid targets, but either features in the environment, such as trees and the ground, or ghost objects which are usually caused by multi path reflections. Identification and removal of the non-target objects is the first step in the process. This is done using the Doppler values of the reported targets. Moving targets cause a Doppler shift in the reflected signal, which is detected by the radar. This is used to discriminate moving targets from the static environment, eventually leaving out only mobile targets in the environment. A drone, even if stationary for some time, has to move to come to a stationary position, and in this early stage of motion it can be detected. This detection can later be used in localizing the drone in the static clutter if necessary.

The second step is tracking the target drone, which will allow detection even if the target drone decides to keep fixed, and also allow the follower drone to actively follow the target drone wherever it goes when becomes necessary. This step is achieved using an Extended Kalman Filter (EKF). Although reporting just the distance and the bearing of the detected target is a limitation of radar that at first sight makes it difficult to localize an object in 3D, this limitation can be overcome by moving the follower drone, which allows observing the target drone from different positions, and eventually the reconstruction of the full path of the target drone.

In this work, a constant speed target model is assumed, hence the state, representing the latitude, longitude and altitude of the target drone in local coordinates as well as the corresponding speeds, is given by  $\mathbf{x} = (x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3)^T$ . The measurement  $\mathbf{z} = (\rho, \phi)^T$  corresponds to the distance measurement  $\rho$  and the bearing measurement  $\phi$  which are given by

$$\rho = \sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2} \quad (1)$$

$$\phi = \arctan \frac{x_2 - \xi_2}{x_1 - \xi_1} + \beta \quad (2)$$

where  $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3, \dot{\xi}_1, \dot{\xi}_2, \dot{\xi}_3)^T$  is the state of the follower drone and  $\beta$  is its heading. The state of the target drone evolves according to the discrete time linear model given as

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \zeta \quad (3)$$

$$\mathbf{z}_t = h(\mathbf{x}_t) + \eta \quad (4)$$

where  $\zeta$  is a term representing the process noise assumed as white Gaussian (assuming no input), and  $A$  is the state transition matrix given in discrete time by

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$h(\mathbf{x})$  represents the nonlinear measurement process, and  $\eta$  is the noise in the measurement.

The EKF estimates the new position of the target drone in two stages. It first predicts the new state, given by the mean  $\bar{\mu}_t$  and covariance  $\bar{\Sigma}_t$  at time  $t$ . For this, Eqs. (6) and (7) are used.

$$\bar{\mu}_t = A\mu_{t-1} \quad (6)$$

$$\bar{\Sigma}_t = A\Sigma_{t-1}A^T + Q_t \quad (7)$$

In the second stage, the filter corrects the estimate of the first stage using an innovation term and the Kalman gain using Eqs. (8), (9) and (10).

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R)^{-1} \quad (8)$$

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \quad (9)$$

$$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \quad (10)$$

where  $K$  stands for the Kalman filter gain, continuously evaluated by the filter at each step.  $Q$  and  $R$  are covariances of the noise in the state and measurement equations corresponding to the variables  $\zeta$  and  $\eta$  respectively.  $H$  stands for  $h(\mathbf{x})$  linearized around the current state estimate and it is given by

$$H = \begin{bmatrix} \frac{1}{\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2}} & 0 \\ 0 & \frac{1}{\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2}} \\ \begin{bmatrix} x_1 - \xi_1 & x_2 - \xi_2 & x_3 - \xi_3 & 0 & 0 & 0 \\ -(x_2 - \xi_2) & x_1 - \xi_1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{bmatrix}. \quad (11)$$

## 4 Experimental Setup

In this work two quad-copters of different sizes were used. The follower drone is a Sky Hero Spyder X4. It has a carbon fiber frame with a frame size of 0.85 m (Fig. 1a). It is controlled by a Pixhawk flight controller and it is powered by 400 RPM/V motors and a 6S 16000 mAh LiPo battery. The drone is additionally equipped with two IMUs, whose measurements are fused by the flight controller, a Drotek DP0601 GNSS receiver with an external GNSS antenna, a Drotek

RM3100 compass, and a Texas Instruments AWR1642 radar, which was used to track the target drone. The drone weighs in total 5.5 kg, and despite this, thanks to the large LiPo battery, it can operate up to 45 min long, allowing search and detection of targets for an extended period. The target drone, a smaller Sky Hero Little Spyder drone, has a frame size of 0.45 m. It was similarly built out of a carbon fiber frame (Fig. 1b). It is controlled by a Pixhawk flight controller, powered by 950 RPM/V motors, and carries a much smaller 4S 3700 mAh LiPo battery. This drone has two IMUs and integrated u-blox M8N receiver with an integrated GNSS antenna.

The AWR1642 is a 2D radar with 2 TX and 4 RX antennas. It operates in the 76–81 GHz frequency range and reports range and bearing of detected objects. It is a customizable radar which allows configuring various parameters such as range, resolution and thresholds. In the first set of experiments, the follower drone was equipped with a AWR1443 3D radar from the same company. The AWR1443 reports range, bearing and at a lower resolution the elevation angle. However, in our preliminary tests it was not able to detect the target drone from all angles, forcing us to use the AWR1642, which showed better detection performance. Although AWR1642 reports detections in 2D using range and bearing, it transmits a 3D electromagnetic wave enabling detection in a 3D cone, but not reporting.

## 5 Experimental Tests and Results

The tests were performed on a field with some trees (see Fig. 1c). The first test aimed not only to detect but also to properly localize the target drone. Since the radar reports only range and bearing, 3D localization of the target drone requires moving the follower drone and fusing the collected data in an EKF, as was described. Therefore for this test, the target drone kept flying at a constant position as the follower drone was moved up and down along a line, and also side to side along a horizontal line (Fig. 2a). The accuracy of fusion is shown in Fig. 2b. The error, which is the distance between the GNSS position and the position estimated by the EKF, in the vertical direction can be seen to be less than 1 standard deviation of the vertical GNSS error, and the horizontal error is close to 2 standard deviations of the GNSS error, showing that moving the drone vertically and fusing the measurements using the EKF produces accurate position estimates. In this work RTK GNSS was not available and therefore raw GNSS values reported by the drones had to be used as ground truth.

The second test aimed to evaluate the maximum detection range of the radar. For this, the follower drone was programmed to keep its pose at a constant height and fixed position. The target drone was programmed to ascend to the same height as the follower drone and fly 40 m away from it, and then return back and land at the starting position (Figs. 1c and 3a). Although the radar was set to detect objects up to 60 m away, the reflections from the drone were too weak to be detected by the radar after 25 m. The radar was able to detect the target drone sufficiently well in the first 25 m, which allowed successful reconstruction



(a)

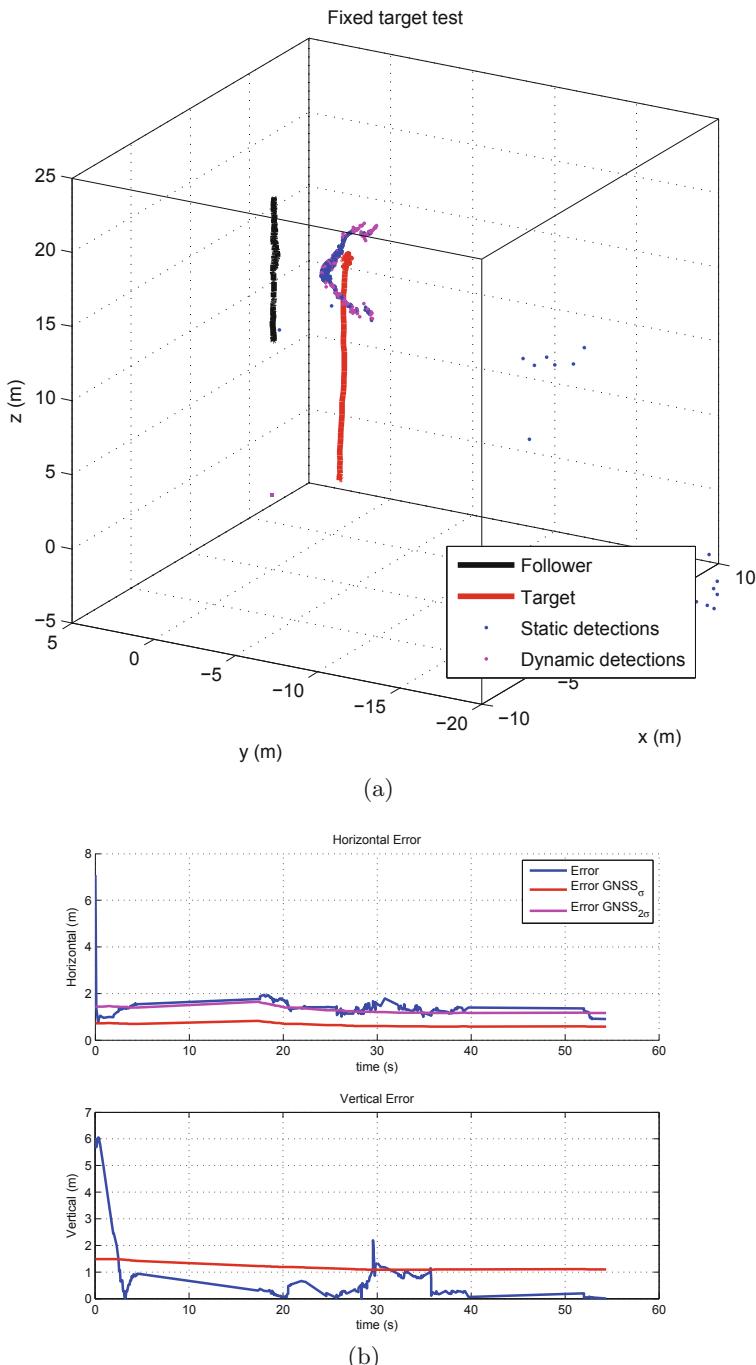
(b)



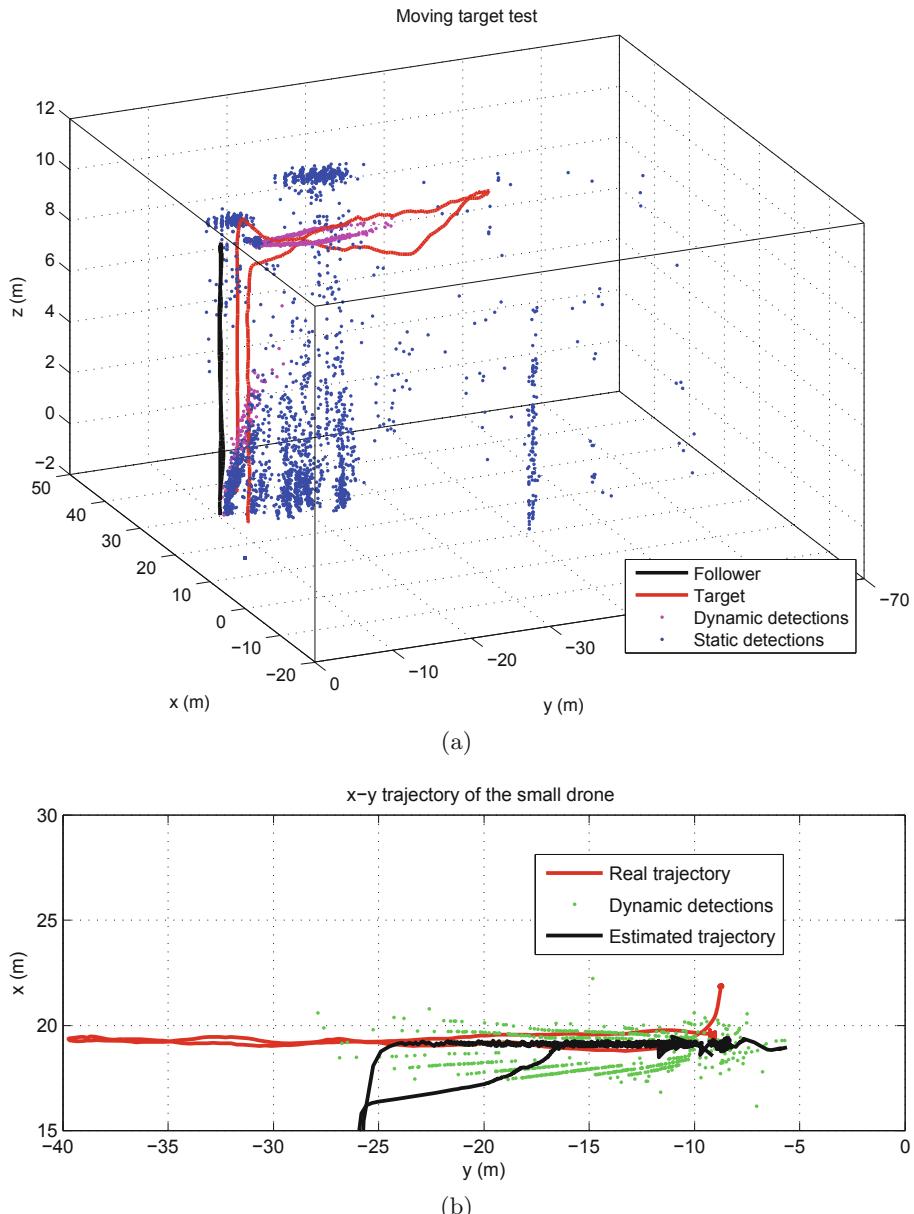
(c)

**Fig. 1.** (a) The follower drone (b) The target drone (c) Both drones in flight during a test. The follower drone is kept at constant height, the target drone is moving away from and then back to the follower drone.

of the target drone's trajectory (Fig. 3b), especially along the outgoing direction. In the return direction, the EKF takes some time to relocate the drone, which is normal because the drone was lost after 25 m, and as the drone reenters the detection range there are not enough detections to help the EKF to quickly converge. Note that, in this test the altitude of the target drone was taken to be fixed in the EKF when fusing the measurements.



**Fig. 2.** Target localization test. (a) The target drone ascends and flies at a fixed position, the follower drone flies up and down reading the shown reflections. (b) Errors in the vertical and horizontal directions of the data fusion process.



**Fig. 3.** Maximum range estimation test. (a) The target drone ascends, and then flies 30 m away from the starting point, and then returns and lands at the starting position. The follower drone ascends, and keeps flying at a fixed position. (b) View from the top of the trajectory of the target drone and the trajectory estimated through radar readings using EKF.

## 6 Conclusion

In this paper, we have presented preliminary results of tracking a drone using a radar mounted on another drone. Although the radar used in this work was not able to detect the drone at long distances, the results are promising. Further improvements to the detection algorithm in the radar, which was programmed to detect much larger objects by the manufacturer, will help increase the range. Additionally, actively following the target will reduce the importance of having a much larger range. Following this work, we are planning to develop active tracking algorithms on the follower drone that will help follow the target drone at a safe distance, eventually reconstructing its full trajectory from detection to landing. Additionally, we are planning to integrate a camera in order to help the identification of drones, avoiding possible erroneous identification of large birds as drones, and eventually help in data fusion in favorable light conditions.

## References

1. Birch, G.C., Woo, B.L.: Counter unmanned aerial systems testing: evaluation of VIS SWIR MWIR and LWIR passive imagers. Sandia Report (921) (2017)
2. Christnacher, F., Hengy, S., Laurenzis, M., Matwyschuk, A., Naz, P., Schertzer, S., Schmitt, G.: Optical and acoustical UAV detection. In: Electro-Optical Remote Sensing X, vol. 9988, p. 99880B (2016)
3. de Quevedo, D., Urzaiz, F.I., Menoyo, J.G., López, A.A.: Drone detection with X-band ubiquitous radar. In: 2018 19th International Radar Symposium (IRS), pp. 1–10 (2018)
4. Dressel, L., Kochenderfer, M.: Hunting drones with other drones: tracking a moving radio target. In: 2019 IEEE International Conference on Robotics and Automation (ICRA) (2019)
5. Farlik, J., Kratky, M., Casar, J., Stary, V.: Radar cross section and detection of small unmanned aerial vehicles. In: 2016 17th International Conference on Mechatronics - Mechatronika (ME), pp. 1–7 (2016)
6. Hoffmann, F., Ritchie, M., Fioranelli, F., Charlish, A., Griffiths, H.: Micro-Doppler based detection and tracking of UAVs with multistatic radar. In: 2016 IEEE Radar Conference (RadarConf), pp. 1–6 (2016)
7. Hu, S., Goldman, G.H., Borel-Donohue, C.C.: Detection of unmanned aerial vehicles using a visible camera system. *Appl. Opt.* **56**(3), B214–B221 (2017)
8. Müller, T.: Robust drone detection for day/night counter-UAV with static VIS and SWIR cameras. In: Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VIII, vol. 10190, p. 1019018 (2017)
9. Nguyen, P., Ravindranatha, M., Nguyen, A., Han, R., Vu, T.: Investigating cost-effective RF-based detection of drones. In: Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, DroNet 2016, pp. 17–22. ACM, New York (2016)
10. Nguyen, P., Truong, H., Ravindranathan, M., Nguyen, A., Han, R., Vu, T.: Matthan: drone presence detection by identifying physical signatures in the drone's RF communication. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2017, pp. 211–224. ACM, New York (2017)

11. Rozantsev, A., Lepetit, V., Fua, P.: Flying objects detection from a single moving camera. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4128–4136 (2015)
12. Shi, X., Yang, C., Xie, W., Liang, C., Shi, Z., Chen, J.: Anti-drone system with multiple surveillance technologies: architecture, implementation, and challenges. *IEEE Commun. Mag.* **56**(4), 68–74 (2018)
13. Vinogradov, E., Kovalev, D.A., Pollin, S.: Simulation and detection performance evaluation of a UAV-mounted passive radar. In: 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1185–1191 (2018)

# **Rehabilitation and Assistive Robotics**



# Classification of Daily Activities Using an Intelligent Tip for Crutches

Asier Brull, Aitor Gorrotxategi, Asier Zubizarreta<sup>(✉)</sup>, Itziar Cabanes,  
and Ana Rodriguez-Larrad

University of the Basque Country (UPV/EHU), Bilbao, Spain  
`{asier.brull,aitor.gorrotxategi,asier.zubizarreta,itziar.cabanes,  
ana.rodriguez}@ehu.eus`

**Abstract.** Individualization of rehabilitation therapies for gait recovery is usually guided by subjective assessments based on the perception of the patient or the therapist. The use of sensorized devices can provide more objective data to adapt the therapy to each patient. In this work, an approach to recognise different daily activities is proposed based on the data captured by an instrumented tip for crutches. The developed approach is based on two steps: a pre-processing based on obtaining a set of statistical indicators, and an artificial neural network that process them. The proposed methodology has been tested in a group of 13 volunteers, allowing to recognize three critical daily activities (walking, standing still and going up/down stairs) with a success rate of 88%.

**Keywords:** Gait monitoring · Rehabilitation · Artificial neural network

## 1 Introduction

Neurological disorders have become one of the main concerns in developed countries due to their high impact on society and economy. The World Health Organization (WHO) calculates that nearly 1/6 of world population is affected by these disorders, representing nearly 12% of the deaths in a year basis [12].

One of the most challenging neurological disorders is Multiple Sclerosis (MS) due to the physical and cognitive decline it causes. It is estimated that nearly 2.3 million people are affected by this disease, which usually affects young women and men in their thirties [5]. The MS is a chronic autoimmune disease in which the immune system attacks the nervous system, destroying the capability of sending signals to the different parts of the body. This way, depending on which parts of the nervous system are damaged, the evolution and symptoms of the disease are different in each patient. Usually, motor functions are diminished (force loss, spasticity, loss of balance,...), sensory and perceptual alterations appear and a gradual cognitive decay arises. In addition, muscular fatigue is one of the main symptoms of the disease, increasing in its intensity as the illness evolves.

Pharmacological treatments are under research with the aim of slowing down or even stopping the degeneration caused by this illness. However, currently there is still no cure for it. Therefore, palliative treatments and neurorehabilitation therapies are critical to reduce the illness progression and maximize the quality of life of patients. Recent studies [10] have demonstrated that appropriate rehabilitation therapies can maintain or even improve the gait ability and strength of the people with MS, and reduce the fatigue perception. However, the available evidence does not offer objective information about the volume and intensity of the assessed therapies. Thus, it is difficult to reproduce the interventions to further advance in the search for efficient therapies directed to the particular characteristics and needs of patients with MS.

Therapy individualization depends on the particular state of each patient, which is related with the capacity to perform everyday activities [4]. For instance, a more active patient that walks a couple of kilometers every day will present different needs than one that does not perform such activity. Hence, an objective quantification of the physical activity of the patients performed during the day may be a valuable tool for the efficient planification of the therapy. For this purpose, technological solutions can be used to monitor the patient. One of the most widespread solution is the use of wearable sensors [15, 16], and in particular, the use of IMUs (Inertial Measurement Units). These devices are attached to different parts of the body and measure their angular velocity and acceleration. Although their cost is not high, they require to be positioned properly in the body, and its placement is invasive for the patient. Other solution, which is more appropriate for those patients that require walking assistance, such as MS patients, is the use of instrumented crutches or canes [3, 14]. This solution is less invasive, and allows to integrate IMUs and force sensors within the cane in order to monitor gait.

Once the monitoring device provides the raw data, it has to be processed to perform the activity classification. Some researchers have designed classification approaches that use the unprocessed raw data. These usually require more complex activity recognition algorithms with increased computational cost, such as Convolutional Neural Networks [6] or traditional Multi-Layer Perceptron (MLP) neural networks with a high amount of hidden layers [18]. In order to reduce the computational burden, a two-step approach is usually carried out by the works proposed in the literature. First, the raw data is reduced by extracting a set of indicators that may be statistical (mean square, maximum, kurtosis, standard deviation, peak to peak values or interquartile range) [1, 7, 8, 13], frequency based [19] (average oscillation cycle) or phase analysis [2]. Second, a neural-network based classifier is designed based on the previously established indicators. In general, MLP neural networks provide good results for this purpose [8, 9, 11, 17]. However, their topology (hidden layer and neurons) varies significantly in each proposal as it depends on the number of indicators used to feed the neural network, the quality of the training samples and the number of activities that are able to classify (walking, running, standing still, going up the stairs, going down the stairs, etc.). Moreover, most of the approaches are based only on the data

provided by wearable IMUs, neglecting other sensor sources such as force or barometric pressure that may enhance the classification capabilities.

Hence, in this work a novel classification approach is presented based on the data provided by an instrumented tip which provides both IMU data and force sensor data. The instrumented tip has been designed to fit any regular crutch or cane, so that it can monitor daily activities of MS patients, being this work a first step toward this goal. The developed activity recognition algorithms are computationally efficient and is based on six significant statistical indicators, that are able to classify three critical activities: standing still, walking and going up/down stairs.

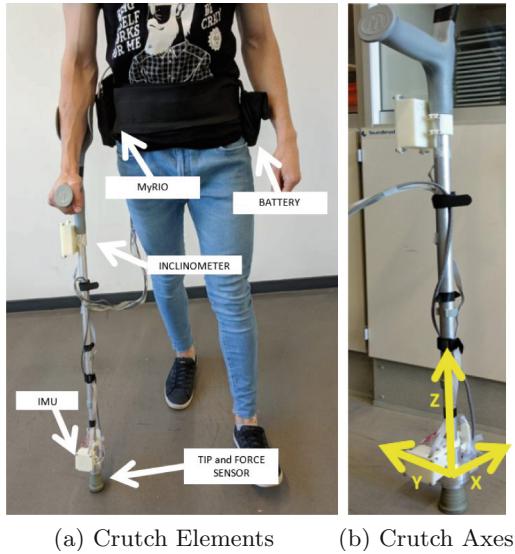
The rest of the paper is structured as follows. In Sect. 2 the instrumented tip design, the integrated sensors and its acquisition system are detailed. Section 3 details the dataset used to develop the recognizing algorithm. Section 4 shows the indicators selected for the activity classification. Section 5 defines the methodology to train the neural network to perform the activity recognition. In Sect. 6 the designed activity recognition system is tested, and its success rate detailed. Finally, the most important ideas are summarized in Sect. 7.

## 2 Instrumented Tip

Figure 1a details the proposed instrumented tip developed for monitoring daily activities. The tip, whose custom-designed structure has been manufactured in light aluminium, is designed to fit any regular crutch or cane easily using a bolt-based attachment. Its structure houses a series of sensors and conditioning circuits that provide information about the movement of the crutch or cane it is attached to. The acquisition system and the power source are housed in a belt that is connected using a cable to the tip. This disposition eliminates the need to allocate this extra weight in the tip of the crutch or cane, making its motion easier for a patient.

As previously stated, the instrumented tip allows monitoring the motion and force of the crutch or cane it is connected to. In order to measure the longitudinal axis of the crutch, an HBM C9C piezoelectric force sensor has been integrated in the aluminum structure. A mobile part within the tip transmits the force from the rubber tip of the crutch/cane to the sensor, causing small deflections that are measured by it. The selected sensor can measure up to 1200 N. However, the generated voltage is small and requires a conditioning circuit and an amplification Integrated Circuit (IC) to be properly measured.

The motion of the tip is measured using two additional sensors. A MPU-6000 Inertial Measurement Unit (IMU) has been used to provide the values of the linear acceleration and angular velocities. This IC integrates a triaxial accelerometer, a triaxial gyroscope and its corresponding Motion Processing Unit, providing information of the amplitude of the motion of the tip on its reference system. This reference system is detailed in Fig. 1b, where the  $z$  axis is aligned with the longitudinal axis of the crutch and the  $x$  axis is aligned with the front part of the crutch/cane. On the other hand, a SCA100T-D02 two axes inclinometer is also



**Fig. 1.** (a) Crutch and elements that compose it. (b) Reference Axes of the crutch.

used, which provides absolute information of the anteroposterior and lateromedial inclination angles of the tip. Please note that although the latter provides absolute inclination with respect to the ground, inclinometers do not perform properly in dynamic scenarios (accelerations, impacts,...), so its measurement will not be used in this work.

Regarding the Data Acquisition System (DAQ), a myRIO device from National Instruments has been used due to its compact form factor. This embedded system is powered by a battery that is fitted into a belt (Fig. 1a). The myRIO is programmed in Real-Time with an application to read and process the data of the aforementioned sensors periodically (50 ms acquisition cycle). This data is stored into the myRIO, and also sent via WiFi to a computer, which can be used for monitoring purposes.

### 3 Activity Database Generation

The design of a proper recognition approach requires first to define a proper database that for each of the different cases to be considered and classified, stores its corresponding raw data. This data are provided by the different sensors of the aforementioned instrumented tip. In addition, due to the differences of each person, an appropriate population of volunteers is needed to perform the tests.

Note that although the motivation behind this work is to develop solutions for MS patients, a set of healthy people were selected for developing the initial database. This was justified due to the increased complexity of carrying out tests with MS patients, and the need of validating first the methodology prior

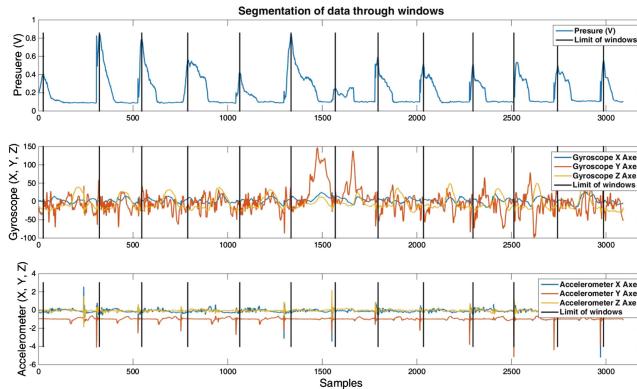
to patient testing. Hence, a total of 13 healthy people (9 Male and 4 Female) have been selected for the database generation.

As the goal is to get enough data to define each category (walking, standing still, going up/down stairs), two tests have been defined to emphasise each one of these activities:

1. Walking 27 m in a straight line, and at a constant, regular walking speed. The acceleration and deceleration phases were not considered in these 27 m.
2. Going up/down 11 stairs. First, the volunteer goes up the stairs, stops, and then goes down back to the starting point. Each volunteer repeats twice each test, generating 26 sets of data for each test

In addition, before each test was started, the volunteers were asked to remain still 5 s in the starting position, to capture data for the *standing still* activity.

Once the tests were performed, the obtained time evolution was segmented in smaller windows associated to each step. This was carried out using the force sensor signal from the crutch. In the signal captured by the instrumented tip, the stance phase (when the crutch is in contact with the ground and the maximum force is exerted) and the swing phase (when the crutch is displacing in the air with no contact) can be easily detected by searching for the maximum force peak, as seen in Fig. 2.



**Fig. 2.** Force Sensor Data and segmentation procedure illustration.

Using this procedure, a total of 553 sets of data associated to steps in a straight line (approximately 44 steps for each volunteer), 192 sets of data associated to steps going up stairs (14 steps for each volunteer) and 203 sets of data associated to steps going down stairs (15 steps for each volunteer, approximately), making a total of 395 samples for this activity (the different amount of steps up and down is due to the fact that the volunteers did not always go up/down stairs one by one). As volunteers did not perform steps when *standing still*, a worst case scenario was considered to create a *null step*. Volunteers

presented a time-step from 0.5–3 s, hence, a 3 s window was used to segment the *standing still* data, obtaining a total of 2150 sets associated to this activity. Then, this set was randomly reduced to 430, in order to equilibrate the number of samples in each activity.

Note that all tests were carried out with the appropriate Ethics Committee approval (code HC180138) and the required informed consents, in which each volunteer was informed of the use that will be given to the data obtained during the tests.

## 4 Statistical Indicators Based Preprocessing

Each of the sets associated to a step in the database stores the time evolution of the 7 different variables measured by the sensors, this is: linear accelerations in X, Y and Z axes, rotational speed in those axes and longitudinal force. Even after segmenting, they are an important amount of data that needs to be processed and analysed in order to properly sort the different activities.

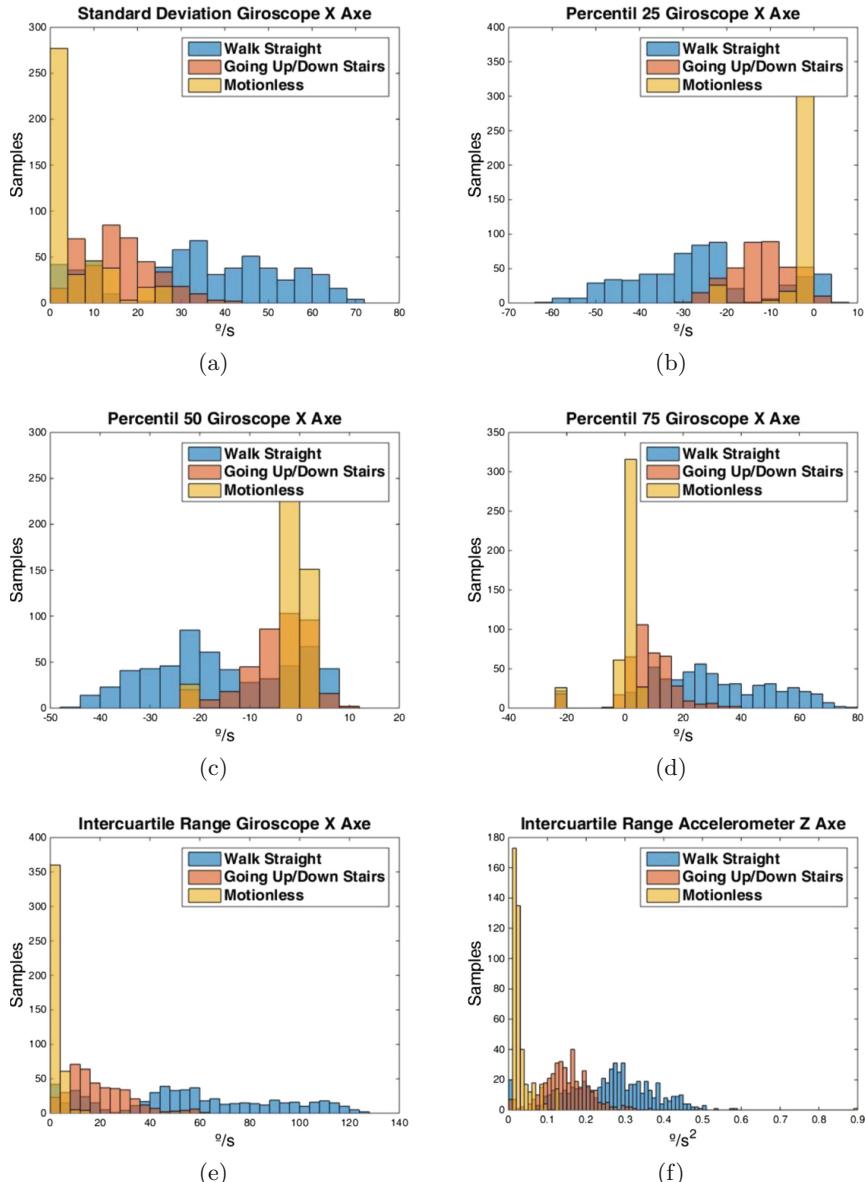
As previously detailed in the introduction, there are two approaches when dealing with the design of activity recognition. One approach considers feeding the raw data into the designed algorithm, which usually implies higher computational cost. The second one, tries to reduce the raw data to a set of indicators or features, usually statistical ones, so that designed approach is more efficient. In this work, the second approach has been selected, requiring a first analysis to define the best indicators.

In the literature, statistical operators have been usually proposed to extract significant data from the raw time evolution of gait parameters. In this work the following parameters have been analysed for each of the 7 aforementioned time variables: the mean value, the standard deviation, the kurtosis value, the 25, 50 and 75 percentiles and the interquartile range. Each of these operators is applied to the segmented set associated to a step that are stored in the database, then, considering all the sets associated to an activity (going up/down stairs and walking) their statistical distribution is analysed.

The most relevant indicators are summarised in Fig. 3. Note that some indicators have not been shown (entirely or for a given time variable) as they do not provide relevant information that allows to classify the different activities. The selected six indicators were selected after analyzing the differences in the statistical distributions (mean, deviation, distribution type) associated to each study cases. Specifically, those that presented more differences in their distribution were selected, as the hypothesis was that this would help better classification.

The relevant indicators are represented by a histogram (Fig. 3). This way the difference in the data distribution can be analysed for all the volunteers. Note that two main variables are used to define the indicators: the rotational velocity around the *x* axis (Fig. 1b) measured by the gyroscope and the linear acceleration of the *z* axis (Fig. 1f), both related with the anteroposterior movement of the crutch/cane. Five different statistical indicators are derived from the gyroscope *x* motion, as they present different statistical distributions: the

standard deviation (Fig. 3a), the 25th (Fig. 3b), 50th (Fig. 3c), and 75th (Fig. 3d) percentiles, and the interquartile range (Fig. 3e). For these indicators, the statistical distribution of the different activities present significant differences in their mean value or mode, suggesting that these could be used to differentiate the

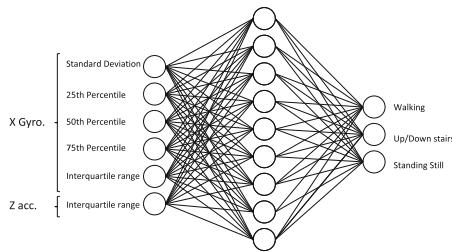


**Fig. 3.** Most relevant indicators histogram.

different scenarios. Finally, the interquartile range (Fig. 3f) of the  $z$  acceleration provides also relevant data, as this variable is related with the up/down motion of the crutch, and it is assumable that the user will present higher values in this variable when going up/down stairs than in the rest of the cases.

## 5 Neural Network Design for Activity Recognition

Once the set of indicators has been selected, a proper activity recognition algorithm has to be defined. As stated in the introduction, Artificial Intelligence approaches, such as Artificial Neural Networks (ANN), are one of the most popular solutions for this purpose.



**Fig. 4.** ANN layers structure

Hence, the proposed algorithm is based on a Multi-Layer Perceptron ANN which will process the set of six indicators defined in the previous subsection, and identify which of the proposed three activities (walking, standing still or going up/down stairs) is the user performing at each step. In order to determine the identified activity, three outputs will be defined in the ANN (one associated to each activity). This way, for each activity, its corresponding output will be activated with a high (1) value, while the other two will remain inactive (0 value). A single hidden layer has been defined for the network, with a variable number of neurons that will be experimentally adjusted. The overall structure of the network is depicted in Fig. 4.

A supervised learning approach using the Levenberg-Marquardt algorithm is used to train the network (maximum of 200 epochs, no early-stopping, 0 target error, 500 iterations, Hyperbolic tangent sigmoid activation transfer function,  $\mu = 1E-5$ ). This approach requires to define proper training and validation sets, composed by the inputs (extracted from the database) and the targets (generated from the identified activity, using the previously defined boolean codification). In order to preserve the generalizing capabilities of the ANN with different users, the training and validation sets are generated using data from different volunteers (the data from 9 volunteers is used for the training set, while the other 4 are used to generate the validation set, the selection of these volunteers has been

**Table 1.** Number of samples

	Training set samples	Validation set samples
Straight	370	183
Stairs	283	112
Still	298	132

done randomly). The summary of the samples for each activity to be identified and set is shown in Table 1.

In order to determine the best hidden layer configuration, the network is trained considering different amount of neurons in its hidden layer, these experiments had been carried out using 5 to 100 neurons. For each configuration, the ANN is trained 500 times with randomized initial weights and considering the samples within the training set.

Once trained, each ANN is fed with the samples of the validation set, and its outputs are compared with the target ones. It is to be noted that the proposed ANN does not provide boolean outputs. Hence, each output of the ANN needs to be postprocessed, considering a high value if the value provided in that output is equal or greater than 0.5, and a low value if less. A global success rate for each network is then calculated, i.e. the percentage of samples within the validation set the ANN is able to classify correctly. This global success rate will be used as a metric to select the best ANN for a given number of hidden layer neurons.

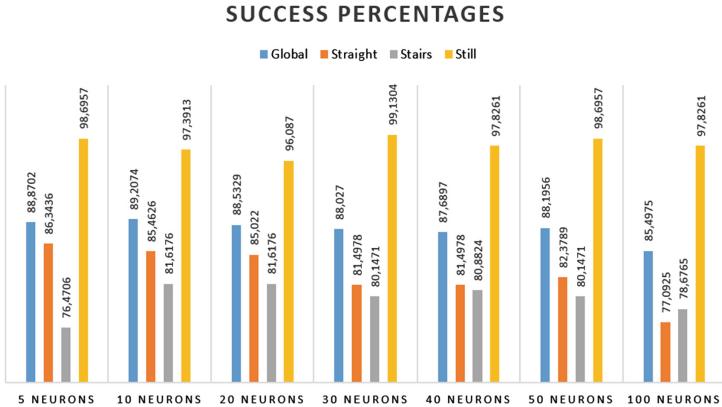
## 6 Results and Discussion

Following the methodology detailed in the previous section, a set of ANNs has been trained considering 5, 10, 20, 30, 40, 50 and 100 hidden layer neurons. For each hidden layer neuron configuration, 500 ANN have been trained and the one with the best global success rate has been selected as the representative for that configuration.

For each selected ANN, the partial success rates for each activity have also been calculated, i.e., the number of samples the ANN is capable of classifying correctly versus the total number of samples of that activity. This will allow to perform a better analysis on the capabilities of the developed approach.

Results are summarised in Fig. 5, where the global and partial success rates versus the number of hidden neurons is detailed. From these results it can be concluded that if the global success rate is considered, all the analysed hidden neuron configurations provide good results (over 88%).

However, the partial success rates presents important differences. For instance, it can be seen that in the general case, the ANNs are capable of recognizing when the person is still with at least 97% success rate, as it is the clearer scenario, being the dynamics of this activity very different from the other two.



**Fig. 5.** Success Rates vs Hidden layer neurons

On the other hand, the differences between walking straight and going up/down stairs are more subtle, as seen in Sect. 4. The success rates in these scenarios are around the 80%, which is an acceptable rate. However, it is to be noted that the ANNs have difficulties recognizing the patterns between the two activities. The particular case of 5 hidden neurons is to be noted, as its performance when recognizing the up/down stairs activity is lower than the 80%. It can be concluded that less than 10 neurons do not provide enough processing capability to learn the differences between the aforementioned two activities.

However, increasing the number of neurons does not provide increased success rate (the differences with 10 to 50, and even 100, neurons are quite small). It is to be noted that all ANNs finished training before the maximum number of epochs due to low gradient in their overall evolution. Hence, as increasing the number of hidden layer neurons does not provide significant improvements, the conclusion to be drawn is that further research is required on the selection of the indicators, and the source sensor data, to obtain better success ratios for these activities.

Looking at the results obtained, the best ANN configuration is considered to be the one with 10 neurons. Note that even if better results are achieved with a greater number of neurons, the computational cost increase is higher than the increase in the success rate. On the other hand, a less number of neurons does not guarantee proper sucess rate.

## 7 Conclusions

Physical activity monitoring outside clinical environment is a key issue when assessing neurological disorder patients. This requires a individualization of this therapy, which depends on the daily routines of the patient.

In this work a first step towards this goal is presented, proposing a methodology to develop a neural network based activity recognition algorithm based on

the data provided by an instrumented crutch/cane tip. The approach is based on two steps: a preprocessing phase, in which statistical indicators are evaluated for each step; and a training phase where an ANN is trained to recognize three main activities: standing still, walking and going up/down stairs.

The developed approach provides a success rate over 85%, and establishes a starting point for further developments. However, some drawbacks exist if increased success ratios are required. Therefore, future work will require analyzing the best set of sensor source and indicator processing.

In addition, obtaining data from MS patients is mandatory for future developments, as it is expected that the variability of walking patters will present challenges to be considered by the classifier.

**Acknowledgments.** This research was supported by the University of the Basque Country under grant number PIF18/067, by the University of the Basque Country UPV/EHU under project number PPGA19/48 (GV/EJ IT1381-19), by the European Commission under grant number PN/TG1/UNSW/PhD/18/2017 and by the Ministerio de Ciencia, Innovación y Universidades (MCIU) under grant number DPI2017-82694-R (AEI/FEDER, UE).

## References

1. Aminian, K., Robert, P., Jequier, E., Schutz, Y.: Estimation of speed and incline of walking using neural network. In: Conference Proceedings, 10th Anniversary, IMTC/94, Advanced Technologies in I & M, 1994 IEEE Instrumentation and Measurement Technology Conference (Cat. No. 94CH3424-9). IEEE (1994)
2. Bartlett, H.L., Goldfarb, M.: A phase variable approach for IMU-based locomotion activity recognition. *IEEE Trans. Biomed. Eng.* **65**(6), 1330–1338 (2018)
3. Chamorro-Moriana, G., Sevillano, J., Ridao-Fernández, C.: A compact forearm crutch based on force sensors for aided gait: reliability and validity. *Sensors* **16**(6), 925 (2016)
4. Cohen, J.A., Reingold, S.C., Polman, C.H., Wolinsky, J.S.: Disability outcome measures in multiple sclerosis clinical trials: current status and future prospects. *Lancet Neurol.* **11**(5), 467–476 (2012). <http://www.sciencedirect.com/science/article/pii/S147442212700595>
5. Federation, M.S.I.: The atlas of multiple sclerosis. Technical report (2013)
6. Gadaleta, M., Merelli, L., Rossi, M.: Human authentication from ankle motion data using convolutional neural networks. In: 2016 IEEE Statistical Signal Processing Workshop (SSP). IEEE, June 2016
7. Godfrey, A., Conway, R., Meagher, D., ÓLaighin, G.: Direct measurement of human movement by accelerometry. *Med. Eng. Phys.* **30**(10), 1364–1386 (2008)
8. Gyllensten, I.C., Bonomi, A.G.: Identifying types of physical activity with a single accelerometer: evaluating laboratory-trained algorithms in daily life. *IEEE Trans. Biomed. Eng.* **58**(9), 2656–2663 (2011)
9. Khan, A.M., Lee, Y.K., Kim, T.S.: Accelerometer signal-based human activity recognition using augmented autoregressive model coefficients and artificial neural nets. In: 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, August 2008

10. Latimer-Cheung, A.E., Pilutti, L.A., Hicks, A.L., Ginis, K.A.M., Fenuta, A.M., MacKibbon, K.A., Motl, R.W.: Effects of exercise training on fitness, mobility, fatigue, and health-related quality of life among adults with multiple sclerosis: a systematic review to inform guideline development. *Arch. Phys. Med. Rehabil.* **94**(9), 1800–1828.e3 (2013)
11. Lei, L., Peng, Y., Zuojun, L., Yanli, G., Jun, Z.: Leg amputees motion pattern recognition based on principal component analysis and BP network. In: 2013 25th Chinese Control and Decision Conference (CCDC). IEEE, May 2013
12. World Health Organization: Neurological disorders: public health challenges. Technical report (2006)
13. Preece, S.J., Goulermas, J.Y., Kenney, L.P.J., Howard, D., Meijer, K., Crompton, R.: Activity identification using body-mounted sensors—a review of classification techniques. *Physiol. Measur.* **30**(4), R1–R33 (2009)
14. Sardini, E., Serpelloni, M., Lancini, M., Pasinetti, S.: Wireless instrumented crutches for force and tilt monitoring in lower limb rehabilitation. *Procedia Eng.* **87**, 348–351 (2014)
15. Shull, P.B., Jirattigalachote, W., Hunt, M.A., Cutkosky, M.R., Delp, S.L.: Quantified self and human movement: a review on the clinical impact of wearable sensing and feedback for gait analysis and intervention. *Gait Posture* **40**(1), 11–19 (2014)
16. Spain, R., George, R.S., Salarian, A., Mancini, M., Wagner, J., Horak, F., Bourdette, D.: Body-worn motion sensors detect balance and gait deficits in people with multiple sclerosis who have normal walking speed. *Gait Posture* **35**(4), 573–578 (2012)
17. Wang, N., Ambikairajah, E., Lovell, N.H., Celler, B.G.: Accelerometry based classification of walking patterns using time-frequency analysis. In: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, August 2007
18. Watanabe, T., Yamagishi, S., Murakami, H., Furuse, N., Hoshimiya, N., Handa, Y.: Recognition of lower limb movements by artificial neural network for restoring gait of hemiplegic patients by functional electrical stimulation. In: 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE (2001)
19. Zeng, W., Wang, C.: Classification of neurodegenerative diseases using gait dynamics via deterministic learning. *Inf. Sci.* **317**, 246–258 (2015)



# ExoFlex: An Upper-Limb Cable-Driven Exosuit

David Pont<sup>1</sup> Aldo Francisco Contreras<sup>1</sup> José Luis Samper<sup>1</sup> Francisco Javier Sáez<sup>1</sup> Manuel Ferre<sup>1</sup> Miguel Ángel Sánchez<sup>1</sup> Ricardo Ruiz<sup>2</sup> and Ángel García<sup>2</sup>

<sup>1</sup> Centre for Automation and Robotics (CAR) UPM-CSIC,  
Universidad Politécnica de Madrid, 28006 Madrid, Spain

{david.pont,joseluis.samper,m.ferre,miguelangel.sanchezuran}@upm.es

<sup>2</sup> Escuela Técnica Superior de Ingenieros Industriales ETSII UPM,  
Universidad Politécnica de Madrid, 28006 Madrid, Spain

**Abstract.** This paper presents ExoFlex, an upper-limb flexible exoskeleton (exosuit) intended for assistance in elbow and shoulder rehabilitation therapies. The soft nature of the device allows it to easily adapt to human biomechanics. The presented exosuit is equipped with a cable-driven transmission in which torque is generated by two direct current (DC) motors. A super-twisting sliding mode controller (SMC) has been simulated and implemented for elbow and shoulder flexion and extension movements. ExoFlex has proven to effectively assist its wearer in experimental tests.

**Keywords:** Exosuit · Soft-robot · Sliding-mode control · Cable-driven · Rehabilitation

## 1 Introduction

Advances and research achieved in recent decades have led to the development of exoskeletons, thus giving rise to a broad range of specialised devices to meet different needs, from mobilising large loads in the workplace [1] to assistance of patients [2] in rehabilitation. Flexible exoskeletons or exosuits are defined as fully wearable devices, with low weight and volume that do not restrict the user from making natural movements with the goal of meeting similar attributes to rigid exoskeletons. It is worth mentioning that exosuits do not increase the capacities of the user since the bone-anatomical structure is the direct recipient of the resulting efforts. In addition, weight restrictions applied to these devices usually limit the use of high torque motors that meet the speed requirements.

Moved by this increasing tendency, the authors introduce in this paper ExoFlex, an upper-limb cable-driven exosuit oriented to rehabilitation tasks. The exosuit's control system is analysed in detail.

## 2 State of the Art

This section focuses on the state of the art of upper-limb exosuit's mechanical design and control methods for cable-driven exosuits.

### 2.1 Exosuits Mechanical Desing

The most typical architecture for soft exoskeletons is based on cable-driven transmissions. For example, CRUX [3] is a textile-based robot with motion capture to identify lines of minimal extension in the upper extremity, deriving a cable map from those lines. It uses a mimetic algorithm that synchronises the position of one arm to the measured position of the other one using a set of Inertial Measurement Units (IMUs) for position feedback. Another cable-driven soft wearable device for human assistance is a design developed in [4] which provides support to the elbow joint motion based on Bowden cables and two DC motors, which are torque-controlled through a PD controller. This system combines a serial-elastic-actuator based transmission and a power-storing architecture in a relatively small hardware. This allows to passively sustain high loads. An alternative to this kind of motion transmissions is shown in [5], where a simple design of an inflatable version of a soft robotic shoulder exosuit is presented, achieving position control by applying varying magnitudes of pressure.

### 2.2 Cable-Driven Transmission Control

Bowden cable transmissions are widely used in many robotic applications such as parallel robots, bioinspired robots or exoskeletons. However, to our knowledge, studies applied to control of cable-driven exosuits are relatively new.

There are two major types of controllers applied to exosuits. The first group gathers position controllers. This kind of scheme is used in cases in which the angle of each joint needs to be accurately controlled. The second group of controllers is based on force/torque control. These controllers are usually used as low-level ones.

In [6], a three-level control paradigm for a cable-driven upper limb exosuit is proposed in order to deliver the appropriate assistive torque to the elbow joint: a high-level control which combines an assistive torque estimator and a human arm model with admittance control; a mid-level control to compensate for the backlash in the transmission and convert the arm motion reference to the desired position of the actuator; and a low-level control, which compensates the non-linear dynamics of the Bowden cable to provide the desired assistive torque at the joint. In order to independently control the elbow joints of both arms of the exosuit, using two modules driven by the same prime actuator controlled with a finite-state machine is proposed in [7], while a refined design of an exosuit for the elbow joint driven by two Bowden cables and a single DC motor is presented in [8]. A low-level closed-loop velocity controller with gravity compensation and a high-level assistance estimator based on the subject's intention of motion detection have been implemented.

In another context, for the past few years, several studies of cable-driven systems relying on sliding mode control (SMC) have been presented. In particular, in [9], the authors propose an optimal non-singular terminal sliding mode control scheme using a super-twisting algorithm for the trajectory tracking of cable-driven manipulators. An exoskeleton oriented to passive rehabilitation of upper limbs using SMC is presented in [10]. The proposed approach only uses the exoskeleton nominal model while the system upper bounds are adjusted adaptively. Likewise, a cable-driven robot for upper-limb rehabilitation using a sliding mode controller combined with a non-linear disturbance observer was developed in [11]. Position tracking results were compared with a PID implementation and a traditional sliding mode controller, showing a significant tracking accuracy improvement. Moreover, a cable-driven elbow exoskeleton for use in robotic rehabilitation tasks is proposed in [12]. A low-level sliding mode control of the joint is implemented in this device. This approach is based on a new robotic bio-joint and sensor (named BJS by the authors) and the use of a dynamical model of the system.

### 3 ExoFlex General Description

The exosuit presented in this article (Fig. 1) is made of a base of fabric, over which some small nylon 3D printed rigid pieces are attached in strategical points in order to, in combination with metallic sheaths, route the transmission Bowden cables. Some of those pieces have been directly sewn to the cloth and others are fixed using velcro fastenings to adapt to the wearers' anatomy. The pieces were first placed according to the arms tendons' distribution. Starting from that disposition, obtaining the number, shape and position of those pieces has been fruit of an iterative process in which errors or improvable aspects have been observed and subsequently corrected based on the wearers' experience.

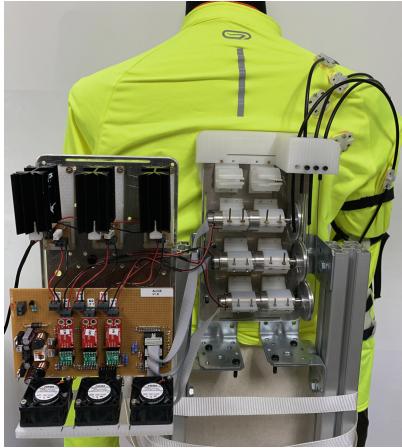


**Fig. 1.** ExoFlex, soft robotic exoskeleton for upper-limbs

The Bowden cables are actuated by DC motors with their corresponding gearboxes. Each cable is responsible of the movement of one independent degree

of freedom (DoF). The actuation system and control electronics have been positioned over a metallic structure (Fig. 2) attached to a heavy wooden board lying on the ground. To ease testing the exosuit, a chair has been attached to the structure. This layout protects the user's back and improves their comfort during the experiments.

Control software has been implemented over a Texas Instruments LAUNCHXL-F28379D board. A prototype shield board which gathers all the necessary electronics for control, such as motor drivers, current sensors or voltage converters has been designed and built. Hardware current limiting has been included to protect the actuation system.



**Fig. 2.** ExoFlex electronics and actuation system

As evidenced by some clinical studies such as [13] and [14], a broad set of daily tasks is performed using a subset of the full range of joint motion of the arm. Consequently, ExoFlex has been designed to cover this main region of the workspace, as shown in Table 1. As a first approach, 2 degrees of freedom (elbow and shoulder flexion and extension) have been effectively controlled and actuated, but ExoFlex' soft nature allows the user moving the arm in other directions. A third motor has been included in the actuation system to control the shoulder abduction-adduction movement in future versions.

In [15], a study comparing an upper-limb traditional rehabilitation methodology to a robot-assisted therapy for people who had suffered stroke was carried out. The results that were obtained showed an improvement of articular range and strength using the robot-assisted therapy. According to this study, three main types of rehabilitation exist depending on the robot and the patient interaction. The first one is passive rehabilitation, in which the patient relaxes his arm, in this case, and the robot moves it according to position targets defined by a programmed trajectory. The second one is the active-assisted mode. In this mode, the patient starts a movement toward the target and the robot provides

**Table 1.** Mobility angles allowed by ExoFlex

Movement	Range
Elevation of the anterior deltoid	From 0° to 120°
Adduction/Abduction	From 0° to 120°
Displacement on horizontal plane	From 0° to sagittal crossing at 30°
Cross elevation with respect to the sagittal plane, ascending direction	From 0° to 100°
Cross elevation relative to the sagittal plane, downward direction	From 0° to 100°
Extension/flexion of the elbow	From 0° to 145°

force in that direction. Finally, in the active-constrained mode, the robot provides a resistance force in the direction of the desired movement. As a starting point, ExoFlex has been oriented to passive-rehabilitation tasks for elbow and shoulder flexion and extension movements.

## 4 Exosuit Modelling and Control

In order to design an appropriate control law for the aforementioned exosuit's movements, it is first necessary to obtain a representative mathematical model of the system. A discrete state-space representation of the model has been selected, since this kind of models are the most adequate for simulation and digital control implementation. Figure 3 shows a sketch of the system. Table 2 gathers the system's main variables with their notation.

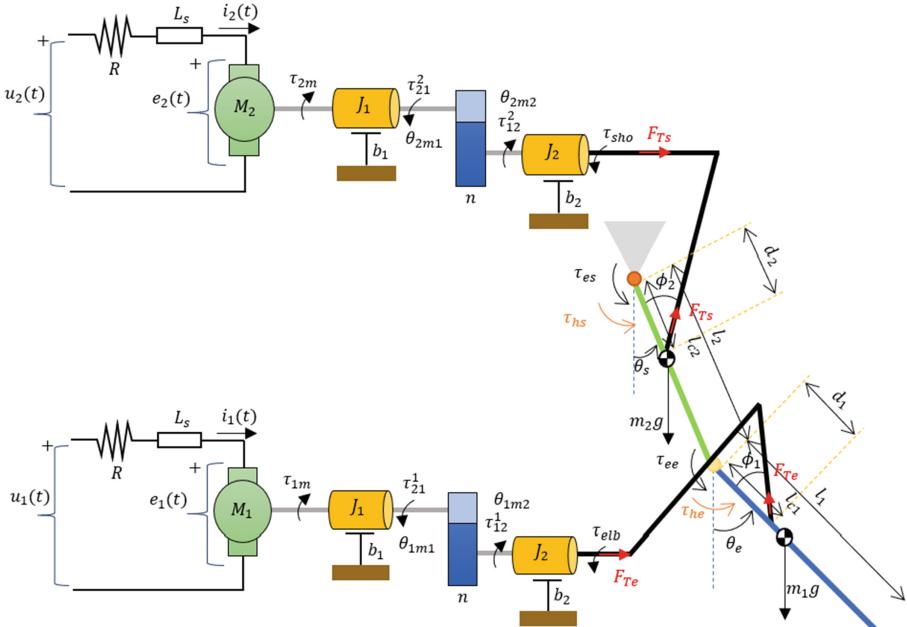
**Table 2.** System variables

Variable	M1 Elbow	M2 Shoulder
$\tau_{\text{gearbox}}$	$\tau_{12}^1$	$\tau_{12}^2$
$\tau_{\text{arm-gearbox}}$	$\tau_{elb}$	$\tau_{sho}$
$\tau_{\text{exo-articulation}}$	$\tau_{ee}$	$\tau_{es}$
$\tau_{\text{human-articulation}}$	$\tau_{he}$	$\tau_{hs}$
$\tau_{\text{total articulation}}$	$\tau_{etot}$	$\tau_{stot}$
Cable tension	$F_{Te}$	$F_{Ts}$
Gearbox angle	$\theta_{1m2}$	$\theta_{2m2}$
Articulation angle	$\theta_e$	$\theta_s$

### 4.1 Modelling Considerations

When addressing the modelling of the system, two main problems arise. The first one is the model parameters' dependence on the exosuit wearer's arm complexion. Since the user's arm is a part of the model, parameters such as mass and size

of the limb are highly variable among subjects. Moreover, the flexibility of the exoskeleton makes practically impossible to place it over the arm always in the exact same position. The second problem is the existence of hard non-linearities such as slacks, hysteresis and dead zones derived from the exosuit's cable-driven transmission and its soft nature.



**Fig. 3.** Model sketch

These factors make the exosuit presented in this work a complex system to highly-accurately model, so some simplifications have been applied. In relation to the first issue, intermediate standard arm dimensions, masses and inertia moments have been chosen to model the system. Moreover, the upper-arm (green bar in Fig. 3) and forearm (blue bar in Fig. 3) links have basically been modelled as simple uniformly-distributed-mass bars, whose mass centre is located in the geometrical centre of each link. As the movements in which ExoFlex can support the user until the date are flexion and extension of elbow and shoulder, both articulations have been modelled as 1 DoF revolute joints.

The nature and number of hard non-linearities inherent to the system make modelling them a really hard task. Furthermore, these non-linearities are greatly dependent on the way the exosuit fits every user's arms. For these reasons, the hard non-linearities have not been included in the state-space model. Finally, the possible interaction of the user and the exosuit in terms of user's torque application to the system has also been considered. The wearer's torque contributions in both elbow and shoulder have been modelled as external perturbations.

## 4.2 Mathematical Modelling

In order to obtain the model of the system, the Euler-Lagrange approach has been used, as shown in (1)

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta} = \psi = \tau_e + \tau_h - B\dot{\theta} \quad (1)$$

where  $\psi$  is the vector of non-conservative generalized torques performing work on the articulations  $\theta_i$ . This term includes the torque exerted by the exosuit ( $\tau_e$ ), the one exerted by the human ( $\tau_h$ ) and the viscous friction torque determined by the diagonal matrix  $B$ .

With the objective of not being too exhaustive, the calculations of the different terms involved in the model are not presented in this paper. The structure of the expression of the obtained discrete space-state model is shown in (2)

$$x[k+1] = F(x[k]) + Gu[k] + P\tau_h[k] \quad (2)$$

where  $x$  is the state vector,  $F$  is the system matrix,  $G$  is the input matrix and  $P$  is the perturbation matrix.

The resulting model is constituted by six state variables, whose physical meanings are described in Table 3.

## 4.3 Control Law Design

The exosuit is a discretized non-linear MIMO (multi-input multi-output) system. Additionally, it is a system with many uncertainties due to its adaptation capability to every wearer's arms. Furthermore, the controller must be stable in presence of user's torque application.

As previously said, the exoskeleton is focused on passive-rehabilitation tasks. In this kind of tasks, the user of the exoskeleton leaves his arm inert and the exoskeleton is responsible for moving the arm articulations as preprogrammed by the therapist. These movements can be monoarticular or multiarticular, and the duration and angle swept by each articulation can be variable. In this context, an articular position control is required.

**Table 3.** State variables description

State variable	Description
$x_1$	M1 gearbox angle
$x_2$	M1 gearbox angular speed
$x_3$	M1 current
$x_4$	M2 gearbox angle
$x_5$	M2 gearbox angular speed
$x_6$	M2 current

Taking into account all the intrinsic characteristics of the exosuit, the simplifications considered in modelling the system in Sect. 4.1 and the control objective, a position super-twisting sliding mode controller has been proposed. This kind of controllers is well known for its robustness to non-linearities, perturbations and model uncertainties. The main drawback of SMC is the chattering, a high frequency component present in the control action signals. However, the super-twisting controller considerably attenuates this problem.

The equations defining the super-twisting controller are given by (3) and (4)

$$\begin{cases} u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} c_1 |\sigma_1|^{\frac{1}{2}} \text{sign}(\sigma_1) + w_1 \\ c_2 |\sigma_2|^{\frac{1}{2}} \text{sign}(\sigma_2) + w_2 \end{bmatrix} \\ \dot{w} = \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} = \begin{bmatrix} b_1 \text{sign}(\sigma_1) \\ b_2 \text{sign}(\sigma_2) \end{bmatrix} \end{cases} \quad (3)$$

$$\begin{cases} c_i = 1.5\sqrt{C_i} \\ b_i = 1.1C_i \\ \sigma = \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} \sigma_{11}e_1 + \sigma_{12}e_2 + \sigma_{13}e_3 \\ \sigma_{21}e_4 + \sigma_{22}e_5 + \sigma_{23}e_6 \end{bmatrix} \end{cases} \quad (4)$$

where  $u$  is the vector of voltages applied to the motor,  $\sigma$  is the sliding manifold,  $C_i$ ,  $b_i$  and  $c_i$  are design parameters and  $e_i$  is the error associated to  $x_i$ . Note that, in this controller, the discontinuous function  $\text{sign}(\sigma_i)$  is integrated in  $w$ , which results in chattering reduction.

## 5 Simulation Results of Exosuit Controllers

Two different sliding manifolds have been proposed for the controller, as shown in (5) and (6). In both cases  $C_i$  has been set to a value of 5.

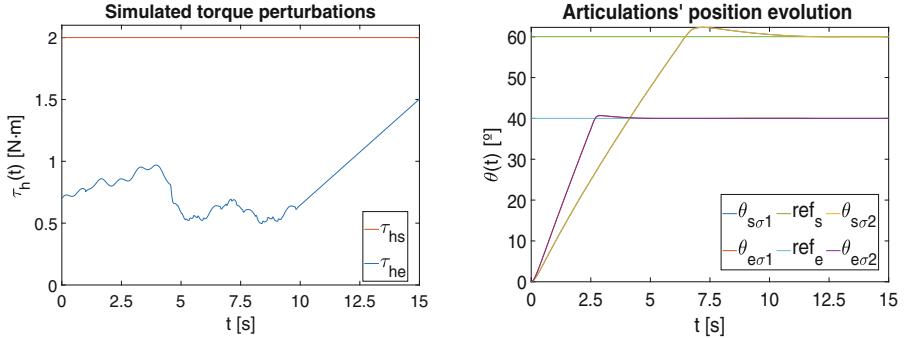
$$\sigma_1 = \begin{bmatrix} 800 & 25 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 800 & 25 & 1 \end{bmatrix} \quad (5)$$

$$\sigma_2 = \begin{bmatrix} 800 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 800 & 25 & 0 \end{bmatrix} \quad (6)$$

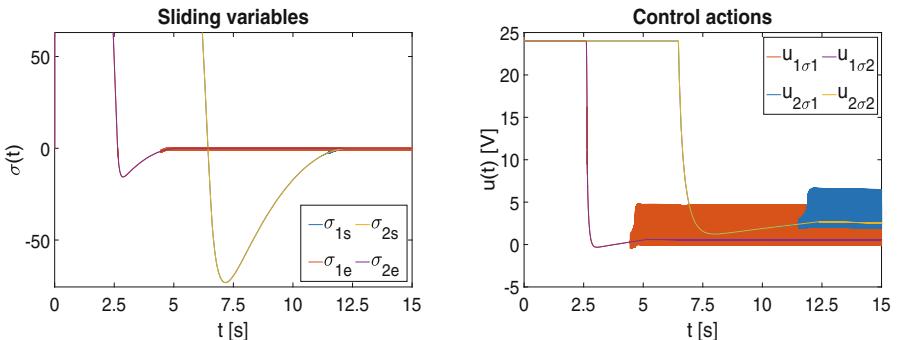
The difference between these two proposed manifolds is that, while  $\sigma_1$  depends on all the model's state variables,  $\sigma_2$  does not take into account the values of the state variables associated to the motor's currents.

In order to compare the capabilities offered by both sliding manifolds and to evaluate the performance of the designed controller, a simulation of both is presented. This simulation proposes a flexion movement for the elbow and shoulder, with an initial repose state and the angular references set to  $40^\circ$  and  $60^\circ$  respectively. Additionally, in order to evaluate the effect of torque perturbations introduced by the exosuit wearer, a constant torque of 2 Nm has been applied to

the shoulder and a pseudorandom variable torque of amplitude lower than 1 Nm during the first 10 s which then increases linearly with time has been simulated over the elbow joint, as shown in Fig. 4a.



**Fig. 4.** (a) Simulated torque perturbations (left), (b) Articulations' position (right)



**Fig. 5.** (a) Sliding variables (left), (b) Articulations' control actions (right)

No significant differences between the temporal response of the two simulated controllers are appreciable, as can be noticed in Fig. 4b. The position error is non-existent, but a slight overshooting (lower than 4%) is present in both articulations.

The temporal evolution of the sliding variables is shown in Fig. 5a. The sliding variables associated to the manifold  $\sigma_2$  are practically chattering-free, while the ones related to the controller given by  $\sigma_1$  have a significant chattering.

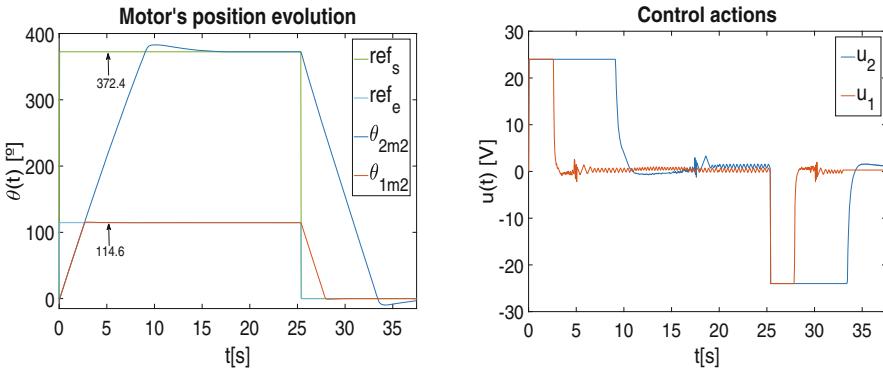
In relation with the control actions applied by the exosuit, as can be seen in Fig. 5b, the use of  $\sigma_2$ , which does not consider the values of the motors' currents in the computation of the sliding variable, produces a significant reduction of chattering (less than 200 mV) in comparison with  $\sigma_1$ . This relies on the

fact that the motor current has intrinsically coupled chattering due to the sliding controller, and feeding that information to the sliding variable computation amplifies this undesired high-frequency component.

In the light of the results obtained in the simulations, it has been proven that  $\sigma_2$  represents a better choice than  $\sigma_1$ , since both manifolds offer practically the same time response, but  $\sigma_2$  highly reduces chattering in the control action, what is hugely beneficial for the actuators and makes the controller implementable over hardware. Moreover, both sliding manifolds have proven to maintain stability and a minimal or null position error in presence of external perturbations of different nature.

## 6 Exosuit Experimental Results

The position super-twisting controller based on the sliding manifold  $\sigma_2$  has been implemented in the physical system at 333 Hz. In the experiment carried out over 4 healthy participants, the subjects left their arm completely relaxed and the exosuit positioned the shoulder and elbow in  $60^\circ$  and  $40^\circ$  of flexion respectively, returning then to the initial position. Each participant performed 5 repetitions. The exosuit was accurately placed over every user, using the velcro fastenings to fit the best over each arm. The angular positioning and control actions for one of the subjects are presented in Fig. 6.



**Fig. 6.** (a) Motors positions (left), (b) Control actions (right)

The ExoFlex position sensory system is based on motor incremental encoders, so the magnitudes that have been monitored are the elbow and shoulder encoder positions. When reaching the final positions, the elbow and shoulder positions have been measured with a goniometer. The average error in shoulder positioning has resulted in  $0.8^\circ$  with a standard deviation of  $3.83^\circ$ , and  $-0.15^\circ$  with a standard deviation of  $3.56^\circ$  for elbow positioning. The RMSEs are  $3.82^\circ$  and



**Fig. 7.** Subject performing the experiment

3.47° respectively. The overall absolute maximum error was of 5°. The overshooting was inferior to 3% in all the cases and the amplitude of the chattering was reasonably low, as respectively shown in Fig. 6a and b.

## 7 Conclusion

The cable-driven exosuit ExoFlex has been presented in this paper. A position super-twisting sliding mode controller has been designed and simulated. Two sliding manifolds have been proposed, and it has been proven that not considering the motor currents causes a chattering attenuation. The exosuit has successfully been applied to assist different healthy users in simple flexion and extension movements of both elbow and shoulder. In future works, the shoulder adduction-abduction will be considered. An articular position sensory system is already being developed. Furthermore, a force control loop will be introduced to work in parallel with the position control.

**Acknowledgements.** This work has been supported by the Spanish Ministry of Economy, Industry and Competitiveness, under the grant ExoFlex (DPI 2015-68842-R) and the I+D+I Own Program of the Universidad Politécnica de Madrid.

## References

1. Marcheschi, S., Salsedo, F., Fontana, M., Bergamasco, M.: Body extender: whole body exoskeleton for human power augmentation. In: 2011 IEEE International Conference on Robotics and Automation, pp. 611–616. IEEE (2011). <https://doi.org/10.1109/ICRA.2011.5980132>
2. Veneman, J.F., Kruidhof, R., Hekman, E.E., Ekkelenkamp, R., Van Asseldonk, E.H., Van Der Kooij, H.: Design and evaluation of the lopes exoskeleton robot for interactive gait rehabilitation. IEEE Trans. Neural Syst. Rehabil. Eng. **15**(3), 379–386 (2007). <https://doi.org/10.1109/TNSRE.2007.903919>
3. Lessard, S., Pansodtee, P., Robbins, A., Trombadore, J.M., Kurniawan, S., Teodorescu, M.: A soft exosuit for flexible upper-extremity rehabilitation. IEEE Trans. Neural Syst. Rehabil. Eng. **26**(8), 1604–1617 (2018). <https://doi.org/10.1109/TNSRE.2018.2854219>

4. Cappello, L., Pirrera, A., Weaver, P., Masia, L.: A series elastic composite actuator for soft arm exosuits. In: 2015 IEEE International Conference on Rehabilitation Robotics (ICORR), pp. 61–66. IEEE (2015). <https://doi.org/10.1109/ICORR.2015.7281176>
5. Natividad, R., Yeow, C.H.: Development of a soft robotic shoulder assistive device for shoulder abduction. In: 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp. 989–993. IEEE (2016). <https://doi.org/10.1109/BIOROB.2016.7523758>
6. Dinh, B.K., Xiloyannis, M., Antuvan, C.W., Cappello, L., Masia, L.: Hierarchical cascade controller for assistance modulation in a soft wearable arm exoskeleton. *IEEE Robot. Autom. Lett.* **2**(3), 1786–1793 (2017). <https://doi.org/10.1109/LRA.2017.2668473>
7. Canesi, M., Xiloyannis, M., Ajoudani, A., Bicchi, A., Masia, L.: Modular one-to-many clutchable actuator for a soft elbow exosuit. In: 2017 International Conference on Rehabilitation Robotics (ICORR), pp. 1679–1685. IEEE (2017). <https://doi.org/10.1109/ICORR.2017.8009489>
8. Chiaradia, D., Xiloyannis, M., Antuvan, C.W., Frisoli, A., Masia, L.: Design and embedded control of a soft elbow exosuit. In: 2018 IEEE International Conference on Soft Robotics (RoboSoft), pp. 565–571. IEEE (2018). <https://doi.org/10.1109/ROBOSOFT.2018.8405386>
9. Wang, Y., Yan, F., Ju, F., Chen, B., Wu, H.: Optimal nonsingular terminal sliding mode control of cable-driven manipulators using super-twisting algorithm and time-delay estimation. *IEEE Access* **6**, 61039–61049 (2018). <https://doi.org/10.1109/ACCESS.2018.2875685>
10. Riani, A., Madani, T., Benallegue, A., Djouani, K.: Adaptive integral terminal sliding mode control for upper-limb rehabilitation exoskeleton. *Control Eng. Pract.* **75**, 108–117 (2018). <https://doi.org/10.1016/j.conengprac.2018.02.013>
11. Niu, J., Yang, Q., Chen, G., Song, R.: Nonlinear disturbance observer based sliding mode control of a cable-driven rehabilitation robot. In: 2017 International Conference on Rehabilitation Robotics (ICORR), pp. 664–669. IEEE (2017). <https://doi.org/10.1109/ICORR.2017.8009324>
12. Jarrett, C., McDaid, A.: Robust control of a cable-driven soft exoskeleton joint for intrinsic human-robot interaction. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**(7), 976–986 (2017). <https://doi.org/10.1109/TNSRE.2017.2676765>
13. Magermans, D., Chadwick, E., Veeger, H., Van Der Helm, F.: Requirements for upper extremity motions during activities of daily living. *Clin. Biomech.* **20**(6), 591–599 (2005). <https://doi.org/10.1016/j.clinbiomech.2005.02.006>
14. Namdari, S., Yagnik, G., Ebaugh, D.D., Nagda, S., Ramsey, M.L., Williams Jr., G.R., Mehta, S.: Defining functional shoulder range of motion for activities of daily living. *J. Shoulder Elbow Surg.* **21**(9), 1177–1183 (2012). <https://doi.org/10.1016/j.jse.2011.07.032>
15. Lum, P.S., Burgar, C.G., Shor, P.C., Majmundar, M., Van der Loos, M.: Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. *Arch. Phys. Med. Rehabil.* **83**(7), 952–959 (2002). <https://doi.org/10.1053/apmr.2001.33101>



# Using a Collaborative Robot to the Upper Limb Rehabilitation

Lucas de Azevedo Fernandes<sup>1</sup>(✉) , José Luis Lima<sup>2</sup> , Paulo Leitão<sup>2</sup> ,  
and Alberto Yoshiro Nakano<sup>1</sup>

<sup>1</sup> Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brazil  
[lucas\\_azevedo33@hotmail.com](mailto:lucas_azevedo33@hotmail.com)

<sup>2</sup> CeDRI - Research Centre in Digitalization and Intelligent Robotics,  
Polytechnic Institute of Bragança and INESC TEC, Porto, Portugal

**Abstract.** Rehabilitation is a relevant process for the recovery from dysfunctions and improves the realization of patient's Activities of Daily Living (ADLs). Robotic systems are considered an important field within the development of physical rehabilitation, thus allowing the collection of several data, besides performing exercises with intensity and repeatedly. This paper addresses the use of a collaborative robot applied in the rehabilitation field to help the physiotherapy of upper limb of patients, specifically shoulder. To perform the movements with any patient the system must learn to behave to each of them. In this sense, the Reinforcement Learning (RL) algorithm makes the system robust and independent of the path of motion. To test this approach, it is proposed a simulation with a UR3 robot implemented in V-REP platform. The main control variable is the resistance force that the robot is able to do against the movement performed by the human arm.

**Keywords:** Robotics rehabilitation · Collaborative robots · Simulation · Reinforcement learning algorithm

## 1 Introduction

According to World Health Organization (WHO), the absolute number of Years Lived with Disability (YLDs) increased more than 17 million between 2005 and 2015, from this value about 74% of YLDs is linked to health conditions which rehabilitation can be beneficial [1]. The major dysfunction cases affect the elderly population, being considered the risk group of cardiovascular diseases, respiratory diseases, and other disabling conditions. As stated by [2], the world's population aged 60 years and over is set to increase from 841 million in 2013 to more than 2 billion in 2050.

The definition of rehabilitation according to [3] is as it follows: “*The use of all available means to reduce the impact of disabling situations and allows disabled individuals to achieve full social integration*”. Given the growing concerns around

physical therapy, various new technologies have been put to the test with patients to improve their treatment or assist them in their ADLs.

Rehabilitation, therefore, is a relevant process to help a large percentage of the population suffering from some dysfunction previously presented. Thus, the main goal of this project is to demonstrate how a collaborative robot can help the patients with non-paralysing dysfunctions, principally in the joints and muscles of the shoulder. The proposed system is composed of a collaborative robot (UR3) and a computer to perform the calculations referring to a possible insertion of an autonomous control module, called self-control module. The UR3 provides resistance to the patient's arm motion, i.e., when the patient execute some force, in turn, the robot applies an opposite force to this movement.

This paper presents the results measured through a simulation in the V-REP platform, as the force acquired in the movement and the variation of the UR3's resistance. In the virtual platform, an interface was developed to perform the control of some variables in the rehabilitation process, such as the applied torque in UR3 joints and the force detected by the sensor. Also, the obtained information of the contact between human and robot is showed, and represents a tool to therapists to analyse the advance of treatment. The force data about the movement will be acquired by a force sensor attached to the UR3. The data measured by the sensor represents the feedback of the environment and will be used to analyse the system.

The rest of this paper is organised as follows. Section 2 briefly explains some similar approaches. In Sect. 3, it is presented the development of the simulation, the analysis of the main concerns in the implementation of these systems, and the control through the self-control module. Section 4 exposes the obtained results in V-REP platform. Finally, Sect. 5 presents some conclusions regarding this problem and future works.

## 2 Related Work

Reflected by the evolution of robotics and control systems, there is an increasing number of research and development of products that address robotics for rehabilitation. The main studies in rehabilitation field use the following control methods: Electromyography (EMG) biofeedback, robot-assisted therapy, Virtual Reality (VR) based interventions, Constraint-Induced Movement Therapy (CIMT), and Functional Electric Stimulation (FES) [4]. This work focuses on a robot-assisted therapy method.

An aspect that must be considered is the acceptance of rehabilitation robots by patients and therapists. A study conducted in [5] shows a report of the use of a robot-assisted therapy method with 22 patients and 11 occupational therapists. The exercises with the patients were simple and consisted of the movement of a robotic arm with a linked button. The patient should press the button in some previously established positions. The system measures the time between two button presses and other information. After the exercises, various questionnaires are filled to evaluate the system.

Nowadays, it is still difficult to find works that relate simulation of robotic system with rehabilitation robots, due to the difficulty of modelling human limbs and complex robotic systems. In [6] is presented many types of robot-based systems and their controls. One of these control methods is the impedance-based assistance, in which the robots measure the position in a determined trajectory and are capable of inducing a force to correct patient's movements. The major difficulty is to adapt the system for an individual patient. One of the systems that shows this control is proposed in [7], using a robotic system to improve coordination of upper and lower limbs in elderly people.

Systems as MIT-Manus, one of the most well-known rehabilitation system, are composed by manipulators robots. The use of robotic manipulators is troublesome since many of these systems that are used in the industry have not been adapted and prepared to work with humans and, even less, with the human touch, but some adjustments can be made to become these systems suitable to rehabilitation. Table 1 presents some examples of rehabilitation systems with manipulators robots.

**Table 1.** Rehabilitation Robots based in Robotic Manipulators.

System	Developer	Treatment of...
Braccio di Ferro [8]	University of Genova, Italy	Shoulder + elbow
Gentle/S [9]	University of Reading, UK	Shoulder + elbow + forearm + wrist
iPAM [10]	University of Leeds, UK	Shoulder + elbow + forearm
MIT-MANUS [11]	Massachusetts Institute of Technologies, USA	Shoulder + elbow
REHAROB [12]	Budapest University of Technology and Economics, Hungary	Shoulder + elbow + forearm

### 3 Proposed System

One of the concerns in this field is the wasted time dedicated to learn and set it up the rehabilitation robots for each patient. Therefore, to develop a new device or method in the rehabilitation field is relevant to think about the utility, usability and facility of the system on both sides, patient and therapist.

The developed work is designated as an end-effector system. These type of systems have a unique fixation point, the physiotherapy exercises and movements are defined in Cartesian coordinates  $XYZ$  and the assistance level is modulated by control of impedance or admittance [13].

#### 3.1 Simulation Model

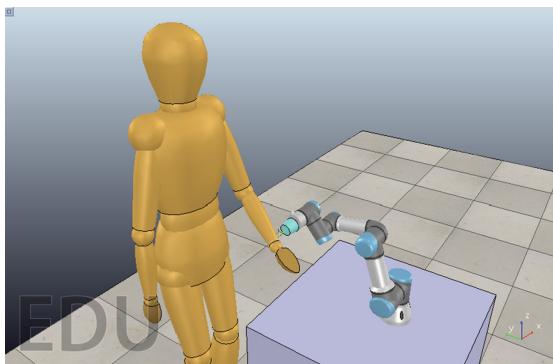
The proposed simulation will be developed in V-REP platform, and its objective is being a tool to test the system with a model of the human arm and predict

the possible failures and corrections needs. The simulation of human arm takes into account the possible consequences of a dysfunction, i.e., the joints of the shoulder may exhibit a small force on the movement or variations of it.

The V-REP is a simulation software that integrates virtual robots with a development environment and can simulate robotic arms, mobile robots, sensors, tools and many different scenes. The Integrated Development Environment (IDE) is programmed in LUA language, and also, presents tools for communications with remote Application Programming Interface (API) as MATLAB [14].

The simulation has three essential parts: The scene with the robot (UR3) and the manikin (model of the human body), the codes that control the simulation, and the control interface. Basically, the simulated elements are composed of mechanical parts and joints. The relevant dynamic properties of the first part that may be changed in the simulation are, mainly, mass, and inertial moment. The joints can also be configured, and its dynamic features refers to the motion: as maximum torque, target velocity, and position control. All these technical features in V-REP linked to a physics engine results in a simulation closest to the real environment.

The developed scene is shown in Fig. 1. It contains two main objects, the robot and the manikin, and both models are developed by outsources. The simulated robot is the UR3, a manipulator with six Degrees of Freedom (DOF), and its joints can be simulated separately. At the end effector of the UR3, a force sensor show the exact force of the human arm during the entire simulation. The UR3 was chosen because it may execute a sufficient torque for the human contact when the joints are used together. The maximum joint torques of this robot are presented in [15].



**Fig. 1.** The Simulated System in V-REP platform.

The goal of the UR3 is to provide a resistance force. In the simulation, the speed of the robot's joint is set as zero, and the torque is increased or decreased, the UR3's expected behaviour is to stand still, so when the human arm exerts its force, the robot applies the opposite force based on the maximum torque set

in the simulation. If the torque provided by the robot is larger than the human arm, there will be no motion; otherwise, if the torque is smaller, the force will not fulfil its function.

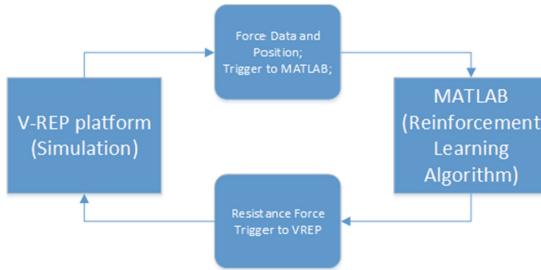
The simulated manikin is a special case because only the model of the arm is used. Thus, the main joints for this approach are those that represent the shoulder. The configuration of the manikin takes into account the real values of a human upper limb. A study accomplished in [16] shows a model for estimating the force and the torque during the abduction of the shoulder. The experiment consists in the execution of the movement by the subject and the measure of the performed force through a dynamometer. The subject is male, 28 years old, 1.78 m and 85 kg. In that case study, the maximum torque is 41 Nm. Therefore, the simulation is configured to get closer to the real case.

The codes were developed in the LUA language and were divided into two main scripts. The threaded-child script and the non-threaded child script. The main difference between both scripts is how each one runs during the simulation. The non-threaded child script runs just once after the start and when is called by other function, such as the press of a button, in this script are found the codes referring to the development of the control interface and all functions linked to it. The threaded script runs in loop during the entire simulation, and it contains the codes about the movement, the data acquisition, the messages that must be sent to another script, and the API connection.

The control interface is developed in XML and was used to configure the main parameters in the simulation, as the maximum torque of shoulder, the target position for the shoulder's joint (degree), the resistance torque of all UR3 joints, the possible failures in the movement, and other parameters related to communication with external API. The link with the external API (MATLAB), is built to execute the self-control module. To establish communication, V-REP must be configured in synchronous mode. Thus, MATLAB will execute a part of its code and send a trigger to V-REP, in turn, the simulator executes one step and returns to MATLAB the requested value. The simulation step is 100 ms. The variables returned to MATLAB are the position and the current force measured through the force sensor. The connection persists until the simulation stop.

### 3.2 Self-control Module Approach

The self-control module is developed to improve the usability and facility of the system; the code of this control was implemented in the MATLAB language. The communication between MATLAB and VREP is exemplified by the Fig. 2 through a block diagram. When the autonomous control is active, the robotic arm is able to recognise the force of the patient and change joint's torque values, making the system responsible, in real-time, by itself and adaptable to any patient. Besides, as the premises of the system work with the patients that are capable of performing some upper limb's movement and the biofeedback is the force of this movement, no path planning is required on the robot's side, but the therapists shall indicate to the patient which are these motions.



**Fig. 2.** Block Diagram of the proposed simulated system

The self-control module is developed based on a machine learning technique (RL). The RL is used to make the system learns with the environment, and it is desired that the robot performs a resistance against the human arm force. As presented in [17], basically the algorithm is separated in states and actions. In this case, the states are all forces performed by the human arm, measured through the sensor attached in the UR3's end-effector. The actions are represented by the possible decisions that the system can make, working with the resistance, the decisions are: increase the force of all joints of UR3, decrease the same variable or hold this.

The goal of an RL algorithm is to choose the best decision based on the system's current state. However, this behaviour is not known at first; therefore, the method of trial-and-error is performed. Initially the system is in some state  $s_x$ , makes a decision (some action  $a_x$  is performed) and goes to another different state  $s_y$ . When the system act in the environment some feedback signal is collected and this value is used to evaluate the action  $a_x$  in the certain state  $s_x$ . So, if the action results in an expected state (considered positive for the system), then the algorithm assigns a positive reward to the set “previous state - action - next state”, else the reward is negative to the same set. The positive reward, in this work, will be awarded when the force measured by the sensor is within a range of 70% to 85% of the maximum force the patient can perform, as the simulation is based on the experiments accomplished by [16], the positive reward is assigned each time the force sensor reads a value between 29 and 35 Nm. If the arm presents some dysfunction in the movement and also presents a reduced force, the system did not assign positive rewards.

Thus, the rewards acquired are used to calculate a matrix, named Q-matrix, which relates the states with the actions. Therefore, if an action  $a_x$  in the state  $s_x$  get many positive rewards, the Q-matrix value corresponding to this pair (state, action) will increase. To make a decision, the algorithm searches in the matrix which action has a larger value in the current state and chooses this. After the occurrence of a few number of episodes, the value for correct actions in Q-matrix will be larger than the wrong actions and, thus, completing the learning of the system. The episodes in this approach are the set of iterations that comprehend an entire movement of the human arm, i.e., the movement of

raise up and bring down the limb represents an episode. As the system learns by itself, it is not necessarily a human control over the joints, completing the objective of providing a resistance force to movement without supervision.

## 4 Results

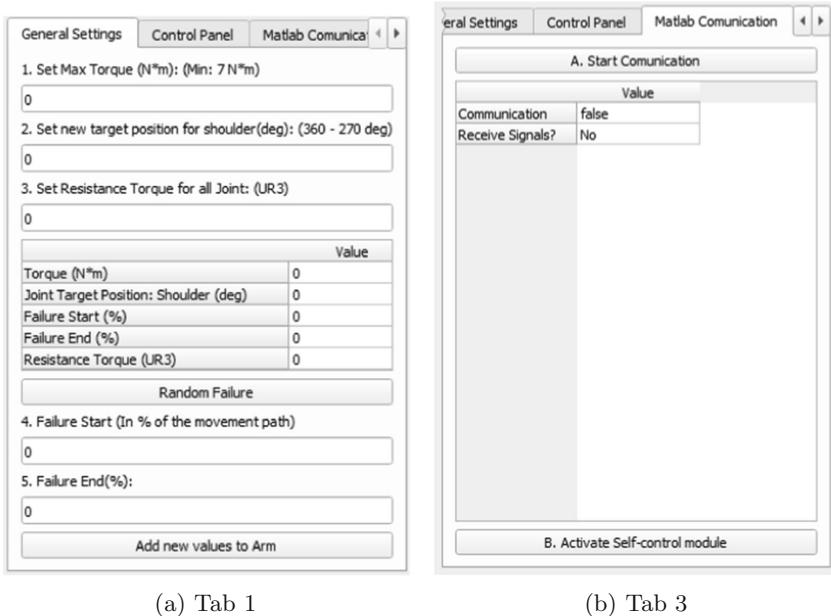
This section explains the obtained results by simulation performed with the self-control module activated and deactivated. The presented values refer to the movement of the arm and the robot. The simulation is developed to comprehend some lacks in the rehabilitation field, being a tool to test the system before its practical application and help the therapist learn about the utilities of the robotics applied to this problem.

To help in the usability, an interface is built to control the entire simulation and presents some data related to the movement. Figures 3a and b show the tabs “General Settings” and “MATLAB Communication”, sequentially. The first tab configures the main parameters linked to human arm movement (1 and 2), the resistance torque of the UR3 (3), and the possible variation of the patient force, which can be random (button “random failures”) or given by the user (4 and 5). When the button “Add New Values to Arm” is pressed, the simulation starts and the table is filled with the configured parameters. The second tab, called “Control Panel”, presents some results of the movement as torque of the robot’s joints and torque of the shoulder. The second tab is showed in Fig. 5. The third is responsible by the communication with the external API, basically when the button “A” is pressed the simulator is able to send and receive signals from MATLAB. If the button “B” is activated, the V-REP send a request to the external API to initiate the self-control module.

To test the simulated system through the interface, the values considered in the arm are from [16], thus the maximum torque of shoulder is 41 N.m. The failures in the movement are set to occur in 17% of the path and stop in 42% and are presented as a decrease in force that the shoulder is capable of performing. A value of 6 N.m is configured to be the maximum torque that the robot joints can execute in opposition to movement. Figures 4a and b show the movement of the robotic arm linked to the human, then Figs. 5a and b show some values related to the performed movement.

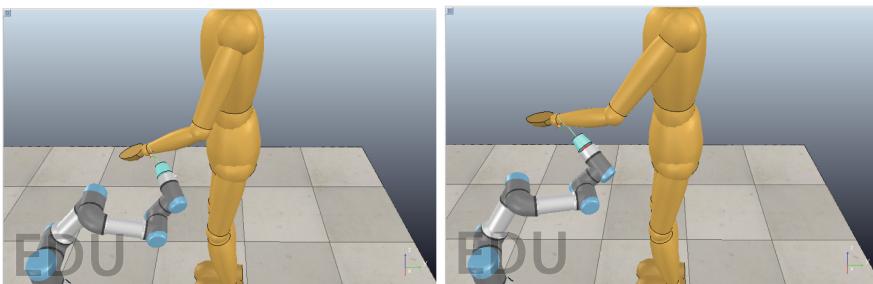
In the occurrence of a failure in the movement of the patient’s arm, immediately the force sensor records this information. As the robot dynamically change its position, should be considered the force over the three Cartesian axes  $XYZ$ , therefore, it is shown in Fig. 6 the resultant force in the movement. The movement has as reference the human shoulder, so the x-axis in the image shows the angulation of it, so the failures occur between  $300^\circ$  (17%) and  $315^\circ$  (42%). The other variations of the value measured by the sensor happen due to the direction of the movement, i.e., the value of force in the end effector will be different according to the angulation of the UR3’s joints and human arm.

When the self-control module is active, the human arm will execute the movement of raise up, using the shoulder as the main joint. The movement of



(a) Tab 1

(b) Tab 3

**Fig. 3.** Human-Machine interface for simulation control

(a) Simulated System - 40% of the path (b) Simulated System - 67% of the path

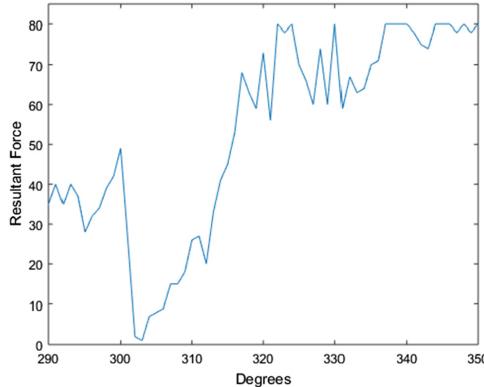
**Fig. 4.** Simulated Movement of human arm

the human arm pulls the robot attached in the wrist. However, the raising up of arm suffers a resistance force provided by the link with the robotic system, and this value is acquired and sent to MATLAB. The objective of the self-control algorithm is to provide resistance to the movement. If the force of human arm decreases, also the opposite force should decrease, then the total range of movement is reached. The algorithm also should be able to identify the improvement of the patient and provide a larger force in order to correct the variations in the movement of the arm.

General Settings		Control Panel		Matlab Comunica	
	Value		Value		Value
Joint Torque: Shoulder(N*m)	41	Joint Torque: Shoulder(N*m)	41	Joint Torque: Shoulder(N*m)	41
Joint Torque: Elbow(N*m)	24.242820739746	Joint Torque: Elbow(N*m)	19.747577667236	Joint Torque: Elbow(N*m)	19.747577667236
Joint Position: Shoulder (deg)	335	Joint Position: Shoulder (deg)	320	Joint Position: Shoulder (deg)	320
Joint Position: Elbow (deg)	326	Joint Position: Elbow (deg)	326	Joint Position: Elbow (deg)	326
UR3 Torque: Joint1 (N*m)	2.6670660972595	UR3 Torque: Joint1 (N*m)	2.6864399909973	UR3 Torque: Joint1 (N*m)	2.6864399909973
UR3 Torque: Joint2 (N*m)	-2.6541259288788	UR3 Torque: Joint2 (N*m)	-1.6248977184296	UR3 Torque: Joint2 (N*m)	-1.6248977184296
UR3 Torque: Joint3 (N*m)	-3.7374215126038	UR3 Torque: Joint3 (N*m)	-1.7394469976425	UR3 Torque: Joint3 (N*m)	-1.7394469976425
UR3 Torque: Joint4 (N*m)	1.669199347496	UR3 Torque: Joint4 (N*m)	4.8537645339966	UR3 Torque: Joint4 (N*m)	4.8537645339966
UR3 Torque: Joint5 (N*m)	5.4660406112671	UR3 Torque: Joint5 (N*m)	5.4401502609253	UR3 Torque: Joint5 (N*m)	5.4401502609253
UR3 Torque: Joint6 (N*m)	10.536356925964	UR3 Torque: Joint6 (N*m)	11.092317581177	UR3 Torque: Joint6 (N*m)	11.092317581177
Force Sensor: X (N*m)	37.997840881348	Force Sensor: X (N*m)	52.030006408691	Force Sensor: X (N*m)	52.030006408691
Force Sensor: Y (N*m)	-48.321956634521	Force Sensor: Y (N*m)	-22.969562530518	Force Sensor: Y (N*m)	-22.969562530518
Force Sensor: Z (N*m)	40.695529937744	Force Sensor: Z (N*m)	58.828010559082	Force Sensor: Z (N*m)	58.828010559082

(a) Data - 40% of the path

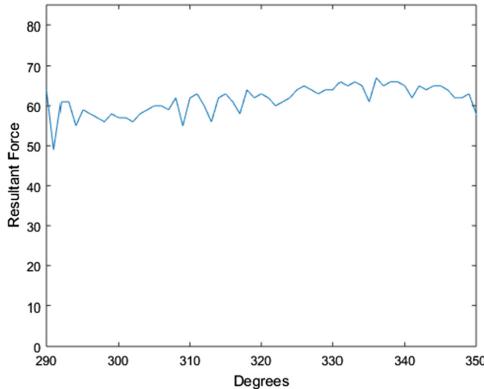
(b) Data - 67% of the path

**Fig. 5.** Data related to movement**Fig. 6.** Failures in the force of Movement

With the data acquired to the contact with the patient, the RL algorithm updates its values and find a better resistance force that the UR3 should perform, then this information return to V-REP through the synchronous communication and the UR3's maximum torque is changed. This process occurs in loop until the simulation stop or the maximum number of iterations in the RL algorithm is reached. This value of resistance torque is given as a percentage of maximum force that human arm can execute.

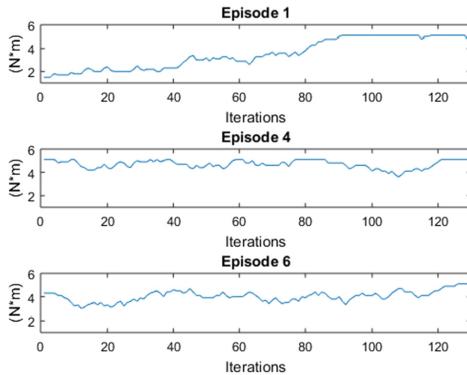
In the first episode, the behaviour of the measured force is not constant and can vary a lot, however after a few number of episodes this force tends to be constant, in Fig. 7 the acquired force is shown, after a few number of episodes.

As mentioned before, the RL insertion of the autonomous mode works with the update of a maximum torque in the collaborative robot in order to correct the resistance force. This update is shown in Fig. 8, and as expected in the first



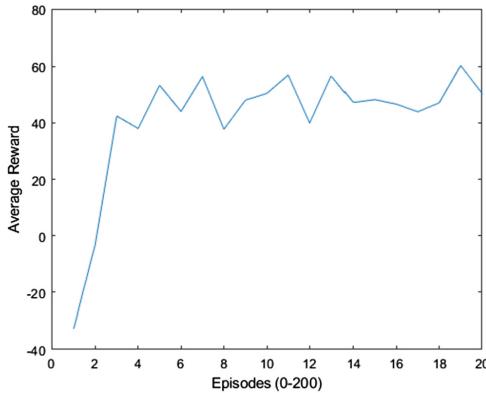
**Fig. 7.** Resultant Force with the insertion of self control module

episode, the torque increase until a value and stands around it, remaining in this condition in the other episodes. The maximum torque applied in the test was set to 5 Nm because if the value were larger, the arm could not move.



**Fig. 8.** Maximum Torque Update of UR3 in some episodes

Lastly, it is relevant to show the learning of the self-control module. That behaviour of the system can be presented through the average reward obtained over the episodes. This information represents the evolution of the system after a number of episodes and can be seen in Fig. 9, note that in figure the y-axis represents the average reward to a set of 10 episodes and the x-axis exposes this set, i.e., the first value in the x-axis represents 10 episodes and so on.



**Fig. 9.** Average reward of the RL algorithm

## 5 Conclusion and Future Works

This paper presented the used of a collaborative robot (UR3) as a robotic rehabilitation system, the simulation of it and the insertion of an autonomous mode in a virtual system via an external API. The main contribution of this work is the possibility of test the system with a model of human contact before its application. Besides, the insertion of a self-control module removes the need for the robot's path planning and its configuration to each patient. In this case, the dynamic simulation can provide a great gain for therapists, because it helps in the learning of the proposed system, besides it allows the therapists to test new methods in the simulation environment.

Another important point to consider is that simulation can be performed before the application in a real environment, thus, for new technologies, it is a way to identify possible failures and make adjustments, principally when this technology is applied together to human touch. Following the execution of this work, some relevant aspects emerge for a more detailed approach. In the simulation, the sensor force presents some noise, because of the simulation engine, which is used the Bullet 2.83 engine. Therefore, it is desired to create a model closest to the real component. For the simulation environment, it would be interesting to implement a real model of shoulder that would bring a better understanding of this approach. For a complete study, the algorithm will be implemented in a real robotic system to evaluate its performance and compare the measured data in the real environment with the simulation to verify which variables are influenced.

## References

1. Rehabilitation 2030: A call for action plan: Then need to scale up rehabilitation (2017)
2. Chatterji, S., Byles, J., Cutler, D., Seeman, T., Verdes, E.: Health, Functioning, and disability in older adults - presents status and future implications. *Lancet* **385**(9967), 563–575 (2015)
3. Union Européenne des Médecins Spécialistes (UEMS) e Académie Européenne de Médecine e Réadaptation: Livro Branco de Medicina Física e de Reabilitação na Europa. Sociedade Portuguesa de Medicina Física e de Reabilitação, Coimbra (2009)
4. Turolla, A.: An overall framework for neurorehabilitation robotics: implications for recovery. In: *Rehabilitation Robotics*, pp. 15–27. Elsevier (2018)
5. Djikcers, M., deBear, P., et al.: Patient and staff acceptance of robotic technology in occupationl therapy: a pilot study. *J. Rehabil. Res. Dev.* **28**(2), 33–44 (1991)
6. Novak, D., Riener, R.: Control strategies and artificial intelligence in rehabilitation robotics. *AI Mag.* **36**(4), 23–33 (2015)
7. Yap, R., Kono, D., et al.: Development of wereable gait assist robot using interactive motor rhythmic stimulation to upper and lower limbs. *AI Mag.* **36**(4), 23–33 (2015)
8. Casadio, M., Sanguineti, V., et al.: Braccio di Ferro: a new haptic workstation for neuromotor rehabilitation. *Technol. Health Care* (14), 123–142 (2006)
9. Amirabdollahian, F., Taylor, M., et al.: The Gentle/S project: a new method of delivering neuro-rehabilitation. *Assistive Technology - Added Value to the Quality of Life* (10), 36–41 (2001)
10. Kemna, S., Culmer, P., et al.: Developing a user interface for the iPAM stroke rehabilitation system. In: *IEEE International Conference on Rehabilitation Robotics*, Kyoto, Japan (2009)
11. Hogan, N., Krebs, H.I., Charnnarong, J., Srikrishna, P., Sharon, A.: Mit-manus: a workstation for manual therapy and training. I. In: *Proceedings IEEE International Workshop on Robot and Human Communication*, pp. 161–165. IEEE (1992)
12. Reharob: Reharob (2000). <http://reharob.manuf.bme.hu>. Accessed 06 May 2019
13. Loureiro, R., Harwin, W., et al.: Advances in upper limb stroke rehabilitation. *Med. Biol. Eng. Comput.* **49**(1), 1103–1118 (2011)
14. Coppelia Robotics. <http://www.coppeliarobotics.com>. Accessed 17 June 2019
15. Coppelia Robotics Homepage: Max. joint torques - 17260. <https://www.universal-robots.com/how-tos-and-faqs/faq/ur-faq/max-joint-torques-17260/>. Accessed 02 Sept 2019
16. Ribeiro, D.C., Estivalet, M.G., Loss, J.F.: Modelo para estimativa da força e torque muscular durante a abdução do ombro. *revista portuguesa de ciências do desporto* **8**(3), 321–329 (2008)
17. Sutton, R., Barto, A.: *Reinforcement Learning: An Introduction*, 2nd edn. A Bradford Book, London (2015)



# Grasping Angle Estimation of Human Forearm with Underactuated Grippers Using Proprioceptive Feedback

Francisco Pastor<sup>(✉)</sup>, Juan M. Gandarias, Alfonso J. García-Cerezo,  
Antonio J. Muñoz-Ramírez, and Jesús M. Gómez-de-Gabriel

Telerobotic and Interactive Systems Laboratory, System Engineering and Automation  
Department, University of Málaga, 29071 Málaga, Spain  
[{fpastor,jmgandarias,ajgarcia,aj,jesus.gomez}@uma.es](mailto:{fpastor,jmgandarias,ajgarcia,aj,jesus.gomez}@uma.es)  
<http://www.taislab.uma.es/>

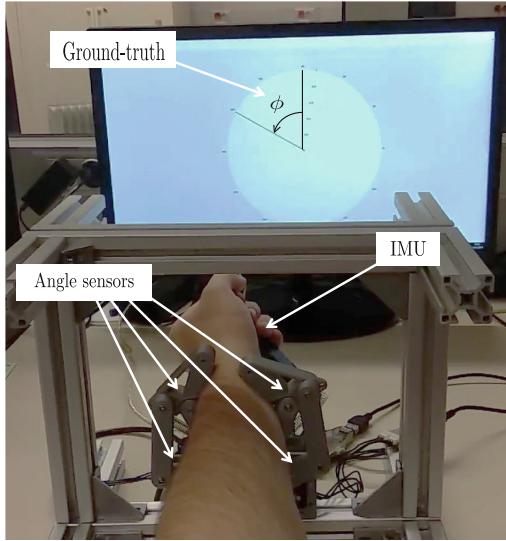
**Abstract.** In this paper, a method for the estimation of the angle of grasping of a human forearm, when grasped by a robot with an underactuated gripper, using proprioceptive information only, is presented. Knowing the angle around the forearm's axis (i.e. roll angle) is key for the safe manipulation of the human limb and biomedical sensor placement among others. The adaptive gripper has two independent underactuated fingers with two phalanges and a single actuator each. The final joint position of the gripper provides information related to the shape of the grasped object without the need for external contact or force sensors. Regression methods to estimate the roll angle of the grasping have been trained with forearm grasping information from different humans at each angular position. The results show that it is possible to accurately estimate the rolling angle of the human arm, for trained and unknown people.

**Keywords:** pHRI · Underactuated gripper · Proprioceptive sensors · Regression · Haptic perception

## 1 Introduction

Triage and initial care are crucial for victims in disaster scenarios [10]. These tasks frequently need safe human limb manipulation. Although visual methods for the estimation of the location of human limbs provide the coordinates of the human joints [7], the angle around the forearm axis (roll angle) is not included. Moreover, the roll of the forearm may change during the grasping process.

Despite the fact that the applications where robots are able to manipulate people are very interesting for robotics researchers, there are few studies that consider direct human-robot physical contacts. Most research studies that are related to physical Human-Robot Interaction (pHRI) consist of control techniques of teleoperated systems [5], exoskeletons [16], prosthetic parts or rehabilitation robots [20].



**Fig. 1.** Adaptive gripper for grasping angle estimation of human forearm. Proprioceptive sensors (angle sensors) are used to estimate the roll-angle, while an IMU is used as ground-truth to train the estimation methods

Regarding these few pHRI applications, a robot that cleanses human limbs of disabled people is presented in [14] and a robot that manipulates human limbs with a non-preassil actuator and an impedance Model Predictive Control is presented in [8]. More recent studies have considered the application of artificially intelligent techniques for robotically assisted dressing but without robot-human contact [9].

One aspect that has to be addressed in pHRI refers to robotics grippers. Although multiple ad-hoc end-effectors have been considered in previous pHRI works [22], it is still necessary to develop grippers or hands that allow the robot to carry out autonomous and safe grasps [1], with enough robustness and softness to manipulate human limbs. Some studies about pHRI grippers are based on the use of Variable Stiffness Actuators (VSA) [18], and the integration of tactile sensors and deep learning methods in robotic grippers to distinguish contacts with humans and inert objects is presented in [13].

The use of adaptive or flexible grippers enhance the in-hand manipulation by reducing the maximum pressure applied over the grasped objects [12]. However, the precision of this kind of mechanism is lower than in rigid grippers. Other existing solutions are based on underactuated rigid hands that adapt their shape over the contact surface. The underactuated and fully rigid PaCome gripper [2], originally thought for industrial applications, could also be used for pHRI applications. In [17], OpenHand is presented. It is an hybrid rigid-soft and open hardware gripper made of 3D printed polylactic acid (PLA) and polyurethane

rubber. In a recent study of the same authors, proprioceptive and tactile information are combined to classify grasps with a hybrid underactuated gripper [19].

In this paper, the problem of human forearm manipulation is addressed. To face this task, a gripper with two independent underactuated fingers has been designed and built using additive manufacturing technology. Each finger has two phalanxes, a single motor and an angle sensor placed in the underactuated joint. The main contribution of this study is to solely use proprioceptive information given by angle sensors integrated in the gripper to grasp the distal forearm and estimate the roll angle using machine learning approaches based on non-linear regression methods as shown in Fig. 1. According to this study, a robot with an underactuated gripper could grasp a distal forearm and estimate the grasping roll angle using haptic feedback only, which could open the door to new research lines into safety human manipulation.

This paper is structured as follows: In Sect. 2 the designed prototype of the underactuated gripper and its validation is presented. In Sect. 3, the new method developed for the estimation of the forearm roll angle, based on machine learning techniques is then described, and the conclusions are shown in Sect. 4.

## 2 Adaptive Gripper

There are two main approaches for the implementation of the driving mechanism: tendons and rigid linkages [4]. In this application, the use of tendons (e.g. Yale OpenHand Model T42), as in [17] has been discarded, because the tendons are located in the internal side of the fingers, so the contact surfaces of the fingers tend to pinch the skin of the forearm. Rigid linkages has been used, because they are located on the back of the finger, leaving a cleaner contact area.

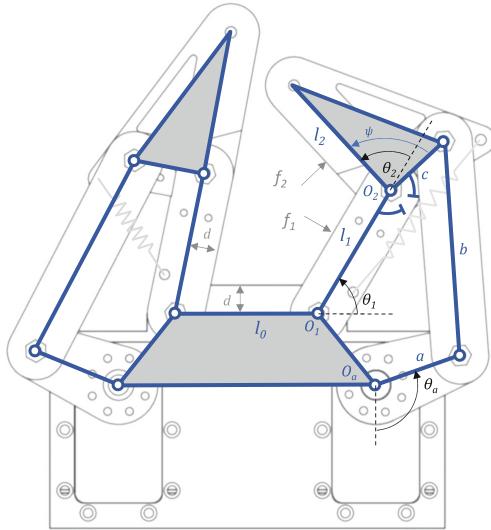
### 2.1 Kinematics

A gripper with two independent underactuated fingers with two phalanx and a single actuator has been designed with the kinematics described in Fig. 2. The parameter values, summarised in Table 1, have been designed to adapt to the shape and size of a human upper-forearm with a perimeter between 15.3 and 18.8 cm.

**Table 1.** Parameter values of the underactuated finger

Parameter	$O_1O_a$ [mm]	$L_0, L_1, L_2$ [mm]	$a$ [mm]	$b$ [mm]	$c$ [mm]	$d$ [mm]	$\psi$ [ $^\circ$ ]	Width [mm]
Value	[16, -20]	40	25	60	20	8	90	15

A mechanical limit makes the distal phalanx angle  $\theta_2$  always positive. The actual position of the finger depends on the balance between external forces  $f_1, f_2$  and the actuator torque. The spring ensures contact between the finger



**Fig. 2.** Kinematics of the underactuated fingers. Each finger is independent, with two DOF's ( $\theta_1$ ,  $\theta_2$ ) and a single actuator  $\theta_a$ . A mechanical stop makes  $\theta_2 > 0$ . The actual position of the finger depends on the external forces  $f_1$ ,  $f_2$  and the actuator torque. A spring ensures contact between the finger pads and external objects

pads and external objects and makes the finger stable when  $f_1$  or  $f_2$  are 0. The extension springs are made with  $0.6\text{ mm}\varnothing$  steel wire and a stiffness of  $164\text{ N/m}$ .

The prototype has been manufactured in PLA plastic in a *Prusa MK2* 3D-printer, and its design has been made accessible in a public repository<sup>1</sup>.

## 2.2 Proprioceptive Sensing

Analog angle sensors have been placed to measure the distal joint angles  $\theta_2$ . The actuators provide feedback on the servo position, so the full position of the adaptive fingers can be estimated. This way, proprioceptive joint sensors provide information about the final gripper position that is related to the shape and size of the grasped object, without the need of external contact or force sensors.

Miniature potentiometers from *muRata* (model SV01 10 k $\Omega$  linear) have been used successfully for the measurement of the distal joints of both fingers ( $\theta_{2l}$ ,  $\theta_{2r}$ ). The analog signals are measured using a micro-controller with 10-bits ADC, (0.26° resolution) at a rate of 50 Hz.

The *Dynamixel MX-28* servos have a magnetic encoder with 12-bits (0.088° resolution) at a rate of up-to 50 Hz with our current set-up. They provide feedback of the servo positions  $\theta_{al}$  and  $\theta_{ar}$ .

---

<sup>1</sup> <http://github.com/TaISLab/umahand>.

### 2.3 Grasping Forces

The forces at the center of the phalanxes contact areas in an underactuated gripper depend also on the joint values. Moreover the spring stiffness [3] is also in the equation.

$$f = J^{-T} T^{-T} t \quad (1)$$

where  $f$  are the contact forces,  $J^{-T}$  is the inverse of the transposed Jacobian matrix that relates the finger joint velocities to the speed of the contact points, and  $T$  is the transfer matrix, that relates the velocities of the actuators to the joint velocities. Both matrices depend on the joints and actuator positions [4]. However, for a position of the gripper given ( $\theta_1$  and  $\theta_2$  remains constant), the magnitude of the closing force of each finger ( $f_1 + f_2$ ) is proportional to the actuator torque  $\theta_a$ . Closing forces have been experimentally measured for different actuator torques at the same grasping position. With a maximum stall torque for each of the *Dynamixel MX-28* servos of 2.5 Nm, the effective closing forces for each finger range from 4.9 N (20%) to 27.4 N (100%).

### 2.4 Getting Ground-Truth Data

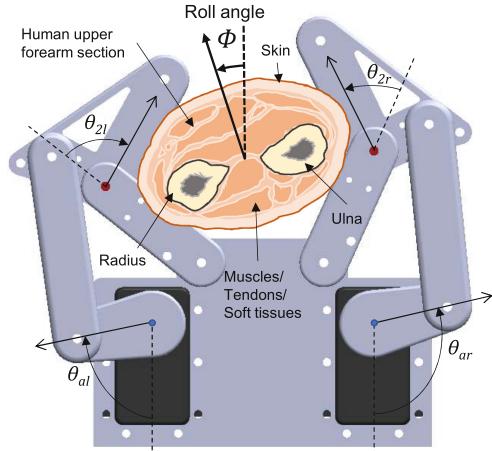
In order to obtain the ground-truth angular measurement of the human forearm, a device that includes an accelerometer has been implemented. The device is held by the volunteers with their hand during the experiments. As the attitude of the robot gripper is known, the relative rolling angle of the human forearm with respect the gripper can be obtained and used as a reference data for training and performance evaluation of estimation methods.

## 3 Forearm Roll Angle Estimation

The proposed method is based on the differences in the final grasping positions of the finger joints when grasping a human forearm, thanks to the internal bone structure, as seen in Fig. 3. The human forearm is supported by Ulna and Radius, which specially at the upper section of the forearm (near the wrist) provide an elliptical shape. The roll angle is measured with respect to the pose in which Ulna and Radius are parallel to the gripper base.

### 3.1 Measurements

During the measurements, the volunteer subject holds a 3D printed handle which integrates the accelerometer, while the gripper closes repeatedly around their forearm. The subject has to rotate the forearm in each iteration, so that the accelerometer measures this new angle, which corresponds to the roll angle. The whole gripper has been mounted on a square frame to perform experiments in different positions. The actual roll angle is computed as the orientation difference between the gripper and the readings from the inertial sensor bar. This process is



**Fig. 3.** Cross section of a human upper forearm grasped by the underactuated gripper, showing the variations in the passive ( $\theta_{2l}$ ,  $\theta_{2r}$ ) and active ( $\theta_{al}$ ,  $\theta_{ar}$ ) DOF's, based on the roll grasping-angle ( $\Phi$ )

shown in Fig. 4, where the arm angle and joint values are recorded for a sequence of six grasps.

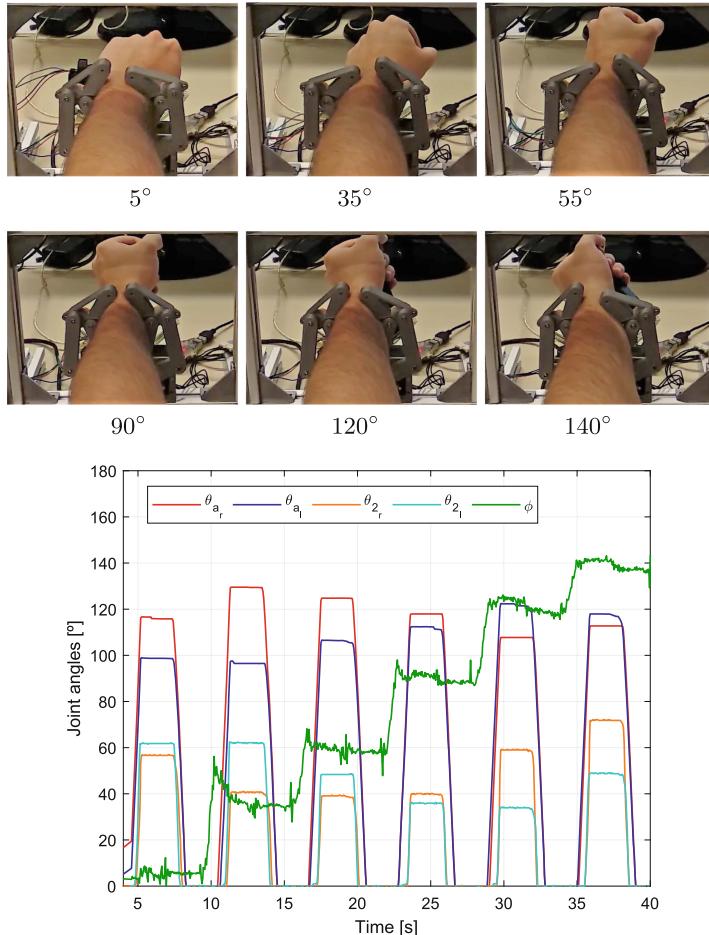
Performing many different grasps at different angles, the relationship between the gripper joints and the roll can be obtained. In Fig. 5, the data of 36 grasps, measured at different roll angles, on the left arm of a volunteer with a perimeter of 17.9 cm, is shown.

### 3.2 Machine-Learning Regression Methods

To estimate the roll angle ( $\phi$ ), machine learning methods, like regression and Deep Learning have been considered. Regression methods have been implemented due to they have lower computational costs [11]. Three machine learning approaches based on non-linear regression methods are used to obtain three models: Gaussian Process Regression (GPR) [21], Regression Tree (RT) [6] and Bagging Regression Tree (BRT) [15]. These models receive the 4-input angles ( $\theta_{ar}$ ,  $\theta_{al}$ ,  $\theta_{2r}$ ,  $\theta_{2l}$ ) and predict  $\phi$ .

All these models are trained in a large dataset. In the training process, features are composed by sets of ( $\theta_{ar}$ ,  $\theta_{al}$ ,  $\theta_{2r}$ ,  $\theta_{2l}$ ), while expected responses composed by  $\phi$  are measured with the accelerometer for each set of features. A cross-validation has also been included during the training process to prevent overfitting.

The training and evaluation processes have been carried out using *Matlab R2018b*, the *Statistics and Machine Learning Toolbox* and the *Regression Learner application*. The code and datasets have been made available in the repository referred in Sect. 2.

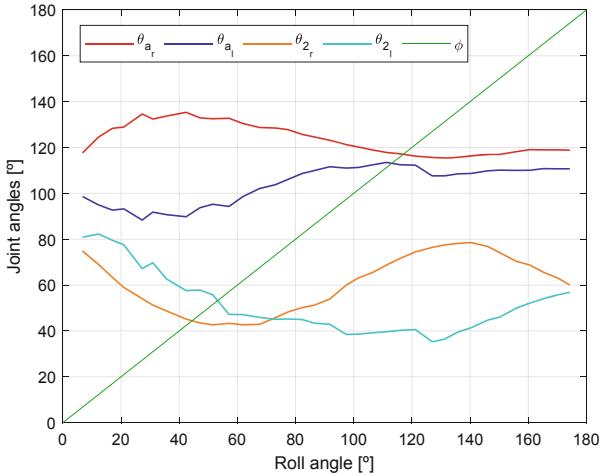


**Fig. 4.** Sequence of 6 grasps with different roll angles during the data collection process of one volunteer (top). Motors, joints and roll angle positions during the data collection process (bottom). Note that in this graph the open-and-close process is repeated each 6 s

The training process have been carried out using the *Parallel Computing Toolbox* in a 4-core Intel i7-7700HQ CPU @ 2.80 GHz.

### 3.3 Data Collection

During this process, a User Interface developed in Matlab shows goals and current angles so that the subject could rotate their forearm until goal and actual angle match. In each step, the gripper opens and closes, so while the gripper is closed, ( $\theta_{ar}, \theta_{al}, \theta_{2r}, \theta_{2l}$ ) and ( $\phi$ ) values are collected.



**Fig. 5.** Joint values for 36 grasps, measured at different angles, on the left arm of a volunteer with a perimeter of 17.9 cm, against the roll angle

There are two types of data collection processes which have been performed: sequential and random. In the sequential process, the subject is asked to rotate their arm in steps of  $5^\circ$ . In the latter, the goal angle is set randomly.

Following this process, two data-sets have been collected: the first dataset contains information of a single subject, while the second dataset contains information of five subjects. To train and test the models, each dataset is split into training and test sets respectively. Therefore, training and test data are different, even if they have been collected from the same person.

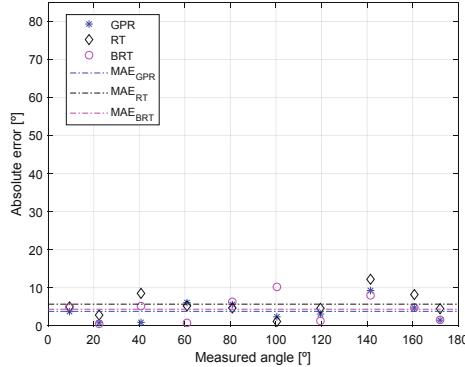
The dataset collected from one subject contains an amount of 555 sets of  $(\theta_{ar}, \theta_{al}, \theta_{2r}, \theta_{2l})$  and  $(\phi)$ , while the dataset collected from 5 subjects is composed by 2775 sets of data.

### 3.4 Results

Three experiments have been carried out to measure the performance of the regression models estimating the roll angle:

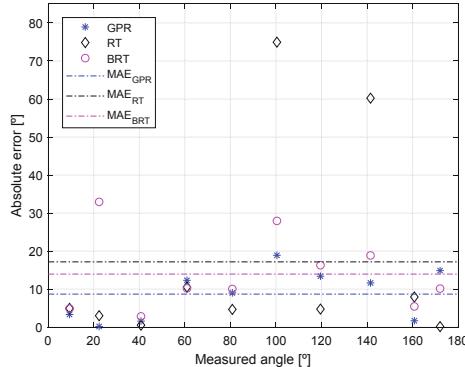
1. **Single-known:** Training and test sets contain data from one subject.
2. **Multiple-known:** Training and test sets contain data from four subjects.
3. **Multiple-unknown:** Training set contains data from four subjects and test set contains data from a fifth subject which is not used to train the models.

In the single-known experiment, machine learning models have been trained with data from a single volunteer and have tested with different data from the same subject. The results of this experiment are shown in Fig. 6. This figure shows the good performance of the regression models, with an almost negligible error, and the best results obtained by GPR.



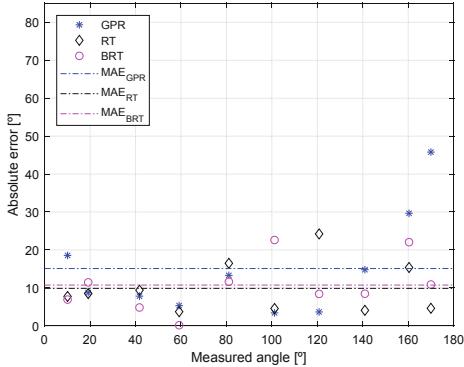
**Fig. 6.** Regression errors results of non-trained data of a known subject when models are trained in this subject dataset only

In the multiple-known experiment, machine learning methods have been trained with data from 4 volunteers and have been tested with different data from one of these 4 subjects. In this case, predictions from RT and BRT include a pair of outliers, however the GPR model still presents a good performance as can be seen in Fig. 7.



**Fig. 7.** Regression errors results of non-trained data of a known subject when models are trained in a dataset obtained from 4 volunteers

In the multiple-unknown experiment, regression models have been trained with data from 4 subjects and tested with completely new data from a fifth volunteer that had not been used in the training process. Results of this experiment are presented in Fig. 8 and show that RT and BRT are more robust than GPR because they generalize better, and the outliers predicted in the previous experiment vanish.



**Fig. 8.** Regression errors results of non-trained data of an unknown user when models are trained in a dataset obtained from 4 people

Results are summarized in Table 2. The Maximum Error (ME) and Mean Absolute Error (MAE) from each model in each experiment are represented in degrees. As is commented before, GPR obtains the best results when the subject is known, with a MAE of  $3.77^\circ$  in the case of a single user and  $8.68^\circ$  in the case of multiple subjects. However, for the third experiment, the best results are achieved by the RT model with a MAE of  $9.86^\circ$  since this model generalizes better than the others. The maximum error shows the outliers in experiments.

**Table 2.** Summary of the errors of the regression models in degrees

Models	Single-known		Multiple-known		Multiple-unknown	
	ME	MAE	ME	MAE	ME	MAE
GPR	9.24	3.77	18.89	8.68	45.79	15.09
RT	12.20	4.33	74.94	17.17	24.22	9.86
BRT	10.19	5.66	32.94	13.95	22.61	10.73

## 4 Conclusions

With this method, new applications of robot-initiated pHRI can be developed without the need of external force/tactile sensors that are expensive or hard to deploy and maintain. These applications may include assistive, rescue or surgical robotics. With this approach, information on the location of the human limbs can be enhanced without the need of additional sensors. In pHRI applications, an accurate location of the human forearm is important not only for a safe human arm manipulation, but also for the placement of biomedical devices such as heart-rate or glucose sensors.

The results demonstrate the good performance of the regression methods used in this application. The errors obtained denote these predictors can be used for pHRI applications. However, in future works, a larger dataset, and a greater number of volunteers may be considered, as well as the use of other prediction methods as deep learning.

Future works after this method may include forearm width estimation for the recognition of people, quality of grasping estimation and skin compliance identification for health evaluation.

**Acknowledgment.** This work was supported by the Spanish project DPI2015-65186-R, the European Commission under grant agreement BES-2016-078237, the Telerobotics and Interactive Systems Laboratory (TaIS Lab) and the Systems Engineering and Automation Department, University of Málaga, Spain.

## References

1. Armendariz, J., García-Rodríguez, R., Machorro-Fernández, F., Parra-Vega, V.: Manipulation with soft-fingertips for safe pHRI. In: Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 155–156. ACM (2012)
2. Birglen, L.: Enhancing versatility and safety of industrial grippers with adaptive robotic fingers. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2911–2916 (2015)
3. Birglen, L., Gosselin, C.: Optimal design of 2-phalanx underactuated fingers. In: Proceedings of the 2004 International Conference on Intelligent Manipulation and Grasping, pp. 110–116 (2004)
4. Birglen, L., Laliberté, T., Gosselin, C.M.: Underactuated Robotic Hands. Springer, Heidelberg (2008)
5. Bowyer, S.A., y Baena, F.R.: Dissipative control for physical human–robot interaction. *IEEE Trans. Robot.* **31**(6), 1281–1293 (2015)
6. Breiman, L.: Classification and Regression Trees. Routledge, Abingdon (2017)
7. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: real-time multi-person 2D pose estimation using part affinity fields. arXiv preprint [arXiv:1812.08008](https://arxiv.org/abs/1812.08008) (2018)
8. Chow, K., Kemp, C.C.: Robotic repositioning of human limbs via model predictive control. In: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 473–480. IEEE (2016)
9. Erickson, Z., Clever, H.M., Turk, G., Liu, C.K., Kemp, C.C.: Deep haptic model predictive control for robot-assisted dressing. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1–8 (2018)
10. Frykberg, E.R.: Medical management of disasters and mass casualties from terrorist bombings: how can we cope? *J. Trauma Acute Care Surg.* **53**(2), 201–212 (2002)
11. Gandarias, J.M., Gómez-de Gabriel, J.M., García-Cerezo, A.J.: Human and object recognition with a high-resolution tactile sensor. In: IEEE Sensors, pp. 1–3 (2017)
12. Gandarias, J.M., Gómez-de Gabriel, J.M., García-Cerezo, A.J.: Enhancing perception with tactile object recognition in adaptive grippers for human-robot interaction. *Sensors* **18**(3), 692 (2018)

13. Gandarias, J.M., García-Cerezo, A.J., Gómez-de Gabriel, J.M.: CNN-based methods for object recognition with high-resolution tactile sensors. *IEEE Sens. J.* **19**(16), 6872–6882 (2019). <https://doi.org/10.1109/JSEN.2019.2912968>
14. King, C., Chen, T.L., Jain, A., Kemp, C.C.: Towards an assistive robot that autonomously performs bed baths for patient hygiene. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 319–324 (2010)
15. Breiman, L.: Bagging predictors. *Mach. Learn.* **24**(2), 123–140 (1996)
16. Li, Z., Huang, B., Ye, Z., Deng, M., Yang, C.: Physical human-robot interaction of a robotic exoskeleton by admittance control. *IEEE Trans. Ind. Electron.* **65**, 9614–9624 (2018)
17. Ma, R.R., Odhner, L.U., Dollar, A.M.: A modular, open-source 3D printed underactuated hand. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2737–2743 (2013)
18. Memar, A.H., Mastronarde, N., Esfahani, E.T.: Design of a novel variable stiffness gripper using permanent magnets. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 2818–2823 (2017)
19. Spiers, A.J., Liarokapis, M.V., Calli, B., Dollar, A.M.: Single-grasp object classification and feature extraction with simple robot hands and tactile sensors. *IEEE Trans. Haptics* **9**(2), 207–220 (2016)
20. Stilli, A., Cremoni, A., Bianchi, M., Ridolfi, A., Gerii, F., Vannetti, F., Wurdemann, H.A., Allotta, B., Althoefer, K.: AirExGlove - a novel pneumatic exoskeleton glove for adaptive hand rehabilitation in post-stroke patients. In: IEEE International Conference on Soft Robotics (RoboSoft), pp. 579–584 (2018)
21. Williams, C.K.: Prediction with Gaussian processes: from linear regression to linear prediction and beyond. In: Learning in Graphical Models, pp. 599–621. Springer (1998)
22. Yang, C., Zeng, C., Liang, P., Li, Z., Li, R., Su, C.Y.: Interface design of a physical human-robot interaction system for human impedance adaptive skill transfer. *IEEE Trans. Autom. Sci. Eng.* **15**(1), 329–340 (2018)



# Integration of a Surgical Robotic Co-worker in an Endoscopic Neurosurgical Assistance Platform

Álvaro Muñoz García<sup>1</sup>, Irene Rivas<sup>2</sup>, Javier P. Turiel<sup>1(✉)</sup>,  
Víctor Muñoz<sup>2</sup>, Juan Carlos Fraile Marinero<sup>1</sup>, Eusebio de la Fuente<sup>1</sup>,  
and Jose María Sabater<sup>3</sup>

<sup>1</sup> ITAP, Universidad de Valladolid, Valladolid, Spain

alvaro.munoz.garcia@alumnos.uva.es,  
{turiel, jcfraile, efuente}@eii.uva.es

<sup>2</sup> Escuela de Ingenierías Industriales, Universidad de Málaga, Málaga, Spain  
{irerivbla, vfmm}@uma.es

<sup>3</sup> Escuela Politécnica Superior de Elche, U. Miguel Hernández,  
Elche, Alicante, Spain  
j.sabater@umh.es

**Abstract.** Endonasal Endoscopic Approach (EEA) is a minimally invasive technique for interventions in the skull base in which specific surgical tools and an endoscope are introduced directly through the nose and sinuses. This approach avoids scars in the patients face and reduces the recovering time compared with other techniques. However, it requires expertise and high accuracy movements, since in the operating area there are critical anatomical structures as well as displacements of the brain matter due to the change of the internal pressure during the craniotomy (brain-shift). The CRANEEAL proposal addresses the problem of developing a co-worker robotic system for minimally invasive neurosurgery, under a learning cognitive scheme. The system will provide assistance to the neurosurgeon with automatic, collaborative or shared-control behaviors, as well as providing an augmented reality environment. In this contribution we present the project fundamentals, with special detail in the functional and integration aspects of the surgical co-worker robot that provides autonomous assistance to the surgeon during the intervention.

**Keywords:** Surgical robotics · Neurosurgery · Endoscopic surgery · Cooperative systems · Mechatronics

## 1 Introduction

Neurosurgery assistant robots focus on tele-operated or shared-control systems, designed for restricting movements in intracranial surgery, or for the insertion of electrodes for deep brain stimulation using a key-hole approach. However, there are brain lesions at the skull base such as pituitary tumors, chordomas, craniopharyngioma, cysts or meningiomas that are difficult to access using the techniques described and, for this, an endonasal approach is used. This technique is complicated by the high

precision required in the movements of the neurosurgeon, as well as by the presence in the operative field of critical anatomical structures such as the optic nerves or the carotid. Also, when accessing the cranial cavity through a hole, there is a deformation of the brain called brain-shift (by changing the internal pressure of the skull) that completely displaces all structures of this organ, and therefore, the brain locations that should be treated in the intervention.

Pituitary tumors represent about 25% of all endo-cranial tumors. Traditionally, this type of surgery was performed through transcranial or transfacial access, which requires large holes in the patient's forehead or cheekbones. However, the endonasal approach allows for less invasive interventions, with less trauma for patients, fewer postoperative complications and shorter operation time [1]. Despite the numerous advantages of this type of approach, less than 50% of pituitary surgeries are performed following this philosophy. This is due to the great complexity involved in this type of intervention for surgeons, who have to handle the tools through very small holes [2]. In the so called "two-surgeons-four-hands" technique, the surgeon introduces two tools through the right nostril, while the assistant handles the endoscope and a surgical aspirator through the left nostril [3]. To extract the pituitary tumor, first, the nasal septum is removed, which allows to increase the working space for the tools. Next, in order to access the tumor, it is necessary to remove the anterior wall of the sphenoid sinus and then perforate the sella turcica to expose the dura mater. Once the dura mater is open, the tumor is already accessible and it can be extracted.

The use of robotic assistants specially designed for this type of surgery allows to increase the accuracy and safety during the movements of the tools, also allowing to reduce the operating times [4]. In comparison with the investigations carried out in the implementation of robots for intravascular interventions [5] or for surgery through other natural orifices [6–8], very few studies have focused on the endonasal approach. Some of these robots designed especially for endonasal surgery have focused on ensuring safety during the process of bone drilling [9], while others have been used to provide assistance during the operation of the endoscope [10, 11].

Another of the main problems of the procedures of EEA for surgical robotics resides in the restriction of movements for the tools posed by the nasal septum and, more specifically, for the endoscope [12]. Unlike abdominal Minimally Invasive Surgery (MIS), where the point of insertion in the abdominal wall or fulcrum allows some flexibility, in EEA the efforts exerted in the vicinity of the nasal orifice should be minimized, for its greater rigidity and delicacy of the tissues. Up to now, the works published for the navigation of instruments in EEA have been reduced to the study of the intranasal work space [13]. Another line of research addresses the problem of planning safe trajectories in neurosurgery [14] taking into consideration the movement restriction discussed.

In this article a robotic assistant is proposed to guide, through assisted teleoperation, the surgeon during the process of perforation of the sella turcica, prior to the removal of tumors in the pituitary gland. The proposed scheme is intended to guide the surgeon both in the control of the position of the tool and in the control of the force exerted during the drilling.

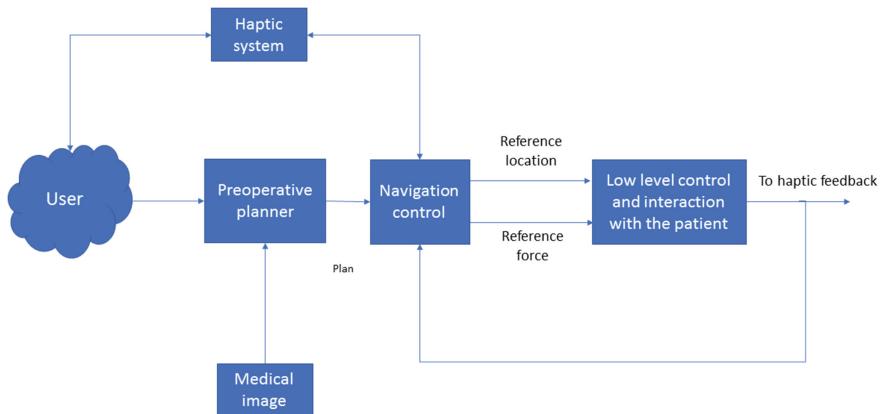
The rest of the document is structured as follows: in section two we describe the approach of the CRANEEAL platform, its main components and the low-level control

strategy proposed. In section three the functional and system integration aspects of the autonomous manipulator that acts as surgical assistant of the surgeon are presented. Finally, in section four, the conclusions of the work are summarized.

## 2 CRANEEAL: A Robotic Collaborative Platform for Endonasal Endoscopic Surgery

### 2.1 Approach of the CRANEEAL Project

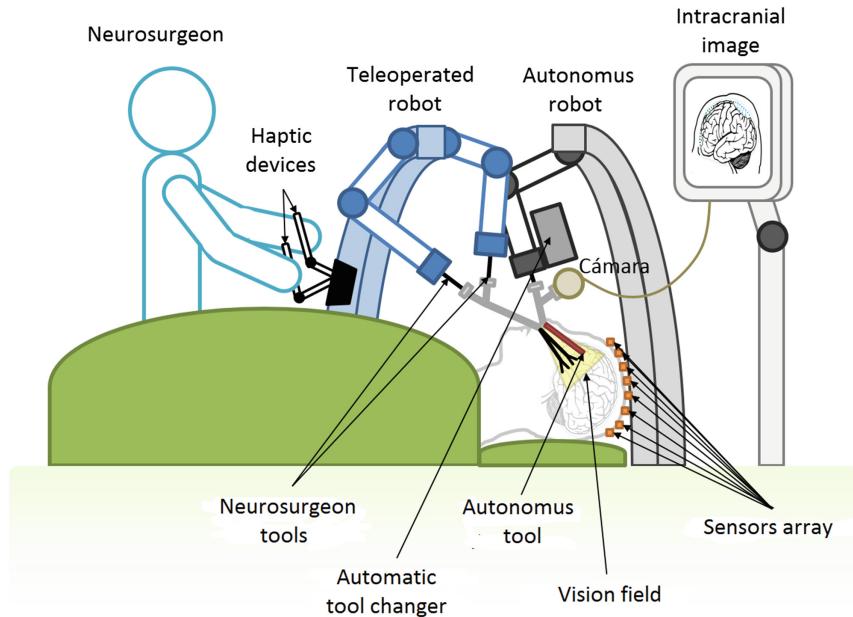
The CRANEEAL (Collaborative Robotic system for brAin-shift corrections in Neurosurgical Endoscopic Endonasal AppLicationS) project proposes a collaborative robotic system concept capable of assisting the surgeon in a scenario of endonasal endoscopic surgery. This scenario starts from the information obtained in the preoperative, which includes both the three-dimensional model of the patient's skull and the identification by the neurosurgeon of the anatomical areas of interest. In addition, the robotic system will allow off-line planning of surgical interventions through a simulation and modeling of the intervention. The overall functionality of the platform is summarized in the block diagram of Fig. 1.



**Fig. 1.** Block diagram of the main sub-systems of the CRANEEAL platform

The information gathered in the preoperative period will serve as a basis for the robotic system to act during the intraoperative phase following the scheme presented in Fig. 2. As can be seen, the Neurosurgeon tools, controlled through the arms of the Teleoperated Robot, either in shared mode or tele-operated with the use of Haptic Devices, perform the necessary tasks to free the way to the anatomical area of interest. In turn, the Autonomous Robot provides assistance with an additional instrument navigated autonomously and supervised by the Neurosurgeon at all times. The information obtained from this supervision will be used to carry out online learning that will correct future autonomous decisions. Other functions that the autonomous robot can

perform are the mapping of neurological activity through an electrode that indicates the proximity to the affected tissue; and the autonomous navigation of the aspiration instruments.



**Fig. 2.** Main components of the robotic sub-system

On the other hand, the Autonomous robot also controls the movement of the camera and its navigation depending on the phase of the intervention [15, 16], while the Intra-cranial image obtained integrates the elements of interest identified in the pre-operative using augmented reality algorithms. These elements will have modified their locations, mainly due to the perforation of the dura mater and the loss of pressure of the cranial cavity (brain-shift), which is why an array of sensors is used to record certain points of the brain with which to correct these deformations.

## 2.2 Low Level Control Strategy

One of the main challenges of the CRANEEAL project is the coordination between the two robotic devices that are integrated into the surgical platform. This section describes the low level control strategy needed to achieve this coordination.

**Tele Operated Robot:** This robot incorporates a force-torque sensor and carries the drilling tool. Its maneuvers and movements are controlled directly by the surgeon using a haptic device. Control “inputs” for this robot are:

- Ptd: The drilling position to which we wish to carry the robot (Ptd - Desired drilling position).

- F<sub>td</sub>: The force that we want to exert with the TCP of the drill on the surface of the body/organ that we drill (F<sub>td</sub> - Desired drilling force).

The vision system provides the position (X<sub>f</sub>, Y<sub>f</sub>, Z<sub>f</sub>), of the nasal orifice. When the surgeon introduces the drill through the nostril (using a haptic device), contact forces may appear at the fulcrum point (because the entry is not made exactly by the point (X<sub>f</sub>, Y<sub>f</sub>, Z<sub>f</sub>), otherwise by a point (X<sub>e</sub>, Y<sub>e</sub>, Z<sub>e</sub>)). Therefore, the possibility of a positioning error at the fulcrum point ( $\Delta X$ ,  $\Delta Y$ ,  $\Delta Z$ ) = (X<sub>e</sub>-X<sub>f</sub>, Y<sub>e</sub>-Y<sub>f</sub>, Z<sub>e</sub>-Z<sub>f</sub>) must be taken into account.

On the other hand, when the surgeon is driving the drill with the haptic device, in order to avoid that the TCP of the drill touches the walls of the nasal cavity, an algorithm to avoid collisions based on the potential method is implemented.

As soon as the drilling tool reaches the target point, it begins to drill and contact forces appear at this point (F<sub>t</sub>) (measured by the force sensor). It must be taken into account that at the fulcrum point there can also be contact (due to position error at the fulcrum point). Therefore, at that moment the force sensor is measuring a total force (F<sub>s</sub> - total force measured by the sensor), which is the sum of two forces: the force at the fulcrum point (F<sub>f</sub>), and the force at the Drilling point (F<sub>t</sub>) → F<sub>s</sub> = F<sub>f</sub> + F<sub>t</sub>.

The force at the fulcrum point (F<sub>o</sub>), can be calculated using a model from the positioning error. Therefore, knowing F<sub>s</sub> (provided by the sensor) and F<sub>f</sub>, we can know what force F<sub>t</sub> we are applying at the drilling point and compare it with the input force F<sub>td</sub>.

**Autonomous Robot:** This robot manages the laparoscopic camera and the surgical aspirator. It incorporates a torque-force sensor and works in an autonomous way. The control “inputs” for this robot are:

- P<sub>d</sub>: The desired position for the laparoscopic camera or the aspirator
- F<sub>d</sub> = 0. Restriction that implies that the laparoscopic camera or the aspirator should not contact any point of the nasal cavity.

When the robot begins to introduce the camera through the nostril, to reach the target point P<sub>d</sub>, two situations can occur:

- If there is contact with the wall of the nasal cavity, a contact force will appear (F<sub>c</sub>).
- If at the fulcrum point there is also contact (by position error), the force at that point will be calculated (using a model, as indicated above) (F<sub>f</sub>).

At that moment, the force sensor is measuring the total force (F<sub>s</sub> - total force measured by the sensor), which is the sum of two terms: the force at the fulcrum point (F<sub>f</sub>), and the force at the contact point. (F<sub>c</sub>) → F<sub>s</sub> = F<sub>f</sub> + F<sub>c</sub>. The control algorithm must implement a strategy such that

- F<sub>c</sub> is canceled (F<sub>c</sub> = 0), that is, the robot must move the camera to a point where there is no contact with the nasal cavity, and
- F<sub>f</sub> is also canceled (F<sub>f</sub> = 0), that is, the robot must move the camera to the fulcrum point.

**Low Level Control:** In order to reach the goals detailed in the two previous points, a behavior-based control scheme is used. Figure 3 shows a generic autonomous camera movement, where it will be displaced an alpha angle by rotating over the  $Y_m$  axis of the frame {M}, fixed to the base of camera where the force sensor is placed.

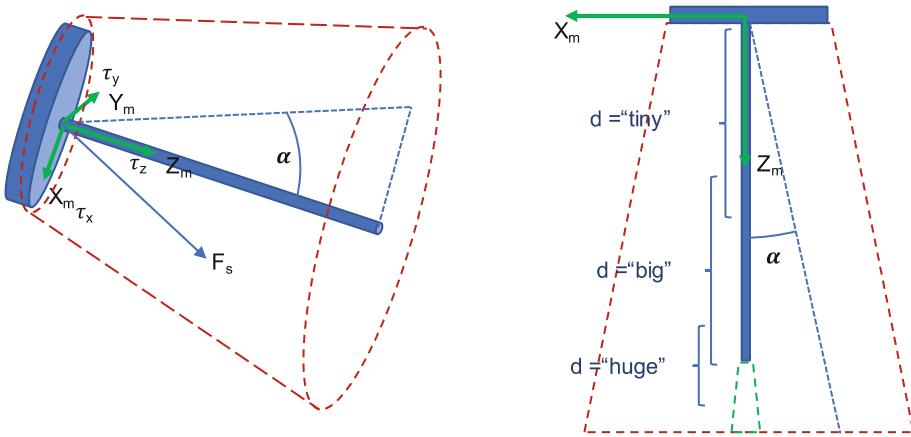


Fig. 3. A generic camera displacement.

The red dotted area represents the endonasal working space. The scheme on the left is the three-dimensional representation of an alpha angle rotation around  $Y_m$ , while the force sensor registers  $F_s = (f_x, f_y, f_z)$  forces and  $\tau_s = (\tau_x, \tau_y, \tau_z)$  moments due both for contact forces and the fulcrum interaction. This information will be used to compute the virtual distance “d” from the sensor, where the resultant force  $F_s$  is applied.

Distance “d” will be used to activate a control system behavior in order to fulfill the above requirements,  $F_s = F_c = F_f = 0$ . In this way, the right scheme at Fig. 3 shows the same camera movement where “d” can classify as a fuzzy variable as tiny, big and huge depending on the virtual distance from the sensor:

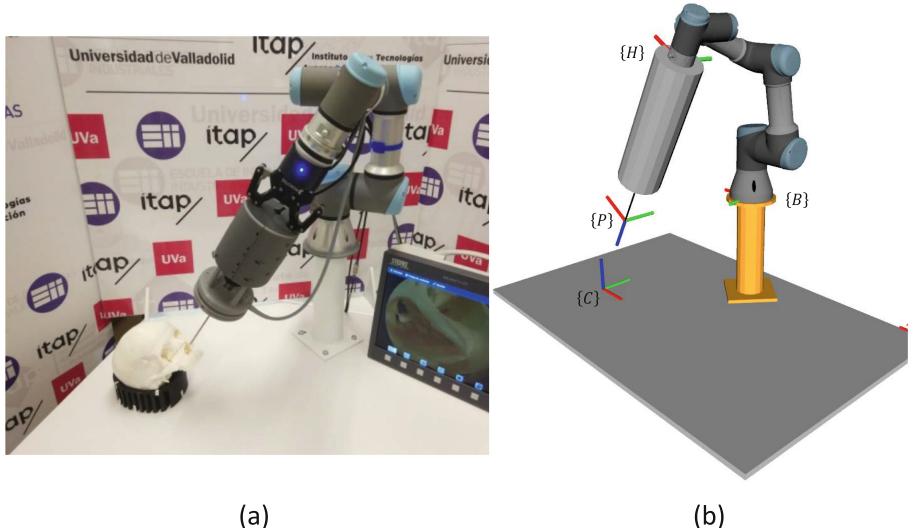
- If “d = tiny” then the virtual application point of  $F_s$  is placed close to the fulcrum point, that is,  $F_f$  is big. In this situation the control system will displace the tool along the  $-Y_m$  axis without changing the orientation. This action is designed to decrease  $F_f$ .
- If “d = big” then the virtual application point of  $F_s$  is placed close to the camera tip. In order to decrease the contact forces, the control system rotates the camera around its  $X_m$  axis, the control system trying to separate the camera tip from the nasal septum.
- If “d = huge” then there are not interaction forces neither on the tip of the camera nor on the fulcrum point. The camera can perform the programmed movement.

### 3 Integration of the Autonomous Co-worker Robot

#### 3.1 Description of the Experimental Platform

For the integration of the different subsystems of the CRANEEAL platform, we have decided to use the ROS (Robot Operating System) environment [17] given its wide diffusion, the availability of functional modules and the tools it provides for the addition of new devices. Specifically, for the integration of the autonomous robot we have used the MoveIt framework. MoveIt is an open-source framework which runs on top of ROS and provides various functionalities, such as motion planning, manipulation, inverse kinematics, control, 3D perception and collision checking.

Figure 4 shows the experimental setup that we have implemented in our laboratory (a) and the different reference systems that have been used (b). The reference system  $\{O\}$  is common to both robotic arms, and is located in the upper right corner of the operating table. The positions and orientations will always refer to this system  $\{O\}$ .



**Fig. 4.** Experimental setup used in our lab (a) and orthonormal reference frames considered (b)

The base of the robot is associated with the reference system  $\{B\}$ , which shares orientation with  $\{O\}$ . In the end effector of the robot is attached a force/torque sensor that has assigned the reference system  $\{H\}$ , whose Z axis is aligned with the longitudinal direction of the endoscope and its end is represented by  $\{P\}$ . The orientation of  $\{H\}$  and  $\{P\}$  is the same. The position and orientation of the skull on the operating table will be determined by  $\{C\}$ .

Finally, and although it is not shown in the figure,  $\{F\}$  is defined as the fulcrum point with respect to which the tool pivots and which coincides with the insertion point and the center of the nasal orifice.

The reference systems are related to each other through homogenous transformation matrices  ${}^aT_b$ , being  $a$  the origin reference system, and  $b$  the destination.

The notation used to describe the position and orientation of a reference system is as follows:  $\{b\} = \{{}^a p_b, {}^a o_b\}$ , being  ${}^a p_b$  a dimension 3 vector representing the XYZ position, and  ${}^a o_b$  a dimension 3 vector representing the RPY Euler angles (*Roll*, *Pitch* and *Yaw*, that is, rotation with respect to the XYZ axes, respectively). In both cases, superscript  $a$  indicates the RD to which both position and orientation are referred.

The forces and moments obtained by the F/T sensor in  $\{H\}$  will be represented by  $\vec{f}_H$  y  $\vec{\mu}_H$ , respectively. Both vectors are of dimension 3 and indicate the force and momentum on the XYZ axes. In the simulation model shown in Fig. 4, a large block between the endoscope and the end of the robot can be seen. This is due to the fact that we have decided to group the F/T sensor, the clamp and the tool holder in the same geometry, in order not to make the model more complex and to speed up the calculation of trajectories and especially collision detection.

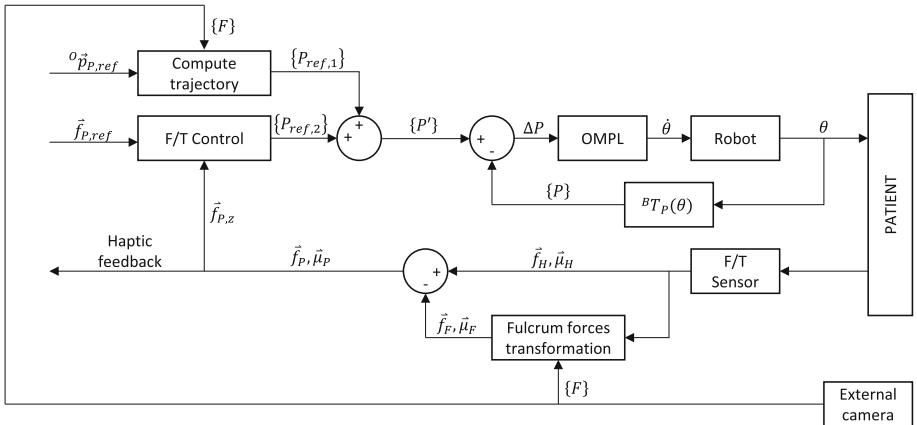
The equipment we use in our experimental set-up includes a UR3 model collaborative robotic arm from Universal Robots®, a 2F-85 clamp from Robotiq®, a 3D printed tool holder, an endoscope (ref 28132AA), a vacuum cleaner (ref 649179C) and a coagulation electrode (ref 11161AB), all of them from Karl Storz®, an HEX-E force/torque sensor from OnRobot® and a model tof640-20gm time-of-flight camera from Basler®.

### 3.2 Low Level Control Architecture

As previously said, the control system is based on the ROS (Robotic Operating System) framework, using MoveIt as a robot control package and trajectory planning. The main computer communicates with the robot and the F/T sensor through sockets, and with the clamp via USB.

The low level control architecture is common to both robotic arms and is shown in Fig. 5. The system has two inputs, position and force references, both in relation to the  $\{P\}$  reference system. The position reference  ${}^0 \vec{p}_{P,ref}$  is provided by another high-level sub-control system. On the other hand, the force reference  $\vec{f}_{P,ref}$  will always be zero in the case of the robot carrying the endoscope, while the reference for the teleoperated robot will depend on the operation phase and will be given by another high-level subsystem.

The external camera will provide the position of the fulcrum point  $\{F\}$  at the beginning of the operation, which coincides with the center of the nostril and the insertion point. This information will be used to obtain the forces and moments in the fulcrum ( $\vec{f}_F$  y  $\vec{\mu}_F$ ) from the forces in  $\{H\}$ , where the F/T sensor is coupled ( $\vec{f}_H$  and  $\vec{\mu}_H$ ). If we subtract from the total force in  $\{H\}$  the forces calculated in the fulcrum  $\{F\}$ , we obtain the force exerted by the end of the endoscope  $\{P\}$ . These forces and moments  $\vec{f}_P$  y  $\vec{\mu}_P$  are used as haptic feedback for the teleoperated robot and the force in the longitudinal axis of the tool  $\vec{f}_{P,z}$  is an input for the F/T control system. Together with the force command, an objective position called  $\{P_{ref,2}\}$  is calculated.



**Fig. 5.** Force–position control scheme

The position of the fulcrum point  $\{F\}$  is also used by the trajectory generator, which, together with the position set point, calculates a target position  $\{P_{ref,1}\}$ . Combining both objective positions, we obtain the final position to which we want to move the TCP of the robot, which takes into account both force and position restrictions and references  $\{P'\}$ .

Once the target position and orientation have been calculated, the trajectory is planned from the current position using OMPL (*Open Motion Planning Library*), which is the *MoveIt* default planner. The order is sent to the robot in the form of angular velocities, and the loop is closed by calculating the end position of the endoscope from the angular values of the axes, comparing it with the commanded position.

At any time the surgeon will be free to move the robot if he/she so wishes, making use of the *freedrive* function of the UR3. This function detects the efforts made in each of the axes and allows the individual movement of each of them. Even so, it is advisable to use this function only when you are not inside the nasal cavity, given the small size of the workspace and the limited precision that you have in this mode.

### 3.3 Low-Level Movement Primitives

Although the final order received by the robot is always in the form of the axes angular speeds, four low-level movement primitives have been developed that encompass all the possible movements of the robot. All movement primitives generate trajectories that prevent collisions between the robot's joints and between the robot's joints and its environment.

#### *goto\_joints*

Move the robot to an axes configuration received as an argument. Since a robotic arm with 6 degrees of freedom is being used, the angles in radians of each of the axes will be specified. It is the lowest level primitive, and it will be useful to reach known positions in a low computation time.

*goto\_joints*( $\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5, \vartheta_6$ )

#### *goto\_position*

Move the robot *Tool Center Point* (TCP) to XYZ coordinates received as a parameter. The orientation is not specified and so it has a random value. By default, the reference system associated with the robot TCP is  $\{P\}$ , and the coordinates can be referred to any of them,  $\{C\}$ ,  $\{P\}$ ,  $\{H\}$ ,  $\{B\}$  or  $\{O\}$ , received as a parameter. Having no orientation restrictions, it can be useful when it is not relevant or unknown.

*goto\_position*( $x_{SR}, y_{SR}, z_{SR}, SR$ )

#### *goto\_pose*

Move the TCP of the robot to XYZ coordinates (specified in millimeters) and orientation, received as a parameter. Orientation can be expressed in the form of a quaternion or through Euler RPY angles (*Roll Pitch Yaw*) in radians. The reference system to which both the position and the orientation are referred can be any, and will be a parameter. There will be, therefore, two possible calls to the primitive, depending on how the orientation is expressed.

*Quaternion:* *goto\_pose*( $x_{SR}, y_{SR}, z_{SR}, q_{x,SR}, q_{y,SR}, q_{z,SR}, q_{w,SR}, SR$ )

*Euler:* *goto\_pose*( $x_{SR}, y_{SR}, z_{SR}, \emptyset_{SR}, \vartheta_{SR}, \Psi_{SR}, SR$ ),

where  $\emptyset_{SR}$  is the rotation with respect to the  $x_{SR}$  (*Roll*) axis,  $\vartheta_{SR}$  is the rotation with respect to the  $y_{SR}$  (*Pitch*) axis y  $\Psi_{SR}$  is the rotation with respect to the  $z_{SR}$  (*Yaw*) axis.

This is the most used and exchanged between the different control blocks primitive, since it unequivocally determines the configuration of the robot.

#### *goto\_pose\_straight*

As with the *goto\_pose* primitive, a position and orientation is specified, with respect to any reference system, to which the movement of the TCP is planned. While using the *goto\_pose* primitive he path followed to get from the origin point to the destination point is indifferent and not controllable, using the *goto\_pose\_straight* primitive it will follow a straight line.

The notation of the parameters will, therefore, be the same:

*Quaternion:* *goto\_pose\_straight*( $x_{SR}, y_{SR}, z_{SR}, q_{x,SR}, q_{y,SR}, q_{z,SR}, q_{w,SR}, SR$ )

*Euler:* *goto\_pose\_straight*( $x_{SR}, y_{SR}, z_{SR}, \emptyset_{SR}, \vartheta_{SR}, \Psi_{SR}, SR$ )

This primitive will be used mainly in phases of the operation in which the endoscope is inside the nasal cavity and you want to perform forward and backward movements with respect to the reference system  $\{P\}$ .

In addition to these movement primitives, two functions have been developed that facilitate the relative positioning of the tool, as opposed to the absolute positioning used up to now.

*offset\_joints*

It allows obtaining the configuration of robot axes from angular increments expressed in radians. The position taken as origin would correspond to the current position of the robot at the time of calling the function.

$$\text{offset\_joints}(\Delta\vartheta_1, \Delta\vartheta_2, \Delta\vartheta_3, \Delta\vartheta_4, \Delta\vartheta_5, \Delta\vartheta_6)$$

*offset\_pose*

Returns the position and orientation of the robot's TCP ( $\{P\}$ ) with respect to an arbitrary reference system, displacing a position and orientation received as a parameter. The variation of the position will be determined by the XYZ coordinates, expressed in mm, and the orientation can be specified in the form of quaternion or using Euler angles, RPY, in radians.

$$\text{Quaternion: offset\_pose}(\Delta x_{SR}, \Delta y_{SR}, \Delta z_{SR}, \Delta q_{x,SR}, \Delta q_{y,SR}, \Delta q_{z,SR}, \Delta q_{w,SR}, SR)$$

$$\text{Euler: offset\_pose}(\Delta x_{SR}, \Delta y_{SR}, \Delta z_{SR}, \Delta\emptyset_{SR}, \Delta\vartheta_{SR}, \Delta\Psi_{SR}, SR)$$

## 4 Conclusions

The CRANEEAL initiative is intended to develop a co-worker robotic system for minimally invasive neurosurgery, combining different technologies under a single platform. The system will provide assistance to the neurosurgeon with automatic, collaborative or shared-control behaviors, and it will perform an accurate navigation based on the on-line information provided to measure the brain-shift phenomena and the pre-operative planification.

The robotic system will be based on a fault tolerance cognitive architecture, including a movement's control system to avoid damages in the nasal cavity, a collaborative movement's planner with learning capabilities and a multi-dimensional model used to predict the tridimensional displacements of the brain, based on intra-operative information.

In this work we have presented the methodologies developed to control the introduction of surgical tools by the nasal cavity and the realization of the basic movements associated with the intervention. During its execution, it is guaranteed that forces will not be exerted that will damage the aforesaid cavity either by an ill-defined movement or by external forces exerted accidentally on the robot. Likewise, a set of low level movement primitives has been defined, from which to approach the generation and control of trajectories and displacements in a homogenous manner, facilitating the integration of the robotic co-worker with the rest of the system elements.

The development of this project is, at present, a work in progress from which we hope to have an operating prototype soon that will allow us to begin testing under operating conditions.

**Acknowledgments.** This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under project DPI2016-80391-R.

## References

- Nogueira, J.F., Stamm, A., Vellutini, E.: Evolution of endoscopic skull base surgery, current concepts, and future perspectives. *Otolaryngol. Clin. North Am.* **43**(3), 639–652 (2010)
- Burgner, J., Rucker, D.C., Gilbert, H.B., Swaney, P.J., Russell, P.T., Weaver, K.D., et al.: A telerobotic system for transnasal surgery. *IEEE/ASME Trans. Mechatronics* **19**(3), 996–1006 (2014)
- Dubey, S.P., Munjal, V.R.: Endoscopic endonasal transsphenoidal hypophysectomy: two hand versus four hand technique: our experience. *Indian J. Otolaryngol. Head Neck Surg.* **66**(3), 287–290 (2014)
- Chalongwongse, S., Chumnanvej, S., Suthakorn, J.: Analyzation of Endonasal Endoscopic Transsphenoidal (EET) surgery pathway and workspace for path guiding robot design. *Asian J. Surg.* **42**, 814–822 (2019)
- Camarillo, D.B., Milne, C.F., Carlson, C.R., Zinn, M.R., Salisbury, J.K.: Mechanics modeling of tendon-driven continuum manipulators. *IEEE Trans. Robot.* **24**(6), 1262–1273 (2008)
- Piccigallo, M., Scarfogliero, U., Quaglia, C., Petroni, G., Valdastri, P., Menciassi, A., et al.: Design of a novel bimanual robotic system for single-port laparoscopy. *IEEE/ASME Trans. Mechatronics* **15**(6), 871–878 (2010)
- Escobar, P.F., Kebria, M., Falcone, T.: Evaluation of a novel single-port robotic platform in the cadaver model for the performance of various procedures in gynecologic oncology. *Gynecol. Oncol.* **120**(3), 380–384 (2011)
- Minjares-Granillo, R.O., Dimas, B.A., LeFave, J.P.J., Haas, E.M.: Robotic left-sided colorectal resection with natural orifice IntraCorporal anastomosis with extraction of specimen: The NICE procedure. *Am. J. Surg.* **217**(4), 670–676 (2019)
- Xia, T., Baird, C., Jallo, G., Hayes, K., Nakajima, N., Hata, N., et al.: An integrated system for planning, navigation and robotic assistance for skull base surgery. *Int. J. Med. Robot.* **4**(4), 321–330 (2008)
- Nimsky, C., Rachinger, J., Iro, H., Fahlbusch, R.: Adaptation of a hexapod-based robotic system for extended endoscope-assisted transsphenoidal skull base surgery. *min – Minim. Invasive Neurosurg.* **47**(1), 41–46 (2004)
- Eichhorn, K.W.G., Bootz, F.: Clinical requirements and possible applications of robot assisted endoscopy in skull base and sinus surgery. In: *Intraoperative Imaging*, pp. 237–240. Springer, Vienna (2011)
- Schneider, J.S., et al.: Robotic surgery for the sinuses and skull base: what are the possibilities and what are the obstacles? *Curr. Opin. Otolaryngol. Head Neck Surg.* **21**(1), 11–16 (2013)
- Chumnanvej, S., Suthakorn, J.: Workspace based design proposal for surgical robot with endoscopic endonasal transsphenoidal approach. In: *Proceedings of the CRIT 2012*, pp. 79–81 (2012)
- Shamir, R.R., et al.: Reduced risk trajectory planning in image-guided keyhole neurosurgery. *Med. Phys.* **39**(5), 2885–2895 (2012)
- Paraskevopoulos, D., Roth, J., Constantini, S.: Endoscope holders in cranial neurosurgery: part I—technology, trends, and implications. *World Neurosurg.* **89**, 343–354 (2016)
- Villaret, A.B., Doglietto, F., Carobbio, A., Schreiber, A., Panni, C., Piantoni, E., Guida, G., Fontanella, M.M., Nicolai, P., Cassinis, R.: Robotic transnasal endoscopic skull base surgery: systematic review of the literature and report of a novel prototype for a hybrid system (Brescia Endoscope Assistant Robotic Holder). *World Neurosurg.* **105**, 875–883 (2017)
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., et al.: ROS: an open-source Robot Operating System. In: *ICRA Workshop on Open Source Software*, vol. 3, no. 2, p. 5 (2009)



# Smart Companion Pillow – An EPS@ISEP 2019 Project

Alexandre Soares dos Reis<sup>1</sup>, Elien Gielen<sup>1</sup>, Ko Wopereis<sup>1</sup>, Marcel Pasternak<sup>1</sup>, Vaido Sooäär<sup>1</sup>, Tobias Schneider<sup>1</sup>, Abel J. Duarte<sup>1,4</sup>, Benedita Malheiro<sup>1,2(✉)</sup>, Jorge Justo<sup>1</sup>, Cristina Ribeiro<sup>1,3</sup>, Manuel F. Silva<sup>1,2</sup>, Paulo Ferreira<sup>1</sup>, and Pedro Guedes<sup>1</sup>

<sup>1</sup> ISEP/PPorto – School of Engineering, Polytechnic of Porto, Porto, Portugal  
epsatisep@gmail.com

<sup>2</sup> INESC TEC, Porto, Portugal

<sup>3</sup> INEB – Instituto de Engenharia Biomédica, Porto, Portugal

<sup>4</sup> REQUIMTE/LAQV, ISEP, Porto, Portugal

<http://www.eps2019-wiki4.dee.isep.ipp.pt/>

**Abstract.** This paper describes the design and development of a Smart Companion Pillow, named bGuard, designed by a multinational and multidisciplinary team enrolled in the European Project Semester (EPS) at Instituto Superior de Engenharia do Porto (ISEP) in the spring of 2019. Nowadays, parents spend most of the day at work and become naturally worried about the well-being of their young children, specially babies. The aim of bGuard is to provide a 24-hour remotely accessible baby monitoring service, contributing to reduce parenting stress. The team, based on the survey of related products, as well as on marketing, sustainability, ethics and deontology analyses, developed a remotely interactive Smart Companion Pillow to monitor the baby's health and room air quality. The collected data, once it is saved on an Internet of Things (IoT) platform, becomes remotely accessible. The bGuard pillow, thanks to its shape, reduces the risk of the baby rolling from back to tummy, lowering the risk of Sudden Infant Death Syndrome (SIDS).

**Keywords:** Baby well-being · Parent stress relief · Room air quality · Smart pillow · Sudden Infant Death Syndrome

## 1 Introduction

The bGuard smart companion pillow was an EPS@ISEP project [1] developed by a team of undergraduate students in the spring of 2019. The team was composed by students from different study fields and countries as follows: biomedical engineering from Germany; electrical engineering from Estonia; mechanical engineering from Germany and Portugal, industrial product engineering from Netherlands and product development from Belgium. This project provided the team with the opportunity to create an out-of-the-box object while contributing to infant and parent well-being.

Parents are always worried about the health of their children. According to a study looking into the parental worries of 2000 parents, conducted by OnePoll in conjunction with Lice Clinics of America [2], the average parents worry about their children around 37 h a week. In addition, according to the American Academy of Pediatrics, Sudden Infant Death Syndrome (SIDS) remains unsolved [3]. As referred by Kinney and Bradley [4] and the National Institute of Child Health and Human Development of the United States of America [5], the biggest risk factors of SIDS are the sleeping of the baby on the stomach or side, overheating, exposure to tobacco smoke and bed sharing. Therefore, the problem statement of the project focus on worried parents and SIDS.

Based on the problem statement, the objectives of this project are to lower the risk of SIDS and parent stress by monitoring the baby's sleeping environment. The next chapters report the teamwork performed to achieve these objectives.

This document includes a background section, where related projects, marketing, sustainability, ethics and deontology analyses are presented. Then, the design and development are explained. Afterwards, planned functional tests are described. Finally, the conclusions of the project are summarised.

## 2 Background

The background studies, which included a survey on related products, together with marketing, sustainability and ethics analyses, allowed the team to derive bGuard's requirements.

### 2.1 Related Products

The survey on domestic products addressing the monitoring of the room environment and vital signs, sleep and motion of babies contemplated:

- **Smart Pillows** are designed to improve the quality of the night rest. The more advanced pillows, like the iSense Sleep [6], contain sensors that monitor the pulse and respiratory rate. These pillows are often considered as gadgets because they include built-in speakers for streaming music or playing audio books, as is the case of ZEEQ [7] and Sunrise [8].
- **Sleep Trackers** are focused on giving detailed information about sleeping patterns. Smart pillows are often equipped with a sleep tracker. These devices come in different forms, such as wearable accessories (Fitbit Versa [9]), bed accessories (Withings Sleep [10]) or bed-side accessories (S+ [11]). They track the different sleep stages, the duration of sleep and overall sleep quality. The researched products also measure the pulse of the user.
- **Environment Monitors** gather information about their location. Although the market has a wide offer, this study selected three types of contact-less monitors: GLOCO, Sense Sleep, and Withings Aura. GLOCO [12] monitors temperature, humidity, and carbon monoxide levels of the child's room. It consists of a home station, which indicates when something is wrong, and an

app with additional information. The Sense Sleep [13] and Withings Aura [14] monitor the room conditions, the sleeping pattern and give advice on how to improve them.

- **Baby Monitors** are focused on the baby well-being. They monitor aspects such as temperature, movement and respiration of the baby. This is the case of Owlet [15], Nanit Plus [16] and Sproutling [17]. In addition, Sproutling alerts when the baby rolls over during sleep and Nanit Plus displays live images of the child.

Table 1 compares the products analysed, ordered by category.

**Table 1.** Product comparison

Category	Product	Media <sup>a</sup>	Pulse (bpm <sup>c</sup> )	Room			App
				RH <sup>b</sup> (%)	Temp. (°C)	CO <sup>2</sup> (ppm <sup>d</sup> )	
Smart Pillows	ZEEQ	✓					✓
	Sunrise	✓					✓
	iSense Sleep		✓				✓
Sleep Trackers	Withings Sleep		✓				✓
	S+		✓			✓	✓
	Fitbit Versa		✓				✓
Environment Monitors	Sense Sleep	✓		✓	✓		✓
	GLOCO			✓	✓		✓
	Withings Aura	✓	✓		✓		✓
Baby Monitors	Owlet		✓				✓
	Nanit Plus	✓		✓	✓		✓
	Sproutling			✓	✓		✓

<sup>a</sup> Image, Music, Sound or Video

<sup>b</sup> Relative Humidity

<sup>c</sup> beats/minute

<sup>d</sup> parts/million

According to this research, the market lacks devices which, simultaneously, reduce the risk of rolling from back to tummy, allow interaction, monitor the health of the baby, measure the room air quality and provide access to all this information through a mobile application. Taking into account these findings, the team embraced the concept of a four in one product for babies: (*i*) pillow with a special shape (to reduce the risk of rolling); (*ii*) microphone and speaker integrated in the pillow; (*iii*) pulse sensor incorporated in the sock; and (*iv*) room sensors (temperature, relative humidity (RH) and CO<sub>2</sub>) in the home station. To contribute to relief the stress of parents, the team chose to create a mobile application to display the collected data and allow remote interaction.

## 2.2 Marketing

The saying “Customers do not buy products, they buy benefits” was the motto of the team. Concerning the psychological segmentation, nowadays, parents worry considerably about the well-being of their children [2]. Consequently, bGuard

aims to lower the stress of parents by providing remote access to the baby's health and environment parameters. The benefits provided by bGuard comprise lower risk of SIDS, sleeping environment quality monitoring and parenting stress relief. Thus, the proposed promotional claim is "Sleepcurity for your baby".

bGuard was positioned with the help of the Porter model [18]. This model describes three different strategies a company can use from an outside-in point of view: cost leadership, differentiation and focus strategy. The target group is concerned parents (25 to 35 years), ranging from middle-class through upper-class, who were raised with technology and constantly rely on their smartphone. The team decided on differentiating the product, compared to its competitors, by creating a unique product combination. bGuard stands out from the crowd, inducing potential customers to pay more.

The four P marketing mix [19], which considers Product, Price, Place and Promotion, has been followed. First, bGuard is a pillow with a special shape to reduce the risk of rolling from back to tummy, with a speaker to sooth the baby when crying, a sock to measure the baby's pulse and a home station to monitor room air quality. The combination of all these elements makes it a unique Product in the market. Secondly, most of the comparable products vary between 113.00 € and 313.00 €. The estimated Price should be around 249.00 €. Being a smart product, the target group should be familiar with online shopping. Thus, in terms of Place, bGuard should only be sold through its website, saving money negotiating with large retail shops or re-sellers. Finally, advertisement on the Internet and social media will be considered for Promotion.

On the whole, bGuard will be promoted on the Internet and social media, sold on its own website and, due to its unique combination of features, the target group will be willing to pay 249.00 € for its benefits.

### 2.3 Sustainability

The team considered the 3 Pillars of Sustainability [20], the 12 Principles of Green Engineering [21] as well as the Good Health and Well-Being, Industry, Innovation and Infrastructure and Responsible Consumption and Production Sustainable Development Goals defined by the United Nations Educational, Scientific and Cultural Organization (UNESCO) [22]. Finally, the Life Cycle Analysis [23] was performed.

All bGuard parts were thought to fulfill these sets of principles. Firstly, the parts on the home station, the pillow and the sock are easily disassembled, allowing the maintenance of electronic components and part recycling. The main raw material of the home station – Acrylonitrile Butadiene Styrene (ABS) –, is recyclable up to 99% [24]. A further advantage of the home station, once the baby grows, is that it outlives the pillow and the sock since it will keep on monitoring the room air quality – temperature, RH and CO<sub>2</sub>. Secondly, the cover textiles of the pillow and sock are made in Lyocell (Tencel) [25,26] fabrics that are naturally biodegradable. During manufacturing, 99% of the water and non-toxic solvents are recyclable, resulting in an eco-friendly production. Thirdly, the foam of the pillow is certified with OEKO-TEX STANDARD 100 [27], class I, which

corresponds to the strictest baby requirements. This certified foam contributes to high and effective product safety.

Thus, bGuard will contribute to relieve the stress of the parents and improve the quality of life of the family, with minimum waste over its life cycle, without jeopardising the needs of future generations.

## 2.4 Ethics and Deontology

Ethical issues inevitably arise when developing a product. Engineering Ethics [28], Sales and Marketing Ethics [29], Environmental Ethics [30] and Liability [31–35] help to deliver a safe and environmentally friendly product, both for the users and the world. Objectively, this means the team wants to limit the commercialisation of the product to the European market in order to keep the footprint small, work with local providers and design according to the Design for Disassembly (DfD) [36] principles. Furthermore, bGuard wants to deliver a safe and user-friendly product. The aim is to ensure that users and manufacturers are guarded from dangers and liabilities, respectively. To help the user, bGuard is to be shipped with a manual detailing operation and maintenance instructions. Last but not least, the team envisages launching marketing campaigns to build a realistic image and promote this solution.

Above all, bGuard wants to be safe and live up to customer expectations.

## 3 Design and Development

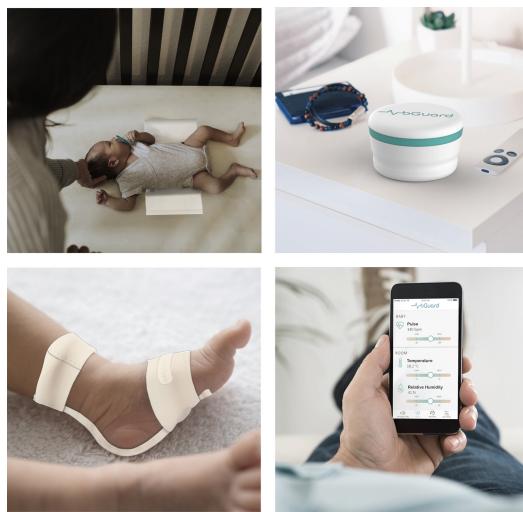
Based on the conducted background studies, the team defined the design concept and proceeded with the development of the bGuard proof-of-concept prototype.

### 3.1 Design

Figure 1 displays the bGuard design, which was driven by safety, sustainability and aesthetics concerns.

The pillow and the sock are covered with Lyocell (Tencel) [25, 26]. Lyocell is a breathable hygienic material, which means it is less prone to the growth of bacteria. The pillow has two side bumpers to reduce the risk of rolling from back to tummy. These side bumpers are made of polyurethane foam. The home station, to avoid sharp angles, has a round shape. In terms of sustainability, bGuard was designed according to the DfD [36] principles like, for instance, the manual screwing of the home station parts. This means that when the product is at the end of its life it can be easily disassembled and the parts recycled. Considering the aesthetics, the home station has a clean look, presenting a smooth and bright surface made of ABS.

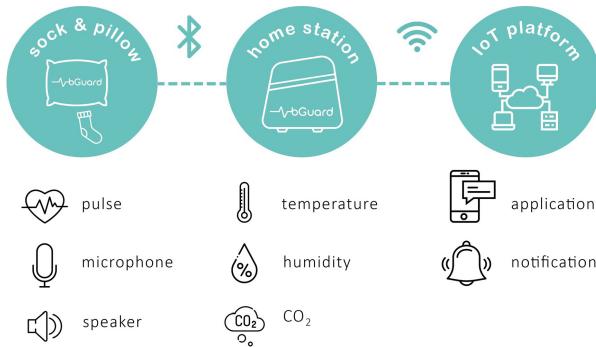
As a result, bGuard will be a safe, eco-friendly and attractive product.



**Fig. 1.** bGuard design

### 3.2 Concept

Figure 2 displays the bGuard concept, with the sock and the pillow on the left, the home station in the middle and the IoT platform and mobile application on the right.



**Fig. 2.** Concept of the prototype

The smart pillow has two side bumpers to minimise the risk of the baby rolling from back to tummy. One of the bumpers contains a micro-controller, a battery, a microphone, an MP3 player and a speaker. This equipment allows the pillow to play music or voice recordings when the baby cries. In the case of the end product, the sock integrates a pulse sensor, a small battery and a thin

micro-controller. In the case of the prototype, the sock pulse sensor is wired to the micro-controller of the pillow.

The home station measures the room temperature, RH and CO<sub>2</sub>, the sock reads the pulse of the baby and the pillow records the sound of the baby. The data collected by the sock and pillow are sent to the home station through a Bluetooth link and the home station sends all the gathered data to the IoT platform through a Wi-Fi link.

Finally, the baby and room data can be monitored online and seen on a smartphone. When the measured values are outside the predefined ranges, a notification is sent by the mobile app. Because of the limited time, the app will not be developed for the prototype. Nevertheless, the measurements will be monitored on the IoT Platform to prove the concept.

### 3.3 Development

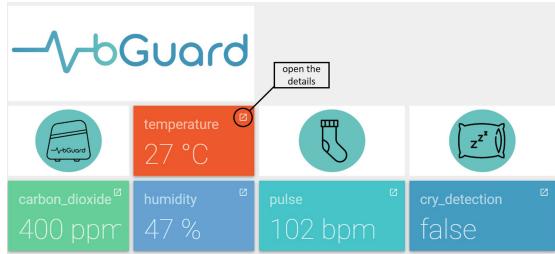
The development of bGuard, a four in one product for babies, was a great challenge for the team, not only in technological, but also in design terms. The home station has to allow ventilation, for the sensors to measure the air quality, the pulse sensor has to be encapsulated inside the sock and the pillow must be covered with a waterproof fabric with sound permeability to protect simultaneously from the potential spillage of liquids and to allow the microphone and the speaker to work properly. Figure 3 depicts the final proof-of-concept prototype, with the smart sock, the smart pillow and the 3D printed home station.



**Fig. 3.** Final proof-of-concept prototype

Once the connection with the IoT platform is established, it is possible to remotely access and monitor all sensor readings (Fig. 4).

All things considered, the design and development decisions, which took into account ethics, sustainability, marketing and the state of the art, resulted in the creation of a real scale proof-of-concept prototype.

**Fig. 4.** IoT dashboard

## 4 Tests and Results

Immediately after the specification of the prototype requirements, the team devised a set of functional tests, concerning the physical product, the software and the hardware, to verify the correct operation of the prototype. Table 2 lists these tests and the results.

**Table 2.** Tests and results

Test	Results
DHT22 Sensor Accuracy	The temperature and RH readings obtained with the DHT22 sensor for 20 °C, 26 °C, 28 °C, and 30 °C were compared with those of a reference device. The calculated and the expected accuracy matched ( $\pm 2.5\%$ for the temperature and $\pm 5\%$ for the RH)
CO <sub>2</sub> Sensor Calibration	The sensor was calibrated for a CO <sub>2</sub> concentration of 400 ppm [37]. The readings, however, were not compared with those of a reference device due to a lack of resources
Pulse Accuracy	The pulse measurements of a young adult matched the data sheet specifications. Next, the sensor accuracy was determined using the medical device Silvercrest SPO55. All readings were within $\pm 5\%$ accuracy
Microphone	The microphone was calibrated using the Sound Level Meter and Spectrum Analyser SC310 from CESVA. Since the baby cry spectrum ranges typically from 336.9 Hz to 502 Hz, the frequency stability was tested. The microphone displayed an acceptance range of 5%
Cry Detection	With a baby crying recording playing, the software analysed the microphone output signal and correctly identified the situation. As a result, the MP3 player played the stored sound recordings through the loud speakers
IoT Platform	The sensor readings submitted in real-time to the IoT Platform Thingsboard were available through the user dashboard. In addition, the user was able to visualise the historical data and preset the minimum and maximum notification values. Whenever the values were out of range, a notification was issued

## 5 Conclusion

Based on preliminary analyses, the team designed an out-of-the-box product, providing parents with remote access to information about the health and environment of their baby. Moreover, even when the infant naturally outgrows the

pillow and the sock, the home station will continue to monitor the air quality of the room. The bGuard Smart Companion Pillow is a product of the IoT era, where smart devices become part of daily lives. The target group is made of worried parents (25 to 35 years old), ranging from middle-class through upper-class, who were raised with technology and depend on their smartphone. Taking everything into account, the team is confident that bGuard has the potential to become a real product in the market due to its unique combination of features that lower the risk of SIDS, monitor the quality of the sleeping environment and reduce parenting stress.

EPS@ISEP, as a cross-cultural and multidisciplinary engineering capstone programme, challenged the team to execute an integrated design-implementation-and-business solution – bGuard. The collective effort of the team was the key for the success. Besides the different kinds of expertise and the distinct visions of the problem – worried parents and SIDS, the team was able, with the help of the supervisors and the teachers, to establish its own way to plan and navigate along the semester – in an agile way, using SCRUM.

Considering that EPS@ISEP aims to promote hard and soft engineering skills in undergraduates, here are the testimonials left by the team members at the end of the semester:

- “Living in a globalised world, the EPS gives students the possibility to spend a semester in a foreign country while working with people from all over the world in close cooperation. Putting together a team from different academical backgrounds, different countries and different strengths, has the potential to create something big. The project allowed me to learn many things with the others and to improve my skills just by recognising how they tackle things and find solutions. What I liked especially about the EPS is that it simulates a start-up company in quite a realistic way. All the know-how was transferred in a practical way to be directly used for the project. It is great to see how motivated colleagues can be and how much effort they put inside when they have a great project which is really progressing.”
- “EPS@ISEP was really a great experience for me, specially because of the different fields of expertise of the team members. Furthermore, it was very interesting how the team always managed the time in order to distribute tasks towards the success of the project. On the whole, I must say that EPS@ISEP was really above my best expectations.”
- “Choosing to participate in the European Project Semester was one of the best choices of my life. I learned a lot during these few months. Not only did I develop new skills as a Product Designer, but I also learned so much about teamwork. It was not always easy to work with people who do not speak the same language as you or study the same courses, but we always managed to communicate one way or another. I would definitely say that our different backgrounds were not a weakness but a strength, which made us stronger as a team.”

- “EPS is a good way to interconnect with students from different countries and cultures as well as the way to create a complete advised product for the real market is a great challenge.”
- “I am grateful for the experience EPS has provided. It is such a nice opportunity to meet so different people from other countries, see how they work and get to know other areas than my speciality. EPS is a good way to learn how to make things work in a totally new environment and in a group where each person has different ideas and viewpoints. It has taught me to be patient, tolerant and to consider much more the ideas of others. The opportunity to see and learn about this beautiful country is also priceless.”
- “I am thankful for having the opportunity to take part on the European Project Semester. During the semester I learned a lot about the different cultures and their way of working. It was great to see that every part of the team was much focused to create a product that supports young parents in the daily life. Also, I liked the way of thinking about ethical and sustainable topics during the project. This experience gave me a new view of generating solutions during the project. Moreover, the EPS enabled me to improve my English skills.”

**Acknowledgement.** The authors thank their home institution, Instituto Superior de Engenharia do Porto, as well as the faculty and staff involved in EPS@ISEP, for the support and contribution to the success of the programme.

**Funding.** This work was partially financed by National Funds through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT), within project UID/EEA/50014/2019.

## References

1. Malheiro, B., Silva, M., Ribeiro, M.C., Guedes, P., Ferreira, P.: The European project semester at ISEP: the challenge of educating global engineers. *Eur. J. Eng. Educ.* **40**(3), 328–346 (2015). <https://doi.org/10.1080/03043797.2014.960509>
2. Gervis, Z.: Parents spend an insane amount of their lives worrying about their kids. <https://nypost.com/2018/09/10/parents-spend-an-insane-amount-of-their-lives-worrying-about-their-kids/>
3. Task Force on Sudden Infant Death Syndrome: SIDS and Other Sleep-Related Infant Deaths: Updated 2016 Recommendations for a Safe Infant Sleeping Environment. *Pediatrics* **138**(5) (2016). <https://doi.org/10.1542/peds.2016-2938>
4. Kinney, H.C., Thach, B.T.: The sudden Infant death syndrome. *N. Engl. J. Med.* **361**(8), 795–805 (2009). <https://doi.org/10.1056/NEJMra0803836>
5. Office of Communications: What causes SIDS? (2017). <https://www.nichd.nih.gov/health/topics/sids/conditioninfo/causes>. Accessed March 2019
6. iSenseSleep: iSense Smart Pillow (2019). [https://isensesleep.com/products/smart-pillow?sscid=31k3\\_20f12](https://isensesleep.com/products/smart-pillow?sscid=31k3_20f12). Accessed March 2019
7. ZEEQ Smart Pillow: ZEEQ Smart Pillow (2019). <https://sleeptrackers.io/zeeq-smart-pillow/>. Accessed March 2019

8. Sunrise Smart Pillow: Sunrise Smart Pillow: The Future of Sleep & Wake Technology (2019). <https://www.kickstarter.com/projects/modem/the-sunrise-smart-pillow-sleep-smart-wake-naturall>. Accessed March 2019
9. FitBit: FitBit Versa (2019). <https://www.fitbit.com/shop/versa>. Accessed March 2019
10. Withings Sleep: Withings Sleep (2019). <https://www.withings.com/mx/en/sleep>. Accessed March 2019
11. ResMed: S+ (2019). <https://splus.resmed.com/>. Accessed March 2019
12. First Alert: First Alert Onelink GLOCO Wi-Fi Environment Monitor (2019). <https://www.firstalert.com/product/Wi-Fi-environment-monitor-apple-homekit-enabled/>. Accessed March 2019
13. Hello: Sense Sleep System - Cotton (2019). <https://www.amazon.com/Sense-Sleep-System-Previous-Generation/dp/B016XBL2RE>. Accessed March 2019
14. Withings Sleep: Withings Aura Smart Sleep System (2019). <https://sleeptrackers.io/withings-aura-review/>. Accessed March 2019
15. Owlet Baby Care: Owlet Baby Monitor (2019). <https://owletbabycare.co.uk/>. Accessed March 2019
16. Nanit: Nanit plus camera (2019). <https://store.nanit.com/products/nanit-plus-camera-wall-mount>. Accessed March 2019
17. Sproutling: Sproutling Wearable Baby Monitor (2019). <https://www.digitaltrends.com/mobile/sproutling-baby-wearable-tracks-sleep/>. Accessed March 2019
18. MindTools: Porter's Generic Strategies. [https://www.mindtools.com/pages/article/newSTR\\_82.htm](https://www.mindtools.com/pages/article/newSTR_82.htm)
19. OnDemandCMO: Do You know Your Marketing Mix? (2019). <https://www.ondemandcmo.com/blog/know-marketing-mix/>. Accessed March 2019
20. Purvis, B., Mao, Y., Robinson, D.: Three pillars of sustainability: in search of conceptual origins. *Sustain. Sci.* **14**(3), 681–695 (2019). <https://doi.org/10.1007/s11625-018-0627-5>
21. American Chemical Society: 12 Design Principles of Green Engineering (2019). <https://www.acs.org/content/acs/en/greenchemistry/principles/12-design-principles-of-green-engineering.html>. Accessed March 2019
22. UNESCO: UNESCO and Sustainable Development Goals, September 2015. <https://en.unesco.org/sdgs>. Accessed March 2019
23. avnIR: avnIR Lyfe Cycle Thinking (2019). <http://www.avnir.org/EN/>. Accessed March 2019
24. Plastics Insight: ABS Plastic (ABS): Production, Market, Price and its Properties (2019). <https://www.plasticsinsight.com/resin-intelligence/resin-prices/abs-plastic/>. Accessed March 2019
25. Sleeping Lucid: Tencel (lyocell) vs bamboo sheets (2019). <https://sleepinglucid.com/tencel-sheets-vs-bamboo/>. Accessed March 2019
26. Good On You: Material guide: How ethical is tencel? (2019). <https://goodonyou.eco/how-ethical-is-tencel/>. Accessed March 2019
27. OEKO-TEX: Standard 100 for oeko-tex. [https://www.oeko-tex.com/en/business/certifications\\_and\\_services/ots\\_100/ots\\_100\\_start.xhtml](https://www.oeko-tex.com/en/business/certifications_and_services/ots_100/ots_100_start.xhtml). Accessed May 2019
28. National Society of Professional Engineers: NSPE Code of Ethics for Engineers. <https://www.iep.utm.edu/envi-eth/>. Accessed April 2019
29. Sales and Marketing Executives International: Sales & marketing creed: The international code of ethics for sales and marketing. <https://www.smei.org/page/16/Marketing-Code-of-Ethics.htm>. Accessed April 2019
30. Science Direct: Environmental ethics. <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/environmental-ethics>. Accessed April 2019

31. EUR-Lex: Document 32006L0042. <https://eur-lex.europa.eu/legal-content/GA/TXT/?uri=CELEX:32006L0042>. Accessed April 2019
32. European Commission: The Low Voltage Directive (LVD). [http://ec.europa.eu/growth/sectors/electrical-engineering/lvd-directive\\_en](http://ec.europa.eu/growth/sectors/electrical-engineering/lvd-directive_en). Accessed April 2019
33. European Commission: Radio Equipment Directive (RED). [https://ec.europa.eu/growth/sectors/electrical-engineering/red-directive\\_en](https://ec.europa.eu/growth/sectors/electrical-engineering/red-directive_en). Accessed April 2019
34. European Commission: Electromagnetic Compatibility (EMC) Directive. [https://ec.europa.eu/growth/sectors/electrical-engineering/emc-directive\\_en](https://ec.europa.eu/growth/sectors/electrical-engineering/emc-directive_en). Accessed April 2019
35. European Commission: Restriction of Hazardous Substances (RoHS) in Electrical and Electronic Equipment Directive. [http://ec.europa.eu/environment/waste/rohs\\_eee/index\\_en.htm](http://ec.europa.eu/environment/waste/rohs_eee/index_en.htm). Accessed April 2019
36. Soh, S.L., Ong, S.K., Nee, A.Y.C.: Application of design for disassembly from remanufacturing perspective. Procedia CIRP **26**, 577–582 (2015). <https://doi.org/10.1016/j.procir.2014.07.028>
37. AirTest: A White Paper: CO<sub>2</sub> Ventilation Control and Measurementf of Outside Air. <https://www.airtesttechnologies.com/support/datasheet/CO2MeasurementAndOutsideAir.pdf>. Accessed June 2019



# Advantages of the Incorporation of an Active Upper-Limb Exoskeleton in Industrial Tasks

Andrea Blanco<sup>1</sup>✉, José M. Catalán<sup>1</sup>, Jorge A. Díez<sup>1</sup>, José V. García<sup>1</sup>, Luis D. Lledó<sup>1</sup>, Emilio Lobato<sup>2</sup>, and Nicolás M. García-Aracil<sup>1</sup>

<sup>1</sup> Department of Systems Engineering and Automation,  
Miguel Hernández University, Elche, Spain  
[a.blanco@umh.es](mailto:a.blanco@umh.es)

<sup>2</sup> MovilFrio S.L., Alicante, Spain

**Abstract.** The main objective of the study collected in this paper is to study the changes in the execution of a repetitive task of industrial field when it is carried out with the help of an active upper-limb exoskeleton. To do this, an experimentation has been carried out in which the aim is to compare the differences when performing a task with and without the exoskeleton. Results show that the incorporation of the active upper-limb exoskeleton supposes an advantage in the execution of the task, in terms of precision and control against disturbances.

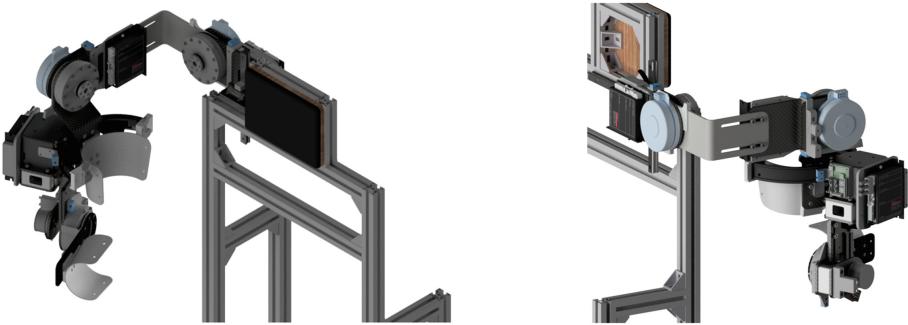
**Keywords:** Upper-limb exoskeleton · Validation · Industrial tasks · Assistive devices

## 1 Introduction

Musculoskeletal disorders of occupational origin affect body structures such as muscles, ligaments, joints, nerves, tendons and bones, and are mainly caused by the working conditions and environment like loads, posture, duration of the task, recurring movements among others [1]. This kind of disorders are among the most important health problems at work, and they have a high impact in the quality of life of many workers as well as suppose an important annual cost. In particular, in the European Union, musculoskeletal injuries are a significant and growing health problem, since more than 40 million workers in Europe are affected, representing an estimated cost to society of between 0.5 and 2% of the gross domestic product (GDP) [2].

Fully automation of the tasks or using robotic manipulators would help to reduce the risk of musculoskeletal injuries. However, a great number of jobs would require an unaffordable investment to be fully automated due to the complexity of the task or the environment. In this regard, workers assisted by wearable robots, specially exoskeletons, are a compromise solution that combine both human skills to deal with complex environments with the load-carrying and sensing capabilities of the robotic devices [3].

The ExIF project (Intelligent Robotic Exoskeleton and Advanced Interface Systems Man Machine for maintenance tasks in the Industries of the Future) arises from the necessity to eliminate or reduce the musculoskeletal injuries associated with the installation and maintenance of industrial facilities, proposing as a solution the development of an active upper-limb exoskeleton that will be supported by a passive lower-limb exoskeleton structure that transmits the weight of the robotic system to the ground, described in [4] and shown in Fig. 1.



**Fig. 1.** Active upper-limb exoskeleton design.

The study presented below aims to analyze changes in the execution of a repetitive task of industrial field induced by the incorporation of an active upper-limb exoskeleton, comparing the results obtained with those that appear when executing the same repetitive task in free mode (without the help of the arm exoskeleton).

## 2 Materials and Methods

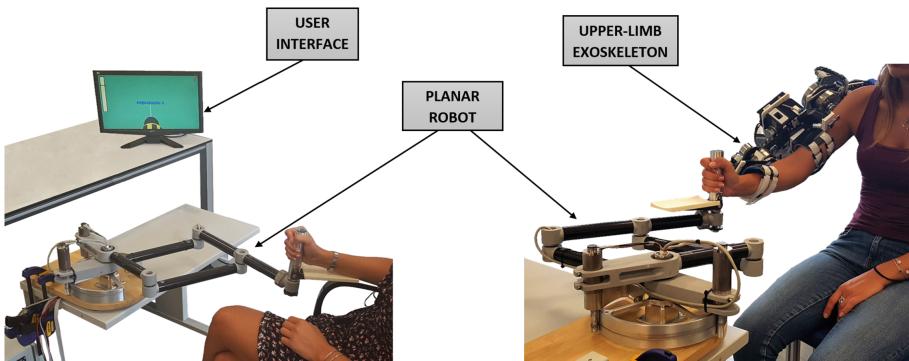
### 2.1 Subjects

10 subjects have participated in this experimentation, 5 male and 5 female with no motor or cognitive impairment. All the subjects are right handed, with ages between 24 and 48 years ( $29.8 \text{ years} \pm 6.795 \text{ years}$ ), with heights ranging between 1.60 and 1.83 m ( $1.69 \text{ m} \pm 0.067 \text{ m}$ ), and a weight between 55 and 86 kg ( $65.9 \text{ kg} \pm 12.087 \text{ kg}$ ).

### 2.2 Experimental Setup

In this study two robots have been used, the upper-limb exoskeleton mentioned above and described in [4], and a planar robotic device for post-stroke home tele-rehabilitation [5] that has already been validated in other works [6]. The experimental setup is shown in Fig. 2.

The subjects sit in front of a screen that shows the instructions to carry out the activity. Depending on the condition they are performing, subjects will take the final-effector of the robotic rehabilitation device, wearing or not the arm exoskeleton.



**Fig. 2.** Experimental setup, with and without the upper-limb exoskeleton.

### 2.3 Task

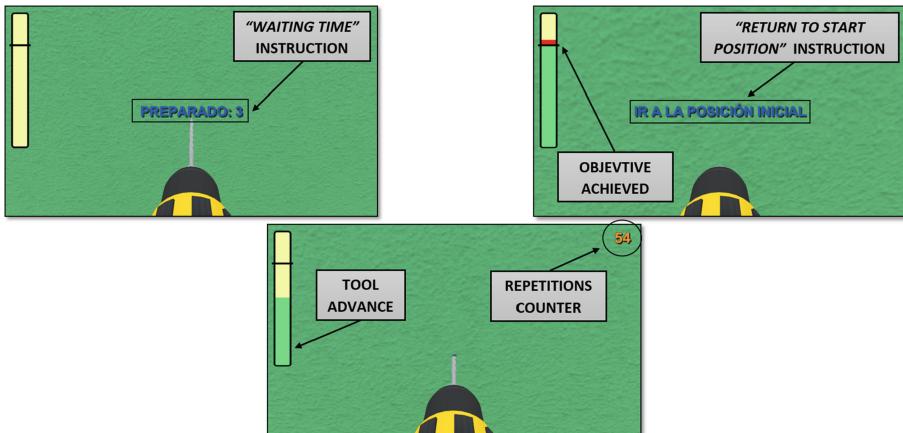
The task consists in the simulation of drilling a wall in which it must reach a depth of 10 cm. To simulate the different scenarios, the following assumptions have been established:

- Continuous wall
- Thick wall + air
- Thin wall + air

Each of these assumptions is carried out with different levels of force, a low level (8 N) and a high level (12 N), thus obtaining 6 different classes. Each of these classes is executed 10 times in random order, so each subject will perform 60 repetitions for each condition (60 repetitions with exoskeleton and 60 freely).

The planar robot performs the force on its axis  $Y$  toward the subject simulating the wall which opposes the advance of the tool, while the position of the end-effector represents the drill to be moved from the home position to achieve the objective. To simulate the wall has been crossed, the robotic system stops applying the force, and the subject should be able to control the end-effector for reaching the target not exceeding 10 cm.

The interface shown on the screen (Fig. 3) indicates to the subject at all times the actions to be carried out during the experimentation: waiting time, executing movement, objective achieved and returning to the starting position.



**Fig. 3.** User interface.

## 2.4 Acquired Data

All the data analyzed in the study are obtained from the planar robot, and were collected during the sessions. Since the exoskeleton performs the activity by rigid position control, it is not necessary to collect any data from this device.

At the beginning of each session, the planar robot sends the sequence of the classes that is carried out during the 60 repetitions, in order to identify which class is going to be given and then be able to compare the results obtained by the subjects in each of them.

On the other hand, the device sends the data associated to the distance traveled by the end-effector as a function of time, which allows us to know when the subject reaches the objective (speed when executing the task), maximum distance traveled (subject precision) and trajectories performed (repetitiveness when executing the task).

## 2.5 Study Protocol

Once we have explained to the subject what the experimentation consists of, and verified that the communication with the devices works correctly, we proceed to begin the experimentation. The purpose of the subject, as detailed above, is to achieve an objective by simulating a drill on a wall, in which the bit must be inserted a depth of 10 cm.

The task is carried out by means of two conditions that are executed in a random way: in one condition the activity is performed freely, while in the other one the exoskeleton of the upper-limb executes the movements of the subject's arm. For each of these conditions, the subject performs 60 repetitions, making a total of 120 trajectories.

## 2.6 Statistical Analysis

Taking into account the 10 trials of every class, the mean value of the position deviation with respect to the objective has been extract for every subject. The average speed of the speed profile has also been obtained for each of the classes. Therefore, each user has a characteristic mean value of each parameter for every class.

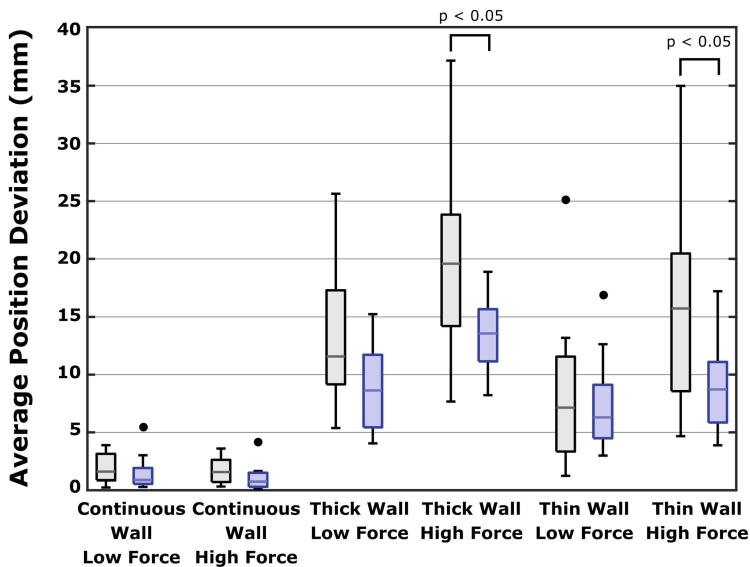
Two-sample t-test was apply to compare whether the difference between the same class in both conditions are really significant.

## 3 Results

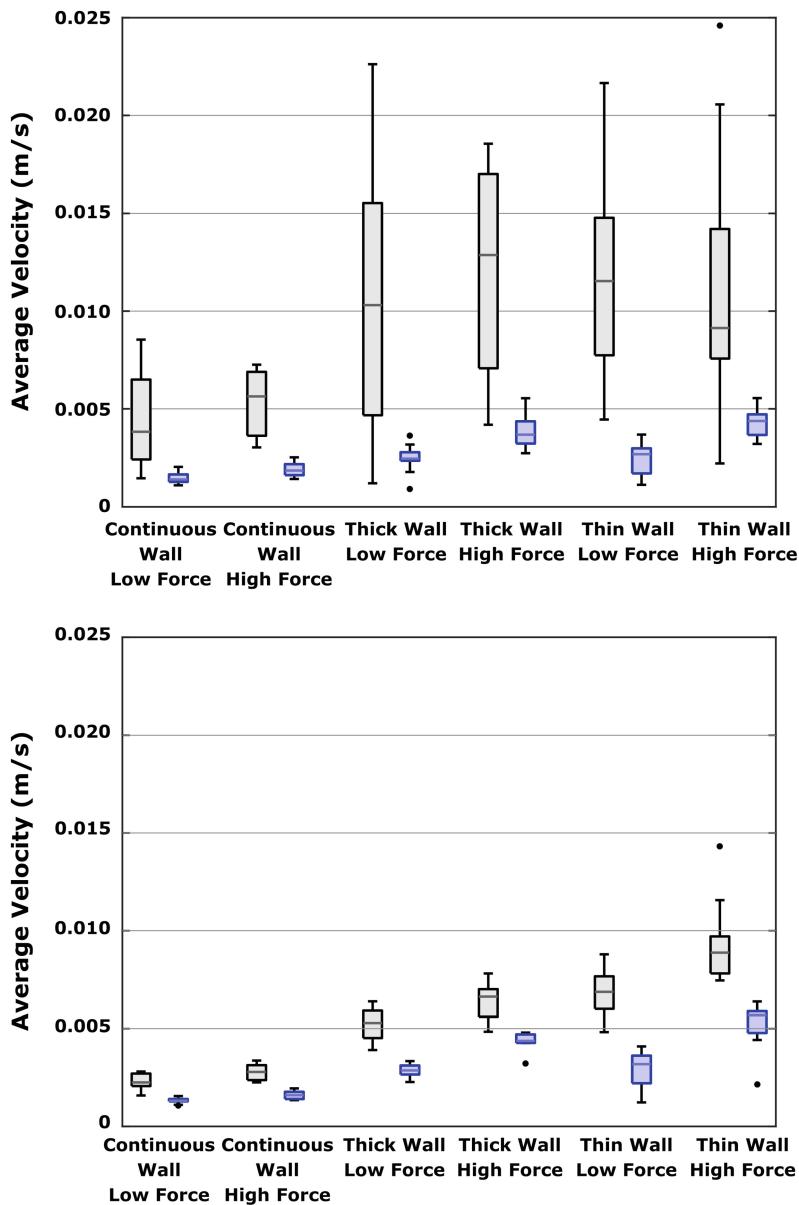
### 3.1 Average Position Deviation

In Fig. 4, the average of the position deviations with respect to the objective position by all users for each of the classes is shown. This parameters shows the influence of the perturbation when users try to reach the objective accurately.

Analysis shows that there is a significant difference between both conditions in case of the Thick Wall High Force (Two-sample t-test  $p = 0.0365$ ) and Thin Wall High Force (Two-sample t-test  $p = 0.0379$ ).



**Fig. 4.** Boxplot of the mean values of the position deviations with respect to the objective position for each of the classes, where the freely condition is represented in gray and the exoskeleton one in blue.



**Fig. 5.** Boxplot of the mean speed profile of two subjects for all the classes, where the freely condition is represented in gray and the exoskeleton one in blue. The results corresponds to the subject with the worst performance (image above) and to the subject with the best performance (image below).

### 3.2 Average Speed

Figure 5, shows the mean values of the speed profile of each trial in case of two subjects, the one with the best performance and the one with the worst, considering a better performance when the dispersion in freely condition is more similar to the exoskeleton one.

## 4 Discussion

### 4.1 Accuracy in the Task Execution

Analyzing the data obtained, it can be seen that the average of the deviations with respect to the position reached is higher when the task is carried out freely, which results in greater accuracy in the task execution when it is performed with the help of the active upper-limb exoskeleton.

This deviation, as shown in Fig. 4, is significant in the classes where the applied force is higher (12 N), so we can affirm that the fact of executing the task with the help of the active upper-limb exoskeleton designed implies an advantage in terms of accuracy achieved by the user over performing the same activity without assistance.

### 4.2 Response After Unexpected External Perturbation

Taking a look to the Fig. 5, for both subjects (the ones that has better and worse performance), the standard deviation of the mean values of the speed profile of all the classes, is much higher in freely condition than in the exoskeleton one, reaching approximately 3 times greater in certain classes of the user with better performance, up to a maximum of approximately 15 times greater in some classes of the user with worse performance.

Another aspect that can be observed is that regardless of the user's performance in the freely condition, in general, users reach almost a identically velocity profile respectively for each classes of the exoskeleton condition.

These effects reflects how the exoskeleton is able to mitigate the unwanted effects caused by disturbances during the task. All this favors not only to have a better accuracy, as seen in the previous section, but also to avoid injuries due to unexpected sudden movements, to avoid making failures due to these disturbances or to have a better control due to the most stable and continuous movement.

## 5 Conclusion

In the present work, the differences of using an active upper-limb exoskeleton with respect not to using it to perform a certain simulated task of industrial field has been evaluated, in this case, to drill a wall horizontally.

Analyzing the results obtained in the experimentation, we can affirm that, in general, the incorporation of the active upper-limb exoskeleton supposes an

advantage in the execution of the studied task, and in particular, it is observed that it is significantly beneficial to obtain a greater precision and more control among the disturbances that may appear.

**Acknowledgments.** This work has been supported by the AURORA project through the grant DPI2015-70415-C2-R of the Ministerio de Economía y Competitividad of Spain, by Centre for the Development of Industrial Technology (CDTI) and by Conselleria d'Educació, Cultura i Esport de la Generalitat Valenciana and by the European Social Fund “Investing in your future”, through the scholarships ACIF 2018/214, ACIF 2016/216 and APOTIP 2017/001.

## References

1. Núñez, G., Mevic, M., García Martín, M.C., Sánchez Lemus, G.: Factores de riesgo laboral para tenosinovitis del miembro superior. *Medicina y Seguridad del Trabajo* **61**(241), 486–503 (2015)
2. Roquelaure, Y., LeManach, A.P., Ha, C., Poisnel, C., Bodin, J., Descatha, A., Imbernon, E.: Working in temporary employment and exposure to musculoskeletal constraints. *Occup. Med.* **62**(7), 514–518 (2012)
3. De Looze, M.P., Bosch, T., Krause, F., Stadler, K.S., O’Sullivan, L.W.: Exoskeletons for industrial application and their potential effects on physical work load. *Ergonomics* **59**(5), 671–681 (2016)
4. Blanco, A., Díez, J.A., López, D., García, J.V., Catalán, J.M., García-Aracil, N.: Human-centered design of an upper-limb exoskeleton for tedious maintenance tasks. In: International Symposium on Wearable Robotics, pp. 515–519. Springer (2018)
5. Díaz, I., Catalán, J.M., Badesa, F.J., Justo, X., Lledo, L.D., Ugartemendia, A., Gil, J.J., Díez, J., García-Aracil, N.: Development of a robotic device for post-stroke home tele-rehabilitation. *Adv. Mech. Eng.* **10**(1), 1687814017752,302 (2018)
6. Catalán, J., García, J., López, D., Ugartemendia, A., Díaz, I., Lledo, L., Blanco, A., Barrios, J., Bertomeu, A., García-Aracil, N.: Evaluation of an upper-limb rehabilitation robotic device for home use from patient perspective. In: International Conference on NeuroRehabilitation, pp. 449–453. Springer (2018)



# Assistance Strategies for Robotized Laparoscopy

Alicia Casals<sup>(✉)</sup>, Albert Hernansanz, Narcís Sayols, and Josep Amat

Universitat Politècnica de Catalunya, BarcelonaTech, Jordi Girona, 1-3,  
08034 Barcelona, Spain  
alicia.casals@upc.edu

**Abstract.** Robotizing laparoscopic surgery not only allows achieving better accuracy to operate when a scale factor is applied between master and slave or thanks to the use of tools with 3 DoF, which cannot be used in conventional manual surgery, but also due to additional informatic support. Relying on computer assistance different strategies that facilitate the task of the surgeon can be incorporated, either in the form of autonomous navigation or cooperative guidance, providing sensory or visual feedback, or introducing certain limitations of movements. This paper describes different ways of assistance aimed at improving the work capacity of the surgeon and achieving more safety for the patient, and the results obtained with the prototype developed at UPC.

**Keywords:** Surgical robotics · Robot laparoscopic surgery · Cooperative robotics · Virtual feedback · Safety in robotics surgery

## 1 Introduction

The introduction of robotics in laparoscopic surgery started in the early 2000s, initially with systems equipped with three robotic arms, two for operation and the third for the support and guidance of the laparoscope. Subsequently, a fourth arm was introduced so that the surgeon could manipulate an additional instrument for holding other auxiliary tasks without the need of an external cooperation. Since the introduction of robotic laparoscopic surgery, with the Da Vinci system being so far the only robotic system with certification to be marketed since 2000, Fig. 1, about 5,000 systems have been installed, of which about 50 in Spain. In the coming years this figure will increase significantly, especially with the appearance of new systems, such as the CMR Versus now recently certified [1], Fig. 2. It is also expected that in the coming years the offer diversifies with the appearance of other systems that can become more competitive. The system developed at UPC [2], Fig. 3, is expected to be one of them in the time frame of two years, the minimum necessary to fulfil all legal requirements.

The systems currently available have been incorporating new features, basically aimed at three objectives: facilitating the work of the surgeon, increasing the patient's safety, and making this type of surgery less and less invasive. In the first aspect, in order to provide a better assistance to the surgeon, new instruments have been developed. To put an example, by using staples the number of sutures to be performed can be reduced, with the advantage this entails in terms of required ability and

operation time. On the other hand, assistance systems have been developed to streamline the set-up, a process that still takes too long. This aspect is relevant, since the introduction of robots has been based on the high levels of surgical efficacy achieved, more than on efficiency, due to the high cost of current systems and the longer operation time still required in front of conventional surgery.

In the second aspect, the increase of patient's safety, visualization systems are developed, which provide the surgeon with a better vision of the position of each instrument or with means of fixing protection boundaries, virtual fixtures. These systems tend to compensate the lack of tactile feedback in teleoperated systems, since despite being a demand of the surgeons, it has not been achieved due to the impossibility of developing sensory systems integrable in the instruments that measure interaction efforts and are compatible with the required miniaturization and the needed robustness in front of sterilization processes. In the third aspect, to minimize trauma to the patient, suitable articulated instruments have been developed to be able to operate through a single hole, which has allowed to develop both, tools oriented to the manual SILS (Single-Incision Laparoscopic Surgery), and robotic systems oriented to this type of surgery, as the TSpot of Titan Medical, Fig. 4.

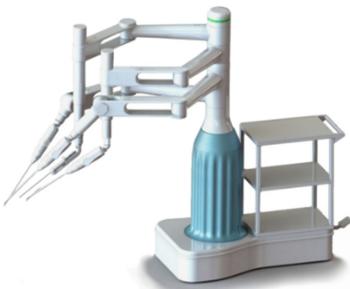
There is therefore a line of progress that aims to look for improvements based on the design of architectures, or new mechanical devices, such as new stapler tweezers or specific tools to operate on a single port, Fig. 4, and another that with the available architectures and systems, enable the introduction of better control strategies or supervisory algorithms that allow improving the medical practice or patient safety. And this is the focus of this work, surgical improvements using the resources provided by multisensory perception, learning techniques and artificial intelligence or big data.



**Fig. 1.** *Da Vinci* system



**Fig. 2.** *Versius* system



**Fig. 3.** The *Bitrack* system (UPC)



**Fig. 4.** *TSpot* system for *SILS* surgery

## 2 Robot Based Assistance Systems for Laparoscopic Surgery

In robotic laparoscopic surgery the typology of soft tissues, the object of the intervention, does not allow a preprogramming of the tasks to be performed. Robotization here is implemented by means of teleoperation, which with the classical control structure can provide advantages over manual execution, despite not having a capacity for sensory feedback. Apart from the assistance functions already developed, the fact of robotizing allows introducing clear advantages since the trajectories of each arm can be provided with correction factors, either a gain factor or keeping the orientation of the working axes during a movement, which allows to increase precision and the work comfort.

The lack of force feedback does not prevent providing the operator with an alternative visual information, for example on the status of the instruments when they are not in the visual field or when they reach extreme positions within their working space.

### 2.1 Aids Based on the Geometric Control of Trajectories

Due to the large computing capabilities available in teleoperation control systems, many aids based on geometric calculations can be introduced, surpassing the capacity of a human operator, and thus facilitate teleoperation. The ability to memorize trajectories allows automatic and secure removing of an instrument and its automatic return to the point of origin. Considering this point as safe, which is reasonable if the removal has been carried out without incompatibilities with the environment, and if there have been no other changes in position in the surgical field, it is unlikely that the inserting and accessing movements change. The automatic execution of the removal and re-entry movements to a specific working point is of especial interest to speed up the operation of cleaning the camera's objective when there is drowsiness that makes vision difficult, or to release an instrument of tissues that may have remained adhered during the intervention.

To guarantee a safe automatic realization of trajectories, such as the return to a certain point previously visited, and in order to anticipate the presence of unforeseen obstacles, some assisted control is necessary. That is, the generation of trajectories that are modifiable referring to both, execution speed and position, in which the position in each point of the trajectory is the vector sum of the planned or memorized trajectory, and the gesture control orders made by the operator, either directly on the arm of the robot, or on the controls, Fig. 5.

The geometric computing capacity provided by the control unit is also applicable to make the geometric measurements that help certain operations, such as the sizing of meshes or other tools, and to be able to automatically guide the camera, focusing the field of view on the tools working area, which position is known since these tools are governed by the own control unit.

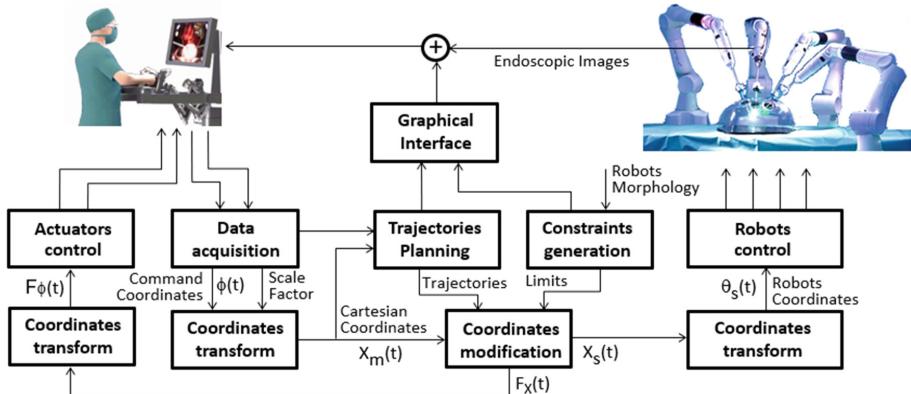


Fig. 5. Control structure with cooperative telecontrol

## 2.2 Aids Based on Haptics

All teleoperated robotic systems in laparoscopic surgery are operated from a workstation with motorized controls, both to be able to do the initial positioning and to guide the operator in order to maintain the orientation of the instrument along a change of position, as well as to perceive limit situations. The perception of efforts in manual laparoscopic surgery, even being affected by the friction of the tool with the trocar, which appreciably distorts their values, is lost in robotized surgery due to the difficulties involved in the incorporation of force sensors in the different working tools. Thus, at present, the robotic systems of application in laparoscopic surgery do not have yet haptic capacities, although they are common in other teleoperation fields.

Direct force/torque measurement, F/T, is achieved placing a F/T sensor on the laparoscopic tools to detect the interaction forces between tool and tissues. The placement of the F/T sensor is crucial to obtain the best measurements. Two are the feasible solutions: interpose the sensor between the tool actuator and the shaft or between tool and robot. The first solution offers optimal results, but its implementation passes through complex mechanical solutions. The most successful approach was developed by DLR, [3], where a F/T sensor based on a Stewart platform was placed to obtain direct measures. More recently, in [4] a grasping force sensor placed on the wrist of a robotic laparoscopic grasper was proposed. The sensor is based on a double layered capacitance sensor attached to a deformable wrist and a torque sensor on the pulley. This approach requires high precision and miniaturization, resulting into high cost solutions with sterilization and electromagnetic compatibility problems that are still pending to be solved.

Placing the F/T sensor outside the patient relaxes the restrictions of sensor size enabling the use of industrial force sensors. This solution suffers from measurement distortions induced by the trocar and tool flexions [5]. The contribution of these two factors is difficult to be modelled. In [6] a force sensed robotic trocar is presented where the force sensor is out of the patient. The force sensor is placed between a double concentric access channel. Laparoscopic tool interacts with the internal channel while

the external one is passively guided. This difference is measured as the interaction force between the tool and the surgical environment.

Unlike direct sensing, indirect F/T is estimated from sensors that directly measure deformations using other parameters. Most of the proposed solutions are based on measuring the intensity of the current of the motors that control the end effector driving cables, [7]. Following a similar principle, it is possible to estimate F/T measuring the tension of the driving cables. In [8], authors guide the grasper actuators cables through rigid channels. Tension variations on that cables generate measurable deformation of the elastic bodies. Using this deformation, the contact forces of the gripper can be estimated.

The use of surgical environment visual information obtained from the laparoscopic camera allows estimating the force from images. Tissues deformation produced by the contact between laparoscopic tools and tissues can be used to estimate interaction forces. Visual force estimation methods present several benefits and are a promising research field. Unfortunately, they require from a correct surface reconstruction to determine the depth and direction of the contact; tissue modelling of dynamic properties to estimate reaction forces against contacts and, finally, occlusions preclude correct penetration computation.

Although not having haptic capacity, as a perception of the efforts, interaction forces, made during the execution of the task, these robotic systems are provided with the necessary motorization at the joints of the master device to generate some haptic perception. This force feedback allows the human operator to perceive the mechanical limits of movements derived from the mechanical structure of the robotic arms, the virtual operation limits that may have been defined by the operator in advance or the security blockages that can be generated through the monitoring and supervision of the task.

### 2.3 Aiding Systems Based on the Supervision of the Task

Other support systems can be based on the autonomous or supervised execution of tasks, or part of tasks, which due to their predictability can be generated from a planner, based on predetermined procedures that can be executed with visual feedback. Many efforts are currently being done for their autonomous execution, or to guide the assisted manual execution of these procedures.

The study of an intervention workflow allows acquiring the knowledge of the phases of the procedure and find the way to automatize some of them. In [9] this information is used for autonomously guiding the camera and for the planning of minimal invasive port positions in combination with an initial setup. In [10] a series of simple tasks are studied aiming to be automated and thus releasing the surgeon from the need to execute systematic and repetitive operations, allowing him or her to focus on the most difficult aspects. The operations analyzed were puncturing, cutting and suturing. A wider study for the modelling of surgical procedures is described in [11] aimed to be applied for the evaluation of surgeon skills, analyzing clinical team workload, optimizing the operating room management and predicting next surgical task or their duration.

### 3 Aiding Systems in the Frame of the Bitrack Project

Bitrack is a teleoperated robotic system for laparoscopic surgery, a system that is expected to be able to reach the market after obtaining the certifications of the CE and the FDA. The prototype has been tested *in vivo*, with pigs, with various surgeons in different surgical specialties with satisfactory results and with a short learning curve, which is less than two sessions, thanks to the friendly user interface. Apart from regulations, further research is going on to provide new teleoperation aids, more than those conventional ones. The new aids developed that the surgeon can use when appropriate consist in haptic perception and guiding assistance for certain pre-established tasks.

#### 3.1 Force Estimation for the Generation of Haptic Feedback

To provide some feeling of the efforts exerted during the intervention, the interaction forces can be estimated from the visual perception of the deformation caused by the end effector in the working environment and its quantification generates the equivalent reactive forces to be fed back to the master device.

The contact with the environment to be detected, which must generate the estimated force to perceive, can be either due to the contact between the instrument and the tissues, or between an element held by the tweezers and the environment too, being this element either a suture needle, a clip or so. The perception system developed should be able to segment the image distinguishing between the instrument handled by the robot arm, its grasped element and the detectable elements of the environment. With this aim, the perception system operates from a pair of contour images, obtained from the stereo endoscope used. To speed up the computing time for obtaining the perception of efforts in real time, these images are simplified. First, using working windows around each instrument and second extracting some singular points to reduce the amount of data to process. Figure 6 shows the resulting segmentation. The instrument (labeled in red) has been identified from the window corresponding to the perimeter of each instrument and registered with the contour obtained by the projection of the CAD model of each instrument, seen from the 3D point corresponding to the position of the optics within the work space. This position is known by the control unit of the robot arm that guides it.

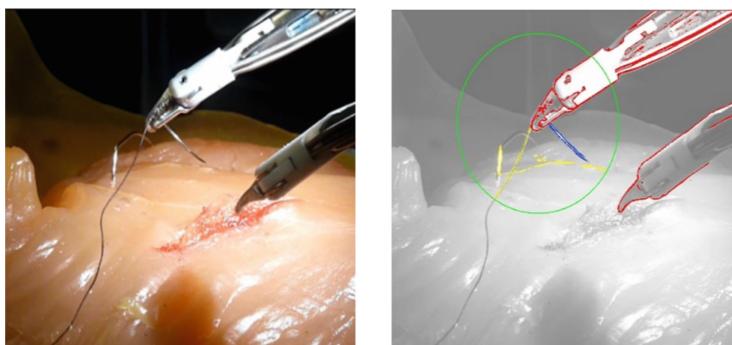
When performing the growth of the points in contact with the instrument, it is also necessary to carry out the 3D points matching with the geometric model of the needle used, to detect its presence (in blue). Once isolated the singular points corresponding to the instrument, a growth of the rest of the singular points found identifies those considered as belonging to the background (in yellow).

Once achieved the segmentation that allows to obtain the singular points of the instrument, or of any element transported by the instrument, which interact with the environment, the image processing system obtains the displacement vector  $\overrightarrow{P(t)}$  of each singular point, and proceeds to subtract each vector from the average of all the vectors considered, in order to eliminate the movement of the scene, Fig. 7. Once the movement is filtered, compensating in part the movement of the scene, the exerted forces are inferred, those that better justify the displacement vectors obtained from the points in

the environment. These calculated forces, which must emulate the F/T sensor that the tool should have had, generate the corresponding reaction forces on the control device, information that can be complemented with synthetic images on the screen.

### 3.2 Guidance Aids

The developed system is also provided with guidance aids, if this function is activated by the user, which allows generating forces of attraction that facilitate the surgeon to carry out these tasks. These guiding aids functions can be geometric, or they can be oriented to goals. The first ones are those that can generate geometrically trajectories from known passing points. The second ones are those needing a planner, able to generate a trajectory that allows the execution of a concrete objective.

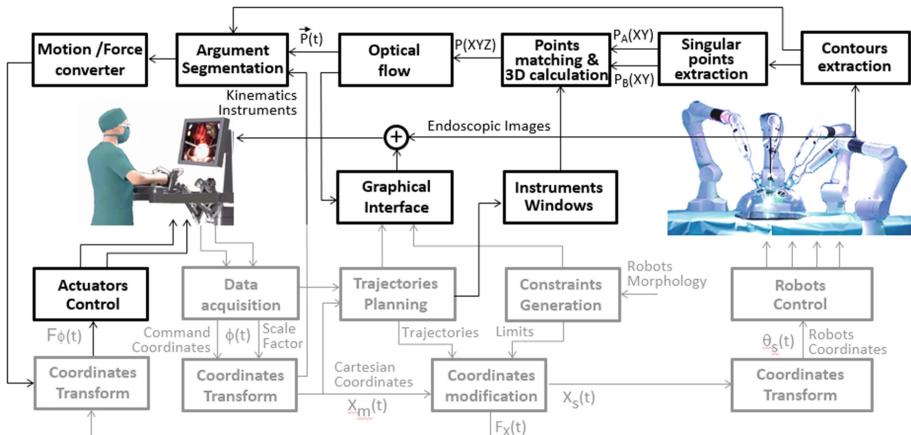


**Fig. 6.** Detection and tracking of the tools and the tissues around them.

Initially, the aiding functions introduced are of geometric type to be able to achieve the high reliability that surgical robotics requires. In this line, the developed aids which have given good and reliable results have been the guidance in operations of insertion and retrieval of the instruments, and in the realization of sutures.

For the retrieval assistance the system generates the forces that maintain the trajectory of the instrument axis along the entire path, since there are no other elements in the space occupied by the tool. The guided entry has two clearly differentiable variants by the own system: a first entry or a new entry towards the previous working point. In the first case, the guidance is oriented towards the center point of the field of view of the camera. This operation facilitates the advance when the instrument is still outside the camera field of view, and despite being blinded, the initial placement of the trocars guarantees that each instrument can be visible to the camera if the trajectory of the entry follows this direction. In the second case, the return to the working point is very useful both for cleaning the camera's optics, and for the cleaning or replacement of an instrument. With respect to the suture assistance, the surgeon after the puncture must rotate the needle as much centered as possible on the geometric center of the needle used. Since the segmentation allows to perceive an object at the tip of the instrument, the aiding system can register it with the different needles that can be used and

determine the position in space of its center with respect to the distal point of the instrument used, Fig. 8. The accuracy in the computation of the virtual center of rotation improves as the movements of the surgeon allows to get more points of view of the needle. The turning made along the suture can then be assisted at the surgeon's will, with a guiding force that maintains the rotation without displacement, which facilitates the task accomplishment with less stress of involuntary displacement of the tissue to suture.



**Fig. 7.** Structure of the vision system that can generate forces to emulate haptic perception

## 4 Future Possibilities

Since the beginning of laparoscopic surgery in the 2000, its progress has been more focused on the mechanics, passing from a robot with three arms to a new with four, and later on in modifying the robot architecture in order to increase accessibility and compatibility between the arms to avoid constraints and collisions.



**Fig. 8.** Detection of the needle geometric center in space for the guidance of the isocentric rotation along a suture

Currently, much work is oriented to introduce new aiding systems to more evolved robotic systems and to the robots that have appeared in the market recently or that will appear in the next years.

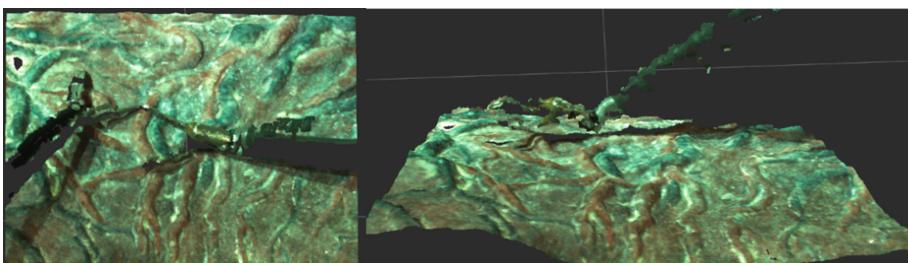
In this sense, different research efforts have been done from different perspectives. One of them is the Horizon 2020 SARAS project, *Smart Autonomous Robotic Assistant Surgeon*, aimed to provide autonomous assistance to the main surgeon, that teleoperates the robotic arms, with two additional robot arms acting as auxiliary surgeons. We are contributing with the development of a vision-based force estimator to be able to provide haptic feedback, and with the system architecture.

In what refers to the haptic feedback, all Robotic Assisted Minimal Invasive Surgery (RAMIS) include an endoscope to visualize the scene during a procedure. Therefore, this visual information from the endoscope can be used to relate the deformation in the tissue surface to estimate the applied force [12]. From the conservation principle of the continuum mechanics, the forces applied to an elastic object is directly proportional to its change of shape. Following this principle, we propose to reconstruct the surface of the tissue using the visual data from the endoscope and find the relationship between the deformation and the applied forces.

Then, the force estimation is developed in a three steps process. First, a stereo reconstruction of the surface of the tissue that generates a dense point cloud. In the second step, a computation of the surface deformation of the tissue from the point cloud. Finally, in the third step, the computation of the force estimation using the relationship with the computed deformation and the type of tissue.

**Surface Reconstruction.** In this part of the algorithm, we deal with the reconstruction of the workspace. Our approach is based on stereovision using the stereo endoscope of the system. This method uses the calibration of two different cameras, where the relative position of both is known, to search in one image all the pixels from the other via maximum correlation. These correspondences create a map of depths of the image pixels. Afterwards, this depths map is transformed in a 3D point cloud of the object. Figure 9 shows the reconstruction of the surface of a placenta.

**Deformation Computation.** From the previous recovered surface in this step the algorithm computes the deformation in the surface as the changes produced in a set of linearly independent vectors (3D lattice). We use this lattice in order to obtain a compromise between reducing computation cost and accuracy. So, in consequence, we use the lattice as our deformation model.



**Fig. 9.** Stereo reconstruction of a placenta surface

The 3D lattice  $\Gamma$  can be parametrized by the following formula:

$$\Gamma(x; P) = \sum_{l=1}^{y1} \sum_{m=1}^{y2} \sum_{n=1}^{y3} P_{lmn} \prod_{K=1}^K \xi_k(x_k) \text{ for } k = 1, \dots, 3$$

where  $\xi_k$  are a cubic basis spline and  $P_{lmn}$  denotes the displacement of a control point with  $y_1 y_2 y_3$  number of points. So, the lattice is calculated from the reconstruction by minimizing the following equation:

$$E_t(P) = E_\phi(\Gamma(x; P), R) + \gamma E_\psi(\Gamma(x; P)) + E_A(\Gamma(x; P))$$

where  $E_\phi$  is the discrepancy measure term,  $\gamma$  is the regulation parameter,  $E_\psi$  is the penalization term and  $E_A$  is a term to preserve the lattice shape.

Finally, the parameters  $P_{lmn}$  of the lattice are the inputs to the neural network used to determine the relation function between the deformation and the applied force of the last part of our approach.

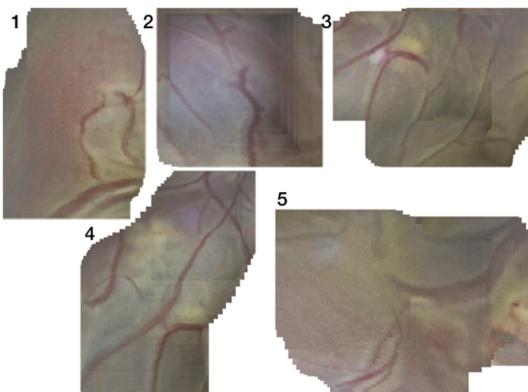
**Force Estimation.** As previously mentioned, our strategy to compute force feedback relies on using a Recurrent Neural Network (RNN) to get the relationship between the deformation and the applied force. The input to this neural network will be the lattice parameters and the geometric information. Specifically, we propose the usage of a Long-Short Term Memory (LSTM) based architecture to compute the force feedback.

Other approaches to calculate the force from the deformation are based on an experimental relationship between the deformation and the force [13]. In [14], the force is estimated from a combination of the deformation from the images and the electrical current from the motor during the interaction of the tool with the tissues.

**Application to Reconstruct Surface by Mosaicking.** An application of the computer vision in surgery is its application in a robust real time tracking for Fetsoscopic Photocoagulation (FLP) surgery for Twin-Twin Transfusion Syndrome (TTTS) [15]. TTTS is a syndrome where both twins share the same placenta and the blood flood has an unbalanced intertwin from the donor twin to the recipient twin through some vessel connections called anastomoses. The treatment of the TTTS relies on a fetoscopic laser to photocoagulate these anastomoses.

The system has been tested using videos from a real TTTS intervention. One of the principal problems of these videos is the low quality of images due to the dirty environment and liquid inside the amniotic sac jointly with the low quantity of light produced by the fetoscope. In consequence, the system must deal with images that are noisy, blurred and poor colour components.

This method assists in the robotic system to track and stabilize the region of interests during the coagulation of the anastomoses by compensating the placenta's movement by calculating the relative position of the fetoscope tool tip with respect to the placenta surface via local vascular structure registration.



**Fig. 10.** Placenta's surface reconstruction mosaics from the image sequences used in tests.

More precisely, the system works in the following three steps. First, the method binarizes the images from the endoscope to segment the local superficial vascular structures of the placenta. Second, using these vascular structures, the algorithm searches for a sequence of relevant points of interest (POIs) that are distributed among the image. Third, the system uses these points to calculate the transformation between two consecutive images. Figure 10 shows some placenta mosaics created using this method.

## 5 Conclusions

The aiding systems developed, based on geometric criteria, have shown that they facilitate the task of the surgeon after a relatively short learning time by performing a pyeoplasty and a clamp in the renal artery, after trying with different surgeons. These aids that robotization of laparoscopic surgery can provide, constitute just a first step that robotics can bring in the future in the field of surgery. In this first stage, only the aids for guidance were addressed since these control strategies can be justified rigorously in the necessary certification process. The trajectories that are taken as reference of a task can be geometrically defined from points provided by vision systems or from the calculation of the kinematics chain robot-trocars-instrument. On the other hand, performing tasks in a more autonomous way, based on expert systems, experience acquired by learning or by databases are currently more difficult to certificate. However, the good results obtained and the acceptance it deserved by the professionals that have collaborated, foresee great advances in this field in the coming years. Much can be done; however, the challenge is to achieve the required reliability that surgery requires.

**Acknowledgements.** The authors thank the contribution of Dr Antonio Alcaraz and Lluís Peri of Hospital Clínic, Barcelona, for their assessment along the clinical evaluation process. The ongoing research is partly funded from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 779813 (SARAS).

## References

1. Versius homepage. <https://cmrsurgical.com/versius/Versius.html>. Accessed 2019
2. RobSurgical Homepage. <https://www.robsurgical.com/bittrack/>. Accessed 2019
3. Ortmaier, T., Deml, B., Kübler, B., Passig, G., Reintsema, D., Seibold, U.: Robot assisted force feedback surgery. In: Advances in Telerobotics, vol. 31, pp. 361–379. Springer, Heidelberg (2007)
4. Choi, H.-R., Lee, D.-H., Kim, U., Gulrez, T., Yoon, W.J., Hannaford, B.: A laparoscopic grasping tool with force sensing capability. *IEEE/ASME Trans. Mechatronics* **21**(1), 1 (2015)
5. Trejos, A.L., Patel, R.V., Naish, M.D.: Force sensing and its application in minimally invasive surgery and therapy: a survey. In: Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 224, no. 7, pp. 1435–1454 (2010)
6. Zemiti, N., et al.: Force controlled laparoscopic surgical robot without distal force sensing. In: HAL Archives-Ouvertes (2015)
7. Zhao, B., Nelson, C.A.: A sensorless force-feedback system for robot-assisted laparoscopic surgery. *Comput. Assist. Surg.* 1–8 (2019)
8. Xue, R., Ren, B., Huang, J., Yan, Z., Du, Z.: Design and evaluation of FBG-based tension sensor in laparoscope surgical robots. *Sensors (Switzerland)* **18**(7), 1–18 (2018)
9. Weede, O., et al.: Towards cognitive medical robotics in minimal invasive surgery. In: ACM International Conference Proceeding Series (2013)
10. Muradore, R., et al.: Development of a cognitive robotic system for simple surgical tasks. *Int. J. Adv. Robot. Syst.* (2015). <https://doi.org/10.5772/6013>
11. Padoy, N.: Machine and deep learning for workflow recognition during surgery. *Artif. Intell. Oper. Room* **28**(2), 82–90 (2019)
12. Aviles, A.I., et al.: Towards retrieving force feedback in robotic-assisted surgery: a supervised neuro-recurrent-vision approach. *IEEE Trans. Haptics* **10**(3), 431–443 (2016)
13. Giannarou, S., Ye, M., Gras, G., et al.: Vision based deformation recovery for intraoperative force estimation of tool-tissue interaction for neurosurgery. *Int. J. CARS* **11**, 929 (2016). <https://doi.org/10.1007/s11548-016-1361-z>
14. Dong-Han, L., Hwang, W., Lim, S.C.: Interaction force estimation using camera and electrical current without force/torque sensor. *IEEE Sens.* (2018)
15. Sayols, N., Hernansanz, A., Parra, J., Eixarch, E., Gratacós, E., Amat J., Casals, A.: Vision based robot assistance in TTTS fetal surgery. In: IEEE-Engineering in Medicine and Biology Conference (2019)



# Suitable Task Allocation in Intelligent Systems for Assistive Environments

Manuel Vinagre, Joan Aranda, and Alicia Casals<sup>(✉)</sup>

Universitat Politècnica de Catalunya, BarcelonaTECH, Barcelona, Spain  
[{manuel.vinagre, joan.aranda, alicia.casals}@upc.edu](mailto:{manuel.vinagre, joan.aranda, alicia.casals}@upc.edu)

**Abstract.** The growing need of technological assistance to provide support to people with special needs demands for systems more and more efficient and with better performances. With this aim, this work tries to advance in a multirobot platform that allows the coordinated control of different agents and other elements in the environment to achieve an autonomous behavior based on the user's needs or will. Therefore, this environment is structured according to the potentiality of each agent and elements of this environment and of the dynamic context, to generate the adequate actuation plans and the coordination of their execution.

**Keywords:** System architectures · Cognitive systems · Tasks manipulation · Semantic nets

## 1 Introduction

The increasing demand of assistive services due to the continuous aging population carries with it a deficit on the availability of human resources able to cope with these needs in the future. Assistive technologies are thus the potential solution to these growing requirements of human welfare in an ageing population. By facilitating the different assistive services to attend daily life tasks, this technology is expected to increase the user's autonomy and their own self-esteem. Looking at what technology offers today, one can observe that technology in this area is quite diverse, ranging from simple devices that help in communicating and in mobility, those which are currently available, to more intelligent devices, as robotic systems capable of perceiving certain environment conditions and dynamically adapt to their changes.

These more advanced systems differ in the way of managing the perceived information. While reactive systems respond to stimulus or orders in an immediate manner, deliberative systems perform an abstraction of the perceived information and do some reasoning to provide a behavior adapted to every situation. Being those latter systems able to achieve capacities closer to those of humans, they become of interest in assistive systems since they allow increasing usability and efficiency.

When dealing with daily live activities, gesture and activity recognition aiming to interpret the user's will, or their needs, allows the generation of robot actuation strategies in a proactive way [1, 2]. These proactive strategies may even require the interaction of multiple robots, when more than one arm is needed to accomplish a given task [3]. At present, many applications require the generation of intelligent environments from the

combination and coordination of different technologies. Industry 4.0 is an example of intelligent environment, which goal is the increase of automation, flexibility and scalability of industrial settings. The advantages of this technology that provides companies with the ability to quickly adapt to production changes are identified in [4], emerging from here the concept of smart retrofitting. Other environments that are experimenting significant changes are those related to surgery, which lead to the concept of surgery 4.0. These new advances in surgical innovation, constitute a step forward in the progressive implantation of Minimally Invasive Surgery (MIS). As explained in [5], the key factor is the intelligent collaboration between assistant personnel, surgeons, and assistance and autonomous systems in the Operating Room. The aim is to support decision making in surgery, contextual assistance and surgical training. When referring to daily life, the concept of Healthcare 4.0 appears too. In this context the efforts focus on the implantation of intelligent services to support daily life therapies. In [6], an open code system is presented aimed at generating an intelligent environment able to monitor the patient from diverse sensors, analyze the obtained data and manage the actions to perform in an intelligent manner through a complex events processor.

The increasing availability of intelligent systems and robots at home will make them part of our daily life. Then, it will be necessary an adequate interconnection and coordination of all these systems to generate an adaptable and useful ecosystem. Thus, the aim of the work is the development of methodologies that allow this coordination through the design of a platform composed of intelligent agents, in a domestic environment.

## 2 Intelligent Environments

The way to generate intelligent environments in any of the above ambits can follow different methodologies. Those more frequently used are the implementation of cyber-physical systems, multirobot environments and heterogeneous environments, which combine the previous ones.

### 2.1 Cyber-Physical Systems

Most intelligent environments are composed of cybernetic means that process information in the digital space by means of physical devices, which by means of sensors and actuators acquire information and act on the real world. The so-called Cyber-Physical Systems (CPS) integrate three fundamental parts: computation, communication and physical control. They allow building environments that can exhibit multiple actuation modes in function of the context through digital or virtual information and the real world.

Internet of Things (IoT) is a kind of CPS where a connection is created between objects and humans, processing information in real time. An example of an intelligent assistive environment for elder is that presented in [7], which provides a fluid interaction between the user and the environment and the care givers.

Although CPS allow a wide range of services, these services are mainly oriented to the connection and communication through internet. In most IoT platforms, the coordination of devices to generate services is pre-programmed, and the links between inputs and outputs are established to generate some given behaviors.

## 2.2 Multirobot Systems

A Multirobot system (MRS) can be characterized as a set of robots operating in the same environment, having been widely applied in different domains [8]. An MRS can improve the effectiveness of a robotic system, providing better performance in the execution of some tasks, more robustness and reliability, thanks to its modularity. This happens not only when the robots have different functions, but also when they all have the same capacities. However, most frequently robots are heterogeneous to exploit their respective characteristics according to the tasks and thus, obtain more efficient solutions, although this implies complex coordination strategies.

In this work, much attention is paid to MRS developments that operate in dynamic environments, where uncertainty and unforeseen changes can occur due to the presence of other external agents (for instance the presence of humans). Even considering uniquely the subset of MRS in assistive robotic applications, it is not easy to identify a common frame for comparing the technical solutions known up to now [9]. This is a common phenomenon that occurs in other application domains, even easier to specify, as shown for instance with the teams of Robot-Soccer in the RoboCup competition.

An MRS cannot be considered simply as the generalization of a single robot system, and the proposed approaches must be characterized in terms of how suppositions over the environment are established and how the system is internally organized [10].

Among the most common problems of multi-robot systems, the focus of this work is the design of new methods of tasks assignment, selecting the minimum number of robots most suitable to perform a certain task [11].

## 2.3 Heterogeneous Systems

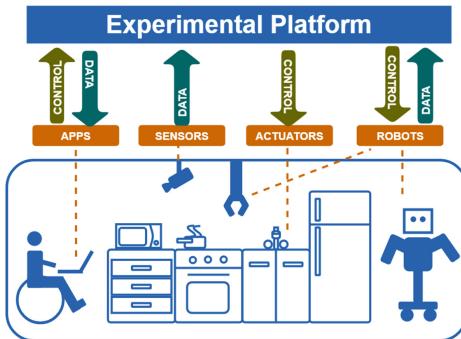
The combination of cyber-physical and robotic systems in an environment generates a great added value. On the one hand, multi-robot systems can benefit from accessing the resources provided by a CPS system, such as information from the environment and the user. This allows robots to have more information to elaborate their decision processes and more capacity of acting on the environment. For instance, if the CPS contains a set of cameras in the scene and a software component that detects objects in it, then the robot can use this information to locate a certain object that cannot be easily found through its perception system. In another situation, if a given actuation requires that the robot must access inside a refrigerator with automatic opening, the robot can take advantage of this resource to access its content in a simpler way. On the other hand, the CPS system benefits from the resources provided by the robot. For instance, the CPS

system can acquire not available information from the robot scanning actions to areas inaccessible by the CPS.

In the interrelation process between CPS and robotic systems different aspects must be considered: communications protocol, interoperability method, information sharing, modeling and capacity coordination, among others. One of the fields that stands out in the development aid of this type of systems is robotics in the cloud. In this field cloud-based technologies are investigated in order to give the robots access to digital resources. The most relevant advantages of this type of systems is the access to computing resources as servers for data analysis or big data and flexible data storage resources. Now, a new paradigm pursues the convergence of cyber-physical and robotic systems, the Internet of Robotic Things (IoRT). This paradigm is based on three fundamentals fields: IoT, robotics and cloud computing.

### 3 Architecture Design for Intelligent Environments

To assist people efficiently and user-friendly, an intelligent environment architecture has been designed based on different devices, which can be sensors, actuators, applications or robots (Fig. 1). This platform has been implemented and tested within the project AURORA, with the aim to create services in a dynamic way in healthcare settings.



**Fig. 1.** Experimental platform

This platform offers different features. The first is the connectivity and management of devices in the environment. The second is the modeling of devices from a general model that contains their relevant information as their capabilities. Third, a modeling of the environment describes its state which is used as context information. There is also a dynamic service generator that depends on the capabilities of available devices and on the context. Then, the assistive services available at a certain time are presented to the user through an interactive interface. Once the user demands for a specific service, or a pro-active service is offered and accepted, the platform controls and monitors its correct

execution. The platform is composed of three main modules: the module of knowledge management, the module of devices' abstraction and the module of services.

The knowledge management module is responsible for maintaining all the information about the environment status and the relationships and rules between the different elements. This information is stored in classes and instances in ontological models using OWL (*ontology web language*). For this module, the knowledge processing system developed in the KnowRob project is used [12]. It provides mechanisms of storage and recovery of actions, objects, processes, events, properties and relations. The knowledge of agents' capacity is established as a property owned by actors, which represents the possibility to perform a certain task. Capabilities are structured hierarchically from two basic abilities: information and operation. At functional level, this module is responsible for the knowledge of the 3D model of the different objects and the dynamic model of the environment.

The device abstraction module manages all existing devices in two layers: an interconnection and a modeling layer. The first layer is responsible for handling communications between all devices using ROS middleware (Robotic Operating System [13]). To build a heterogeneous system, a communication from ROS to the Open-Hab platform is used. Open-Hab is a platform for the interconnection of smart devices (IoT).

Above this layer, the device modeling layer contains a definition of the types of devices and their capabilities. These definitions are managed considering the information provided by the knowledge management module. At functional level, this module allows the interconnection of the actors: control agents, interface agents and recognition and location agents.

The service module handles the definition, updating, progress and achievement of services. Contextual information is managed in a similar way to [14], with the adaptation of the concept of skill as a quantifiable unit of a service. The services are then the result of the planning through the composition of different possible skills depending on the state of the environment and the capabilities of the actors. These skills have as objective to fulfill a service and are evaluated by a factor of competence of the skill. This factor is calculated by the evaluation of each skill and the available capability. These evaluations are calculated under the current context conditions.

We distinguish between three different types of evaluation functions. First, those related to actions that modify the environment. For example, there is an evaluation on the manipulation of objects depending on its shape, performed action and environment status. Second, functions for evaluating actions addressed to the user, mainly skills and abilities of person-robot interaction. And third, those that assess the acquisition of the information about the state of the environment and the user, as well as its interpretation.

## 4 Implementation

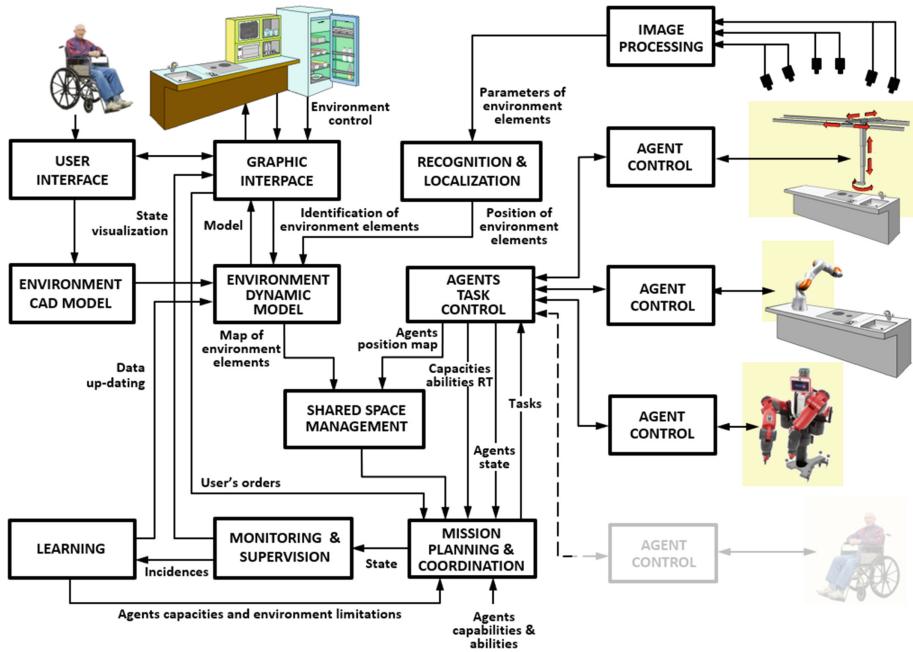
### 4.1 Hardware

An experimental platform has been implemented to validate the architecture. This platform emulates a kitchen living environment with the automation of different

devices. A table surface with an induction cooker, an automated water tap of a sink, a cabinet and a refrigerator equipped with a motorized system for opening/closing doors. Two 3D cameras cover the whole environment and a user interface allows the interaction with the environment from an interactive graphical interface to select elements of the environment and trigger the actions necessary to achieve the proposed objectives.

To support actions over the environment, the system counts on three robots with very different and complementary features: a Cartesian robot hanging on the ceiling (CAPDI), a second double-handed robot for complex manipulation operations (BAXTER) and a single robot arm over the table surface (MICO).

The implementation of the proposed control strategy is summarized in Fig. 2. This figure shows the agents currently incorporated and in attenuated mode the possibility to add new agents.



**Fig. 2.** System architecture showing the main physical components, the agents, and the modules

## 4.2 Multi-robot Task Allocation System

In the setup considered in this work, the team of robots have different architectures. They coordinately assist a disabled user performing daily tasks in a kitchen. Several task levels can be considered. The first level is constituted by the high-level tasks, or services, such as “making a coffee” or “pouring a glass of water”. These types of actions are those that will most likely be triggered by the user through a user-machine interface. These services or goals are compound or even complex tasks which require

the execution of various sub-tasks. For instance, “pouring a glass of water” will require (1) to fetch an empty glass, (2) to grasp a bottle of water, (3) to pour water from the bottle into the glass. These lower level tasks can themselves be decomposed into sub-subtasks such as “picking the glass from its location”, “placing the glass at a given location”...

The services are decomposed into a *precedence graph* of simple or elementary tasks by means of preprogrammed receipts considering some ordering constraints. These tasks must be allocated to the convenient agent at any time, permitting the services to overlap in time, by starting one service even before others finish. Every task will be assigned considering three constraints: capability, capacity and precedence.

The Multi-Robot task allocation problem can be stated as:

*Finding an optimal allocation A of a set of tasks T to a subset of robots R, that will be in charge of carrying it out:  $A: T \rightarrow R$*

The implemented solution is based on an iterated multi-robot auction process inspired in the work by [15] conveniently modified for our precedence-constrained task scheduling. Despite certain drawbacks, auction-based methods are indeed the most appropriate to work with heterogeneous teams, since they are the most suited to consider the different capabilities [16]. The optimization objective will be to minimize the make span, i.e. the total schedule duration.

In the proposed solution, the tasks are scheduled in batches. It is assumed that in each batch, the tasks are pairwise independent. This enables to schedule every task of a batch without considering the scheduling of the other tasks in the same batch. Batch selection is performed by the auctioneer, which takes as input the *precedence graph* defined by receipts. At each iteration, it selects the tasks that have either no predecessor or which predecessors have all been scheduled.

Once the batch of tasks to auction has been selected, it is broadcasted to all bidders. To take into account heterogeneity, McIntire’s solution is modified by adding the following step: before computing the bid for a task, the robot evaluates if it can be done with respect to its *capabilities*. Each robot then computes a bid for each feasible task in the batch and for each possible position in its current schedule. The best result of each robot is sent to the auctioneer, which selects the best “offer” and communicates the winner to the robots. The winner updates its schedule and all participants remove the assigned task from the tasks to be auctioned. The process is repeated until all the tasks from the auctioned batch have been scheduled.

The task schedules generated in this way are represented using Simple Temporal Networks (STN), introduced in [17]. STNs are data structures similar to graphs, in which nodes symbolize events and edges symbolize timing constraints on the events.

Notice that in this stage the system follows the *decompose-then-allocate* paradigm, in which the services are decomposed into tasks that are then allocated optimizing duration under capability, capacity and precedence criteria. However, before task execution, the *allocate-then-decompose* approach will be used, in which the tasks assigned to robots can be decomposed locally using a spatial 3D suitability function.

### 4.3 Supervising and Optimizing Robot Capabilities

Once all tasks have been assigned to each agent, an assessment of the best execution by means of the degree of adequacy of the capabilities of each agent is calculated over the workspace. For that, there is a common data structure of the workspace for general coordination and management. This data is a 3D map represented by the voxelization of the workspace volume and a given value for each voxel (workspace map). One is the occupancy map that provides the occupancy of the workspace that includes static elements loaded from the pre-known CAD of the environment and dynamic elements that are updated in real-time from the processing of the images provided by the different.

After the task allocation has been established, the implemented evaluation of capacities and abilities aims to be spatially evaluated on each action. To achieve this objective, a spatial three-dimensional suitability function is used, calculating for each free agent a suitability value in each voxel of the workspace map using Eq. 1.

$$\Phi_a(x, y, z) = \frac{1}{n} \sum_{i=1}^n p_i(x, y, z) \quad (1)$$

where  $a$  is the evaluated agent, and  $p_i$  a function based on the capabilities of each agent given by quantitative and qualitative parameters. Those quantitative that have been considered initially for the validation of the proposed strategy are shown in Table 1.

**Table 1.** Quantitative parameters

$p_1 = \frac{(P_{max} - P_r)}{P_{max}}$	<b>Load margin around workspace.</b> $P_{max}$ is the load capacity of robot and $P_r$ is the load of the object to be manipulated
$p_2 = 1 - \frac{(T_i - T_{min})}{T_i}$	<b>Execution speed factor.</b> Measured from the origin of the trajectory to each intermediate point of the trajectory
$p_3 = \left(1 - \frac{1}{d_{min} + 1}\right)$	<b>Accessibility to the operating point.</b> Where $d_{min}$ is the distance to a singularity of the articulation, which represents a minimum among all of them
$p_4 = \frac{(h_{max} - h_r)}{P_{max}}$	<b>Grasp capacity</b> , where $h_{max}$ is the opening value of the end effector, and $h_r$ the opening resulting from prehension

The qualitative parameters introduced as indicators of the quality of the task performed by each agent, in this case robotic arms, are shown in Table 2.

**Table 2.** Qualitative parameters

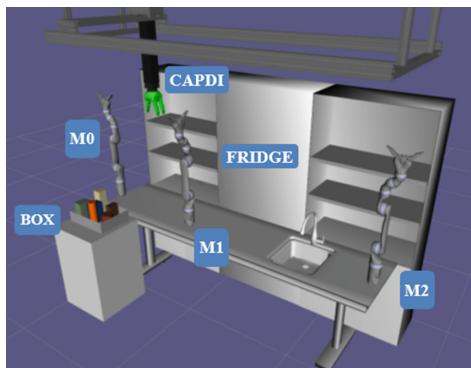
$p_5 = 1 - \frac{Vpp_{max}}{Vpp}$	<b>Smoothness of task execution</b> The value Vpp, is the mean value of the peak values of the frequency signal generated by an accelerometer located at the end effector of each agent, in each path. $Vpp_{max}$ is the maximum value from the signal along the entire trajectory
$p_6 = 1 - \frac{\Gamma_{max}}{\Gamma_r}$	<b>Ease of manipulation</b> throughout the completion of each task, in which $\Gamma_{\{r\}}$ is the pair made by the last articulation of the kinematic chain of each robotic arm, and $\Gamma_{\{max\}}$ the maximum admissible torque

The action will switch to the free agent with the maximum suitability in the workspace map from a predefined trajectory of the actual action. Since performing this process can lead to excessive segmentation and transfer of tasks between agents, a hysteresis factor is introduced in the allocation change algorithm.

## 5 Performance Evaluation

The performance of the system architecture has been tested with a given problem in two simulated scenarios, similar to the real one. These simulations involve modifications and physical constraints with the objective of testing the system performance.

The conditions of the first simulation are slightly different from the implemented setup. We consider the case where the kitchen is equipped with one CAPDI robot and three, instead of one, MICO arms. This enables to test more demanding scenarios, as well as to evaluate the possibility of parallel task execution, which is rather limited in the real Aurora setup. In the rest of this subsection, the three MICOs will be identified as “M0”, “M1” and “M2”. This arrangement, which can be seen in Fig. 3 enables to reach both cupboards to store objects and to create parallelism.



**Fig. 3.** Evaluation scenario

In order to evaluate the interest of having two MICOs on the left side of the table, the second simulation is carried out on the same scenario but without M1.

Given both simulations, the following problem is presented to the system: food items from a box must be unpacked, and each item must be stored to its place. The robots must store six items: Sugar (green box) and coffee (brown box), to be stored on the shelves of the left cupboard. A tomato sauce can (in red) and pasta (in blue), to be placed on the shelves of the right cupboard. A carton of milk (in pale yellow) and a bottle of juice (in orange), that must be stored in the fridge (see Fig. 4).

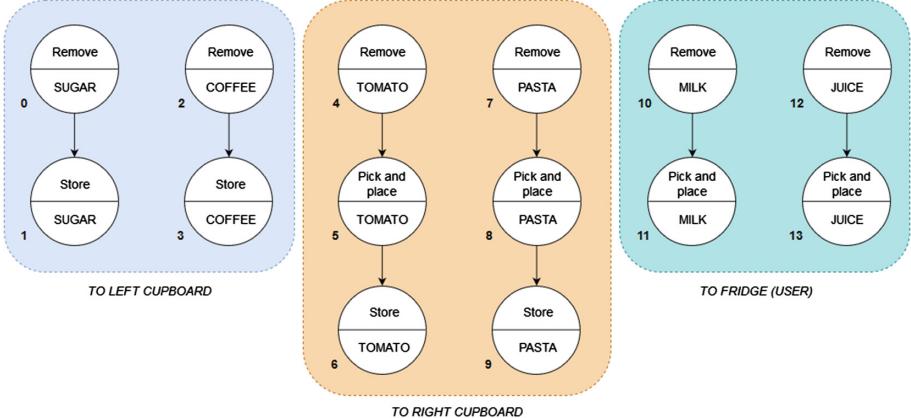


Fig. 4. Task precedence graph

Some of the tasks can be performed by several agents, while others can only be achieved by one of the team members, depending on the agent's capabilities and reachable workspace. The precedence graph is represented in Fig. 4. Each task is represented by a circle divided into two: in the upper half is the action type, and in the lower the associated object. The global precedence graph is composed of six disconnected branches which can operate in parallel. The bold number at the left of each circle indicates the task index. For every task its duration is known (sec.), the object, initial and final point (if needed), and other internal parameters needed for task execution (Table 3).

The output schedules obtained for the proposed problem and for each simulation are shown in Table 4. The first column of each table presents the task ids in ascending order with respect to their start times. For each task, its start and finish times are specified along with the agent to which it has been assigned. The make span is indicated in bold; for all cases, it corresponds to the finishing time of the last task to be started.

This result obviously presents a different task distribution and task sequence. In the first simulation, CAPDI has only three tasks to complete, those of bringing the pasta and tomato sauce to the other side of the table and that of bringing the milk to the fridge. In the second case, CAPDI is also in charge of bringing the juice to the fridge, while M1 is not present so all tasks must be assigned to other robots. M0 oversees the rest of tasks previously assigned to M1, since M2 performs the same tasks in both cases.

Comparing the two variants, results are as expected: the make span is 41% longer with one robot less. This is due to the fact, visible in the task distribution of the first variant, that M0 and M1 work in parallel most of the time.

**Table 3.** Task list definition

Id	Type	Duration(s)	Object	Surface
0	8	30	sugar	Table
1	0	30	sugar	lower left cupboard
2	8	30	coffee	table
3	0	30	coffee	middle left cupboard
4	8	30	tomato	Table
5	0	45	tomato	Table
6	0	30	tomato	middle right cupboard
7	8	30	pasta	Table
8	0	45	pasta	Table
9	0	30	pasta	middle right cupboard
10	8	30	milk	fridge
11	0	30	milk	fridge
12	8	30	juice	fridge
13	0	30	juice	fridge

**Table 4.** (a) Output schedule for simulation 1 and (b) Output schedule for simulation 2

Task	Start time(s)	Finish time(s)	Agent
4	5.7	35.7	M1
0	6.8	36.8	M0
5	35.7	80.7	Capdi
2	40.8	70.8	M1
10	42.8	72.8	M0
12	78.3	108.3	M1
7	79.7	109.7	M0
6	80.7	110.7	M2
11	96.8	126.8	Capdi
13	108.3	138.3	M1
1	110.3	140.3	M0
8	133.6	178.6	Capdi
3	149.6	179.6	M0
9	178.6	208.6	M1+M2

(a)

Task	Start time(s)	Finish time(s)	Agent
0	6.8	36.8	M0
10	42.8	72.8	M0
11	72.8	102.8	Capdi
12	80.3	110.3	M0
13	110.3	140.3	Capdi
7	117.2	147.2	M0
8	147.2	192.2	M0+Ca
2	152.3	182.3	M0
4	188.6	218.6	M0
9	192.2	222.2	M2
5	218.6	263.6	Capdi
1	218.8	248.8	M0
3	258.1	288.1	M0
6	263.6	293.6	M2

(b)

The improvement of the quality has been applied in both simulations. From the variant of four robots, task 9 was assigned only to robot M2, but it has been replaced by a shared task between robot M1 (free) and M2. The position of the pasta placed by CAPDI in task 8 gives a suitability of  $\Phi_{M1} = 0.8$  and  $\Phi_{M2} = 0.6$  mainly due to a better accessibility and easiest manipulation in the workspace. When M1 carries pasta to right direction at the middle between both robots, M1 places pasta and it is picked by M2 because it has a major suitability mainly due to zero accessibility of M1 to reach the right cupboard.

In the second simulation, task 8 firstly allocated to CAPDI it's finally shared between robot M0 (free) and CAPDI. This is due to the less suitability of CAPDI to reach the pasta in the position placed after it has been removed from the box. M0 has a better execution speed and smoothness for the task, so, it translates pasta towards the center of the kitchen until  $\Phi_{CAPDI} > \Phi_{M0}$ .

## 6 Conclusion

When dealing with different kind of agents that collaboratively should perform a task, an adequate management of all the resources considering the capabilities and abilities of the agents, the state of the elements in the scene and the sequence of actions should be considered. This work has been addressed to formalize a methodology that allows considering all the involved elements to establish the sequence of actions and the allocation of the robot assigned to each action. Then, the methodology has been evaluated on slightly modified scenarios of a robotized kitchen.

**Acknowledgments.** The work has been developed in the frame of Project RTC-2015-3926-1, from MINECO and with Feder funds.

## References

1. Vinagre, M., Aranda, J., Casals, A.: An interactive robotic system for human assistance in domestic environments. In: Lecture notes in computer science, vol. 8548, pp. 152–155 (2014)
2. Vinagre, M., Aranda, J., Casals, A.: A new relational geometric feature for human action recognition. In: Informatics in Control, Automation and Robotics, pp. 263–278. Springer, Heidelberg (2015)
3. Aranda, J., Vinagre, M.: Anticipating human activities from object interaction cues. In: IEEE International Symposium on Robot and Human Interactive Communication, pp. 58–63. IEEE (2016)
4. Guerreiro, B.V., Lins, R.G., Sun, J., Schmitt, R.: Definition of smart retrofitting: first steps for a company to deploy aspects of industry 4.0. In: Advances in Manufacturing, pp. 161–170. Springer, Heidelberg (2018)
5. Maier-Hein, L., et al.: Surgical data science: Enabling next-generation surgery. In: arXiv.org. Cornell University (2017)
6. Gonzalez-Usach, R., Collado, V., Esteve, M., Palau, C.E.: May. Aalopen source system for the monitoring and intelligent control of nursing homes. In: IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), pp. 84–89 (2017)
7. Dohr, A., Modre-Opsrian, R., Drobics, M., Hayn, D., Schreier, G.A.: The internet of things for ambient assisted living. In: Seventh International Conference on Information Technology: New Generations, pp. 804–809 (2010)
8. Yasuda, T. (ed.): Multi-robot systems: trends and development. In: TechOpen (2011)
9. Das, G.P., McGinnity, T.M., Coleman, S.A., Behera, L.: A distributed task allocation algorithm for a multi-robot system in healthcare facilities. *J. Intell. Rob. Syst.* **80**(1), 33–58 (2015)

10. Farinelli, A., Iocchi, L., Nardi, D.: Multi robot systems: A classification focused on coordination. *IEEE Trans. Syst. Man Cybern. part B* **34**(5), 2015–2028 (2004)
11. Hernansanz, A., Casals, A., Amat, J.: A multi-robot cooperation strategy for dexterous task-oriented teleoperation. *Robot. Auton. Syst.* **68**, 156–172 (2015)
12. Tenorth, M., Beetz, M.: KnowRob – a knowledge processing infrastructure for cognition-enabled robots. *Int. J. Rob. Res. (IJRR)* **32**(5), 566–590 (2013). Part 1: The KnowRob System
13. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
14. Santofimia, M.J., Villa, D., Villanueva, F.J., Escolar, S., Lopez, J.C.: A semantic middleware architecture for supporting real smartness. In: IECON 2016 – 42nd Annual Conference of the IEEE Industrial Electronics Society, pp. 6925–6930 (2016)
15. McIntire, M., Nunes, E., Gini, M.: Iterated multi-robot auctions for precedence-constrained task scheduling. In: International Conference on Autonomous Agents and Multi-agent Systems, pp. 1078–1086 (2016)
16. Khamis, A., Hussein, A., Elmogy, A.: Multi-robot task allocation: a review of the state-of-the-art. In: Cooperative Robots and Sensor Networks 2015, pp. 31–51. Springer, Heidelberg (2015)
17. Dechter, R., Meiri, I., Pearl, J.: Temporal constraint networks. *Artif. Intell.* **49**(1), 61–95 (1991)

# **Simulation in Robotics**



# Coverage Path Planning Optimization for Slopes and Dams Inspection

Iago Z. Biundini<sup>1</sup>(✉), Aurelio G. Melo<sup>1</sup>, Milena F. Pinto<sup>2</sup>,  
Guilherme M. Marins<sup>1</sup>, Andre L. M. Marcato<sup>1</sup>, and Leonardo M. Honorio<sup>1</sup>

<sup>1</sup> Federal University of Juiz de Fora, Juiz de Fora, Brazil  
[iago.biundini@engenharia.ufjf.br](mailto:iago.biundini@engenharia.ufjf.br)

<sup>2</sup> Federal Center for Technological Education of Rio de Janeiro,  
Rio de Janeiro, Brazil

**Abstract.** In the last decades, there is a growing in the use of UAVs for inspection applications due to their maneuverability, flexibility, and efficiency. UAVs can be used for regular inspections to verify deformities, reconstruct 3D spaces, and mapping through aerial photogrammetry. An important aspect when inspecting buildings and structures is to cover the entire structure as efficiently as possible. Some applications, such as the inspection of electric power generation facilities, demand security constraints to ensure safety in their big structures like slopes and dams. Those inspections will be repeated regularly to ensure that changes in the structures are not occurring over time. The determination of the optimal path planning for those structures can be quite complex. This is due to the presence of obstacles and safety restrictions, such as high voltage power lines, transformers, and water outlets that will be present and can not be modeled previous to the first inspection. Thus, these initial inspections are performed manually by a skilled operator. However, this first path is performed without considering mission time and optimal trajectory. Therefore, this research work proposes a methodology to maximize coverage inspection trajectories generated manually. The results proved the algorithm capacity of optimizing trajectories performed manually by an operator.

**Keywords:** Optimized Path Planning · Inspection · Surface Deformation Analysis · Unmanned Aerial Vehicle

## 1 Introduction

Electric power generation facilities have large structures, such as slopes and dams, that are subject to forces and wear, which demands inspection periodically to ensure security constraints and safety. Large and complex structures like these can be challenging to access and inspect and are best surveyed from the air. In this sense, in recent years, due to the development in the Unmanned Aerial Vehicle (UAV) capabilities, the use of UAVs for regular inspections to verify deformities is increasing. As an example, in the work [9], the authors presented

a UAV application to perform assessment and inspection of large structures. Another use is in [2]. The images acquired by the UAV are georeferenced by markers installed in the inspected structure to increase the inspection reliability. A critical part of the cited works is the proper path planning to provide reliable information. The path planning has to be efficient and also provide enough information for the analysis required. Thus, different from those cited works, this research focus on optimizing coverage path planning for slopes and dams inspection.

Path planning is an active topic in robotics and refers to find an optimal route of a moving object from the starting to a final point under certain constraints [1]. Coverage Path Planning (CPP) is a significant issue in many branches of the robotics field. CPP is the task of determining a path that passes over all points of an affected area. Several applications in the robotic field are variations to the problem of finding optimal paths. For example, [6] showed an implementation of the watchman route problem to find CPP solutions. Other applications applying CPP can be found in cleaning robots [13], underwater inspection [16], mining robots [10], among others.

Besides finding a solution, other requirements are also crucial for CPP. In some cases, distance minimization, energy usage, mission time, or other parameters are essential, and must also be optimized. For example, the work of [3] proposed an energy-aware CPP algorithm that generates optimal sub-paths considering photogrammetric requirements and estimates feasibility for the UAV energy constraints. In [7], a different approach is presented where the two objectives are optimized. The first objective is to find a given target in a coverage problem. The second one establishes a communication link. Despite the good results, both works are limited once they consider that the path provides coverage only to a 2D surface.

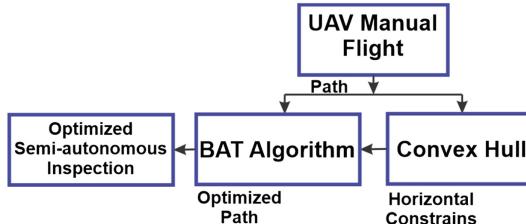
In the topic of 3D structures inspection, it is not always possible to obtain an accurate model of the area to be inspected previous to the activity, making it challenging to generate safe trajectories beforehand. In some scenarios, such as populated areas, in spaces close to machinery and flammable material, this is very critic, and an incident may cause serious health, environmental or financial impacts. In this sense, the first inspection of this kind of area has to be performed by a trained operator. In this context, this research proposes a methodology to optimize trajectories generated manually or through autonomous navigation methods for extensive scale surfaces inspection. The proposed methodology is relevant to complex scenarios that have to be inspected cyclically, and it may contain multiples deformations along the path. This approach allows a trained operator to perform a first inspection covering relevant parts of the inspection. Then, the optimized path can be latter automatically repeated if there are no changes in the obstacle positions. The main contributions of this work can be summarized as follows:

- An optimized approach to provide qualified information for inspection and 3D surface reconstruction for slopes and dams;
- Efficient methodology to optimize coverage path planing applied in 3D structures;

The remainder of this research is organized as follows. Section 2 details the methodology and its foundations for large structures inspections. Section 3 shows the proposed experiments with a proper discussion of the results. The concluding remarks and future works are conducted in Sect. 4.

## 2 Proposed Methodology for Large Structure Inspection

The main objective of this research is to provide a method of optimizing CPP trajectories. These trajectories aim at capturing images using a UAV. Those images are used for a 3D reconstruction. Thus, it is necessary that the UAV be able to obtain enough photos with a proper overlap to generate the required dense point cloud data. Also, the final mission must be able to be reproduced several times with similar characteristics of the first flight to maintain the quality of Point Cloud. A common objective to optimize flight parameters such as power usage, the safety of maneuvers, and flight times. Figure 1 presents a global overview of the proposed methodology. Through the waypoints and distances collected from the first flight (i.e., manual flight), the Convex Hull algorithm [8] is used for choosing a set of all the waypoints acquired in the manual mission to smooth and ease the optimization of the mission. Then, the original data, along with the boundary determined by the Convex Hull algorithm, is feed into an optimization algorithm (i.e., Bat algorithm). This algorithm will determine the optimal path that covers all inspected location. This algorithm has the goal to optimize a multi-objective function involving variables such as inspection time, area, among others. The output is a new set of waypoints and orientations that can be executed to perform subsequent inspections.



**Fig. 1.** Proposed methodology.

### 2.1 Problem Formulation

An essential aspect of the CPP algorithm is the overlap between pictures. This overlap is a requirement once a point has to appear in at least three images to have its position adequately calculated. There are two types of overlap, in this case, i.e., horizontal and vertical ones. The horizontal movement is defined through the points chosen by the convex hull. The minimum distance between

them is defined as ( $D_{min}$ ). At each point, a vertical mission is defined going from a minimum ( $h_{min}$ ) to a maximum height ( $h_{max}$ ). The size of each image depends on the distance of the UAV to the surface ( $Dist_H$ ), and the camera opening angle ( $\theta_{hor}$ ), as shown in Eq. 1, where the horizontal distance to the surface is defined by ( $Dist_{hor}$ ).

$$\tan\left(\frac{\theta_{hor}}{2}\right) = \frac{Dist_{hor}}{2Dist_H} \quad (1)$$

The horizontal coverage rate ( $Coverage_{hor}$ ) is given by the ratio between  $Dist_{hor}$  and  $D_{min}$  as presented in Eq. 2.

$$Coverage_{hor} = 1 - \frac{D_{min}}{Dist_{hor}} \quad (2)$$

The vertical dimension ( $Dist_{vert}$ ) of each image also depends on the distance of the UAV to the surface ( $Dist_H$ ) and the camera opening angle ( $\theta_{vert}$ ). Equation 3 shows this dependency.

$$\tan\left(\frac{\theta_{vert}}{2}\right) = \frac{Dist_{vert}}{2Dist_H} \quad (3)$$

Equation 4 shows that the vertical offset ( $Offset_{vert}$ ) depends on the quantity of waypoints in the vertical mission ( $N_{way-vert}$ ), minimum height ( $h_{min}$ ) and maximum height ( $h_{max}$ ).

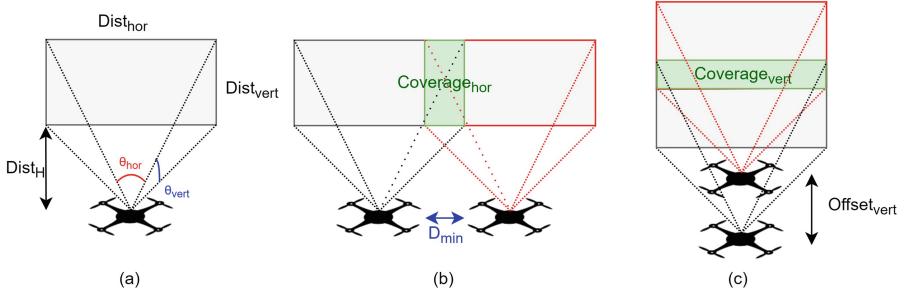
$$Offset_{vert} = \frac{|h_{max} - h_{min}|}{(N_{way-vert} + 1)} \quad (4)$$

The vertical coverage rate ( $Coverage_{vert}$ ) is given by the ratio between  $Dist_{ver}$  and  $Offset_{vert}$  as in Eq. 5.

$$Coverage_{vert} = 1 - \frac{Offset_{vert}}{Dist_{hor}} \quad (5)$$

Figure 2 illustrates the schematic of the problem. It demonstrates all values used in the equations. Figure 2(a) shows the vertical ( $Dist_{vert}$ ) and horizontal ( $Dist_{hor}$ ) distances besides their respective equipment viewing angles( $\theta_{vert}$  and  $\theta_{hor}$ ) along with the surface distance ( $Dist_H$ ). Figures 2(b) and (c) demonstrate the horizontal and vertical covers, where each UAV represents the equipment at different respective times.

The mission time ( $T_{mission}$ ) considers that the UAV hovers for a short period in each waypoint to take pictures and the displacements between them. This time depends on the number of horizontal waypoints ( $N_{way-Hor}$ ), which is related to  $D_{min}$ , the number of vertical waypoints ( $N_{way-vert}$ ), and the speed of the UAV ( $Vel_{UAV}$ ), as presented in Eq. 6. The goal is to make the mission time closest to UAV's battery time with a safety margin (i.e., to return to the start point or land).



**Fig. 2.** Problem schematic.

$$T_{mission} = \frac{Distance_{hor} + (|h_{max} - h_{min}|)N_{way-hor}}{vel_{UAV}} + T_{shot}(N_{way-vert}N_{way-hor}) \quad (6)$$

The parameter  $Distance_{hor}$  is the distance of the course delivered by the convex hull algorithm, while  $N_{way-hor}$  is the number of points with distance  $D_{min}$  between them. The variable  $T_{shot}$  is the time that the UAV hovers to take photos. The two proposed objectives have the following optimization variables: (1) Minimum distance between points ( $D_{min}$ ); (2) distance to surface ( $D_H$ ); (3) number of vertical waypoints ( $N_{way-vert}$ ).

Considering  $D_{min}$ , the coverage objective will tend to decrease this value. However, this decrease generates many  $N_{way-hor}$ , which may increase the mission time. Thus, an increasing in  $D_H$  tends to increase the coverage rate but decreases mission time. In this case, a proximity factor must be added so that the equipment is closer to the surface, respecting a safety margin, and improving the image resolution. In the case of ( $N_{way-vert}$ ), the coverage goal will increase the number of vertical waypoints, increasing the total UAV hover time for capturing photos. These three values will be optimized for the UAV coverage path planning optimization.

## 2.2 Coverage Path Planning Optimization

In the context of mobile robot navigation, the best path is the trajectory that presents the shortest length and traveling time between the start and final positions. Metaheuristic techniques have been satisfactorily applied in a wide range of applications [11]. A sub-group of metaheuristic algorithms employs methods based on biological systems, i.e., computational techniques that are inspired by how problems are solved in nature. For example, the Bat Algorithm (BA), developed by Yang in [17], is inspired by the echolocation behavior of micro-bats. This algorithm has superior accuracy and efficiency compared to Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO) [5]. The problem described in this research work, besides the best path, has a few characteristics about

the surface 3D reconstruction, such as coverage rate and distance to the surface. Considering that the surfaces have convex shapes or closer convex shapes, the Convex Hull algorithm returns a mission with safety, avoiding contact with the explored surface in the aspects of coverage and equipment. This mission was implemented for 3D mapping approach to obtain models of semi-structured environments, such as partially destroyed buildings [14]. The optimization of the task will be performed by using the BA algorithm considering the points delivered by the Convex Hull algorithm.

**Bat Algorithm.** The standard BA has many advantages, and one of the key benefits is that it can provide very fast convergence at a very initial stage by switching from exploration to exploitation. Among the first set of applications of BA, the continuous optimization in the engineering context has been extensively studied, which demonstrates the potential of BA dealing with highly nonlinear problem efficiently [18, 19]. As shown in Eqs. 7, 8 and 9, each bat is associated with a frequency  $f_i$ , velocity  $v_i^t$  and a location  $x_i^t$ , at iteration  $t$ , in non dimensional search or solution space. Among all the bats, there exists a current best solution  $x_*$ . Therefore, the above three rules can be translated into the updating equations for  $x_i^t$  and velocities  $v_i^t$ , where  $\beta$  is a random number of a uniform distribution in  $[0,1]$ .

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \quad (7)$$

$$v_i^t = v_i^{t-1} + (x_* - x_i^{t-1})f_i, \quad (8)$$

$$x_{temp} = x_i^{t-1} + v_i^t \quad (9)$$

Initially, each bat is randomly assigned with a frequency in which is drawn uniformly from  $[f_{min}, f_{max}]$ . For this reason, the BA can be considered as a frequency-tuning algorithm that provides a balanced combination of exploration and exploitation. The loudness and pulse emission rates offer a mechanism for automatic control and auto-zooming into the region that presents reasonable solutions. The bat location is a vector with three positions that can be defined as in Eq. 10.

$$[D_{min}, D_H, N_{way-vert}] \quad (10)$$

where  $D_{min} \in \mathbb{R}^+$ ,  $D_H \in \mathbb{R}^+$  and  $N_{way-vert} \in \mathbb{N}$ . The fitness function of each bat is given by the sum of three Gaussian, as defined in Eqs. 11 and 12.

$$Fitness_{Time} = 1000 * e^{(-(T_{Mission} - T_{Desired})^2)/8)}/8) \quad (11)$$

$$Fitness_{Coverage} = 200 * e^{(-(Coverage - Coverage_{Desired})^2)/0.2)} \quad (12)$$

where  $T_{Desired}$  and  $Coverage_{Desired}$  are the desired mission time and the desired coverage for the mission, respectively. These functions were chosen to approximate the highest fitness points of the desired point. Besides, for times higher than the desired one, they could have problems regarding mission safety, and in this case, the fitness is divided by 10. The values 1000 and 200 were chosen to identify the most critical mission objectives. In this case, they are the

choice of time to avoid accidents with the UAV. The relationship between these values shows the importance to choose the best bat.

For each bat, a value of the sound amplitude is given initially as 1. If the random value is less than the sound rate, i.e.,  $r_i^t$  like Eq. 14, then, the algorithm goes through the local search, where a small perturbation is performed on a temporary bat. This value is linked to the global algorithm search, where the bat is only updated if Personal Fitness is more significant than before or if a random value is less than this amplitude, such as in Eqs. 13, 14 and 15.

$$x_i^t = x_{temp} \quad (13)$$

$$r_i^t = 1 - e^{-\lambda t} \quad (14)$$

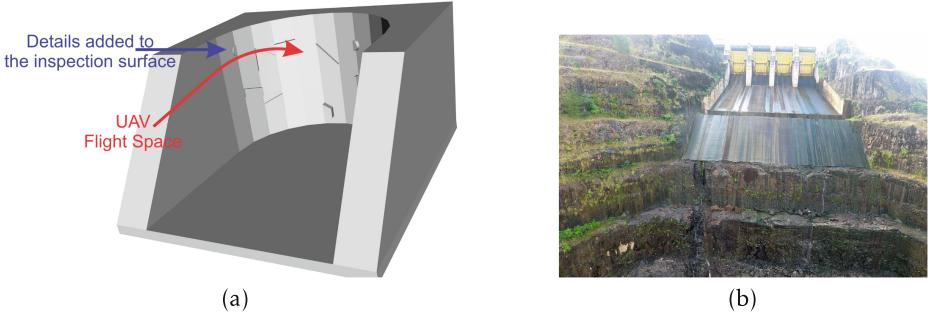
$$A_i^t = \alpha A_i^{t-1} \quad (15)$$

The variables  $\alpha$  and  $\lambda$  are the sound amplitude decrease rate and the increase rate of the pulse emission, respectively. After all the process of local and global search phases of the bats, a new best bat is chosen with the best fitness among them all. The stopping criterion is the algorithm number of iterations.

### 3 Methods, Results and Discussions

The simulation environment was designed in the ROS framework using the software Gazebo to show the proposed method effectiveness. All the code was implemented using the programming language Python. A semi-circular was designed to be the inspected structure, once it requires a more complex 3D movement to be performed by the aircraft. Figure 3(a) and (b) show the structure created in simulation and its real representation, respectively. Note that this structure also mimics the external surface of a dam. A few extrusions were added to the structure surface to allow the comparison among the inspections and the original model.

The UAV used in the simulation is the Hector Quadrotor, proposed by [12]. For the sensor, it was used a  $640 \times 480$  pixels camera pointed in the front direction of the UAV. In this sense, the UAV heading is the same as the camera orientation. Two steps were performed to show the algorithm effectiveness. In the first step, the structure was manually inspected using an XBOX controller. This manual inspection was performed by the operator to capture the relevant details of the structure. After this step, the path was optimized using the proposed methodology. Then, the algorithm uses this path to perform a second inspection in the structure, where the trajectory is used as waypoints to control the UAV. A 3D reconstruction using COLMAP library [15] was applied to both sets of images verifying the equivalence of the 3D reconstructions. The effectiveness of the reconstruction was verified in a real environment. For this, a Phantom 4 quadcopter was used [4]. The first inspection was performed manually by an operator with the objective of a 3D reconstruction, which is used as a basis for a comparison to allow the analysis of the deformations in the structure over time. After that, the path was optimized using the proposed method. The structure



**Fig. 3.** Inspected structure. (a) Simulation (b) Real environment

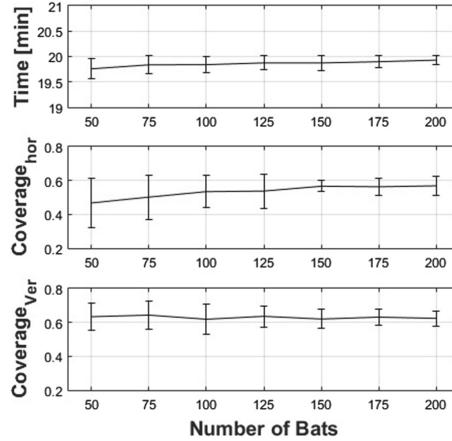
was inspected again using the optimized waypoints and control software in a smartphone.

As previously explained, the first part of the experiment is the manual inspection of the area. It is possible to note that the path in this first inspection is not very stable, or efficient, which can be explained by the dependence on the operator ability to perform spacial location. After the manual flight, the methodology uses the optimization algorithm. The algorithm performance was evaluated varying the bat population. For this test, it was performed 50 optimization exercises for each number of bats to compute the mean and the standard deviation, as shown in Fig. 4. The initial bat population ranges from 0 to 20 randomly for each execution. Table 1 gives the values of the simulation constants. The parameter  $T_{desired}$  was chosen to maintain the flight safely, once the battery time is around 25 min (1500 s). Besides, the coverage was selected to have a balance between the image reconstruction and not to prevent the flight time, since very high coverage will lead to many points with redundancy and the aircraft displacements would be minimal.

**Table 1.** Simulation constants values.

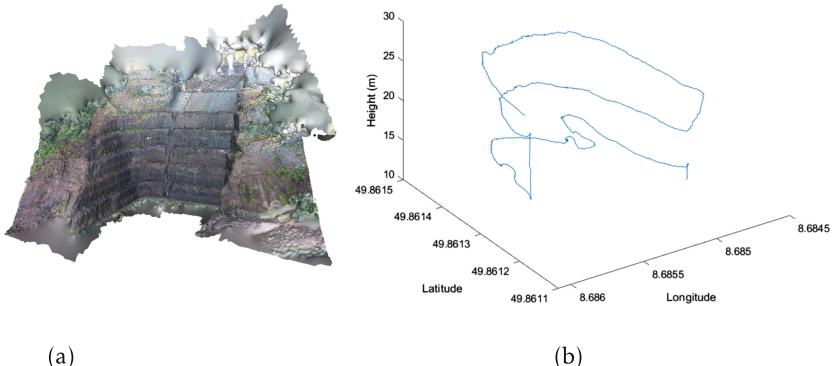
Variable	Value	Variable	Value	Variable	Value
$\alpha$	0.8	$T_{desired}$	1200 s	$h_{min}$	0.5 m
$\lambda$	0.01	$Coverage_{desired}$	0.6	$h_{max}$	25 m
$\theta_{vert}$	50°	$\theta_{hor}$	120°	$Vel_{UAV}$	3 m/s

Figure 4 presents the algorithm performance considering three objectives: time, vertical, and horizontal coverages. Note that the accuracy of the responses increases when the number of bats is high, as can be seen with decreasing standard deviation. Thus, although they are conflicting objectives, the BA behaves well in the search for solutions space for this problem. The optimized trajectory was performed using the Gazebo simulator.

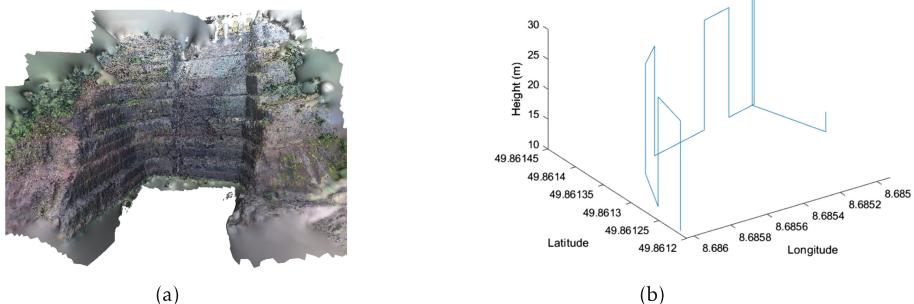


**Fig. 4.** Bat algorithm performance varying the number of bats.

After the simulation results, the next step is an inspection using a real UAV. Figure 5(a) and (b) present the 3D reconstruction and the path, respectively. The path performed is not optimized because the trajectory uses the operator experience instead of any efficiency parameter. Figure 6(b) shows the optimized path for the inspection, which is a shorter path, but similar enough. The resultant reconstruction is in Fig. 6(a). Note that the detail level is similar enough. A drawback for the manual inspection, is the dependability on the operator prior knowledge, i.e., its experience and ability to control the UAV. Besides, the operator can pass points of interest for the mission, causing issues for the 3D reconstruction. Another big challenge for manual inspection is to make the mission reproducible. An advantage of an optimized mission is the guarantee of enough space among the waypoints, the desired coverage characteristics, and the guarantee of mission reproducibility.



**Fig. 5.** Manual inspection. (a) Inspection reconstruction. (b) Trajectory.



**Fig. 6.** Optimized inspection. (a) Inspection reconstruction. (b) Trajectory.

## 4 Conclusions and Future Work

This research proposed a methodology to maximize coverage inspection trajectories that are generated manually. The manual flights are performed by an operator that covers relevant parts of the inspection. The proposed method optimizes this path to ensure optimal energy usage while also ensuring minimal coverage and adequate flight speed in offline mode. The results proved that the algorithm optimizes trajectories manually acquired in simulated environments. The use of simulation software is indispensable for mission reliability and can explore aspects that mission in equipment could generate losses, such as surface approach and UAV speed. A few extensions are foreseen for future works. First, the proposed methodology will be tested in different real environments. Besides, it is also intended to expand the number of cases and obstacles. Other improvements are also expected to improve the algorithm, such as the consideration of energy consumption related to the different paths.

**Acknowledgment.** We would like to thank the following Brazilian Agencies UFJF, CAPES, CNPq, INCT- INERGE, ANEEL P & D Program (CPFL Energia) and INESC Brazil.

## References

1. Bo, X., Liping, C., Yu, T., Min, X.: College of engineering, china agricultural university; Beijing key laboratory of intelligent equipment technology for agriculture; Beijing research center of intelligent equipment for agriculture;; route planning algorithm and verification based on UAV operation path angle in irregular area. 11 (2015)
2. Buffi, G., Manciola, P., Grassi, S., Barberini, M., Gambi, A.: Survey of the ridracoli dam: UAV-based photogrammetry and traditional topographic techniques in the inspection of vertical structures. Geomatics Nat. Hazards Risk 8(2), 1562–1579 (2017)
3. Di Franco, C., Buttazzo, G.: Energy-aware coverage path planning of UAVs. In: 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, pp. 111–117. IEEE (2015)

4. DJI: Phantom 4 quadcopter. <https://www.dji.com/br/phantom-4> (2019). Accessed 06 Sept 2019
5. Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), vol. 1, pp. 81–86 (2001)
6. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. *Rob. Auton. Syst.* **61**(12), 1258–1276 (2013)
7. Hayat, S., Yanmaz, E., Brown, T.X., Bettstetter, C.: Multi-objective UAV path planning for search and rescue. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 5569–5574. IEEE (2017)
8. Jacob, R., Brodal, G.S.: Dynamic planar convex hull. arXiv preprint [arXiv:1902.11169](https://arxiv.org/abs/1902.11169) (2019)
9. Khaloo, A., Lattanzi, D., Jachimowicz, A., Devaney, C.: Utilizing UAV and 3d computer vision for visual inspection of a large gravity dam. *Front. Built Environ.* **4**, 31 (2018)
10. Kim, D.H., Hoang, G., Bae, M., Kim, J.W., Yoon, S.M., Yeo, T., Sup, H., Kim, S.: Path tracking control coverage of a mining robot based on exhaustive path planning with exact cell decomposition. In: 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), pp. 730–735, October 2014
11. Kostiv, O., Demydov, I., Makarenko, A., Tverdohlib, M.: Metaheuristics in cloud platform traffic engineering. In: 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), pp. 533–537, February 2018
12. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., von Stryk, O.: Comprehensive simulation of quadrotor uavs using ros and gazebo. In: 3rd International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR) (2012)
13. Miao, X., Lee, J., Kang, B.: Scalable coverage path planning for cleaning robots using rectangular map decomposition on large environments. *IEEE Access* **6**, 38200–38215 (2018)
14. de la Puente, P., Rodriguez-Losada, D., Valero, A., Matia, F.: 3d feature based mapping towards mobile robots' enhanced performance in rescue missions. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1138–1143. IEEE (2009)
15. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104–4113 (2016)
16. Song, Y.S., Arshad, M.R.: Coverage path planning for underwater pole inspection using an autonomous underwater vehicle. In: 2016 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), pp. 230–235, October 2016
17. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)*, Studies in Computational Intelligence (SCI), vol. 284, pp. 65–74. Springer, Berlin, Germany (2010). chap. 6
18. Yang, X.S., Hossein Gandomi, A.: Bat algorithm: a novel approach for global engineering optimization. *Eng. Comput.* **29**(5), 464–483 (2012)
19. Yang, X.S., Karamanoglu, M., Fong, S.: Bat algorithm for topology optimization in microelectronic applications. In: The First International Conference on Future Generation Communication Technologies, pp. 150–155. IEEE (2012)



# Evaluating an AEF Swimming Microrobot Using a Hardware-in-the-loop Testbed

José Emilio Traver<sup>1(✉)</sup>, Inés Tejado<sup>1</sup>, Cristina Nuevo-Gallardo<sup>1</sup>,  
Javier Prieto-Arranz<sup>1,2</sup>, Miguel A. López<sup>1</sup>, and Blas M. Vinagre<sup>1</sup>

<sup>1</sup> Industrial Engineering School, University of Extremadura, 06006 Badajoz, Spain  
`{jetraverb,itejbal,cnuevog,malopezr01,bvinagre}@unex.es`

<sup>2</sup> School of Industrial Engineering, University of Castilla-La Mancha,  
13071 Ciudad Real, Spain

`Javier.PrietoArranz@uclm.es`

**Abstract.** This paper studies the swimming and control effectiveness of a 4-link artificial eukaryotic flagellum (AEF) swimming microrobot through hardware-in-the-loop (HIL) experiments, which are executed in an environment characterized by high mechanical stress. The tested HIL experiment consists of a simulator of the robot, developed in the MATLAB/ Simulink environment, and a microcontroller Atmel ATmega32u4, where the control of the robot is programmed. Data exchange between the simulator and microcontroller is carried out through serial protocol via universal asynchronous receiver-transmitter (UART). For comparison purposes, two control strategies, namely fractional order proportional-derivative ( $PD^\mu$ ) and integer order proportional-integral-derivative (PID) controllers, are considered for the robot to emulate a non-reciprocal motion. Two types of these controllers are implemented and evaluated.

**Keywords:** Control · HIL · Microswimmer · Robot · Simscape · Simulink · Non-reciprocal motion

## 1 Introduction

Technology advances at micro and nanoscale are promoting recent research in the field of robotics. For example, microrobots are receiving much attention in the last years due to the access to small spaces down to the micro scale, such as inside the human body by making existing therapeutic and diagnostic procedures less invasive [10, 16], and the manipulation and interaction with tiny entities of the environment [4–6, 17, 22].

However, working at micro and nanoscale involves a great number of challenges, among which is worth noting the investigation of new fabrication techniques, ways of propulsion, supply power, and control. Likewise, the principles governing the design of such scale robots rely on the understanding of microscale hydrodynamics, which implies to navigate into spaces at low Reynolds

number ( $Re$ ), i.e., environments dominated by viscous forces; thus, conventional actuation principles do not work. A review of the theoretical framework for locomotion at low  $Re$  can be found in [12, 21].

The analysis of the swimming geometry, as well as ways of generating non-reciprocal motion, have been extensively addressed in the literature from different perspectives [2]. Currently, the main tends are based on the generation of planar waves by means of an eukaryotic flagellum, which can be approached with several methods: (1) by distributed actuation [1, 11, 15, 25], (2) by two-point actuation [9, 18], and (3) by single-point actuation with absorption of the reflected wave [9, 18] or with a non-uniform distribution of mass [14]. Nevertheless, the control strategies applied to the flagellum actuators for the generation of an adequate non-reciprocal motion have been left aside.

In our previous works [24, 25], a comparative study was performed to evaluate how different motion waveforms and control strategies improve locomotion of a 4-link swimming artificial eukaryotic flagellum (AEF) microrobot at low  $Re$  environments, using fractional and integer definitions for waveforms and control. Before testing the control on a microrobot prototype in a real environment, a hardware-in-the-loop (HIL) testbed is built in this paper based on the simulator of the AEF microrobot developed in the MATLAB/Simulink environment [25]. The objective is to validate the control strategies proposed in [24] in a more realistic way rather than only in simulation, and this will allow us to find out possible bugs on digital implementation and the limitations of control system hardware. This testbed will also assess the swimming performance of the microrobot for compliance with the specifications under all possible circumstances, avoiding all the drawbacks that involve the use of a real prototype, such as costs, time-consuming implementation, and certain inflexibility [23].

The remainder of the paper is organized as follows. Section 2 describes the characteristics of the environment and different kinds of propulsion waveforms for swimming robots. Section 3 gives the details of the HIL testbed built for the 4-link swimming robot. Section 4 addresses the control of the robot required to emulate a non-reciprocal motion within low  $Re$  environments. Experimental and simulation results are given and discussed in Sect. 5. Finally, conclusions and future works are drawn in Sect. 6.

## 2 Background

This section describes some details of the hydrodynamics of swimming robots within low  $Re$  environments and other considerations needed to understand the effects of microscopic scale at macroscopic level. It also describes the kinds of motions that allow swimming in such environments.

### 2.1 Hydrodynamics

Hydrodynamics of macroscopic swimming robots is governed by the Navier-Stokes equation. For an incompressible fluid, it is defined by the following

expression:

$$\frac{\partial V}{\partial t} + (V \cdot \nabla) V - \frac{\mu^*}{\rho^*} \nabla^2 V = -\frac{1}{\rho^*} \nabla P + F \quad (1)$$

where  $t$  is time,  $\rho^*$  and  $\mu^*$  are fluid's density and dynamic viscosity, respectively,  $V$  is the velocity vector,  $P$  is pressure, and  $F$  represents other external forces. This equation is only suitable for regimes with high  $Re$ , which means that the inertial terms are far higher than viscous terms. This indicates that robots, within such environments, are based on inertia to swim.

However, viscous forces predominate over inertial at small scale (from milli to micrometers), which results in low  $Re$  and, thus, swimming robots experiment different hydrodynamics. For this regime, the two first terms on the left of (1) are neglected, and it is reduced to Stokes' equation. The hydrodynamics defined by Stokes' equation indicates that the change of velocities does not produce a propulsion motion. Therefore, a non-reciprocal and irreversible in time motion is needed to swim [16, 20].

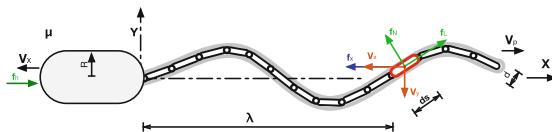
To obtain a low  $Re$  behavior at macroscopic scale, all variables involved have to be scaled through this dimensionless parameter. Taking into account that  $Re$  can be mathematically expressed as:

$$Re = \frac{\rho^* V L}{\mu^*}, \quad (2)$$

setting the size of the robot ( $L$ ), a fluid with the proper  $\rho^*/\mu^*$  relation at the expected forward speed ( $V$ ) must be used.

## 2.2 Waveforms for Propulsion

Propulsion of a robot operating in a low  $Re$  environment achieves a great importance because only a non-reciprocal motion allows the propulsion in such environment.



**Fig. 1.** Free body diagram of planar artificial eukaryotic flagella.

Propulsion of an undulatory system is due to the fact that normal forces generated during the propelled motion along the flagella compensate all tangential forces [7, 8]. Figure 1 shows the main variables involved in that motion and the parameters that define waveforms, in which  $y$  is the displacement along transverse axis,  $x$  is the displacement along main axis,  $f$  is the frequency,  $V_x$  and  $V_y$  are the forward and transverse speeds, respectively, of an infinitesimal element

of flagellum ( $ds$ ), and  $f_N$ ,  $f_L$ , and  $f_h$  are the reaction forces of motion for the considered differential segment in accordance with Stokes' law. The propulsion thrust,  $f_x$ , can be analyzed by considering the reaction forces on  $ds$  as follows [7]:

$$f_x = (f_N \sin \theta - f_L \cos \theta)ds \quad (3)$$

where  $\theta$  is the angle of the infinitesimal element. In addition, the resultant thrust over a complete cycle is equal to the drag force. Thus, the forward speed reached by the wave can be calculated from the relation  $nF_x = 6\pi\mu\frac{R}{2}V_x$  where  $R$  is the diameter of head,  $n$  is the number of simultaneous waves along the flagellum and  $F_x$  is the total propulsion thrust.

From a nature perspective, different kinds of propulsion waveforms can be observed in biological swimmers. The first kind is the planar wave associated to the flagella of eukaryotic cells, which is described by a traveling harmonic wave [7]. Likewise, Carangiform fishes bend their bodies describing linear and quadratic traveling waves that extend from their head to their tail end [13]. These three waveforms can be described by:

$$y(x, t) = (c_0 + c_1x + c_2x^2) \sin\left(\frac{2\pi}{\lambda}(x - V_p t)\right), \quad (4)$$

where  $c_0$ ,  $c_1$  and  $c_2$  are the coefficients that govern the amplitude growing,  $V_p$  denotes the propagation speed of the wave, and  $\lambda$  is the wavelength. For harmonic waves, the amplitude is only given by coefficient  $c_0$  (i.e.,  $c_1 = c_2 = 0$ ), whereas  $c_0 = 0$  for Carangiform swimmers. Furthermore, two important properties are preserved during propulsion for the latter kinds of waveforms: (1) the flagellum head is always maintained at zero amplitude (boundary condition, i.e.,  $y(0, t) = 0$ ), and (2) the wave amplitude along the flagellum can be modulated.

An alternative way for preserving such properties, which has been demonstrated to improve the forward propulsion velocity [25], is the use of a fractional growing power for variable  $x$  as follows

$$y(x, t) = (cx^\alpha) \sin\left(\frac{2\pi}{\lambda}(x - V_p t)\right), \quad (5)$$

where  $c$  defines the amplitude at end of flagellum, and  $\alpha \in \mathbb{R}^+$  ( $0 < \alpha < 1$ ) is the fractional order coefficient of the wave, whose value determines the way of growing of the waveform amplitude. This waveform can be also understood as a generalization of the previous ones, merging the features from planar to Carangiform waves depending on the value of  $\alpha$  [25].

### 3 Testbed of the Robot

This section describes the testbed designed to perform HIL experiments for a 4-link swimming robot able to swim within low  $Re$  environments.



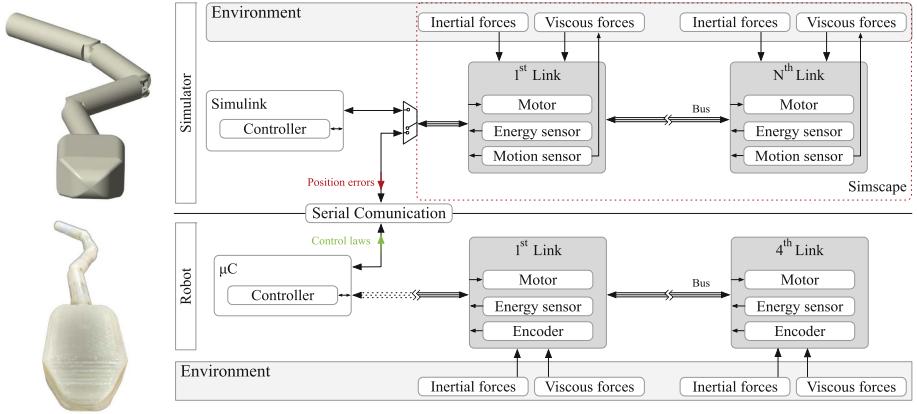
**Fig. 2.** Illustration of the 4-link AEF swimming robot.

Figure 2 illustrates the 4-link AEF swimming robot under study. Based on the flagella of eukaryotic cells and snakes, the robot consists of four cylindrical segments, actuated by motors, and a triangular head. Each segment is a 1-DOF joint that is driven by a Maxon DCX 08M motor with planetary gearbox GPX08 of 64:1 ratio. The output shaft transfers rotation through a miter gear, allowing a rotation up to  $\pm 85^\circ$  perpendicular to the displacement axis of the flagellum. Regarding the frame, it is made of thermoplastic aliphatic polyester; the electronics is distributed inside it along the flagellum to minimize robot volume. It must be remarked that, although this first design is limited to four links, the design is scalable and new links can be added or removed as convenient.

Figure 3 shows a scheme of the robot testbed. As can be seen, it consists of:

- The simulator of the robot presented in [25], which is a configurable N-link swimming robot simulator developed with the Simscape toolbox within the MATLAB/Simulink environment. Likewise, it also allows to simulate the interaction between the robot and the environment where it swims. The geometry of the robot (namely, size, number and length of links in which its flagellum is divided to), as well as the properties of the actuators, are adequately selected in accordance with the robot description given above (see Table 1). This simulator runs on a Macintosh with the following specifications: Intel Core i7 (six core) with a clock speed of 2.2 GHz and 16 Gb DDR4 RAM.
- The microcontroller Atmel ATmega32u4, where the controllers will be implemented.

The data interchange connection between the microcontroller and the PC is done by means of micro USB type B through a serial protocol via universal asynchronous receiver-transmitter (UART) configured as follows: BigEndian byte order, 32-bit data, IEEE Standard 754 floating point for data representation, and 250000 baud rate. On the PC side, an *S-Function* was written to receive/send data. As shown in Fig. 3, the simulator sends the position error of each link to the microcontroller, whereas this calculates and sends back the control laws to the PC.



**Fig. 3.** Connection scheme of the robot testbed to perform HIL experiments.

**Table 1.** Actuator and environment parameters for the robot simulator.

Parameter	Value	Description	Parameter	Value	Description
$c_0$	$5.59 \times 10^{-2}$	Amplitude coefficient (m)	$\alpha$	0.2	Fractional coefficient
$f$	0.5	Wave frequency (Hz)	$\lambda$	$24.86 \times 10^{-2}$	Wavelength (m)
$n$	1	Simultaneous waves in flagellum	$N$	4	Number of joints
$R_m$	12.3	Armature resistance ( $\Omega$ )	$L_m$	$41.1 \times 10^{-6}$	Armature inductance (H)
$K_{em}$	0.0034	Back-emf constant (V/(rad/s))	$J_m$	$3.79 \times 10^{-9}$	Rotor inertia ( $\text{kgm}^2$ )
$B_m$	$9.09 \times 10^{-7}$	Rotor damping (Nm/(rad/s))	$ng$	64	Gear ratio
$J_g$	$4 \times 10^{-10}$	Gear head inertia	$fg$	0.73	Maximum efficiency
$\rho$	809.13	Density of solid segment ( $\text{Kg/m}^3$ )	$d$	$7.5 \times 10^{-3}$	Radius of flagellum (m)
$L_s$	$\lambda/N$	Total length of swimming robot (m)	$\mu^*$	100	Viscosity of environment (cSt)
$\rho^*$	964	Density of environment ( $\text{kg/m}^3$ )			

## 4 Control of the Robot

This section addresses the control required for the robot to emulate a non-reciprocal motion within low  $Re$  environments. Firstly, the procedure established to split the desired waveform (reference) for each actuator is explained. Secondly, the control strategies are designed, and finally, details of controllers implementation are given.

## 4.1 Discretization of Waveforms

The emulation of the desired waveform is carried out dividing such a waveform into the same number of segments that the robot has, taking as criterion that the projection on the propulsion axis is fixed and equal to the length of the segments. Therefore, the segment angle at each sampling time can be calculated by:

$$\theta_{ij} = \tan^{-1} \left( \frac{y_{ij} - y_{i(j-1)}}{x_{ij} - x_{i(j-1)}} \right), \quad (6)$$

where  $i \in [0, T]$  is the instant of discretization ( $T$  is the final simulation time),  $j \in [0, N]$  means the considered segment,  $(x_i, y_i) = y(t, x)$ , and  $N$  is the total number of robot links. More details of this procedure can be found in [25].

## 4.2 Strategies

In order to guarantee a non-reciprocal motion, the robot actuators have to follow the desired position with the same dynamics and error.

In our previous work [24], fractional proportional-derivative (henceforth referred to as  $\text{PD}^\mu$ ) and proportional-integral-derivative (PID) controllers were designed in continuous time using three tuning methods: (a) minimizing the integral time absolute error (ITAE), to be able to evaluate the performance of the system along the time; (b) minimizing the integral absolute error (IAE), to quantify the performance to be sensitive at low errors [3]; and (c) considering robustness to time constant variations with a desired phase margin criteria. Among them, the controllers considered next allow to obtain the best system performance namely those tuned for robustness.

The structure of the fractional controller was given by

$$\text{PD}^\mu(s) = K_p + K_d s^\mu, \quad (7)$$

where  $K_p$  and  $K_d$  are the proportional and derivative gains, respectively, and  $\mu \in (0, 2]$  is the fractional order. For the PID, the parallel form was used, i.e.,

$$\text{PID}(s) = K_p + \frac{K_i}{s} + K_d s, \quad (8)$$

where  $K_p$ ,  $K_i$  and  $K_d$  are the proportional, integral and derivative gains, respectively. The parameters of both integer and fractional order controllers obtained after tuning are given in Table 2.

**Table 2.** Parameters controllers.

Controller	$K_p$	$K_i$	$K_d$	$\mu$
PID	466.69	25.56	3.43	—
$\text{PD}^\mu$	382.32	—	24.51	0.90

### 4.3 Implementation of the Controllers

To implement the above controllers in the hardware of the robot testbed, they must be discretized. For that purpose, the Tustin's method is used with a sampling time of 1 ms. On the other side, in the  $\text{PD}^\mu$  controller, the fractional operator was previously approximated by the Oustaloup method with four poles and four zeros in the frequency range [0.8, 20] rad/s. The discrete models for PID and  $\text{PD}^\mu$  can be written in the z-domain as:

$$C(z) = \frac{\sum_{i=0}^M K_i z^{-i}}{1 + \sum_{i=0}^N L_i z^{-i}} \quad (9)$$

where  $N$  and  $M$  correspond to the number of poles and zeros of the controller, respectively, and the  $K_i$  and  $L_i$  are coefficients that depend on the controller parameters and the sampling time.

With the aim to evaluate the results regardless of the implementation method applied, two digital realizations of the discretized controllers are considered next, namely, direct form II and parallel form. Contrary to what happens with the direct form II, the parallel-form realization reduces the sensitivity to its coefficients due to the use of partial-fraction expansion of the discrete transfer function of the controller [19].

The direct-form-II realization describes the control law as:

$$u(n) = \sum_{i=0}^M K_i w(n-i) \quad (10)$$

with

$$w(n) = e(n) - \sum_{i=1}^N L_i w(n-i)$$

where  $e(n)$  is the error signal. With respect to the parallel form, the control law is given by:

$$u(n) = Ce(n) + \sum_{i=1}^N A_i e(n) + B_i u_i(n-1) \quad (11)$$

where  $A_i$  and  $B_i$  are the coefficients of the partial-fraction expansion, assuming that all the poles are real,  $C$  is a constant, and  $u_i$  is the output of the  $i^{th}$  partial-fraction.

The parameters of the discretized controllers obtained for both realizations are given in Table 3. It is worth to remark that the derivative part of the PID was implemented with the low-pass filter  $F(s) = 50/(s + 50)$ .

**Table 3.** Parameters of the discretized controllers.

Direct-form-II	$K_0$	$K_1$	$K_2$	$K_3$	$K_4$	$L_1$	$L_2$	$L_3$	$L_4$
PID	634.01	$-1.24 \times 10^3$	611.22	—	—	-1.95	0.95	—	—
$PD^\mu$	751.74	$-2.98 \times 10^3$	$4.45 \times 10^3$	$-2.95 \times 10^3$	733.25	-3.96	5.90	-3.90	0.96
Parallel form	$A_1$	$A_2$	$A_3$	$A_4$	$C$	$B_1$	$B_2$	$B_3$	$B_4$
PID	-8.58	$25.55 \times 10^{-3}$	—	—	642.57	-0.95	-1.00	—	—
$PD^\mu$	-6.27	$-165.15 \times 10^{-3}$	$-22.99 \times 10^{-3}$	$-3.12 \times 10^{-3}$	758.21	$-9.80 \times 10^{-2}$	$-9.91 \times 10^{-2}$	$-9.96 \times 10^{-2}$	$-9.98 \times 10^{-2}$

## 5 Experiments

This section discusses the results obtained when applying the above-mentioned control strategies to the swimming robot using the described HIL testbed, and considering different realizations of the  $PD^\mu$  and PID controllers. The experimental results will be compared with the obtained using only the robot simulator. Likewise, results for continuous controllers will be also shown. The performance of the controllers is analyzed in terms of the ratio between the mechanical power developed by the links and the electrical power supplied to actuators as follows:

$$\eta = \frac{P_m(t)}{P_e(t)} = \frac{F_x(t)V_x(t)}{V(t)I(t)}, \quad (12)$$

where  $V_x$  is the generated forward speed as result of the forces caused by non-reciprocal motion in a viscous medium,  $F_x$ ,  $I$  refers to the current demanded by the motor, and  $V$  is the voltage applied to the actuator.

The results obtained for the two implementations of the controllers are summarized in Table 4, where the IAE and ITAE obtained for each actuator, as well as the mean electrical, mechanical power and the efficiency (calculated according to (12)) are given. For comparison purposes, the results obtained in simulation considering the controllers in both continuous and discrete time are also included.

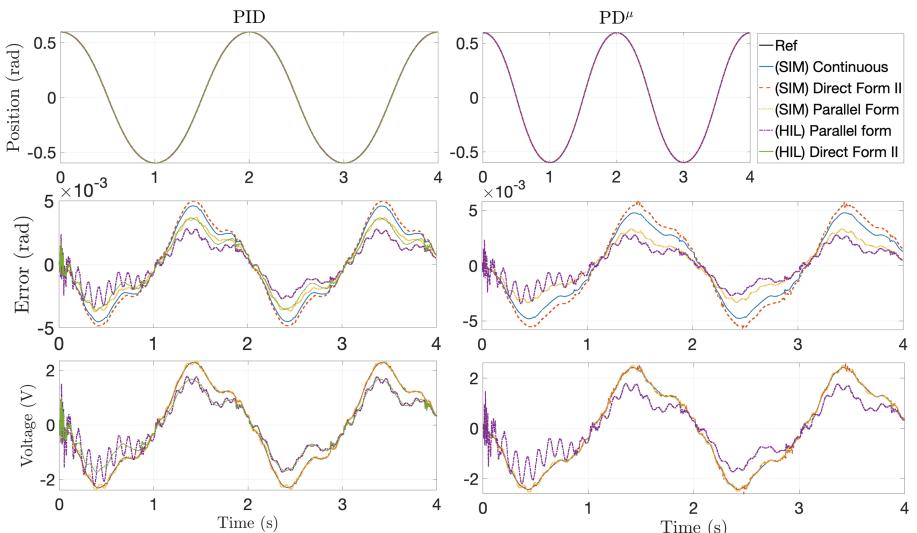
**Table 4.** Performance of the 4-link robot when applying the discretized controllers.

Controller	1 <sup>st</sup> Link		2 <sup>nd</sup> Link		3 <sup>rd</sup> Link		4 <sup>th</sup> Link		$P_e$ (W)	$P_m$ ( $\mu$ W)	$\eta$ (%)
	ITAE ( $\times 10^{-3}$ )	IAE ( $\times 10^{-3}$ )	ITAE ( $\times 10^{-3}$ )	IAE ( $\times 10^{-3}$ )	ITAE ( $\times 10^{-3}$ )	IAE ( $\times 10^{-3}$ )	ITAE ( $\times 10^{-3}$ )	IAE ( $\times 10^{-3}$ )			
Continuous	15.58	7.67	33.23	17.38	33.95	16.84	35.23	17.63	0.79	3.03	$3.77 \times 10^{-4}$
PID Discrete - Direct	17.32	8.52	36.81	19.23	37.77	18.71	39.32	19.62	0.79	3.03	$3.75 \times 10^{-4}$
Discrete - Parallel	11.38	5.65	24.15	12.70	24.79	12.33	25.57	12.88	0.79	3.03	$3.75 \times 10^{-4}$
HIL - Direct	10.73	5.28	23.21	12.11	22.99	11.46	23.84	11.96	1.85	3.03	$1.62 \times 10^{-4}$
HIL - Parallel	6.90	3.52	14.36	7.55	14.16	7.21	14.47	7.50	1.85	3.03	$1.62 \times 10^{-4}$
Continuous	17.57	8.59	37.06	19.27	37.08	18.32	39.39	19.54	0.60	3.03	$4.93 \times 10^{-4}$
$PD^\mu$ Discrete - Direct	21.58	10.52	45.24	23.46	45.38	22.36	48.62	24.01	0.61	3.03	$4.88 \times 10^{-4}$
Discrete - Parallel	9.65	4.79	20.44	10.75	20.96	10.43	21.63	10.90	0.61	3.03	$4.88 \times 10^{-4}$
HIL - Parallel	6.90	3.52	14.35	7.55	14.16	7.21	14.27	7.59	1.85	3.03	$1.62 \times 10^{-4}$

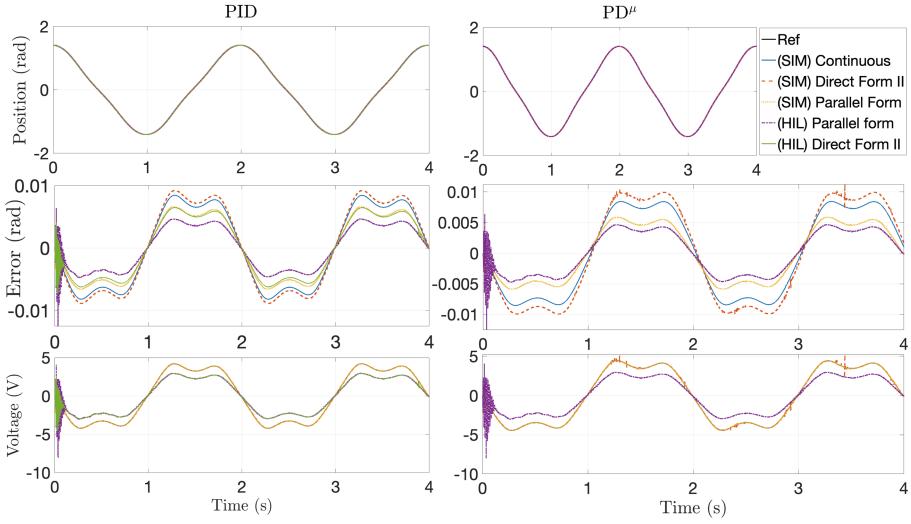
The first conclusion that can be drawn is that the designed controllers are validated: the differences between experimental (with HIL) and simulated results can be considered as acceptable, except of the case of  $\text{PD}^\mu$  implementation in direct form, which provides an unstable control (results are omitted). Independently of the controller realization, another issue that can be concluded is that, taking into account the locomotion performance given by (12), the propulsion of the robot is better when applying the  $\text{PD}^\mu$  controller. As it is evident, the electrical power required by the motors in the HIL experiments raises, and thus, a reduction of the control efficiency is obtained in comparison with simulations. Likewise, if we do an analysis with respect to the segments, it can be observed that the behavior of the controllers deteriorates farther away from the robot head. And with respect to implementations, the results are quite similar.

All these conclusions can be also stated from the plots shown in Figs. 4 and 5, where a comparison of the position, tracking error and voltage applied to the motor of the first and fourth links are also illustrated for both controllers. In what position is concerned, it can be observed that only very slight differences can be found between experimental and simulated results. However, the differences are already visible for the tracking error and the voltage signal, for both the first and fourth link.

To sum up, all these given results prove that the discretization and implementation methods of the controllers, as well as the designed controllers themselves and the hardware used, are suitable for this application.



**Fig. 4.** Tracking, error and control law of first link for different controller realizations when applying the PID (left) and the  $\text{PD}^\mu$  (right).



**Fig. 5.** Tracking, error and control law of fourth link for different controller realizations when applying the PID (left) and the  $\text{PD}^\mu$  (right).

## 6 Conclusions

This paper has presented a hardware-in-the-loop (HIL) testbed for a 4-link artificial eukaryotic flagellum (AEF) swimming microrobot, as a previous step to the fabrication and control of the robot. The testbed consisted of a simulator of the robot, developed in the MATLAB/Simulink environment, and a microcontroller Atmel ATmega32u4, where the controllers of the robot were programmed. Data exchange between the simulator and microcontroller was carried out through serial protocol via universal asynchronous receiver-transmitter (UART). Fractional order proportional-derivative ( $\text{PD}^\mu$ ) and integer order proportional-integral-derivative (PID) controllers were designed for the robot to emulate a non-reciprocal motion within low Reynolds number ( $Re$ ) environments. Two realizations of these controllers were implemented in order to study the effectiveness of the swimming mode and validate the control on the real hardware. The results showed that both the control strategies and the hardware were suitable for this application.

Our future works will go towards building the robot and testing locomotion within a cardiovascular system.

**Acknowledgments.** This work has been supported in part by the Spanish Agencia Estatal de Investigación (AEI) under the project DPI2016-80547-R (Ministerio de Economía y Competitividad), in part by the Consejería de Economía e Infraestructuras (Junta de Extremadura) under the grant “Ayuda a Grupos de Investigación de Extremadura” (no. GR18159) and the project IB18109, and in part by the European Social Fund (FEDER, EU) and the European Regional Development Fund “A way to

make Europe". José Emilio Traver would like to thank the Ministerio de Educación, Cultura y Deporte its support through the scholarship no. FPU16/2045 of the FPU Program. Cristina Nuevo-Gallardo would like to thank University of Extremadura its support through the scholarship "Plan Propio de Iniciación a la Investigación, Desarrollo Tecnológico e Innovación 2018".

## References

1. Abadi, A., Kosa, G.: Piezoelectric beam for intrabody propulsion controlled by embedded sensing. *IEEE/ASME Trans. Mechatron.* **21**(3), 1528–1539 (2016)
2. Alouges, F., DeSimone, A., Giraldi, L., Zoppello, M.: Self-propulsion of slender micro-swimmers by curvature control: N-link swimmers. *Int. J. Non-Linear Mech.* **56**, 132–141 (2013)
3. Arrieta, O., Vilanova, R.: Simple PID tuning rules with guaranteed  $M_s$  robustness achievement. *IFAC Proc. Volumes* **44**(1), 12042–12047 (2011)
4. Bogue, R.: Miniature and microrobots: a review of recent developments. *Ind. Rob. Int. J.* **42**(2), 98–102 (2015)
5. Ceylan, H., Giltinan, J., Kozielski, K., Sitti, M.: Mobile microrobots for bioengineering applications. *Lab Chip* **17**(10), 1705–1724 (2017)
6. Diller, E., Sitti, M.: Micro-scale mobile robotics. *Found. Trends® Rob.* **2**(3), 143–259 (2013)
7. Gray, J., Hancock, G.J.: The propulsion of sea-urchin spermatozoa. *J. Exp. Biol.* **32**(4), 802–814 (1955)
8. Hancock, G.J.: The self-propulsion of microscopic organisms through liquids. *Proc. R. Soc. London A Mathe. Phys. Eng. Sci.* **217**(1128), 96–121 (1953)
9. Hariri, H., Bernard, Y., Razek, A.: A traveling wave piezoelectric beam robot. *Smart Mater. Struct.* **23**(2), 025013 (2013)
10. Hunter, I.W., Doukoglou, T.D., Lafontaine, S.R., Charette, P.G., Jones, L.A., Sagar, M.A., Mallinson, G.D., Hunter, P.J.: A teleoperated microsurgical robot and associated virtual environment for eye surgery. *Presence Teleoper. Virtual Environ.* **2**(4), 265–280 (1993)
11. Kósa, G., Jakab, P., Hata, N., Jólesz, F., Neubach, Z., Shoham, M., Zaoroor, M., Székely, G.: Flagellar swimming for medical micro robots: theory, experiments and application. In: Proceedings of the 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2008), pp. 258–263 (2008)
12. Lauga, E.: Life at high Deborah number. *EPL (Europhys. Lett.)* **86**(6), 64001 (2009)
13. Lighthill, M.J.: Note on the swimming of slender fish. *J. Fluid Mech.* **9**, 305–317 (1960)
14. López, M.A., Prieto, J., Traver, J.E., Tejado, I., Vinagre, B.M., Petrás, I.: Testing non reciprocal motion of a swimming flexible small robot with single actuation. In: Proceedings of the 19th International Carpathian Control Conference, pp. 312–317 (2018)
15. Mancha, E., Traver, J.E., Tejado, I., Prieto, J., Vinagre, B.M., Feliu, V.: Artificial flagellum microrobot. Design and simulation in COMSOL. In: Advances in Intelligent Systems and Computing, vol. 693, pp. 491–501. Springer, Heidelberg (2018)
16. Nelson, B.J., Kaliakatsos, I.K., Abbott, J.J.: Microrobots for minimally invasive medicine. *Ann. Rev. Biomed. Eng.* **12**, 55–85 (2010)

17. Paprotny, I., Bergbreiter, S.: Small-scale robotics: an introduction. In: Small-Scale Robotics. From Nano-to-Millimeter-Sized Robotic Systems and Applications, vol. 8336, pp. 1–15. Springer, Heidelberg (2014)
18. Prieto-Arranz, J., Traver, J.E., López, M.A., Tejado, I., Vinagre, B.M.: Study in COMSOL of the generation of traveling waves in an AEF robot by piezoelectric actuation. Actas de las XXXIX Jornadas de Automática, Badajoz, 5–7 de Septiembre de 2018, pp. 748–755 (2018)
19. Proakis, J., Manolakis, D.: Digital Signal Processing: Pearson New International Edition. Pearson Education Limited (2013)
20. Purcell, E.M.: Life at low Reynolds number. Am. J. Phys. **45**(1), 3–11 (1977)
21. Raz, O., Avron, J.E.: Swimming, pumping and gliding at low Reynolds numbers. New J. Phys. **9**, 437 (2007)
22. Sitti, M.: Microscale and nanoscale robotics systems: characteristics, state of the art, and grand challenges. IEEE Rob. Autom. Mag. **14**(1), 53–60 (2007)
23. Tejado, I., Serrano, J., Pérez, E., Torres, D., Vinagre, B.M.: Low-cost hardware-in-the-loop testbed of a mobile robot to support learning in automatic control and robotics. IFAC-PapersOnLine **49**(6), 242–247 (2016)
24. Traver, J.E., Tejado, I., Prieto-Arranz, J., Nuevo-Gallardo, C., Vinagre, B.M.: Improved locomotion of an AEF swimming robot using fractional order control. In: IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2019) (**Accept**) (2019)
25. Traver, J.E., Tejado, I., Vinagre, B.M.: A comparative study of planar waveforms for propulsion of a joined artificial bacterial flagella swimming robot. In: Proceedings of the 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT 2017), pp. 550–555 (2017)



# Learning Low-Level Behaviors and High-Level Strategies in Humanoid Soccer

David Simões<sup>1,2(✉)</sup>, Pedro Amaro<sup>3</sup>, Tiago Silva<sup>3</sup>, Nuno Lau<sup>1</sup>,  
and Luís Paulo Reis<sup>2,3</sup>

<sup>1</sup> DETI/IEETA, Institute of Electronics and Informatics Engineering of Aveiro,  
University of Aveiro, Aveiro, Portugal

{david.simoes,nunolau}@ua.pt

<sup>2</sup> LIACC/FEUP, Artificial Intelligence and Computer Science Lab.,  
Faculty of Engineering, University of Porto, Porto, Portugal

lpreis@fe.up.pt

<sup>3</sup> DEI/FEUP, Informatics Engineering Department, Faculty of Engineering,  
University of Porto, Porto, Portugal

{up201405210,up201402841}@fe.up.pt

**Abstract.** This paper investigates the learning of both low-level behaviors for humanoid robot controllers and of high-level coordination strategies for teams of robots engaged in simulated soccer. Regarding controllers, current approaches typically hand-tune behaviors or optimize them without realistic constraints, for example allowing parts of the robot to intersect with others. This level of optimization often leads to low-performance behaviors. Regarding strategies, most are hand-tuned with arbitrary parameters (like agents moving to pre-defined positions on the field such that eventually they can score a goal) and the thorough analysis of learned strategies is often disregarded. This paper demonstrates how it is possible to use a distributed framework to learn both low-level behaviors, like sprinting and getting up, and high-level strategies, like a kick-off scenario, outperforming previous approaches in the FCPortugal3D Simulated Soccer team.

**Keywords:** Multi-agent systems · Machine learning · Evolution strategies

## 1 Introduction

The design of a robotic system comprised of multiple elements is a major challenge in the field of robotics research. It is desirable for agents to have near-optimal controllers for different tasks, like kicking a ball, or getting up after falling down, which are both reliable and fast, and which are often learned in simulation. It is also desirable for the team to cooperate using high-level strategies that take advantage of the distributed nature of the environment.

Agents should make decisions that benefit the team, not themselves, and work together to complete the end goal (for example, winning a soccer match).

The scientific community has proposed multiple algorithms [1, 9, 16, 20, 22, 23] to optimize robot controllers with a given objective function. Some of these can be run in distributed environments, in order to increase the speed at which a policy is converged to. We will show that these can also be used to optimize high-level strategies, by optimizing the parameters that define each player's decisions. However, high-level strategies are traditionally optimized with multi-agent reinforcement learning algorithms [5, 11, 14, 18, 21], which use strategies from the multi-agent systems field to achieve coordination between team members. We describe a framework that supports multiple reward-based learning algorithms to be deployed over a network of multiple workers, and is able to optimize both low- and high-level controllers within reasonable time and with outstanding results.

The remainder of this paper is organized as follows. Section 2 introduces the main concepts used in this paper, related to RoboCup and, more specifically, the 3d Soccer Simulation League. Section 3 describes the framework that was developed to optimize low-level and high-level skills. Sections 4 and 5 describe examples of low-level and high-level skills, which were optimized using CMA-ES and A3C3. Section 6 presents and discusses the results of our experiments, and finally, Sect. 7 concludes our work, summarizing the main findings.

## 2 Soccer Simulation League

The Soccer Simulation League is part of the RoboCup initiative, an annual international robotics competition, whose goal is to have a team of fully-autonomous physical robots winning a soccer match against a team of humans, using FIFA standard rules, by the year 2050. The 3d Soccer Simulation League is a complex multi-agent environment where two teams of humanoid simulated robots play a ten-minute soccer match using realistic rules. Each team is comprised of eleven Nao robots [4] with multiple different models, each with different physical characteristics. Each agent perceives the environment and acts only upon its local joints, by sending commands to the environment simulator.

Using low-level controllers that abstract simple tasks, like kicking or moving, has been the *de facto* standard in the league, controllers which are then used by high-level decision-making modules. In other words, agents have a set of behaviors which are chosen according to their strategy. The behavior acts upon the agent's joints, and the strategy defines which behavior to execute and with which parameters. A simple example is a kick-off strategy, at the start of the game, where agents choose their positions, such that the enemy team is covered and the friendly team can pass the ball and attempt to score a goal.

Recently, the league has proposed new rules to constrain the movements of agents. Before now, the body parts of the same agent can intersect with each other, without repercussion. These *self-collisions* are unrealistic and have since become something taken advantage of by many teams to achieve stronger kicks, or faster movements. However, new rules are being implemented to limit

self-collisions, with heavy penalties for agents that have them. Re-optimizing previous behaviors with attention to the new rules is then a necessity for any team that wishes to maintain its performance.

Since changes to low-level behaviors will inadvertently affect their performance, optimizing high-level strategies that depends on them is the logical next step. While these may seem very different optimization problems, we will show that it is possible to describe them as fully parametric problems that can be optimized by the same algorithms.

### 3 Optimization Framework

A framework was developed for optimizing low-level behaviors and high-level strategies, using arbitrary reward-based optimizers, and adaptable to multiple RoboCup teams. The framework supports distributed computing, to speed up the optimization process, and scales linearly with computational power. In other words, the time taken to evaluate a set of samples is inversely proportional to the amount of workers available.

We have focused on the CMA-ES [7] algorithm, the A3C3 [18] multi-agent algorithm, and the FCPortugal3d [10] team. CMA-ES is a genetic algorithm where an arbitrary amount of samples correspond to sets of parameters to optimize. After each generation of samples is evaluated, a new set is generated. This process is repeated until convergence has been achieved. A3C3 is a multi-agent actor-critic algorithm, based on deep learning, where agents learn policies and communication protocols simultaneously. Agents gather mini-batches of experience samples, and optimize a centralized critic to approximate the team's value function, an actor network to output each agent's policy, and a communication network to output each agent's broadcast message. We refer the reader to the original articles for more further information.

In order to optimize behaviors or strategies, fitness functions (also known as reward functions) must be defined for both. These are problem dependent, and take into consideration factors relevant to the scenario, such as the time it takes an agent to stand up, or whether a goal was scored with a given formation.

Due to a noisy environment, the fitness of a given sample must be averaged over a number of trials. We have assumed the reward distribution is a Normal distribution  $\mathcal{N}(\theta, \sigma)$ . Thus, we can calculate the standard error of the mean  $\sigma_e = \frac{\sigma}{\sqrt{n}}$ , where  $n$  is the number of trials used to evaluate the sample's fitness. We now have a trade-off between accuracy of our evaluation, which increases with the amount of trials, and the speed at which each sample is evaluated.

The framework allows samples to be evaluated with three main methods. The first is a trivial centralized implementation, where a single worker evaluates each sample sequentially. This results in low computer utilization and slow computation times. The second is a local distributed implementation, where multiple workers each handle their own environment. This solution takes advantage of the many cores a modern computer has available, and speeds up the convergence process. The third solution is a network distributed one: a master

generates samples and requests other workers to evaluate them. This allows the framework to scale horizontally, and is the fastest method.

We defined three different entities required to successfully run the network distributed system: a master, which runs the optimizer and assigns tasks to workers; a worker, which evaluates samples given by the master; and a manager, which allows communication between the master and the workers. The system is also fault-tolerant against worker failures (which can crash with software or hardware errors). Statistics are automatically collected and displayed, including the performance of each individual worker over time, and histograms of the best parameter sets.

Another crucial feature of the framework is that it is partly continuously integrated. This allows for quick iterations without constantly deploying code running on workers. The framework is published on-line, and the master notifies workers of specific code versions to execute. Worker then fetch that code, compiling it when necessary, and evaluate the samples using multiple threads. Results are sent back to the manager, and this process is repeated until convergence has been achieved. In addition, the core code of the workers is also periodically updated by simply publishing new versions on-line. This allows multiple workers to be deployed and updated with ease.

## 4 Low-Level Behaviors

A behavior that allows agents to stand up after having fallen, which we call a *get-up*, is one of the key behaviors in robotic soccer competitions. Because soccer is a sport of constant physical contact, agents fall frequently, and even champion teams are not yet capable of reliably playing a match without falling. For example, it is quite common to see agents fall after performing a powerful kick or when trying to run fast. Standing up as soon as possible after such falls is of the utmost importance, as an agent that is lying on the ground is not capable of positively impacting the state of the game.

The current *get-up* behaviors used by the FCPortugal3d team have a major flaw - they disregard self-collisions. This means an agent is capable of, for example, moving its arms directly through its legs. As stated previously, new rules are now making such movements illegal and current behaviors must be re-optimized, taking into account the new circumstances. We use a state-of-the-art black-box optimizer, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6], which has been widely used in optimization problems [2,8].

The process of evaluating the parameters of a *get-up* has the following steps: the agent purposefully falls to the ground; the agent tries to get up using the parameters optimized by CMA-ES; once the agent has stopped moving and its position is stable, the fitness of the parameters is evaluated.

The last step is the most crucial one, and requires a good evaluation function, or the agent will not be able to learn how to stand up. During the first experiments, the fitness function was rather simple, and only took into account the final height of the agent. The fitness value  $f$  was given by  $f = \frac{\min(0.54, h_f)}{0.54}$ ,

where  $h_f$  represents the agent's final height, and 0.54 its standing height, such that  $f$  is normalized between 0 and 1. The intuition behind this is to reward the agent based on how well it stood up from a fallen position. However, any action taken during the behavior would be rewarded solely based on the end result, ignoring factors like the time taken. Since speed is valuable,  $f$  was then given by

$$f = \begin{cases} \frac{10}{\Delta_t} & \Delta_t > 10 \\ 1 + \frac{h}{0.54} & \Delta_t \leq 10, h_f < 0.54, \\ 2 + \frac{3}{\Delta_t} & \Delta_t \leq 10, h_f \geq 0.54 \end{cases}, \quad (1)$$

where  $\Delta_t$  represents the time taken for the agent to stand up, from the moment it fell on the ground. The new formula can be thought of as a three-part learning process. First, the agent learns to move in a way that takes less than ten seconds to complete. Then, the agent learns how to get up successfully. Finally, it learns to do so as fast as possible. In order to make reading the results easier, we set the goal for  $\Delta_t$  as three seconds. Thus, a fitness value  $f < 1$  means the agent took too long to move, a fitness value  $1 < f < 2$  means the agent did not manage to get up, a value  $2 < f < 3$  means the agent managed to get up but took more than three seconds, and a value  $f > 3$  means the agent managed to stand up in less time. However, we quickly noticed that rewarding the agent based solely on the end result was not enough for it to consistently learn how to get up. Not only did it take the agent hundreds of generations to get up successfully for the first time, but sometimes it did not even manage to get up at all. Often times, the agent would prioritize a stable final position, even if it meant it was not standing up.

In order to address this problem, we drew inspiration from Q-Learning [24], a classic reinforcement learning algorithm. In Q-Learning, future rewards are discounted by a factor  $\gamma$ , and in doing so, their importance is incorporated into current rewards. In our case, we wanted to give more importance to heights achieved by the agent near the end of the movement, but without completely disregarding earlier heights. With this in mind, we arrived at the current fitness function

$$f = \begin{cases} \frac{10}{\Delta_t} & \Delta_t > 10 \\ 1 + \frac{\sum_{i=0}^n \gamma^{n-i} \times h_{n-i}}{\sum_{i=0}^n \gamma^{n-i} \times 0.54} & \Delta_t \leq 10, h_f < 0.54, \\ 2 + \frac{3}{\Delta_t} & \Delta_t \leq 10, h_f \geq 0.54 \end{cases}, \quad (2)$$

where  $n$  stands for the number of time-steps of the behavior, and  $h_i$  stands for the height of the agent at time-step  $i$ . The sum  $\sum_{i=0}^n \gamma^{n-i} \times 0.54$  represents the maximum possible reward the agent could have achieved throughout the movement, if the agent measured its standing height at every single time-step. As with the previous versions of the fitness function, this is used for normalization purposes, in order to ease readability of the fitness values.

The height of the agent is evaluated once per server time-step, with 50 recordings per second. During testing, we found a good value for the discount factor

$\gamma = 0.98$ . Smaller values would give too much weight to the final height, while greater values would give too much importance to initial heights, where the agent would just learn to lift its torso, but rarely to use its legs in order to reach greater heights.

## 5 High-Level Strategies

The previously exemplified behavior is considered a low-level skill, as defined by Stone et al. [19], which interface directly with the hardware, controlling motor joints to achieve a goal. With these, we can build sets of higher level strategies, which use these behaviors to achieve a larger task, such as passing the ball. This can be repeated to form multiple layers of skills, such as forming an overall game strategy from ball passes and goal kicks. All these components are necessary in a simulated soccer team to achieve victory in a game. At the beginning, much of the attention of the 3D Simulation League was given to the low-level behaviors, focusing on walking, kicking and standing up. As these skills have been refined, more importance is now given to high-level skills.

Optimizing the entire team's behaviors and strategies simultaneously is unfeasible, however, and layered learning strategies are often used [12]. By learning low-level behaviors first, and then optimizing high-level strategies, the complexity of the optimization problem is greatly reduced. Additionally, one can optimize related skills together, even if they are in different layers, such as a kick to the goal and a strategy that ends in a kick to the goal. This allows tuning the parameters of the skills to each other, improving their overall effectiveness.

We initially focused on optimizing a kick-off strategy, used at the beginning of the soccer game, executed by two players, a passer and a kicker. These use low-level behaviors implemented in the FCPortugal3d team: walking, a controlled kick and a long-distance kick. The passer is positioned near the middle of the field, next to the football, and the kicker is positioned behind the passer. The passer uses the controlled kick to pass the ball to the kicker, moves out of the way, and the kicker then tries to score a goal.

The fitness function for this strategy takes into account several factors, such as whether there was effectively a goal, how quickly it was performed, and the height of the ball after it was kicked to the goal (which prevents opponents from blocking it). The fitness value  $f$  is given by

$$f = \begin{cases} gh & t \leq 10 \\ (g - \frac{t-10}{20})h & t > 10 \end{cases}, \quad (3)$$

where  $g \in \{0, 1\}$  represents whether there was a goal,  $t$  the time taken to achieve the goal, and  $h$  the maximum vertical position achieved by the ball. This function was defined in order to reward goals within ten seconds, and penalize strategies up to thirty seconds, at which moment the opponents will have overrun the kicker. We averaged the fitness value over 90 trials, to reduce the variance as much as possible without compromising the computational cost of the learning process.

We define two scenarios to optimize for Kick-Offs: one where we optimize the initial positions of the agents and the kick targets, which we call Kick-Off A; and one where we further optimize the parameters of the kicks used in the strategy, Kick-Off B. In other words, a simpler model with only eight parameters that focuses only on the high-level strategy, and a more complex scenario with 24 parameters that optimizes low- and high-level skills simultaneously.

We also consider the Keep-Away challenge, an environment where, instead of playing a soccer match, teams play a game of keep-away soccer. The defending team, with three members, must keep the ball away from a single opponent for as long as possible. The opponent will always walk towards the ball, and if the defenders take too long to pass the ball, they will lose the game. For this challenge, we consider two scenarios as well: a Passing game, without an opponent, where agents only need to pass the ball in order to score points; and a Keep-Away game, where a hard-coded agent will walk towards the ball, ending the game if he reaches it.

## 6 Results

We evaluate the performance of our framework, the low-level behavior of standing up and the high-level strategy of the initial kick-off, including a comparison with the expected optimization performance for the last optimization.

### 6.1 Framework

We started by evaluation the performance of the developed framework. For each of the three architectures, we averaged the amount of evaluated samples over time. Table 1(a) shows the hardware specifications of the worker computers used for these runs.  $L1$  was used for the first two methods, while all the computers together were used for the distributed runs. The first method uses a single core, and the second used eight cores.

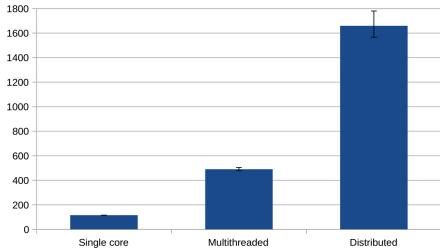
The results of our evaluation are shown in Fig. 1(b). Memory usage never exceeded 4 GB in any machine, and CPU usage was at 100% in all four computers, showing that the optimization process is CPU bound. It is quite clear that the distributed system greatly outperforms the other methods, as expected, since it utilizes all the computer resources available. We can also see a much greater variation in the minimum and maximum values in the distributed system, due to synchronization between threads and networking.

### 6.2 Low-Level Behavior

In FCPortugal3d, behaviors are defined following a key-frame approach [3, 15], with each key-frame being called a *slot*. Each slot defines how the agent's joints are at a given moment. The interpolation of two key-frames and the new joint positions are calculated through the use of a sinusoidal function.

ID	CPU	Memory
L1	i7-7700HQ 3.8GHz	16GB DDR4
L2	i5-M460 2.53GHz	4GB DDR3
D1	i7-4770K 4.3GHz	24GB DDR3
S1	Dual X5650 2.66GHz	8GB DDR3

(a) Hardware specifications of the machines used by the distributed optimization framework.



(b) Mean, minimum, and maximum trials evaluated over six ten-minute intervals.

**Fig. 1.** Specifications and evaluation of the distributed optimization framework.

To optimize a *get-up* behavior, we initially assumed that a high amount of slots would allow the agent to be more flexible. This intuition was derived from the hand-tuned solutions that the team used until now. However, as can be seen in Table 1, that is not the case. In fact, the behavior Getup A with the smaller amount of slots outperformed Getup B, the one with more slots. This may be due to an insufficient amount of trials per episode or of individuals per generation. Theoretically, optimizing both scenarios with an infinite amount of trials and individuals should yield similar or better results in the behavior with more slots. Nevertheless, Getup B took more than twice the amount of generations to achieve the same performance as Getup A, due to the extra amount of parameters to optimize, while still taking longer.

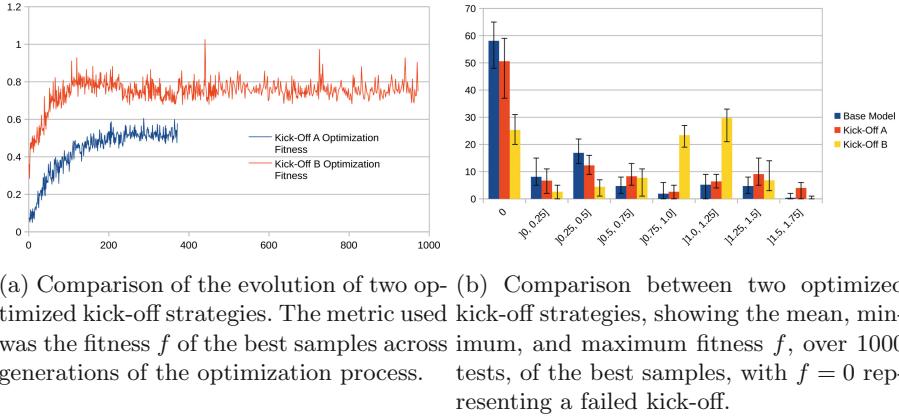
**Table 1.** Comparison of results between two optimized *get-up* behaviors with a different amount of slots. Metrics used include the success rate and behavior duration, as well as the generations where the first and best successful behaviors were found. A summary of the optimization process can be viewed at <https://youtu.be/QILCViewJdM> and <https://youtu.be/dx0fhFiQyg> for Getup A and B, respectively.

	Slots	Success	Duration	1st success gen.	Best solution gen.	Total gens.
Getup A	5	>99%	1.715 s	18	1506	2750
Getup B	10	>99%	1.858 s	43	4837	7360

In FCPortugal3d, a neutral position is used to smoothly transition between behaviors. As such, the results shown above were optimized with the aim to connect with this neutral position. When compared with the current *get-up* behavior in use (which does not link to the neutral position or respect self-collisions rules), Getup A was able to outperform it, being 29% faster. Although we were not able to objectively measure it, it also outperformed the original behavior in terms of consistency, as it would often fail to stand up on a single try.

### 6.3 High-Level Strategies

We optimized both the Kick-Off A and the Kick-Off B strategies, using a baseline as an initial solution, and optimizing until convergence was achieved. For each iteration, Kick-Off A used a default population size of 10 and took approximately 370 generations, and Kick-Off B used a default population size of 13 and took approximately 1000 generations.



- (a) Comparison of the evolution of two optimized kick-off strategies. The metric used was the fitness  $f$  of the best samples across 1000 generations of the optimization process.
- (b) Comparison between two optimized kick-off strategies, showing the mean, minimum, and maximum fitness  $f$ , over 1000 tests, of the best samples, with  $f = 0$  representing a failed kick-off.

**Fig. 2.** Evolution of two kick-off strategies and their comparison with the baseline. The optimized models can be viewed at <https://youtu.be/RILWVmaSETk> for the base model and <https://youtu.be/bO61mAD7rig> for the model with Kick-Off B.

The results of our evaluation are shown in Fig. 2. Both scenarios shows some peaks, which were most likely due to luck in the evaluation of the random trials, and not due to a good set of parameters.

We expected Kick-Off A to converge faster, as the search space is smaller. It took approximately  $440k$  trials to converge, while Kick-Off B took  $1200k$  trials. Using the estimates found in Fig. 1, we have compared the expected time it would take to compute this amount of episodes, and the actual time taken, in Table 2. The actual time taken is larger than the expected time for the distributed platform for many reasons, including  $L1$  not being continuously on-line during the optimization, and failures related to networking and power.

After the optimization, we selected the best sample for each scenario. We then test the samples over 1000 episodes for the base model and the two optimized scenarios. Figure 2 divides these tests based on their fitness, and shows how the optimized scenarios clearly outperform the baseline.

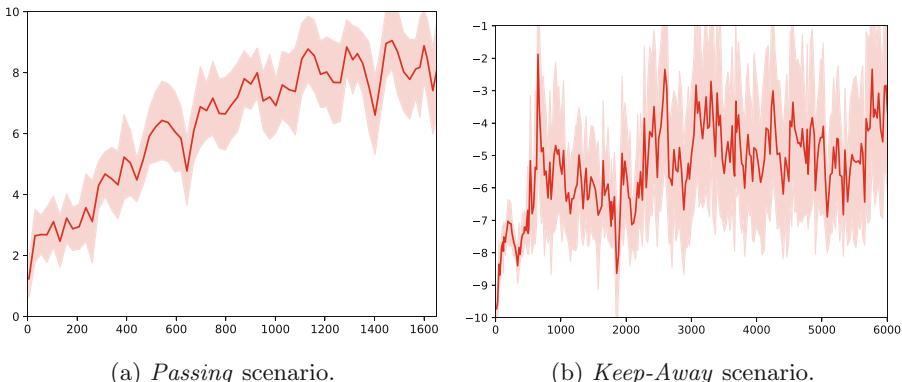
**Table 2.** Expected time according to measurements to converge to the Kick-Off A and B scenarios, as well as the actual time taken.

	Single core	Multithreaded	Distributed	Actual time
Kick-Off A	628 h	148 h	43 h	70 h
Kick-Off B	1770 h	417 h	123 h	175 h

The results are outstanding, with a fitness function average of 0.249 for the base model, which increases to 0.388 for the Kick-Off A, and to 0.701 for the Kick-Off B. Both scenarios show an increased performance with respect to the base model of 55% and 182%, respectively. It is interesting to note that, while Kick-Off A achieved the best goals, Kick-Off B achieved a larger amount of mediocre goals, and so achieved better overall performance.

The Passing and Keep-Away strategies were optimized with A3C3, using a learning rate  $\eta = 10^{-3}$ , a future reward discount factor  $\gamma = 0.9$ , two fully connected hidden layers of 40 nodes with ReLU activations [13] for every network, a tanh-activated two-node communication network, and  $N = 6$  concurrent worker threads.

A sub-set of FCPortugal3d’s behaviors was used, for A3C3 to learn a high-level policy that takes advantage of the low-level behaviors already existing in the team. Agents found ways to exploit the environment, like sequentially touching the ball instead of passing it, which would cause the environment to award them a point for a detected pass. These exploits were corrected, and as can be seen in Fig. 3, agents learned successful strategies for each scenario. For the *Passing* challenge, agents cluster in the center of the field for shorter and quicker passes. In the *Keep-Away* scenario, agents do the exact opposite, and rely on long and slow passes to avoid the opponent from ever reaching the ball.



**Fig. 3.** The evolution of policies in the two tasks of the 3D Soccer Simulation environment. The plots represent the average reward and standard deviation obtained by the team across episodes.

## 7 Conclusion

This paper investigated the learning of both low-level behaviors for humanoid robot controllers and of high-level coordination strategies in the context of the FCPortugal3d team. It proposed a framework that allows both to be optimized in a distributed manner, and demonstrated how it can be used in conjunction with reward-based optimizers like CMA-ES or A3C3 to outperform previous solutions. The framework was implemented with a horizontally scalable architecture, which supports multiple workers distributed over a network for parallel computation.

A low-level *get-up* behavior, and multiple high-level kick-off strategies were optimized. The learned solutions outperformed previous solutions by wide margins, and respects the newly implemented rules of the 3d Soccer Simulation League. The optimized skills were implemented in the team, and will be further tested in real-game environments in the RoboCup competition.

Future work directions include using other algorithms, like PPO [17], to optimize skills, as well as optimizing all other behaviors of the FCPortugal3d team, such that all conform to the new rules.

**Acknowledgements.** The first author is supported by FCT (Portuguese Foundation for Science and Technology) under grant PD/BD/113963/2015. This research was partially supported by LIACC (UID/CEC/00027/2019) and IEETA (UID/CEC/00127/2019).

## References

1. Abdolmaleki, A., Lau, N., Reis, L.P., Neumann, G.: Regularized covariance estimation for weighted maximum likelihood policy search methods. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 154–159. IEEE (2015)
2. Abdolmaleki, A., Simões, D., Lau, N., Reis, L.P., Neumann, G.: Learning a humanoid kick with controlled distance. In: Robot World Cup, pp. 45–57. Springer, Heidelberg (2016)
3. Cruz, L., Reis, L.P., Lau, N., Sousa, A.: Optimization approach for the development of humanoid robots' behaviors. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) Advances in Artificial Intelligence - IBERAMIA 2012, pp. 491–500. Springer, Heidelberg (2012)
4. Federation, R.: Robocup simulation 3d league rules. [https://ssim.robocup.org/wp-content/uploads/2018/12/Rules\\_RoboCupSim3D2018.pdf](https://ssim.robocup.org/wp-content/uploads/2018/12/Rules_RoboCupSim3D2018.pdf). Accessed 21 Apr 2019
5. Foerster, J.N., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. CoRR **abs/1705.08926** (2017)
6. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. **11**(1), 1–18 (2003)
7. Hansen, N.: The CMA evolution strategy: a tutorial. arXiv preprint [arXiv:1604.00772](https://arxiv.org/abs/1604.00772) (2016)
8. Kasaei, S.M., Simões, D., Lau, N., Pereira, A.: A hybrid zmp-cpg based walk engine for biped robots. In: Iberian Robotics Conference, pp. 743–755. Springer, Heidelberg (2017)

9. Kupcsik, A., Deisenroth, M., Peters, J., Neumann, G.: Data-efficient contextual policy search for robot movement skills. In: Proceedings of the National Conference on Artificial Intelligence (AAAI) (2013)
10. Lau, N., Reis, L.P., Shafii, N., Ferreira, R., Abdolmaleki, A.: FC Portugal 3D simulation team: team description paper. RoboCup 2013 (2013)
11. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. CoRR **abs/1706.02275** (2017)
12. MacAlpine, P., Stone, P.: Overlapping layered learning. Artif. Intell. **254**, 21–43 (2018)
13. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)
14. Peng, P., Yuan, Q., Wen, Y., Yang, Y., Tang, Z., Long, H., Wang, J.: Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. CoRR **abs/1703.10069** (2017)
15. Picado, H., Gestal, M., Lau, N., Reis, L.P., Tomé, A.M.: Automatic generation of biped walk behavior using genetic algorithms. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) Bio-Inspired Systems: Computational and Ambient Intelligence, pp. 805–812. Springer, Heidelberg (2009)
16. Rückstieß, T., Felder, M., Schmidhuber, J.: State-dependent exploration for policy gradient methods. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 234–249. Springer, Heidelberg (2008)
17. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint [arXiv:1707.06347](https://arxiv.org/abs/1707.06347) (2017)
18. Simões, D., Lau, N., Reis, L.P.: Multi-agent neural reinforcement-learning system with communication. In: World Conference on Information Systems and Technologies, pp. 3–12. Springer, Heidelberg (2019)
19. Stone, P., Veloso, M.: Layered learning and flexible teamwork in RoboCup simulation agents. In: Robot Soccer World Cup, pp. 495–508. Springer, Heidelberg (1999)
20. Stulp, F., Sigaud, O.: Path integral policy improvement with covariance matrix adaptation. arXiv preprint [arXiv:1206.4621](https://arxiv.org/abs/1206.4621) (2012)
21. Sukhbaatar, S., Szlam, A., Fergus, R.: Learning multiagent communication with backpropagation. CoRR **abs/1605.07736** (2016)
22. Sun, Y., Wierstra, D., Schaul, T., Schmidhuber, J.: Efficient natural evolution strategies. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 539–546. ACM (2009)
23. Theodorou, E., Buchli, J., Schaal, S.: A generalized path integral control approach to reinforcement learning. J. Mach. Learn. Res. **11**(Nov), 3137–3181 (2010)
24. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**(3), 279–292 (1992)

# **Workshop on Physical Agents**



# 3D Hand Joints Position Estimation with Graph Convolutional Networks: A GraphHands Baseline

John-Alejandro Castro-Vargas<sup>(✉)</sup>, Alberto Garcia-Garcia, Sergiu Oprea,  
Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez

3D Perception Lab, University of Alicante, Alicante, Spain  
`{jacastro,agarcia,soprea,pmartinez,jgarcia}@dtic.ua.es`

**Abstract.** State-of-the-art deep learning-based models used to address hand challenges, e.g. 3D hand joint estimation, need a vast amount of annotated data to achieve a good performance. The lack of data is a problem of paramount importance. Consequently, the use of synthetic datasets for training deep learning models is a trend and represents a promising avenue to improve existing approaches. Nevertheless, currently existing synthetic datasets lack of accurate and complete annotations, realism, and also rich hand-object interactions. For this purpose, in our work we present a synthetic dataset featuring rich hand-object interactions in photorealistic scenarios. The applications of our dataset for hand-related challenges are unlimited. To validate our data, we propose an initial approach to 3D hand joint estimation using a graph convolutional network feeded with point cloud data. Another point in favour of our dataset is that interactions are performed using realistic objects extracted from the YCB dataset. This could allow to test trained systems with our synthetic dataset using images/videos manipulating the same objects in real life.

**Keywords:** Synthetic dataset · Photorealism · Hand-object interaction · 3D hand joint estimation

## 1 Introduction

Hands offer useful information about people since they are a natural way used for humans to transmit non-verbal information to express and interact with each other [7, 18]. The correct detection of hands and the interpretation of their poses and gestures have the potential to improve the way in which people interact with machines: in virtual/augmented reality systems, it can help the user to interact in a more fluid and friendly way with the environment [9], in industry, it makes possible to make the control of robots easier [2] aside from improving the security in robot work areas [5], in domestic environments it makes easier to give orders for domotic systems [4] and also it opens the way to interact with service-oriented social robots [15].

Over the last years, several ways of getting information from hands have been studied. These ways have been based on wearables sensors as gloves [25] or vision methods [40]. The first case is distinguished by the use of gloves equipped with sensors, which are used to provide information about the joints positions. The second one makes use of cameras to identify the hands poses, which may or may not use marked gloves to enable image processing [39]. While sensorized glove-based methods require expensive equipment, vision-based ones only need cameras, which can also provide depth information such as Kinect [8] and Intel Realsense [6]. However, the accuracy in vision-based method usually are worse than gloves ones.

Vision-based methods are particularly interesting due to the costs and minimally invasive nature of using cameras. For this reason, researches have been made efforts to improve the accuracy of algorithms based on visual features to obtain information from hands, whether for: pose [24], actions [29], gestures [30], and behaviours [13] among other applications.

Recent approaches based on deep learning methods have outperformed the results obtained in several of the topics mentioned before [1]. However, these methods require labeled large-scale datasets to learn properly and generalize to unseen samples. Furthermore, manually annotating larg-scale datasets is excessively tedious and error-prone [38]. In this line, various datasets have been presented along of the years generated and labeled with the less supervised way possible. Such automatic labelling systems may be based on synthetic data [38] or vision methods with automatic [36] or semi-automatic labelling [27].

The main advantage of synthetic datasets is the ground truth fidelity, but the lack of total realism on hand textures, geometries, and movements usually has an impact on performance when automatic learning algorithms trained with synthetic data are used in real environments. On the other hand, vision based methods do not always get perfect ground truth and most of the times need manual refinement.

Despite their disadvantages, synthetic datasets are imposed due the ability to generate large-scale data at a low cost. Nowadays, we have powerful tools designed to video games like UnrealEngine, which can render in real time photo-realistic scenarios. Also, we have access to virtual reality headsets which offer an immersive experience. These tools added with sensorized gloves have the potential to generate photo-realistic datasets interacting with virtual environments, where ground truth may have a high fidelity. As proof of this, in the literature, several works have successfully applied synthetic datasets using domain transfer techniques [19, 20].

In this project we propose a novel photo-realistic synthetic dataset generated in a virtual reality environment and powered by a video game engine. Also, we propose a first approach for joints position detection by means of graph convolutional networks trained with pointcloud data.

## 2 Related Works

In the state of art can be found different datasets approaches oriented to hand context problems. Most of them are focused on hand detection and pose estimation where different methods are applied to generate and labelling the data. Many of them have been made with manual annotations, what can involve a lack in ground truth accuracy and a reduced number of samples to get representative sets of the problem that we want to solve. In this line, different approaches can be found which have been captured and annotated manually or semi-manually.

Dexter [21, 31, 32] and MSRA [33] are two common datasets used in hand tracking and pose estimation. Both of them have been manually annotated with a relative small number of samples, 2400 frames in the first case and 2137 in the second one. Due to the laborious process of labelling, the number of frames is limited and reduces their applicability in deep learning applications.

Other approaches set up controlled recording scenes in order to get a boost in the number of labelled frames and minimising the error obtained in the process with automatic methods. NYU dataset is one example of this process [35], multiple cameras were deployed and a slow but robust algorithm [22] was applied to make high accurate joints predictions based on synthetic rendered hands, with the addition of one possible post-refinement after predictions. Similarly, MSRA [33] and ICVL [34] first make a pose estimation and then manually reduce the error. In the first case the ground truth was defined by means of a spherical geometric approach [26], where spheres are used to represent hands poses. In the second one, it was used a simulated heuristic method [17], which consists of a preliminary estimation of different poses to finish selecting the most promising.

In order to automate the annotation process and obtain a greater amount of data, commercial devices were used to automatically annotate large datasets. In [11, 37, 38], LeapMotion, ShapeHands and trakSTAR are respectively used to annotate hands poses. This approach has as disadvantage that in the process is annotated the ground truth with the device error. In other words, the estimated pose might not accurately correspond to the real one.

Synthetic datasets have been generated in some proposals. In MSRC [28] random positions are generated on synthetic hands with depth image frames and its corresponding pose information. Halfway between synthetically generated datasets and capture devices, SynthHands [20] dataset use a real-time markerless tracker to track hands positions and re-target them onto a simulated hand. This approach allows to generate an accurate ground truth due to synthetic hands are positioned based on the information received by the device. In contrast with datasets recorded with devices as LeapMotion or ShapeHands, in this case the ground truth is accurately defined due to make the annotation after move the simulation. As a detriment, generated data are not totally realistic, so when it is tried to transfer the knowledge learned from synthetics, it might not respond appropriately.

New approaches based on advances in Deep Learning with GAN (Generative Adversarial Networks) networks [12] have sought to improve the photo-realistic quality of synthetic images. GANerated [19] performs this procedure, firstly they

process SynthHands dataset using GAN for image-to-image translation network to generate images similar to real hands. However, results generated by GAN networks can hold artefacts on the image, distorting features learned by automatic learning algorithms, producing an effect similar to such trained only with synthetic data when are used in real environments.

The datasets mentioned in this section have been used in different hand challenges, like hand pose detection. Different approaches can be found in the state of the art to solve this task like: regression forest [34, 37], cascade pose regression [33] and deep learning methods [10, 11, 38]. The last one has outperformed the results, specially building networks based on CNNs (Convolutional Neural Networks) [11, 38], even though that recent approaches show potential using graph convolutional networks [10].

These approaches usually learn from RGB and/or depth information. But in this project we propose an architecture based on graph convolutional networks to regress pose information from point clouds defined like graphs.

### 3 Dataset

Most of the datasets used to make hand pose detection have little data or are captured in environments with low occlusion. Therefore, it was decided to generate a new dataset with a crowd of poses for both hands. This data was generated interacting with daily objects from a first person perspective, and representing realistic occlusions that can occur with hands from this perspective.



**Fig. 1.** This figure shows a subset of samples scenes recorded with 6 of the objects available on the dataset.

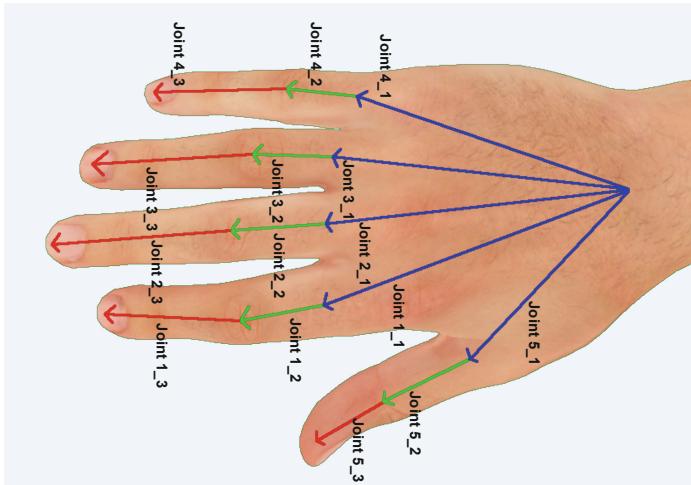
To generate large scale datasets our project has been powered by *UnrealRox* [16] tool, a plugin designed to work with UnrealEngine. This plugin allows to generate color and depth images through synthetic but high-realistic scenes, and

thanks to the use of synthetic data, it is possible to generate segmentation masks with pixel accuracy.

In our dataset 59 scenes were recorded with *UnrealRox*, which corresponds with 59 objects imported from [23]. These objects were published in YCB [3] dataset, which is focused to grasping studies, even though in this project the were used to generate occlusion in each recording. The Fig. 1 shows a subset of the objects used to record our dataset.

In periods of one minute we interact with one of the objects, making movements and rotations with both hands in order to capture different degrees of occlusion depending on the grasped object and the camera position.

The objects interactions were made with the approach defined in [23], which consists in using a Virtual Reality Headset and two hands controller to indicate hands positions in the simulation. Then, when hands are close to the object, the user triggers the grasp action process, and each finger begins its closing animation as far as a collision is detected.



**Fig. 2.** Visualisation of the joints in one of the hands.

The dataset is composed by 59 scenes, 6 cameras have been deployed in each one of them with different positions and orientations around the head position. In this matter, different angles of vision were obtained in each frame with both hands interacting with the object.

### 3.1 Recording

Throughout the recording, *UnrealRox* made captures of positions and orientations in each frame to generate images of color and depth with high realism. In

Left Hand/Right Hand								
Root Hand 3D Vector	Joint 1_1 3DPosition	Joint 1_2 3DPosition	Joint 1_3 3DPosition	...	Joint 4_1 3DPosition	Joint 4_2 3DPosition	Joint 4_3 3DPosition	
Finger 1					Finger 5			
i=0	i=1			...		i=4		

**Fig. 3.** List structure with joints information for each hand.

the same way, it allowed to generate segmentation masks in an automatic and precise way in the process.

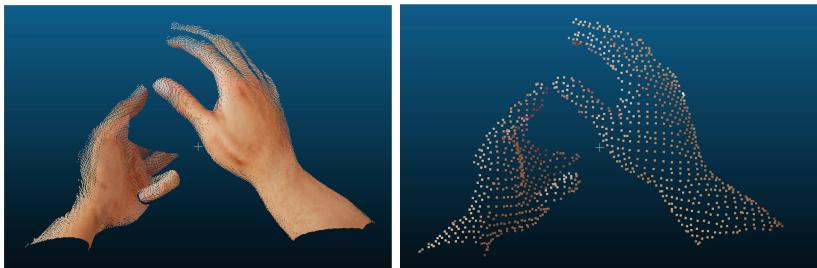
On one hand, the color and depth images are used to generate point clouds. In this process, we use the segmentation mask corresponding to hands to generate only points of this object.

On the other hand, we use the hand joints positions to define the labels for our neural network. *UnrealRox* is prepared to generate a json file with positions and orientations of every object in the scene in UnrealEngine world coordinates, so for each frame we transform world coordinate joints positions to each camera coordinate system.

Each json file contains a dictionary with two keys binded to lists joints for each hand (Fig. 2). For each list the first value is the root to hand position. The following values, are three vectors for each finger (Fig. 3), which are defined by means of a coordinate system transformation from UnrealEngine to camera.

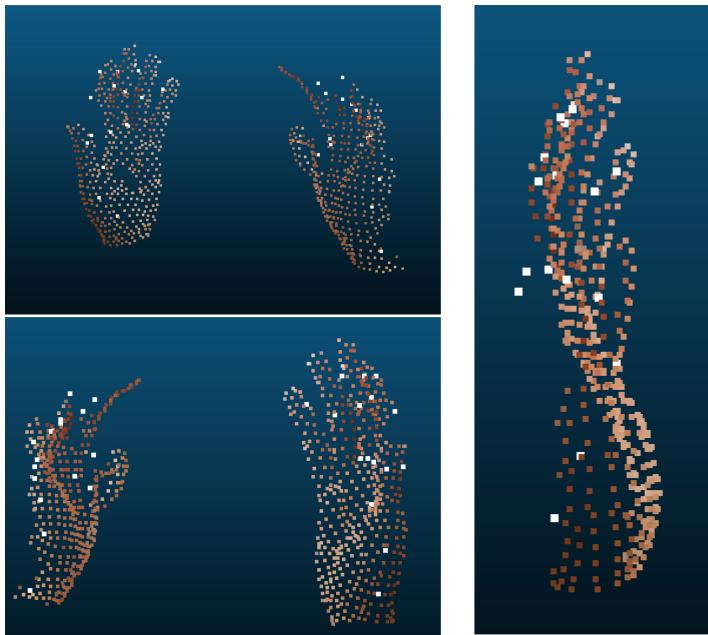
## 4 Network and Training

The method used to extract information and regress a pose from point clouds is based on graph convolutions [14]. In order to define the structure of the graph we used kd-tree with  $k = 7$  to relate each point with its neighbors.



**Fig. 4.** On the left, a representation of the hands point cloud. On the right, the same point cloud after voxel grid downsampling with a voxel size of 0.01.

In order to segment the point clouds we used their segmentation masks. Before training we also downsampled the point clouds using voxel grid technique with a voxel size of 0.01 (see Fig. 4) and also normalized the color values, dividing by 255 each channel. After this preprocessing step, the point clouds were reduced under 1495 points, which represents the biggest point cloud we have used. Moreover, in order to obtain a dense tensor which could be used throughout a fully connected layer, we added points to the graph without edge connections in position (0, 0, 0) with color (0, 0, 0).



**Fig. 5.** This figure shows different perspectives of the positions of the joints learned next to the corresponding point cloud.

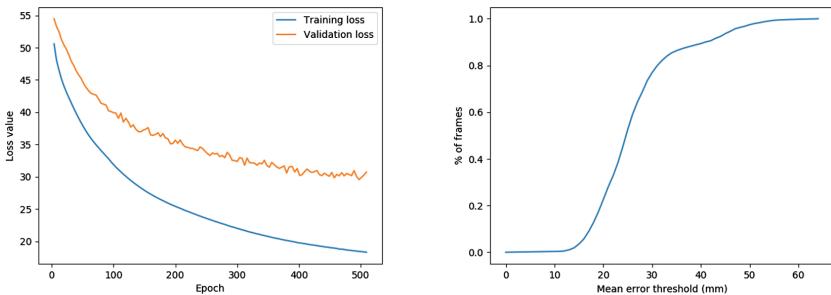
Our network architecture consists of two graph convolutional layers with 16 features based on Kipf and Welling approach [14] using ReLu activation layer. The input to the network consists of the graph represented as a feature matrix ( $N \times M$ ), where  $N$  are the nodes in the graph and  $M$  the number of features. The output of each graph layer is another feature matrix composed by ( $N \times P$ ) where  $P$  is the new number of features. In our case, the first feature matrix is defined by 1495 nodes and 6 features, corresponding with the number of points ( $N$ ) and [color,position] ( $M$ ). The feature matrix obtained by this approach is defined by node-level feature vectors, which means that the network needs to be pooled up to one node or connect the last matrix feature with a fully connected layer.

The output of the network is defined by 3 fully connected layers, two of 4096 nodes and the last one of 96, corresponding with the network output. The activation function used in these layers was leaky relu with a negative slope of 0.2. For training the network, we have used the mean absolute error alongside the Nesterov optimization using a momentum of 0.9.

#### 4.1 Experiments and Results

We conducted a brief experimentation with the aim to evaluate the effectiveness of graph convolution detecting hand joints position, and also to test the dataset itself. This experiment was performed taking six scenes recorded in our dataset: master chef can, cracker box, sugar box, tomato soup can, mustard bottle and tuna fish can. The first one has been used as validation set, the next four as training set, and the last one as test set.

The training set was used to train the network during 1024 epochs with a learning rate of 0.00001 and L1 regularization with lambda = 0.0001.



**Fig. 6.** The figure on the left shows the evolution of the loss along of 512 epochs on the sets of training and validation. The figure on the right shows the percentage of samples below an error threshold in millimetres on the test set.

The Fig. 5 shows the results obtained in the last epoch and how our network achieves regress joints positions successfully in training.

The left side of Fig. 6 shows a comparative between the loss evolution in training set and validation set. This allows observe that the network is not overfitting and there is a tendency to generalise with our learning method.

In order to test our network, we chose the weights with the lowest loss in the validation set, and tested the samples recorded with the object “sugar box”. On the right side of the Fig. 6, we can see the percentage of samples below an error threshold on this set. In this graph we can easily detect a lack of learning when approaching the joints up to their correct positions, since most of them are between 20 mm and 30 mm of error.

## 5 Conclusions

Two main contributions have been made in this project. On the one hand, a novel synthetic dataset with more than one million of frames has been proposed for hand challenges, and on the other hand, a first approach of an architecture based on graph convolutions for hand joints position detection was introduced.

Our architecture has been trained and tested with the developed dataset, obtaining satisfactory results, therefore showing that it can be successfully applied to regression problems, concretely to detect joint positions. Since this project consisted of a first test of the usefulness of our dataset and a brief evaluation of the capacity of graph convolutions to extract features from point clouds to position regression, very positive results can be extracted from it.

In future works, the authors want to perform a deep analysis of the proposed architecture for hand pose detection. This first application shows how our network is prepared to learn features and converge from point clouds generated with our dataset, but it would need to be trained with all the data in order to improve its capacity to generalise.

**Acknowledgements.** This work has been funded by the Spanish Government grant TIN2016-76515-R for the COMBAHO project, supported with Feder funds. This work has also been supported by three Spanish national grants for PhD studies (FPU15/04516, FPU17/00166, and ACIF/2018/197), by the University of Alicante project GRE16-19, and by the Valencian Government project GV/2018/022. Experiments were made possible by a generous hardware donation from NVIDIA.

## References

1. Asadi-Aghbolaghi, M., Clapes, A., Bellantonio, M., Escalante, H.J., Ponce-López, V., Baró, X., Guyon, I., Kasaei, S., Escalera, S.: A survey on deep learning based approaches for action and gesture recognition in image sequences. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), pp. 476–483. IEEE (2017)
2. Barattini, P., Morand, C., Robertson, N.M.: A proposed gesture set for the control of industrial collaborative robots. In: 2012 IEEE RO-MAN, pp. 132–137. IEEE (2012)
3. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The YCB object and model set: towards common benchmarks for manipulation research. In: 2015 International Conference on Advanced Robotics (ICAR), pp. 510–517. IEEE (2015)
4. de Carvalho Correia, A.C., de Miranda, L.C., Hornung, H.: Gesture-based interaction in domotic environments: state of the art and HCI framework inspired by the diversity. In: IFIP Conference on Human-Computer Interaction, pp. 300–317. Springer, Heidelberg (2013)
5. Castro-Vargas, J., Zapata-Impata, B., Gil, P., Garcia-Rodriguez, J., Torres, F.: 3DCNN performance in hand gesture recognition applied to robot arm interaction. In: Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods: ICPRAM, vol. 1, pp. 802–806. INSTICC, SciTePress (2019)

6. Chih, C.Y., Wan, Y.C., Hsu, Y.C., Chen, L.G.: Interactive sticker system with intel realsense. In: 2017 IEEE International Conference on Consumer Electronics (ICCE), pp. 174–175. IEEE (2017)
7. Congdon, E.L., Novack, M.A., Goldin-Meadow, S.: Gesture in experimental studies: how videotape technology can advance psychological theory. *Organ. Res. Meth.* **21**(2), 489–499 (2018)
8. Dong, C., Leu, M.C., Yin, Z.: American sign language alphabet recognition using microsoft kinect. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 44–52 (2015)
9. Garcia-Garcia, A., Martinez-Gonzalez, P., Oprea, S., Castro-Vargas, J.A., Orts-Escalano, S., Garcia-Rodriguez, J., Jover-Alvarez, A.: The RobotriX: an eXtremely photorealistic and very-large-scale indoor dataset of sequences with robot trajectories and interactions. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6790–6797. IEEE (2018)
10. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3D hand shape and pose estimation from a single RGB image. arXiv preprint [arXiv:1903.00812](https://arxiv.org/abs/1903.00812) (2019)
11. Gomez-Donoso, F., Orts-Escalano, S., Cazorla, M.: Large-scale multiview 3D hand pose dataset. *Image Vis. Comput.* **81**, 25–33 (2019)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
13. Kim, H., Lee, S., Kim, Y., Lee, S., Lee, D., Ju, J., Myung, H.: Weighted joint-based human behavior recognition algorithm using only depth information for low-cost intelligent video-surveillance system. *Exp. Syst. Appl.* **45**, 131–141 (2016)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
15. Luo, R.C., Wu, Y.C.: Hand gesture recognition for human-robot interaction for service robot. In: 2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 318–323. IEEE (2012)
16. Martinez-Gonzalez, P., Oprea, S., Garcia-Garcia, A., Jover-Alvarez, A., Orts-Escalano, S., Rodríguez, J.G.: UnrealROX: an eXtremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. CoRR abs/1810.06936 (2018). <http://arxiv.org/abs/1810.06936>
17. Melax, S., Keselman, L., Orsten, S.: Dynamics based 3D skeletal hand tracking. In: Proceedings of Graphics Interface 2013, pp. 63–70. Canadian Information Processing Society (2013)
18. Miwa, H., Itoh, K., Matsumoto, M., Zecca, M., Takanobu, H., Rocella, S., Carrrozza, M.C., Dario, P., Takanishi, A.: Effective emotional expressions with expression humanoid robot WE-4RII: integration of humanoid robot hand RCH-1. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004 (IROS 2004). Proceedings, vol. 3, pp. 2203–2208. IEEE (2004)
19. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: GANerated hands for real-time 3D hand tracking from monocular RGB. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 49–59 (2018)
20. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1284–1293 (2017)

21. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: Proceedings of International Conference on Computer Vision (ICCV) (2017). <http://handtracker.mpi-inf.mpg.de/projects/OccludedHands/>
22. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3D tracking of hand articulations using kinect. In: BmVC, vol. 1, p. 3 (2011)
23. Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J.A., Orts-Escalano, S., Garcia-Rodriguez, J.: A visually plausible grasping system for object manipulation and interaction in virtual reality environments. arXiv preprint [arXiv:1903.05238](https://arxiv.org/abs/1903.05238) (2019)
24. Panteleris, P., Oikonomidis, I., Argyros, A.: Using a single RGB frame for real time 3D hand pose estimation in the wild. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 436–445. IEEE (2018)
25. Pławiak, P., Sośnicki, T., Niedźwiecki, M., Tabor, Z., Rzecki, K.: Hand body language gesture recognition based on signals from specialized glove and machine learning algorithms. IEEE Trans. Ind. Inf. **12**(3), 1104–1113 (2016)
26. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1106–1113 (2014)
27. Rogez, G., Khademi, M., Supančič III, J., Montiel, J.M.M., Ramanan, D.: 3D hand pose detection in egocentric RGB-D images. In: Workshop at the European Conference on Computer Vision, pp. 356–371. Springer, Heidelberg (2014)
28. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., et al.: Accurate, robust, and flexible real-time hand tracking. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3633–3642. ACM (2015)
29. Singh, S., Arora, C., Jawahar, C.: First person action recognition using deep learned descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2620–2628 (2016)
30. Singha, J., Roy, A., Laskar, R.H.: Dynamic hand gesture recognition using vision-based approach for human-computer interaction. Neural Comput. Appl. **29**(4), 1129–1141 (2018)
31. Sridhar, S., Mueller, F., Zollhoefer, M., Casas, D., Oulasvirta, A., Theobalt, C.: Real-time joint tracking of a hand manipulating an object from RGB-D input. In: Proceedings of European Conference on Computer Vision (ECCV) (2016). <http://handtracker.mpi-inf.mpg.de/projects/RealtimeHO/>
32. Sridhar, S., Oulasvirta, A., Theobalt, C.: Interactive markerless articulated hand motion tracking using RGB and depth data. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), December 2013. [http://handtracker.mpi-inf.mpg.de/projects/handtracker\\_iccv2013/](http://handtracker.mpi-inf.mpg.de/projects/handtracker_iccv2013/)
33. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 824–832 (2015)
34. Tang, D., Jin Chang, H., Tejani, A., Kim, T.K.: Latent regression forest: structured estimation of 3D articulated hand posture. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3786–3793 (2014)
35. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph. (ToG) **33**(5), 169 (2014)

36. Wetzler, A., Slossberg, R., Kimmel, R.: Rule of thumb: Deep derotation for improved fingertip detection. In: Xianghua Xie, M.W.J., Tam, G.K.L. (eds.) Proceedings of the British Machine Vision Conference (BMVC), pp. 33.1–33.12. BMVA Press, Durham, September 2015
37. Xu, C., Nanjappa, A., Zhang, X., Cheng, L.: Estimate hand poses efficiently from single depth images. *Int. J. Comput. Vis.* **116**(1), 21–45 (2016)
38. Yuan, S., Ye, Q., Stenger, B., Jain, S., Kim, T.K.: BiGHand2.2M benchmark: hand pose dataset and state of the art analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4866–4874 (2017)
39. Zaman, M., Rahman, S., Rafique, T., Ali, F., Akram, M.U.: Hand gesture recognition using color markers. In: International Conference on Hybrid Intelligent Systems, pp. 1–10. Springer, Heidelberg (2016)
40. Zimmermann, C., Brox, T.: Learning to estimate 3D hand pose from single RGB images. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4913–4921. IEEE (2017)



# Map Slammer: Densifying Scattered KSLAM 3D Maps with Estimated Depth

Jose Miguel Torres-Camara, Felix Escalona, Francisco Gomez-Donoso<sup>(✉)</sup>,  
and Miguel Cazorla

Institute for Computer Research, University of Alicante,  
P.O. Box 99, 03080 Alicante, Spain

jmtorres@dccia.ua.es, {felix.escalona,fgomez,miguel.cazorla}@ua.es

**Abstract.** There are a range of small-size robots that cannot afford to mount a three-dimensional sensor due to energy, size or power limitations. However, the best localization and mapping algorithms and object recognition methods rely on a three-dimensional representation of the environment to provide enhanced capabilities. Thus, in this work we propose a method to create a dense three-dimensional representation of the environment by fusing the output of a KSLAM algorithm with predicted point clouds. We demonstrate with quantitative and qualitative results the advantages of our method, focusing in three different measures: localization accuracy, densification capabilities and accuracy of the resultant three-dimensional map.

**Keywords:** SLAM · 3D maps · Point clouds · Depth perception · Depth estimation · Sensor fusion

## 1 Introduction

One of the key features of any robot is the ability to localize itself in unknown environments as well as to build the corresponding map. This problem is known as simultaneous localization and mapping (SLAM) and is one of the main state of the art methodologies to provide a mobile robot with navigation capabilities without prior knowledge. This is not an easy problem because, in order to create a good map, it is needed an accurate localization. In the same way, to perform accurate localization tasks a map is needed.

There are several approaches to solve the SLAM problem, but the most accurate methods to perform this rely on three-dimensional data inputs. Usually, this kind of data is provided either by a LIDAR sensor, by a stereo pair or by a RGB-D camera. These sensors are commonly built in on mid-size robots and larger like the Turtlebot or the Pepper robot. However, small-size robots do not provide them. There are many reasons that preclude three-dimensional sensors in small-size robots. For instance, size and weight of these sensors could interfere in the movement of the robot or they could drain their batteries soon

enough. Nonetheless, this kind of robots use to come with a regular RGB camera that does not show the mentioned drawbacks and is less expensive. Some recent approaches, under the visual keyframe SLAM (KSLAM) methodology, rely in 2D landmark detection to perform localization and tracking. However, KSLAM methods, whilst performing accurately, do not provide a dense map of the environment so they can not benefit from the navigation methods that takes advantage of 3D data.

The small-size robots are intended to be deployed in locations hard to reach for greater robots and are mainly used for inspection. For instance, they are commonly used for evaluating the state of the gas and water pipelines [2, 14], or even the electric power lines [4]. They are also used in rescue tasks [10] due to their ability to get into narrow paths under collapsed buildings. The main drawback of these robots is that they are fully teleoperated by an expert.

In this paper, we propose a method to provide SLAM capabilities to small-size robots that do not feature three-dimensional sensors. Our proposal and main contribution uses the scattered 3D points provided by a state-of-the-art KSLAM method as support to correctly place 3D point clouds of the environment generated by estimating depth with a deep learning-based algorithm. Our pipeline only takes pairs of color images as an input and returns a dense three-dimensional map and the relative position of the camera. If the proposal is applied to a sequence of frames, it would finally provide the full map of the environment and the corresponding localization checkpoints within it. Our approach could be deployed in any robot equipped with a color camera to enable localization and dense three-dimensional map generation of the environment, which are far more easy to understand for human agents.

The rest of the paper is organized as follows. First, relevant works are reviewed in Sect. 2. Then, our approach is thoroughly explained in Sect. 3. Section 4 describes the experiments we carried out to validate our method and the corresponding discussion of the results. Finally, Sect. 5 draws the conclusions of this work and future research directions as well.

## 2 Related Works

In this section we review the main contributions to solve the SLAM problem and depth prediction from a single color image.

The solutions based on key frames are becoming the most common and efficient approaches for building a monocular visual SLAM system at the expense of filter-based methods [20]. In [16], it was demonstrated that keyframe-based family of methods outperforms filter-based one.

The main idea of the keyframe-based methods is splitting the camera tracking and the mapping in parallel tasks, as originally presented in PTAM [8]. It performed well in real time for small environments, and was used for augmented reality applications. For our work, we are focusing in landmark-based systems. These kind of systems takes advantage of keypoint detection and matching alongside subsequent frames in order to estimate the camera motion.

At the beginning, these methods use a visual initialization module to establish an initial 3D map and the camera pose. When the system captures a new frame, the data association module guesses a new pose using the information taken from previous frames, establishing associations with the three-dimensional map. Then, an error vector between the matches is calculated and minimized using the pose optimization module. If the minimization fails, they usually take advantage of different techniques to recover from this error. In the case of normal frames, the pipeline ends with this step. If the frame was selected as key frame, the system looks for landmarks, triangulates their positions and expands the resulting map. In parallel, a task of map maintenance is running to detect loop closures and minimize their errors.

In order to infer the new pose of the camera, the association module locates 2D features on the image and establish a search window around their locations to find the correspondence on the previous images. Each feature is associated with a descriptor to measure the similarity between them. It has been used low level descriptors as *Sum of Squares Differences (SSD)*, *Zero-Mean SSD* and *Normalized Cross Correlation (NCC)* [7], and high level descriptors as *SIFT* [12], *ORB* [15] or *SURF* [3]. This step requires fast matching structures to ensure real time performance.

For the visual initialization, the most common approaches do not use a known position of the camera with respect to a plane, and use the methods proposed in [11] to remove the depth from the problem, employing the essential and homography matrices. The side effect of this fact is that the reconstructed 3D scenes are scaled by a unknown factor. The depth is initialized randomly with values of large variance and it is updated in a looping process until this variance converges in successive frames.

*ORB SLAM* [13] is an state-of-the-art monocular SLAM keyframe-based technique that uses ORB keypoints and descriptors to perform the association step. It extracts corners using 8 pyramid levels over the entire image and dividing them into cells to calculate the descriptor. Then, this method discretizes the descriptor into a bag of words to speed up the feature matching. The viewpoint could be an issue for the description, so *ORB SLAM* chooses the descriptors from the key frame with the slight viewpoint difference with the current frame.

For pose estimation, *ORB SLAM* considers that the camera moves with a constant speed, and detects abrupt motions if the number of matched features goes under a threshold. If this kind of motion is detected, the map points are projected onto the current frame and matched with the current descriptors. It is important to remark that it defines a local map with the features of the key frames near the current frame, so it allows to carry out real-time processing. As an output, it generates a set of 3D camera poses. For this work, we modified it to also get the 3D scattered map and the 2D keypoints alongside the frame in which they appear.

Regarding depth estimation, in recent years, many deep learning approaches have appeared to estimate depth from monocular images, using an end-to-end

architecture. Additionally, some of them perform motion estimation too, so they can be suitable to solve the pose estimation problem.

In 2014, David Eigen published one of the state-of-the-art methods in this field [6]. First, it uses a network that predict the depth from the monocular frame in a coarse-scale. Later, this prediction is refined passing local regions to another network specialized in fine details.

In 2016, Iro Laina presented another interesting research in this area [9]. It presents a single fully-convolutional network that carries out the depth prediction in a more efficient manner, combining convolutions with the upsampling of feature maps.

However, single-image methods tend to have problems to estimate the depth when dealing with unseen types of images.

In 2017, DeMoN [19] proposed a method to profit the stereopsis. It features a convolutional neural network to estimate the depth and the camera motion from a subsequent pair of images, similar to the *Structure from Motion* technique, but with a learning-based system. This proposal calculates the dense correspondences, the depth and the camera motion between two frames using a single network. This approach has demonstrated that outperforms the reconstruction of *Structure from Motion* with two frames, so the dense representation could be suitable to generating a 3D map for SLAM. Nonetheless, this approach renders the predictions in an arbitrary scale. The scale is not even consistent between two subsequent predictions.

### 3 Approach

Our approach is focused in computing dense 3D maps using KSLAM methods and estimated depth. The proposed pipeline takes as an input two subsequent color images of a sequence and returns the 6 DoF transformation between the two frames and a dense point cloud that represent the 3D map. The pipeline could run in a loop so as the camera moves around the environment it would return an incremental map. Our approach is intended to provide small-size robots with SLAM and three-dimensional mapping capabilities. However, it could be deployed in any robot that features a regular color camera.

To do so, we assume a robot moves while captures color images. As the robot is moving, there exists a transformation between two consecutive frames. This transformation explains the movement of the robot between these two frames. Both frames are forwarded to ORB-SLAM2, which follows a KSLAM approach. This method returns a sparse 3D point cloud of landmarks along with the estimated camera motion.

A depth estimator, which is based in the DeMoN [18] approach, is run simultaneously. This deep learning-based method takes as input a pair of images as well and returns the estimated depth map of the scene with an arbitrary scale. This depth map is then projected into a dense 3D point cloud that represent the environment depicted in the most recent image coordinate frame.

Then, the 3D landmarks returned by the KSLAM method are reprojected into the most recent image of both so a set of 2D points are generated. These 2D

points are looked up into the point cloud generated before using the very same coordinates. This is straightforward as there is a 3D point for each 2D point. At this point we have a set of scattered 3D landmark points computed by the KSLAM and the corresponding 3D points obtained from the estimated point cloud. These correspondences are used to compute a transformation between both subsets of points so the estimated dense point cloud is aligned with the scattered 3D landmarks. It is worth noting that both sets of points yield different, arbitrary scales. It should also be noted that as the dense points clouds are estimated, they yield some error. To compute the correct transformation we use a RANSAC [1] approach. This algorithm allows us to find the best transformation despite having some error in the corresponding points. To deal with the scale issue we compute the transformation using singular value decomposition (SVD) with scale component [5].

The RANSAC process is as follows. First, we choose 10 random correspondences and compute the transformation (rotation, translation and scale) using SVD. The resulting transformation is then applied. Finally, we count the inliers. A correspondence is considered inlier when the corresponding points are under a distance threshold. This process runs in a loop for 300 times. Finally, we return the best transformation, namely the transformation that achieved the greatest number of inliers. This step also helps to filter erroneous transformations. If the best transformation does not explain at least the 25% of the points, this keyframe is discarded. We do this because considering that there are some points that are error prone, the SVD process cannot be performed on the full set of correspondences as it will lead to incorrect transformations.

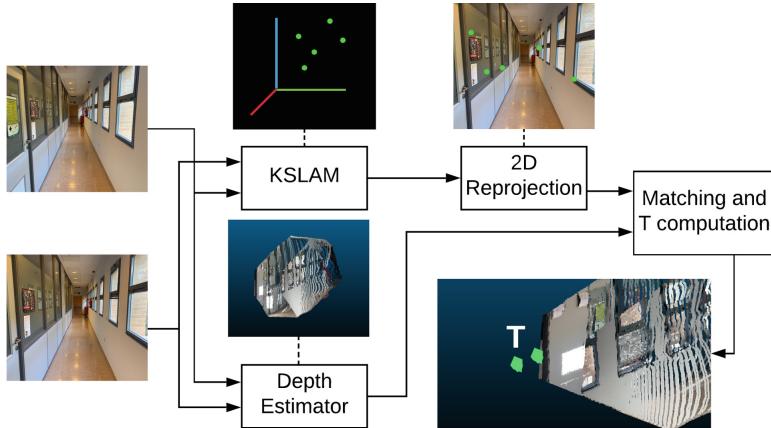
The returned transformation matches the relative motion of the camera between two keyframes. So the proposed algorithm returns the camera localization and the corresponding dense 3D map. This process is applied in a loop, for each frame of a sequence, to finally reconstruct the environment in which the camera is moving. A diagram of the pipeline is depicted in Fig. 1.

## 4 Experimentation and Discussion

In this section, the experimentation methodology and details about the data set are explained. The different benchmark metrics along the corresponding results of the experiments are also given.

### 4.1 Data Sets Description

We have chosen the RGB-D SLAM dataset and benchmark proposed in [17], as it is one of the main state of the art data sets to test SLAM methods. This data set contains the color and depth images of a Microsoft Kinect sensor along the ground-truth trajectory of the sensor. The data was recorded at full frame rate (30 Hz) and sensor resolution ( $640 \times 480$ ). The ground-truth trajectory was obtained from a high-accuracy motion-capture system with eight high-speed tracking cameras (100 Hz). The dataset is composed of several sequences



**Fig. 1.** Pipeline of the proposal. This pipeline is looped as the robot moves to finally build a full 3D map of the environment.

but we evaluated our approach on the following ones: freiburg1\_xyz, freiburg1\_desk, freiburg1\_desk2, freiburg2\_xyz and freiburg2\_desk. These sequences feature a range on different linear and angular velocities which will challenge the benchmarked algorithms.

In addition, we recorded several sequences of different indoor environments for qualitative evaluation. One of them was in one corridor of the building where our laboratory is located. This sequence was the selected one because the challenges it introduces, such as high contrasts, low variation in the structure and a lot of reflections and lightning flashes due to the glasses in the corridor and the windows. The results for this sequence can be seen in Sect. 5.

## 4.2 Localization Accuracy Benchmark

The authors of the dataset also provide some metrics to measure the accuracy of the benchmarked methods. These metrics are thoroughly explained in their work. First, the Absolute Trajectory Error (ATE) measures the difference between points of the true and the estimated trajectory so we can get insight about whether the final global trajectory is accurate or not. Then, Relative Pose Error (RPE) measures the error in the relative motion between pairs of time stamps and so states the accuracy of local trajectory over a fixed time interval. The implementation of both metrics are freely available by the authors of the data set and are used as provided.

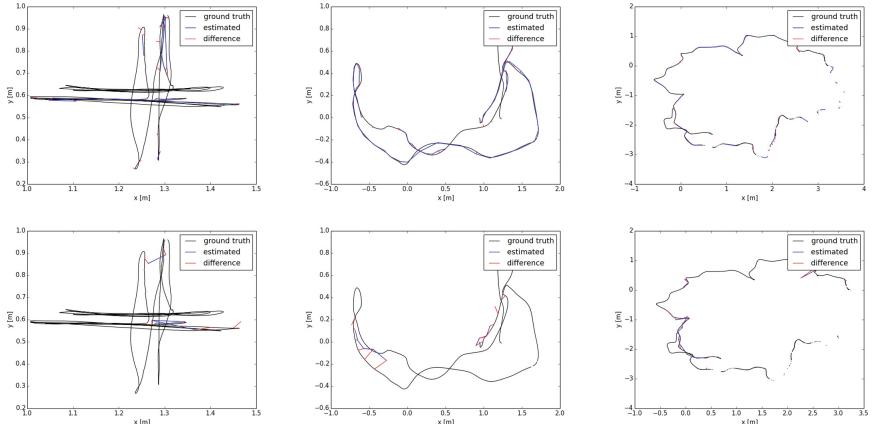
The results we achieve with our proposal are stated in Table 1. The average ATE and RPE of ORB-SLAM2 are 0.011588 and 0.014281, with a variance of 0.000006 and 0.000114, respectively. Given this statistic data it is possible to conclude that the error that the KSLAM commits is over 1 cm yielding a really low variance across our testing data sets. Thus, we consider it a robust method that can provide us with a trustful estimation of the camera pose.

**Table 1.** Absolute average trajectory error and average relative pose error achieved by ORB-SLAM2 (O-S2) and our proposal.

	ATE (O-S2)	RPE (O-S2)	ATE (ours)	RPE (ours)
freiburg1_xyz	0.008126	0.016362	0.032302	0.032302
freiburg1_desk	0.015965	0.018959	0.104802	0.219784
freiburg1_desk2	0.022276	0.028443	0.119201	0.290972
freiburg2_xyz	0.002442	0.001433	0.077768	0.048959
freiburg2_desk	0.009132	0.006209	0.115219	0.161799

Then, the average ATE and RPE of our proposal are 0.089858 and 0.150763, with a variance of 0.001297 and 0.012235. In spite of this data showing a bit more dependence on the data set, this relationship is small enough to trust the obtained averages, which tells us that translations of over 10 cm are needed to fit the point clouds.

It is also worth to mention that it was not possible to find good enough transformations for every keyframe used by ORB-SLAM2, leading to some mismatches in the associations between our trajectory and the ground truth, as can be appreciated in Fig. 2.



**Fig. 2.** Ground truth (black) vs O-S2 (blue top) and ours (blue bottom) trajectories. The distances are shown in red. The representations (from left to right) correspond to fr1/xyz, fr1/desk and fr2/desk.

### 4.3 Densification Capabilities Test

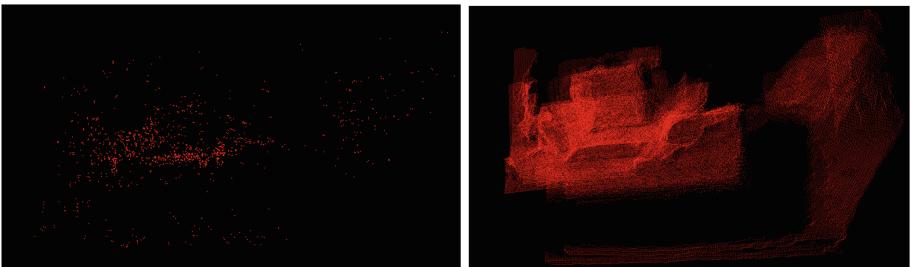
One of the main contributions of this work is that it returns a dense 3D reconstruction of the environment. To measure its densification capabilities, we com-

puted the count of points averaged across all the frames of each sequence of the data set. The results are reported in Table 2.

**Table 2.** Averaged count of points across all the frames of each sequence of the data set for ORB-SLAM2 and our method.

	ORB-SLAM2	Ours
freiburg1_xyz	315,156250	49152
freiburg1_desk	265,337838	49152
freiburg1_desk2	202,824324	49152
freiburg2_xyz	284,484848	49152
freiburg2_desk	270,320988	49152

As it can be seen in Table 2, the ORB-SLAM2 approach yielded few 3D points. This is expectable as these points are the result of matching 2D key points (ORB) across a minimum of two frames over time. Regarding our method, it always extracts 49152 3D points from each frame. That is due to the depth estimation we perform, in which we resize the frames to  $256 \times 192$  and, since predicted values are gotten for each pixel, we end up having always the mentioned amount of 49152 points for each frame. This known amount of data provides us not only with more certainty about our outputs but with almost 200 times more 3D data that ORB-SLAM2 did (the average of O-S2 provided points is 250,537375). This can be clearly appreciated in the Fig. 3.



**Fig. 3.** ORB-SLAM2 scattered output (left) vs our dense one (right). This results correspond to the fr2/xyz sequence.

This fact also justifies our pipeline, in which we only use the points provided by the KSLAM algorithm, which allows us to register the estimated point clouds with much less computational requirements compared with the utilization of a traditional registering pipeline which is based on the extraction and the description of the keypoints, and matching between them.

Furthermore, as the depths and, therefore, the 3D points were extracted using estimations of each pixel, we could assign to each of this points its corresponding color information, achieving the result shown on the Fig. 4.



**Fig. 4.** Our output with RGB information (from fr2/xyz).

#### 4.4 Accuracy of the Returned 3D Map

The previous metric measured the density of points. However, the goal of our approach is not only to produce a dense representation of the environment but also to do it in as accurately as possible. To measure the precision, we computed the mean distance of the nearest neighbor between the full 3D representation achieved by our approach and by ORB-SLAM2 (the sparse map) to the ground truth. We could have done this using the estimated point clouds, but this would have given worst results (taking into consideration the error of the predictions), so we chose to use only the points created by ORB-SLAM2 to be as critical as possible with our method. The results are shown in Table 3.

**Table 3.** Mean distance between the nearest neighbor of each point of the produced 3D representations to the ground truth representation.

	ORB-SLAM2	Ours
freiburg1_xyz	0.086565	0.0310728
freiburg1_desk	0.040083	0.0464012
freiburg1_desk2	0.086565	0.0626974
freiburg2_xyz	0.066125	0.0557921
freiburg2_desk	0.004647	0.0035932

As the results show, the distances to the nearest neighbour of the output of our method is between 3 and 6 cm. Being more precise, the average distance

is less than 4 cm (0.039911), with a variance of 0.000553. These results show a great precision level, keeping in mind that we are using points generated from an estimated depth map.

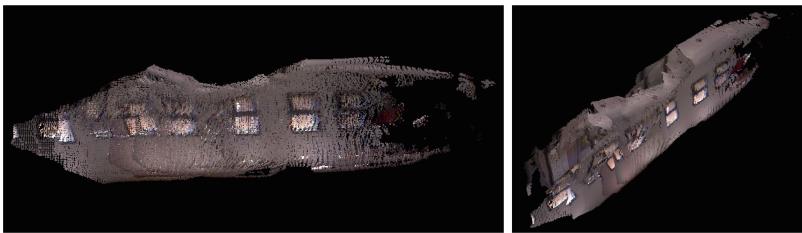
Regarding the ORB-SLAM2 related measurements, we can appreciate higher distances, between 4 and 8 cm, being 0.065162 the average one, with a reduced variance (0.000474). This demonstrates that our method not only provides almost 200 times more data than the used KSLAM method (as demonstrated in Subsect. 4.3), but this data is also more precise.

## 5 Conclusions and Future Work

In this work, we propose a method which fuses a visual KSLAM algorithm with predicted point clouds to generate a dense three-dimensional map of the environment. The proposal also outputs the camera pose within the map. Our method is monocular based, namely, it only requires a color camera to be executed.

Our method achieves a localization error of about 10 cm, which is accurate enough to be deployed in an actual robot. In addition, our method provides about  $50k$  points per frame, which is way more dense than the original ORB-SLAM2 visual SLAM algorithm that provides about 202–284 points per frame. Finally, the three-dimensional map produced by our method yields an error of about 4 cm compared with the ground truth.

In addition, we tested our approach live in different, challenging scenarios for qualitative evaluation. The result can be appreciated in Fig. 5.



**Fig. 5.** Output of our method using our own challenging sequence, seen from the side (left) and with perspective (right).

Nonetheless, our proposal shows some limitations. First, neither the KSLAM algorithm nor the predicted point clouds yield a real world scale. Furthermore, the scale is not even consistent between two consecutive predicted point clouds. However, the KSLAM does provide a constant yet arbitrary scale. Thus, our approach provides the very same scale so it does not represent the environment in real world measure units. In addition, our system relies on the KSLAM localization capabilities. If it fails to provide an approximately correct camera pose, our method will fail, since the KSLAM would locate the points in a wrong place, forcing our method to register the dense point cloud there.

The mentioned limitations are the main focus for the future work. We plan to address the scale issue by appending a calibration step that would easily provide real world units to the three-dimensional representation returned by our method. In addition, we plan to tune the SLAM localization so it can be robust against eventual localization disturbances. Finally we want to include pixel-level semantic information to enable a hybrid traditional and semantic localization. To do this, we will rely in deep learning-based pixel-wise classification algorithms.

**Acknowledgements.** This work has been supported by the Spanish Government TIN2016-76515R Grant, supported with Feder funds and by a Spanish Government grant for cooperating in research tasks ID 998142. This work has also been supported by a Spanish grant for PhD studies ACIF/2017/243 and FPU16/00887. Thanks to Nvidia for the generous donation of a Titan Xp and a Quadro P6000.

## References

1. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**, 381–395 (1981)
2. Yahya, N.A.B.H., Ashrafi, N., Humod, A.: Development and adaptability of in-pipe inspection robots. *IOSR J. Mech. Civ. Eng.* **11**, 01–08 (2014)
3. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: European Conference on Computer Vision, pp. 404–417. Springer (2006)
4. Deng, C., Wang, S., Huang, Z., Tan, Z., Liu, J.: Unmanned aerial vehicles for power line inspection: a cooperative way in platforms and communications. *JCM* **9**, 687–692 (2014)
5. Eggert, D., Lorusso, A., Fisher, R.: Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Mach. Vis. Appl.* **9**(5), 272–290 (1997). <https://doi.org/10.1007/s001380050048>
6. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems, pp. 2366–2374 (2014)
7. Hisham, M., Yaakob, S.N., Raof, R.A., Nazren, A.A., Embedded, N.W.: Template matching using sum of squared difference and normalized cross correlation. In: 2015 IEEE Student Conference on Research and Development (SCoReD), pp. 100–104. IEEE (2015)
8. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 1–10. IEEE Computer Society (2007)
9. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV), pp. 239–248. IEEE (2016)
10. Linder, T., Tretyakov, V., Blumenthal, S., Molitor, P., Holz, D., Murphy, R., Tadokoro, S., Surmann, H.: Rescue robots at the collapse of the municipal archive of Cologne City: a field report. In: 2010 IEEE Safety Security and Rescue Robotics, pp. 1–6, July 2010
11. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. *Nature* **293**(5828), 133 (1981)

12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
13. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Rob.* **31**(5), 1147–1163 (2015)
14. Zin, M.R.A.M., Saad, J.M.D., Anuar, A., Zulkarnain, A.T., Sahari, K.: Development of a low cost small sized in-pipe robot. *Procedia Eng.* **41**, 1469–1475 (2012)
15. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R.: ORB: an efficient alternative to SIFT or SURF. In: *ICCV*, vol. 11, p. 2. Citeseer (2011)
16. Strasdat, H., Montiel, J., Davison, A.J.: Real-time monocular SLAM: why filter? In: *2010 IEEE International Conference on Robotics and Automation*, pp. 2657–2664. IEEE (2010)
17. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: *Proceedings of the International Conference on Intelligent Robot Systems (IROS)*, October 2012
18. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: DeMoN: depth and motion network for learning monocular stereo. *CoRR* abs/1612.02401 (2016). <http://arxiv.org/abs/1612.02401>
19. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: DeMoN: depth and motion network for learning monocular stereo. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5038–5047 (2017)
20. Younes, G., Asmar, D., Shammas, E., Zelek, J.: Keyframe-based monocular SLAM: design, survey, and future directions. *Robot. Auton. Syst.* **98**, 67–88 (2017)



# Integration of the Alexa Assistant as a Voice Interface for Robotics Platforms

Alejandro Hidalgo-Paniagua<sup>2</sup>, Andrés Millan-Alcaide<sup>1</sup>, Juan P. Bandera<sup>1()</sup>, and Antonio Bandera<sup>1</sup>

<sup>1</sup> Universidad de Málaga, Málaga, Spain

{amillan,jpbandera,ajbandera}@uma.es

<sup>2</sup> Altran Innovación S.L., Barcelona, Spain

alejandro.hidalgopaniagua@altran.com,

<http://www.uma.es>, <https://www.altran.com>

**Abstract.** Virtual assistants such as Cortana or Google Assistant are becoming familiar devices in everyday environments, where they are used to control real devices through natural language. This paper extends this application scenario, and it describes the use of the Alexa assistant from Amazon through an Echo dot device to drive the behaviour of a robotic platform. The paper focuses on the description of the technologies employed to set such ecosystem. Significantly, the proposed architecture is based, from the remote server to the on-board controllers, in Low-Energy (LE) hardware and a scalable software platform. This approach will ease programmers integrating different platforms, e.g. mobile-based applications to control robots or home-made devices.

**Keywords:** Virtual assistant · Human-robot interface · Voice commanding

## 1 Introduction

Robotics has made significant progress towards full and shared autonomy in complex tasks, such as navigation and manipulation. Moreover, they are now capable of social interaction with human users, presenting a new opportunity to provide individualized care/assistance/help. A great deal of attention and research is dedicated to assistive systems aimed at promoting ageing-in-place, facilitating independent living in one's own home as long as possible [2]. The success of these systems depends on their acceptability, that can be described in terms of attitude, usability and confidence. Under these premises, it is clear that these robots require to provide natural communication channels to be commanded. However natural, intuitive interaction is usually not easy to establish, specially when engaging a population group (elderly people) in which physical and cognitive limitations are more common, and the use of new technologies,

less frequent [3]. One solution to smooth this engagement could be to include a virtual assistant on the *loop*.

A virtual assistant is a *conversational, computer-generated character that simulates a conversation to deliver voice- or text-based information to an user via a web, kiosk or mobile interface* [1]. Voice-user interface (VUI) employs different technologies to recognize and process natural language, enabling users to interact with an artificial agent by just speaking. Similar to the scenario desired by robotics assistants, the expected result is a more human-like, natural form of user interface. On the other hand, it is also important to note that voice assistants in particular, but also robotic controllers in general (e.g. the Jibo robot<sup>1</sup>), are becoming smaller, with lower energy consumption, and require less computing power to operate. This fact is causing voice assistants to be bursting strongly into the world of IoT (Internet of Things). This paper proposes the use of a virtual assistant as the interface between the human user and the robotic platform. The advantage of using this scheme is twofold. On one hand, the robust voice-based framework of the virtual assistants provides a natural mechanism for interaction, improving the relative low speech recognition rates (specially in real environments) that can provide previous recognition alternatives. On the other hand, the effort to introduce virtual assistants at home may take hand in hand the simultaneous introduction of robotic assistants, able of performing tasks that require manipulation or/and navigation (i.e. agents that go beyond the mentioned Jibo, which is quite similar to a virtual assistant). The integration with the *non-written* guidelines of IoT is considered in our proposal. Thus, it is built over an architecture that, from the remote server to the on board controller, is based in LE (Low Energy) embedded systems (e.g raspberry pi or odroid platforms) and is fully scalable and compatible with other technologies, as mobile-based applications for controlling robots or any home-made device.

The paper is organized as follows: Sect. 2 introduces the software architecture as a block diagram that describes the information flow through all components. Section 3 summarizes the process of natural interaction using voice assistants and how they allow us easily programming new applications in an abstract way. Section 4 shows the set of tools and frameworks for programming voice-based applications. Section 5 comments the hardware used in experiments, from the remote server to the on board controllers. Section 6 shows a real VUI implemented to control a robot using the Amazon's Alexa voice assistant. Alexa has been used to interface robots such as the Lynx<sup>2</sup> one, and is fully integrated in the ES-2A<sup>3</sup> robot. In our scheme, the voice assistant and the robotic platform are included within a heterogeneous and distributed system, which is open to consider other items. Finally, Sect. 7 draws the obtained conclusion of our proposal and the possible future works.

---

<sup>1</sup> <https://www.jibo.com/>.

<sup>2</sup> <https://ubtrobot.com/products/lynx>.

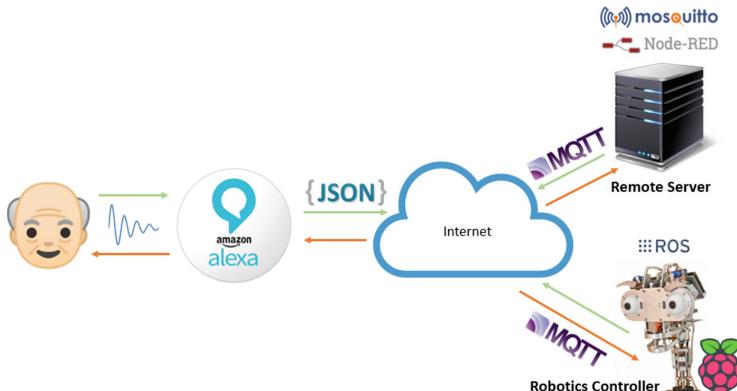
<sup>3</sup> <https://espabot.es/robot-social-es-2a/>.

## 2 Overview of the Proposal

Figure 1 shows the overview of the software architecture. The main goal is to set a communication channel between the user and the actuation devices. This channel should include the ability to have a conversation with the user, providing automatic responses to their entries if required (i.e. a chat bot). In our case, the channel has two clear end parts:

- The interaction channel with the user: composed by the Alexa assistant, embedded in the Amazon Echo Dot device, and connected through Internet to the Alexa Voice Service (AVS), that runs on the Amazon Web Services (AWS) ecosystem<sup>4</sup>.
- The devices in charge of running the required commands, that may be robots or other devices.

Within both end parts, the third part is a remote server, in charge of managing the interaction. Messages will be managed using a publish/subscribe model. The open-source Eclipse Mosquitto<sup>5</sup> broker (MQTT protocol<sup>6</sup>) was chosen for carrying out this channel. Mosquitto is lightweight and very suitable for use with low-power boards. Finally, the required chat bot is implemented using the RedBot platform, based on Node-RED. All technologies will be reviewed in Sect. 4.



**Fig. 1.** Overview of the proposed system architecture

## 3 Natural Language Processing (NLP) Using the Alexa Voice Assistant

Natural Language Processing (NLP) is a field of computer engineering and artificial intelligence (AI) that focuses on voice-based natural interactions between

<sup>4</sup> <https://developer.amazon.com/de/alexa-voice-service>.

<sup>5</sup> <https://mosquitto.org/>.

<sup>6</sup> <http://mqtt.org/>.

humans and computers. While research in NLP started in the early '50s, results have been traditionally constrained due to the complexity of getting useful data from voice. The problem has not a trivial solution and, in fact, in order to use voice for human-robot interaction, several stages related to different problems must be solved. The first stage, ***speech recognition***, deals with the translation of spoken language into text. This process is known as Speech To Text (STT), and it provides a text output that allows a computer processing the potential data contained in voice in a machine-readable format [4]. The second stage is in charge of solving the ***natural language understanding*** (NLU). The aim of the NLU is to post-process the text coming from the STT, to interpret and understand the meaning of a user query or sentence in a non-structured format, and to convert it into an structured representation [5]. In short, NLU tries to understand a whole situation context instead of understanding individual words or phrases. The last stage is known as ***natural language generation*** (NLG). The NLG aims to generate natural language automatically mainly using structured text (although it is also possible to generate natural language from voice). Thus, NLG is the reverse stage of the speech recognition phase, so it is also known as Text To Speech (TTS).

### 3.1 NLP in the Alexa Assistant

Alexa assistant tackles the problem of NLP in an easy way for programmers. Regarding STT, the Amazon's Alexa assistant implements a cloud-based STT engine that delivers the translated voice through its application programming interface (API). But the process is transparent to the programmers: all they need to do is to associate syntactical structures with actions. The syntactical structures correspond with voice translations, the so-called ***utterances***, and the associated actions are called ***intents***. The Amazon's AI engine is designed to learn as the users use the assistant, so the STT becomes better with time. This is the reason why Amazon stores the user queries: to continuously train its AI engine with new data.

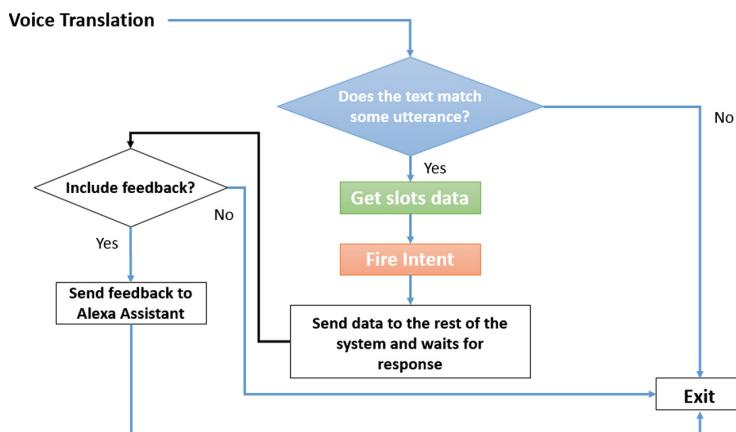
After the voice is translated, Alexa assistant allows programmers to extract useful information from utterances. This is how Amazon brings programmers the NLU capabilities. To extract specific data from utterances, Alexa uses the concept of ***slots***. Slots are variables of a defined type that programmers can embed into utterances, so these ones look like regular expressions or patterns. Alexa assistant only fires an intent when a translation perfectly matches with an associated utterance. In that moment, the assistant delivers to the rest of the system the specific data extracted from utterances through slots. Once an intent is executed, Alexa can deliver a customized spoken feedback to the user. This feedback is generated from text data in a NLG phase. This is a recommended, but optional, feature of this kind of smart devices. The major difference with other competitors is that Alexa assistant eases this work. The Amazon's assistant communicates outside using JSON-formatted messages, and to include a spoken feedback programmers, it only needs to embed their customized messages into the original message received from Alexa and returns it back.

Thus, the whole NLP model of the Alexa assistant can be specified into JSON-formatted data. Figure 2 shows an example of NLP to command a robot to turn on or off the light of the living room. In the left side, the red-colored text corresponds with the application intents. The only one intent has several slots and utterances associated, which appear as green- and blue-colored text respectively. Each slot is of a defined data type, and they appear in figure as yellow-colored text. Note that, in the right side of the figure, each data type defined by the user is associated with a list of values it can take. The Amazon's assistant will only fire the intent *switchLight* when someone tells the echo device “tell him to turn **on** the **living room** light” or “tell him to turn **off** the **living room**

```

{
  "invocationName": "uma robot",
  "intents": [
    {
      "name": "switchLight",
      "slots": [
        {
          "name": "order",
          "type": "orderType"
        },
        {
          "name": "zone",
          "type": "zoneType"
        }
      ],
      "samples": [
        "tell him to turn {order} the {zone} light"
      ]
    }
  ],
  "types": [
    {
      "name": "orderType",
      "values": [
        {"name": {"value": "on"}},
        {"name": {"value": "off"}}
      ]
    },
    {
      "name": "zoneType",
      "values": [
        {"name": {"value": "living room"}}
      ]
    }
  ]
}
  
```

**Fig. 2.** Alexa NLP JSON model example



**Fig. 3.** Alexa NLP flow diagram

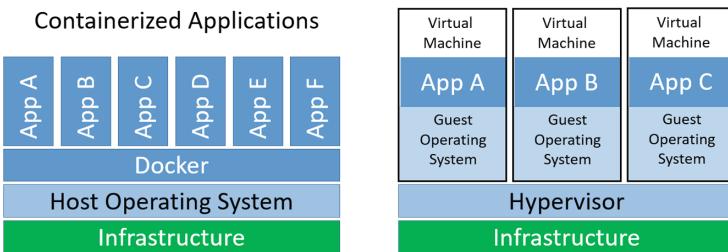
light”. Otherwise, no utterance will match with the given order and the voice assistant will do nothing. Figure 3 shows the Alexa assistant flow. It graphically draws the scheme described above.

## 4 Tools, Frameworks, and Services

To program scalable voice-based software for robotics platforms there are several open-source tools, frameworks, and services available. One of the main advantages of these resources is that they can be self-hosted so, if their source code or executable are removed from their repositories, our applications can continue running. Next subsections present the resources that our proposal uses.

### 4.1 Docker

Docker<sup>7</sup> is an open-source project to develop and deliver software in packages called *containers*. Containers provide a layer of abstraction and automation of virtualization of applications that can run on different operating systems (OS). Docker also provides a mechanism for resource isolation, allowing that independent containers can run over a single instance of an OS. Summarizing, Docker containers are something like lightweight virtual machines (see Fig. 4). In our proposal, Docker containers (in its version 18.06.1-ce) are used to deploy the whole system in an easy way. Thus, each self-hosted service, tool or framework is deployed into an independent Docker container.



**Fig. 4.** Docker vs. Virtual Machines

### 4.2 Serveo

Alexa applications require an end-point URL to send user commands outside the Amazon’s voice service server. This end-point is, in this case, the entry-point of the proposal we present in this paper. Amazon’s voice service is designed to communicate over secure connections, mainly using the https protocol. However, in our case, the entry-point of our system is of type http and not of type

<sup>7</sup> <https://docs.docker.com/engine/docker-overview/>.

[https<sup>8</sup>](https://www.alexa.com). Hence, http queries need to be encapsulated into https queries, providing a compatible end-point to the Alexa applications. This process is known as *tunneling*.

Serveo<sup>9</sup> is a SSH-based service for port forwarding. It generates a valid https URL that anybody can use to connect with a specific remote server or service. In order to get this https URL, it is necessary to execute the following command in the server side (see Command 1).

#### Command 1. Obtaining an https URL using Serveo

```
$ autossh -M 0 -o "ServerAliveInterval=120" -o "
    ↪ ServerAliveCountMax=3" -R <custom_url>:80:localhost:<
    ↪ own_service_port> serveo.net
```

In Command 1, *autossh* is a Linux command that automatically restarts SSH sessions and tunnels. The autossh command is configured by setting several parameters. Among the ssh parameters, we found the *ServerAliveInterval* and *ServerAliveCountMax*. The first one indicates the number of seconds that the client will wait before sending a null packet to the server (to keep the connection alive). The second one indicates how long a client is allowed to stay unresponsive before being disconnected. Also, the -R option tells your SSH client to request port forwarding from the server and proxy requests to the specified host and port (usually localhost).

### 4.3 Node-RED and RedBot

Node-RED<sup>10</sup> is a very popular programming tool for wiring together hardware devices, APIs and online services as interconnected flows. It provides a browser-based editor that eases programming new application flows. Node-RED provides functionality through artifacts called **nodes**. Nodes are configurable artifacts providing a specific functionality. For instance, we can define nodes implementing REST for rapidly deployment online services. A node output can be connected to one or several nodes input. In short, different nodes can interconnect with each other composing **flows**. Moreover, Node-RED supports custom functionality by writing JavaScript code. This feature makes Node-RED a powerful application programming tool. Finally, a flow (or a set of them) models, implements and executes a whole system. Figure 5 shows two http REST services that have been modeled and developed using Node-RED. In our proposal, Node-RED (in its version v0.19.5) is mainly used to process data from Alexa assistant but also to model the interactions among all the components belonging to the system.

RedBot<sup>11</sup> (in our proposal using version 0.16.8) is an open-source Node-RED-based chat bot platform. This platform allows building full chat bots for

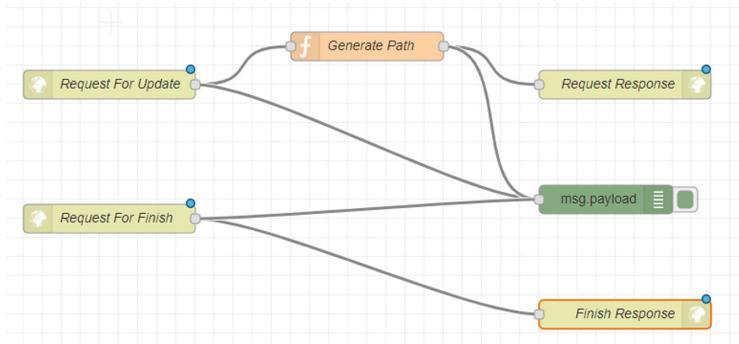
---

<sup>8</sup> We do not have valid signed-certificates for the Alexa SDK and, furthermore, the chosen communication protocol, http or https, do not affect to the designed flow.

<sup>9</sup> <https://serveo.net/>.

<sup>10</sup> <https://nodered.org/>.

<sup>11</sup> [red-bot.io/](http://red-bot.io/).



**Fig. 5.** Two http REST services developed as a Node-RED flow

Telegram, Facebook, Alexa, etc. through the previously described Node-RED's nodes. With respect to Alexa, RedBot provides, among others, nodes that act as entry-points for the Alexa queries; nodes that filter Alexa intents; nodes for inserting the user feedback as plain text (that the Alexa engine will later translate into natural language using its NLG engine); and nodes for sending back the final response to the user query.

Although these Alexa-related nodes are the most important ones for the presented proposal, RedBot provides interesting functionality in the field of NLP. For example, programmers can use RedBot to develop custom chat bots (using RiveScript<sup>12</sup>, Recast.ai<sup>13</sup>, Dialogflow<sup>14</sup> or other NLP algorithms) and to easily extract relevant information (like numbers, email, etc.) from user's sentences.

#### 4.4 Mosquitto

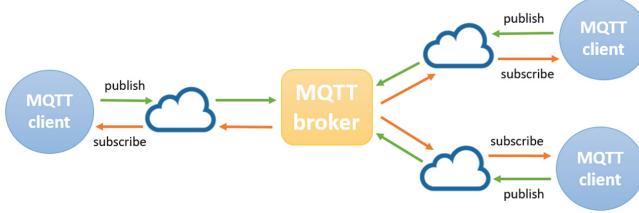
To explain what Mosquitto is, it is necessary to start explaining what MQTT is. MQTT (Message Queue Telemetry Transport) is a lightweight machine-to-machine (M2M)/IoT communication protocol that is based in a publish-subscribe mechanism. The MQTT publish-subscribe mechanism manages messages tagged under a plain name called ***topic***. In short, in a MQTT-based system, some software entities will send messages out using custom topics. Those messages will later be received by all software entities that are subscribed to the same topics. But in order to perform communication between two remote software entities, MQTT needs a ***broker*** (see Fig. 6).

A MQTT broker is the element in charge of managing the network and transmitting messages. The most popular one is the Eclipse Mosquitto (simply called Mosquitto). Mosquitto is an open-source lightweight message broker that is suitable for use on all devices, from low-power single board computers to full

<sup>12</sup> <https://www.rivescript.com/>.

<sup>13</sup> <https://cai.tools.sap/>.

<sup>14</sup> <https://dialogflow.com/>.



**Fig. 6.** The MQTT star topology

servers. It is fully configurable and has plenty of features that allow programmers, for instance, to develop secure communications. Both MQTT and Mosquitto are designed to run over low bandwidth and high latency networks. Moreover, they require low computation capacity so they are one of the best choices when working with LE devices. For these reasons, they have become one of the most popular combinations in the IoT field. In our proposal, they are used to communicate the external agents (distributed robots, in our application scenario) among them, and with the Node-RED server.

#### 4.5 ROS

ROS<sup>15</sup> (Robot Operating System) is an open-source, meta-operating system to program robots. It provides OS features, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management among others. It also provides a set of tools and libraries to obtain, build, write, and run code across multiple computers. ROS models systems as a *runtime graph* where processes are in a peer-to-peer network. The processes in the runtime graph are loosely coupled using the ROS communication infrastructure. ROS implements several different styles of communication, including synchronous RPC-style (Remote Procedure Call) communication over services, asynchronous RTPS (Real-Time Publish-Subscribe) of data over topics, and storage of data on a Parameter Server.

In our proposal, ROS (using the kinetic version) is used in the robot side. Its objective is to model the robot system and execute the necessary logic to carry out the commands received from the user.

### 5 Embedded Systems as Dedicated Services for Voice-Based Applications

In Sect. 4, we presented the open-source-based resources employed in the proposed system. All these tools only require a few computation resources and therefore they are suitable to be hosted in single board embedded computers. In our experiments, two different single board embedded computers have been

<sup>15</sup> <https://www.ros.org/>.

**Table 1.** Main features of the Odroid xu4 and Raspberry pi 3 model B+

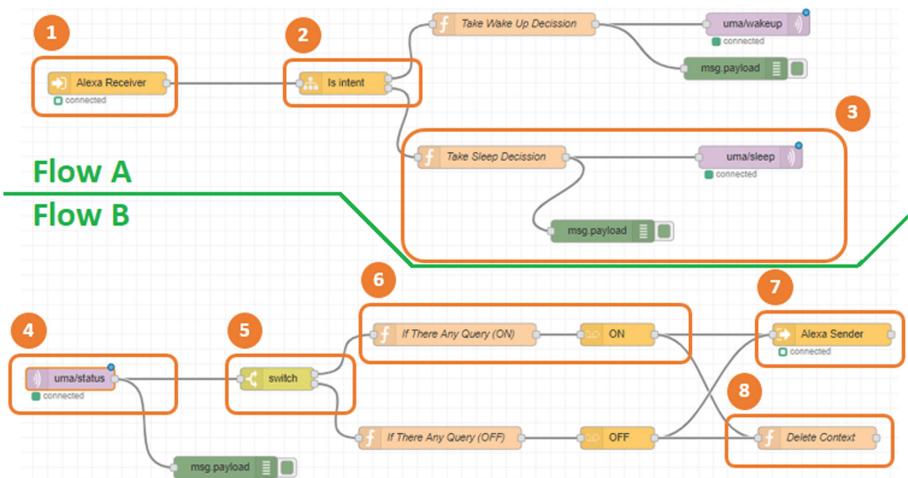
Feature	Odroid xu4	Raspberri pi 3 model B+
Processor	Octa ARM Cortex-A15 Quad 2 GHz Cortex-A7 Quad 1.3 GHz CPUs	Cortex-A53 (ARMv8) 1.4 GHz
Memory	2 GB LPDDR3 RAM	1 GB LPDDR2 SDRAM
Graphics	Mali-T628 MP6	Broadcom Videocore-IV
OS	Ubuntu 18.04	Raspbian
Consumption	up to 15 W	up to 5.1 W

tested: the Odroid xu4 and the Raspberry pi 3 model B+. Table 1 summarizes the specifications for both devices. As the table shows, the Odroid board is worse than the Raspberry one in terms of consumption, however it has a better performance. As the remote server (running Node-RED, RedBot, and the Mosquitto MQTT broker) is always connected to the electrical network, the power consumption is not a critical parameter at this point. Also, and due to the fact that the remote server executes a very important part of the system (the Node-RED flows interconnecting the systems entities), an extra processing capacity is desirable. For these reasons, the Odroid xu4 was the selected board to act as remote server. Robots, on the other hand, are limited by its batteries, so the power consumption is a critical parameter. This is the main reason why the Raspberry pi board was selected as a robot controller, because it has enough computation capacity to meet software requirements (mainly to execute ROS) and it presents lower energy consumption than the Odroid board.

## 6 Designed Node-RED Flows

Several Node-RED flows were designed and developed to interconnect the Alexa assistant with different devices (robots or other actuators). In this section we present one of these Node-RED flows (see Fig. 7). In many cases, these actuators are home-made devices, not officially supported by the Alexa ecosystem. So, this flow can be seen as a starting point for those researchers who want to control their devices using the Amazon’s voice assistant as a hands-free interface. Figure 7 provides a numbered snapshot of our proposed architecture. Numbers correspond to the most relevant parts of the flow, that is detailed below:

- Alexa Receiver:** This is the node acting as the Alexa application’s endpoint. It receives user queries as JSON-formatted documents.
- Is intent:** It connects each Alexa assistant’s intents with a specific node output. In short, it acts as an intents selector capable of modifying the Node-RED flow.
- Take decisions and send data to external agents:** Once an intent is selected, a JavaScript-based function takes a decision depending on the JSON at the input. It extracts the payload and sends the slot variables via MQTT



**Fig. 7.** Designed Node-RED flows (A and B) to interconnect (home-made) devices with the Alexa voice-based assistant

using a custom JSON message. Before the function finishes, it saves the original message into the *Node-RED context* to allow the rest of the nodes to use it later (if needed). Furthermore, a debugging node is used (green-colored node) to test functionality through the Node-RED's *Debugging Window*.

4. **MQTT subscription node:** This node receives an end-device's status via MQTT subscriber. After that, it sends the received response (the message payload) to the next stage in the flow.
5. **Switch node:** This node checks the message payload received from the previous stage, and informs whether the end-device is ready to perform any command sent by the user ('ON'), or not ('OFF').
6. **Alexa Speech and Query-Recognition:** A Javascript-based function checks if the current execution was caused by an user order through the Alexa assistant. This step is necessary as other non voice-based systems, which are also interacting with the robot, could have triggered this order. If the result is 'ON' or 'OFF', a speech is performed by the Alexa node reporting its status to the user. Note that the speech node uses the original message that was stored in context to carry out the NLG phase.
7. **Alexa Sender:** This is an output node to send back responses to the Alexa-based device. A queue with all given commands is set.
8. **Delete Context:** This function deletes messages in context to allow new queries to be executed correctly.

## 7 Conclusions and Future Work

Given the interest of large companies on introducing networked virtual assistants at home, it is expected that these devices will become popular in the

next years. These assistants are equipped with artificially intelligent frameworks, which allow users to interact with them and also to use them as interfaces for commanding other devices hands-free. This scenario offers the user the possibility to interface a smart environment using natural speech. In other words, it increases accessibility of smart environments, becoming a tool to increase autonomy for the most fragile people, such as the elderly or users with disabilities.

In this paper, we describe how the virtual assistant can be connected with other devices to allow users commanding them. There is no novelties in the contribution, but an ordered summary of all technologies involved in our proposed architecture is presented, and as described in the paper, the use of some of these tools is not straightforward. The paper pursues to ease the programmers to create a VUI for controlling their home-made (or not) devices, using tools that have proven their robustness and flexibility (referred to programming) in everyday environments. This is the main contribution of this paper.

Future work focuses on extending this framework to a more ambitious scenario: interacting with elderly people in retirement houses. This application scenario is considered in the projects that are currently being addressed in our research group, and that continue the research topics of recently finished projects such as CLARC<sup>16</sup> or LifeBots<sup>17</sup>. In this scenario, a robust VUI is clearly required to increase the possibilities of a socially assistive robot to be accepted and used. Working with elderly people in retirement homes, and using the proposed VUI, our robots will be able to provide several services such as suggesting physical or cognitive activities, or showing information about the weather or menu. With the help of the virtual assistant and chat bot, we will also integrate the robots with other devices in the environment, increasing the repertoire of activities to be offered to the users, and allowing these robots become a familiar and welcomed presence in the shared environment of the retirement house.

## References

1. Gartner's it glossary. <https://www.gartner.com/it-glossary/virtual-assistant-va>. Accessed 15 June 2019
2. Robotics 2020 multi-annual roadmap for robotics in Europe. Technical report, SPARC: The partnership for robotics in Europe. The EU framework programme for research and innovation (2015)
3. Bandera, A., et al.: CLARC: a robotic architecture for comprehensive geriatric assessment. In: Proceedings of XVII Workshop of Physical Agents (WAF 2016), Málaga, Spain, 16–17 June 2016
4. Chung, Y., Weng, W., Tong, S., Glass, J.: Towards unsupervised speech-to-text translation. Computation and Language [arXiv:1811.01307](https://arxiv.org/abs/1811.01307) (2018)
5. Wachter, M., Ovchinnikova, E., Wittenbeck, V., Kaiser, P., Szedmak, S., Mustafa, W., Kraft, D., Krüger, N., Piater, J., Asfour, T.: Integrating multi-purpose natural language understanding, robots memory, and symbolic planning for task execution in humanoid robots. Robot. Auton. Syst. **99**, 148–165 (2018)

<sup>16</sup> <http://www.clarc-echord.eu/>.

<sup>17</sup> <http://www.plg.inf.uc3m.es/~lifebots/>.



# Adapting ROS Logs to Facilitate Transparency and Accountability in Service Robotics

Francisco Javier Rodríguez-Lera<sup>1</sup>(✉), Ángel Manuel Guerrero-Higuera<sup>1</sup>, Francisco Martín-Rico<sup>2</sup>, Jonathan Gines<sup>2</sup>, Juan Felipe García Sierra<sup>1</sup>, and Vicente Matellán-Olivera<sup>3</sup>

<sup>1</sup> Robotics Group, Department of Mechanical, Computer and Aerospace Engineering, University of León, 24071 León, Spain

fjrodl@unileon.es

<sup>2</sup> Intelligent Robotics Lab, Universidad Rey Juan Carlos, Madrid, Spain

<sup>3</sup> Supercomputación de Castilla y León (SCAYLE), 24071 León, Spain

**Abstract.** Understanding why a particular robot behavior was triggered is a cornerstone for having human-acceptable social robots. Thus, every robot action should be explained in order to audit, recognize, and control expected and unexpected behaviors using usual sources of information provided by robot software components. Enabling intelligibility goes through a correct translation of system logs. This paper presents an initial approach for arranging low-level knowledge extracted from logs available in the *de facto* standard for service robotics ROS in order to be useful for developers and regulators.

**Keywords:** Accountable robotics · ROS · Trusted robotic platform · Accountability · Logging · Continuous auditing

## 1 Introduction

When thinking in commercializing a robot platform, it is necessary to fulfill different regulations attending the field of use. For instance, the Medical Electrical Equipment has the rule EC 60601-1-10, which provides the general requirements for basic safety and essential performance for a set of devices devoted to health-care. This rule presents the idea of an auditing system supported on logs. These logs should provide enough information: to determine the root of a transient behavior on a device; to determine the root of unexpected behaviors when the device is in regular use; to enhance the quality assurance process of solutions integrated on a device; and to promote a mechanism for studying the device status along the time after a catastrophe.

Although logging register has been widely faced in the software industry, beyond the industrial robots, most of the autonomous robots deployed in real-world environments lack standardized mechanisms for auditing and mapping

each robot behavior with the stack of software components that generated it. For instance, automatic credit reporting is heavily regulated, and different laws have created to guarantee people's right to know the logic involved in automatic decision-making, for example, about creditworthiness. When a robot is deployed in a supermarket, it should be a clear protocol for mapping each robot behavior with the set of software components that triggered it, notwithstanding, there is no regulated mechanism for doing it.

Thus, it is necessary to perform manual auditing of those logs generated by the Operative System, any middleware software, and final applications. This process needs to ensure that any robot action may be mapped uniquely to one or many software components. Manual auditing is a complex process and to perform it with a certain level of trust, the logs require a particular level of accuracy, integrity, and consistency of all data generated.

A common strategy used for dealing with accountability in distributed approaches is to perform a chronological record of all logs generated by the network nodes [12]. In a robotic environment, the solution is to perform accountability based on monitoring, registration and secure data-recording mechanisms of all robots logs. This approach allows reconstruction and examination of the sequence of actions and environment status, leading to generate robot behavior from inception to final results [8]. However, the log system has to show accuracy, integrity, and consistency of the information generated by each robot component. Thus, it is easy to find out who is legally responsible for any behavior performed by an autonomous agent.

Our research aims at defining a framework for the challenging problem of performing the accountability process on autonomous robots supported on logs. This work presents the first approach for dealing with ROS mechanisms of logging for its use in auditing environments. ROS logging engine deals with data generated from a given set of internal or external events that come from proprioception, interoception and exteroception. This data is known as logs and corresponds with raw information. This work reviews the current process and adds some features such as centralization, avoiding missing information, and presenting a mechanism for storing the log encrypted.

The rest of this paper is organized as follows. Section 2 describes our proposal, first at the cognitive level and then at the development framework level. Section 3 describes two alternatives for implementing an accountability module. Section 4 discusses the use of both proposals in a basic scenario. Finally, Sect. 5 summarizes the conclusions and further work.

## 2 Accountability Design for Autonomous Agents

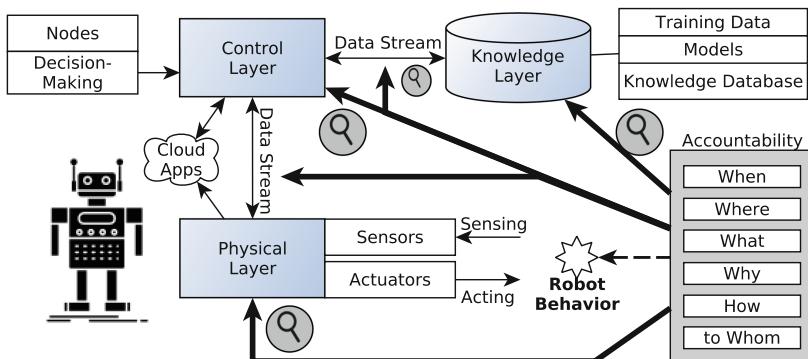
The design of an accountability process has to take into account the system resources such as computational, storage, and communications, and the middleware deployed. Current distributed approaches where the decision-making process is performed off-board or in cloud solutions turn out to be even more problematic because it has associated a vast number of components at a different level of abstraction.

The accountability process is intended to be fair (free from bias, dishonesty, or injustice), transparent, explainable and accountable. It is also essential for an accountability process to be transparent for the robot operation. Finally, it has to be explainable to ensure that every action the robot takes may be audited and associated with a software component.

A solution to this problem involves the use of components that implement the accountability mechanisms integrated into the core of the middleware used. On the other hand, we also think that this module has to be considered in the cognitive part of the control architecture [4]. However, this paper aims to deal with the accountability process at the middleware level, in particular at the ROS core level.

## 2.1 Overview

Figure 1 illustrates the main elements involved in the behavior generation on a service robot. Four components define robot behavior in an environment based on distributed solutions such as ROS and YARP:



**Fig. 1.** Behavior generation in Mobile Service Robotics.

1. **Physical Layer:** Those software components associated with robot hardware that make up a robotic system. They allow us to get information from the environment and to generate robot behaviors.
  - **Sensors:** identification of all input sources of our agent connected to onboard hardware or software sensors.
  - **Actuators:** association of all possible outputs for each actuator. It could define degrees of freedom, navigation possibilities, actions and robot behaviors.
2. **Control System:** This is the engine that determines the robot's behaviors. It is composed of several components which run in parallel.
  - **Nodes:** mechanism and engines used for solving the problems and performing tasks and operations. It includes different functionalities, such as perception and natural language processing.

- **Decision-making:** it includes and defines the architecture used for triggering robot behaviors under predefined circumstances. It means finite-state machines, deliberative architectures, or hybrid approaches.
3. **Database Layer:** Those source of information generated on runtime or beforehand for feeding the control system.
- **Knowledge Database:** knowledge representation of internal/environment robot status.
  - **Training Datasets:** those datasets used by one or many components of the control system that is used for learning how to classify external or internal stimuli. It is necessary to determine the origin of those sources of information used in the system. Their kind, synthetic or real would be gathered at run-time or off-line in controlled, semi-controlled or free world scenarios.
  - **Models definition:** Determine the features and parameters for modeling and defining the knowledge database and algorithms.
4. **Data Stream:** those sequences of data generated for each robot component to receive or transmit information.
- **Connectivity frameworks:** defines those data streams available in the robot for addressing node communication. It could any communication protocol such as Extensible Messaging and Presence Protocol (XMPP), Data-Distribution Service (DDS), ZigBee, CAN (Controller Area Network) or AMQP (Advanced Message Queuing Protocol among others).

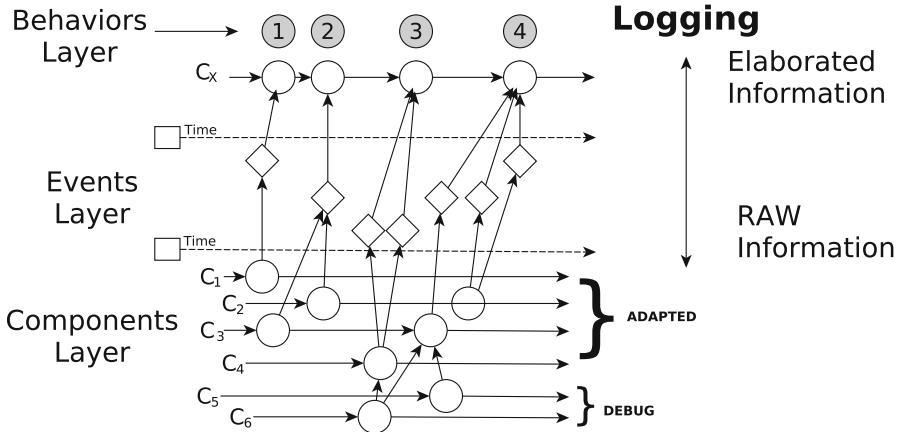
These four elements classically provide a set of different mechanisms for logging their states. Tracking the flow of these elements is possible to infer those components that generate robot behavior. The logging should answer the questions needed for accountability: when and where it happened, what components triggered it, why, how the final behavior was composed and to whom, identifying if other software components are consuming it.

## 2.2 Accountability Design

An AI-based system hides different levels of information abstraction. Thus, to perform the pairing software component-behavior mapping, it is necessary to define the reasoning process:

- (a) Single component, single event: The mapping links a single robot event with a single software component. E.g., the Perception component recognizes a bottle.
- (b) Multiple components, single event: The mapping links a robot event with two or more robot components. E.g., an operator defines the robot role as “Game Mode”, and the robot perceives an individual; as a consequence, the robot talks.
- (c) Single component, multiple event: The mapping links multiple robot event with a single robot component. E.g., the robot triggers two events: tracking humans event and talk to people because a software component perceived an individual in the scenario.

- (d) Multiple components, multiple events: The mapping links multiple robot events with multiple robot components. E.g., several components are triggering events given the interaction with an individual.
- (e) Single event, single behavior: The mapping links an event with a single robot behavior. E.g., a particular event changes the robot behavior (temporal event or recognition event or dialogue event).
- (f) Multiple events, single behavior: The mapping links multiple robot events with a single robot behavior. E.g., multiple events change the robot behavior (temporal event and recognition event and dialogue event).



**Fig. 2.** Accountability overview using logging.

This reasoning process is depicted in Fig. 2. Each software component triggers different events attending its nature. As a result, an *Accountable Logging* mapping is performed through an auditing process of raw information extracted from logs. In this process, it is necessary to cross the events generated by each component. Then, it is possible to link the behavior generated as a result of a set of these events. Based on these assumptions, it is possible to consider three labels for classifying the accountability layers.

The **Component layer** defines the method of accountability based on the information dumped by all software components running on the system. This study proposes two different modes inspired by [2]: DEBUG and ADAPTED. In the scenario of someone trying to understand a robot behavior looking at logs, when accounts at DEBUG, the user needs to check all log engines running in the autonomous robot: OS, middleware and application. Nevertheless, this process is usually performed in a bounded way, and the individual looking for information adopts the ADAPTED way, it means to dump a specific quantity of components logs to pay attention only to critical information about the robot behavior. Consequently, low-level information offered by the middleware is avoided.

The **Events layer** defines the method of accountability based on audit those concrete circumstances that arise as a result of one or many software components. In this scenario, an event represents a piece of information that represents something happening, such as recognizing a bottle or triggering a new inner status.

Finally, the **Behaviors layer** defines the accountable method in which it is possible to infer which components are generating a robot behavior and the events that trigger it. As a consequence of this mapping, it should be possible to map the software components that trigger them. The diagram depicted in Fig. 2 presents four accountable behaviors illustrated by stars, at the top level of the figure. These four behaviors are generated as a consequence of different events (illustrated by diamonds in the event layer) which at the same time are triggered by different software component occurrences.

There is, in fact, sufficient information present in the Component level for performing the accountability. However, the understanding and quality of this log are insufficient for other users different than software developers. Thus, the two extra labels allow the connection of every robot behavior with a set of events and the software components that generated these events. This project aims to prepare the ROS logging system for simplifying the auditing process having in mind those different actors that will require to know more about what is happening in the system.

### 2.3 Example

An example of these three levels could be illustrated with a service task extracted from any @home robotic competition: someone asks the robot for something.

The logging layer presents all information from all components of the system. Different log levels define the amount of information.

- Debug [Timestamp][Comm]Received requests
- Debug [Timestamp][Camera] Optimizing parameters
- Debug [Timestamp][Motor] Sending wheel request
- Adapted [Timestamp][Navigation] Moving the wheels 5 s at speed 1
- Adapted [Timestamp][Localization] Reaching position 3, 3
- Adapted [Timestamp] [Perception] Selected object 16

There is a set of events that are triggered as a result of a set of algorithms-plans:

- Info [Timestamp] Robot recognizes sentence: “Bring me the apple”
- Info [Timestamp] Robot getting current position: living room

There is a set of accountable behaviors that should be

- Info [Timestamp] [state 1] Starting service process.
- Info [Timestamp] [state 2] Attending individual.

This example summarizes dumped information generated by software components, moreover, highlights the main issues associated with the accountability process using logs:

- Verbosity: there is a massive amount of information. The process of mapping is hard to accomplish. The data is even more when working in DEBUG mode.
- Event-component mapping: several components could trigger an event, and this information would be presented with regular logs.
- Multi-layer auditing: every robot behavior is associated with the four layers presented above, with a high-level of verbosity, there is a problem for mapping robot behaviors with the data available. Besides, logging would be presented as a box that contains dumped information from all robot components (OS, middleware, applications).

### 3 Implementation

This section overviews current functionalities established by the ROS logging system and proposes a set of changes for enhance the auditing process. For reproducibility reasons, the set of samples defined in the rostutorials is being used for checking the log process.

#### 3.1 ROS Logging Description

Using logs for accountability implies four different actions: centrally collecting logs, retaining logs, analyzing/reviewing logs, and protecting logs. ROS has its logging methods based on state of the art engines. However, it has different logging weakness that complicates the accountability process.

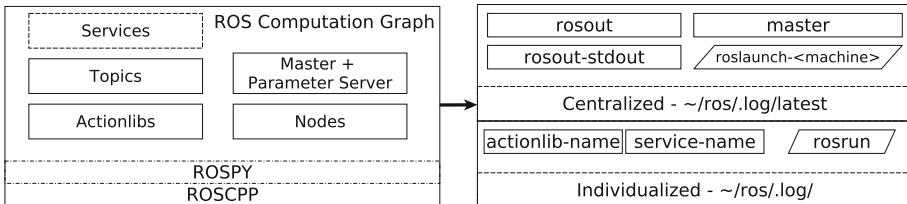
The Computation Graph Logging bounds the software components defined in ROS. It defines the ROS core concepts (ROS Master + Param Server and ROS nodes) and the communication between them (i.e., topics, services and action lib). In addition to that, it also has ROS Bags, a mechanism for storing robot messages and playing back.

ROS does have a centralized method for showing system logs, but it does not show all the process running in the system. The central collecting is performed into the `/home/ < user >/.ros/logs` folder, nevertheless the ROS logging process is seeding information in multiple files.

The logging process in ROS is driven by console log reporting called *rosout*. There are two broadcasting methods associated with the two clients available in ROS: the roscpp, supported on rosconsole, and the rospy's implementation supported on logging module. Rosconsole is a C++ package that supports console output and logging in roscpp client. It provides the interfaces that performs the logging report. This approach is supported on Apache log4cxx which supports hierarchical loggers, verbosity levels and configuration-files. Both of them share five verbosity levels, DEBUG (Received a message on topic X from caller Y), INFO (Node initialized), WARN (Could not save figuration file...), ERROR

(Received unexpected NaN value in transform X. Skipping...”), FATAL (Motor 1 has reached 90°, shutting down).

ROS defines a set of commands for showing the current status of system parameter server (list, get, set and dump), and the API for Python and C++. The outputs for accessing the parameter server are dumped in the *latest* folder in files: *rosout.log* and *latest/master.log*.



**Fig. 3.** ROS logging solution associated with ROS computation graph.

### 3.2 Accountability for ROS Developers

The first step for proceeding in accountability in ROS is the activation of the DEBUG level. Given the twofold approach ROSCPP and ROSPY used in ROS, it is necessary to update the configuration files on both cases for obtaining a fine-grain level of logs.

ROSCPP has by default a configuration file located in the folder defined by `$ROS_ROOT/config/rosconsole.config`. However, it is recommended to work over an environment variable that loads the log level configuration to the system. It means to export the path with the new configuration for setting a new output for ROS. For example, the file `debug_config.conf` would contain `log4j.logger.ros = DEBUG` instead the default `log4j.logger.ros = INFO`.

On the other hand, ROSPY defines the `Python_logging.conf` file, which is available in the `ROS_ROOT` folder. The file uses the standard `fileConfig` format defined by the Python logging module. Besides, ROSPY also presents the environment variable called `$ROS_PYTHON_LOG_CONFIG_FILE` that allows temporal overriding of by default values.

There is a third way of changing ROS logs levels using the tool `rqt_logger_level`. It provides a GUI for adjusting individually the level of each node running on the system.

ROS logging structure (depicted on Fig. 3) presents also an individualized approach based on the path `./ros/logs` and its nested folder: `./ros/logs/latest`. When running the `roscore` standalone, this folder contains four files that show the next output streams: `master.log`, `roslaunch- <hostname> - <roscore-pid>.log`

- Roslaunch: *roslaunch - <hostname> - <roscore\_pid> .log*

```
[roslaunch.<subprocess>] [protocol] [LOGLEVEL] TIMESTAMP: INFO
-- ----- Example -----
[xmlrpc] [INFO] 2019-07-03 09:47:31,973: Started XML-RPC server
[http://127.0.0.1:44151/]

[roslaunch] [INFO] 2019-07-03 09:47:32,091: master.is_running
[http://127.0.0.1:11311/]
```

- Rosout:*rosout.log*

```
TIMESTAMP LOGLEVEL TOPIC NODE_NAME [PATH] [TOPICS] [MESSAGE]
-- ----- Example -----
1562139473.540198537 INFO /talker [~path/talker.cpp:115(main)]
[topics: /rosout, /chatter] hello world 801
```

- Rosmaster

```
[rosmaster.<subprocess>] [LOGLEVEL] TIMESTAMP MESSAGE_TYPE [TOPIC_NAME] NODE_NAME ADDRESS
-- ----- Example -----
[rosmaster.master] [INFO] 2019-07-03 10:05:04,604: +SUB [/chatter]
/listener http://127.0.0.1:33461/
[rosmaster.master] [INFO] 2019-07-03 10:05:04,607: +PUB [/rosout]
/talker http://127.0.0.1:43471/
```

- Rosout (bis): *rosout - 1 - stdout.log*

```
output messages from rosout
-- ----- Example -----
re-publishing aggregated messages to /rosout_agg
subscribed to /rosout
```

The folder also contains a file linked to a node that creates and generates an event, particularly, showing the *rosout* logging info. Nevertheless, those nodes launched with *roslaunch* and those services triggered without invoking *ros.init()* will be located outside the latest folder, which increases the difficulty of a straight forward tracking. In order to obtain more information, we will need to track the filename, which is a sum-up of a *nodeName.pid.timestampSeconds*. This syntax is not working for those nodes launching services without *init*. These nodes just present the name and no extra information. Also, services running in this way, leave no trace in the rosmaster.log. For finding this information it is necessary to change the log level for running in DEBUG mode, while when the node has an *init* the information is presented in INFO mode.

## 4 Discussion

Although the accountability process would be fully supported on logs, it is not a direct translation from them natural language. The logs associated with an

event should be explicit and present detailed information in order to avoid event ambiguity. Furthermore, a good design of these logs should be faced as a mandatory process for enhancing the understanding of the robot accountable process to external experts and law regulators [2, 5, 7] and [9].

The minimal elements for explaining an event can be deduced from current approaches from data fields such in [10]. The elements for facing accountability should be:

1. Origin: Identifying who is generating the log event (where).
2. Timestamp: Information about the temporal window in which something is happening (when).
3. Owner: The user of the information access is revealed (who/toWhom).
4. Access: The method used for generating information (how).
5. Type of action: Define the category of the event (what).
6. Outcome: Results or evidence generated after the event was triggered (what).
7. Reason for access: The reason for the information access is revealed (why).

The scalability of the system also needs to delimit which data are recorded and how long. This meta-information is different depending on the publisher node, so we need to decide it for everyone. We require a configuration data repository to store meta-information data so the Accountability component to access it. Configuration data will include for each publisher node at least the following data: the message type, the periodicity and the circumstances required to record them.

Once decided which messages are going to be stored, we have to decide the moment to do it. Every time a publisher node sends a message to a topic, Accountability component will decide if it saves it according to the meta-information gathered for each publisher node.

In addition to that, it is necessary to face the impact of ciphering information for loggers. Although we prepared a naive proof-of-concept for evaluating the impact on the system, the best option goes for completely remove ROS debug logins, and move it to a specific file where everything is encrypted. The reason for this approach is that the messages identify the IP's of the origin, and this information would be used to identify some users.

Next snippet shows how the information would be presented in current log files for services, that have synchronization and special interest for performing several ROS actions.

```
[rospy.service] [DEBUG] 2019-07-05 01:27:06,895: connection from 127.0.0.1:49740
[accountability] [DEBUG] 2019-07-05 01:27:06,895: gAAAAABdHotKX5i19Fah0ZjP8qoP_JooLLsmr8c-VN9-
PN4xdD0exJ1xKw2qBD3DIwu80FUwq_XskNYmS1KjvxUjSzTb4q1g==
[rospy.tpcros] [DEBUG] 2019-07-05 01:27:06,895: [/add_two_ints]: writing header
[rospy.service] [DEBUG] 2019-07-05 01:27:06,895: /add_two_ints_server, /add_two_ints,
<class 'rospy_tutorials.srv._AddTwoInts.AddTwoInts'>
[accountability] [DEBUG] 2019-07-05 01:27:06,895: gAAAAABdHotKmi3M6NB5i60Xdmb0sd41a8f3ttWJ63CM
L1_8x-Iwc81SLxktUVJvKFj5ubSPYwKOOFYAdMrk1NCjfgpcm_6qJSFkwjJG_cafMOC_vgQ
TZkIWQo7R2tuK1bgIFus_Srk_XuGd14EVc7ZHwadnHEQFrqQ0nhmLnThA5XUnUK4EaBm4h
vEowZ77iaONFZNKDeoMO
```

The problem is that the use of DEBUG mode in ROS has an impact on the system, not only for the nodes but also it has an impact on the *roscore*.

To measure it we launch initially ten times the *roscore* (ROS Melodic) getting a summary of system resource usage using time command on an Ubuntu 18.04 it presents an increase of 75% on Voluntary context switches having from an average of 10500 to an average of 19000. In addition, also the involuntary context switches increases by around 25% rising from 53 to 65 (on average). Thus, we are decided to reduce the number of regular ROS logs and migrate it to our accountable option.

**Table 1.** ROS tools for robot logging and system monitoring.

Technology	Visualization	Data type	Description
ROS system monitor [11]	CLI	Raw data	A system monitoring tool for ROS (HDD, CPU, NTP, networking, and memory)
ARNI [1]	CLI	Raw data	Advanced ROS Network Introspection
Rosmon [6]	CLI	Raw data	A ROS node launcher with monitoring features
RQT logger level [3]	GUI	Raw data graphically	Graphical tools suite that wraps back-end logs

Besides, when the log system is dumping to file in DEBUG modes, the size of files would increase dramatically. For instance, when running turtlebot3 in gazebo world (3 nodes gazebo, gazebo\_gui, rosout), the rosout-1-stdout.log increases substantially, 1 megabyte every 3 s instead of a few bytes just in a basic state of up and running.

Results demonstrate that to perform accountability has overhead effects over the robot. ROS Logging becomes the primary tool for mapping behavior-component. However, there is a considerable number of ad-hoc tools created for evaluating different aspects of the robot, avoiding the impact on ROS logs. Table 1 presents some of these tools. They were selected based on two parameters: they are available in ROS repositories and they present a Github repository.

## 5 Conclusions

We believe that a standardized accountability system would facilitate behavior explanation and forensic analysis of safety and security incidents caused or suffered by service robots. However, to perform the accountability process using logs is not a trivial task. The process of increasing the logging level for mapping the robot behaviors has an impact on the limited resources available in resource-limited robots.

The initial approach for encrypting the logs to be protected from unauthorized access also has a computational cost associated. Besides, it is still missing

basic properties such as tamper resistance and verifiability, which are planned to be applied in the next stages of this research.

**Acknowledgment.** This work has been partially funded by Junta de Castilla y León and FEDER funds, Spain under Research Grant No. LE028P17, by Ministerio de Ciencia, Innovación y Universidades under Research Grant RTI2018-100683-B-I00 and by Ministerio de Economía and Competitividad of the Kingdom of Spain under RETOGAR project (TIN2016-76515-R).

## References

1. Bihlmaier, A., Hadlich, M., Wörn, H.: Advanced ROS network introspection (ARNI), pp. 651–670. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-26054-9\\_25](https://doi.org/10.1007/978-3-319-26054-9_25)
2. Butin, D., Chicote, M., Le Métayer, D.: Log design for accountability. In: 2013 IEEE Security and Privacy Workshops, pp. 1–7. IEEE (2013)
3. Thomas, D., Scholz, D., Kruse, T., Blasdel, A., Saito, I.: RQT common plugins (2013). [http://wiki.ros.org/rqt\\_common\\_plugins](http://wiki.ros.org/rqt_common_plugins). Accessed 29 Aug 2019
4. Guerrero-Higueras, Á.M., Rodríguez-Lera, F.J., Martín-Rico, F., Balsa-Comerón, J., Matellán-Olivera, V.: Accountability in mobile service robots. In: Workshop of Physical Agents, pp. 242–254. Springer (2018)
5. Le Metayer, D., Mazza, E., Potet, M.L.: Designing log architectures for legal evidence. In: 2010 8th IEEE International Conference on Software Engineering and Formal Methods, pp. 156–165. IEEE (2010)
6. Schwarz, M.: Rosmon (2018). <http://wiki.ros.org/rosmon>. Accessed 29 Aug 2019
7. Schneier, B., Kelsey, J.: Secure audit logs to support computer forensics. ACM Trans. Inf. Syst. Secur. (TISSEC) **2**(2), 159–176 (1999)
8. Shirey, R.: RFC 2828: Internet Security Glossary. Internet Soc. **13**, 151 (2000)
9. Waters, B.R., Balfanz, D., Durfee, G., Smetters, D.K.: Building an encrypted and searchable audit log. NDSS **4**, 5–6 (2004)
10. Wickramage, C., Sahama, T., Fidge, C.: Anatomy of log files: implications for information accountability measures. In: 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–6. IEEE (2016)
11. Willow Garage, Inc.: Maye, J., Kaestner, R.: ROS-system-monitor (2015). <https://github.com/ethz-asl/ros-system-monitor>. Accessed 29 Aug 2019
12. Xiao, Y., Yue, S., Fu, B., Ozdemir, S.: GlobalView: building global view with log files in a distributed/networked system for accountability. Secur. Commun. Netw. **7**(12), 2564–2586 (2014)



# A Story-Telling Social Robot with Emotion Recognition Capabilities for the Intellectually Challenged

David Azuar, Guillermo Gallud, Felix Escalona, Francisco Gomez-Donoso<sup>(✉)</sup>,  
and Miguel Cazorla

Institute for Computer Research, University of Alicante,  
P.O. Box 99, 03080 Alicante, Spain  
[{dazuar,ggallud}@dccia.ua.es](mailto:{dazuar,ggallud}@dccia.ua.es),  
[{felix.escalona,fgomez,miguel.cazorla}@ua.es](mailto:{felix.escalona,fgomez,miguel.cazorla}@ua.es)

**Abstract.** In this paper, a story-telling social robot is proposed. The robot is able to modify the evolution of the story considering the emotions the audience is feeling. To do that, the robot uses the user's emotion from his/her face. We have used a deep learning-based model to identify the emotion. This model was trained and tested on state of the art dataset. We also demonstrate that involving generated, realistic samples in the training process as a way of data augmentation does not benefit the model at all. The mentioned samples are generated by a state of the art GAN, which is able to translate neutral faces to a range of emotions.

**Keywords:** Social robotics · Emotion recognition · Data augmentation · GAN · CNN · Deep learning · Intellectually challenged

## 1 Introduction

It is well known that the technology is a big help when it comes to assist disabled people. For instance, there is a range of apps that reads text out loud for the blind using the smartphones' camera, colour identifiers for the daltonics, or maps that show the accessibility level of the buildings. Nonetheless, the amount of solutions for autistic and intellectually challenged people is reduced. Furthermore, the amount of adapted resources is even more reduced when it comes to cultural content like movies or shows despite the recreational and playful time is important for a healthy mind.

In this paper, we propose a story-telling app for autistic and other intellectually challenged people. Our application, which is intended to be deployed in a social robot, modifies the path of the stories it tells by considering the emotions the audience is feeling throughout the evolution of the tale.

To do so, we rely in a deep learning-based model which is in charge of recognizing the emotion of the audience by looking at their faces. The stories are organized as graphs. The nodes are blocks of the story and the edges are the

paths the robot would take considering the emotion. For instance, if the audience is feeling fearful during a certain block, the robot would choose the path that is tailored to calm them.

The facial emotion recognition is a tough task. Even the convolutional neural networks (CNNs) struggle in order to achieve high levels of accuracy. It could be due to the necessity of more data. In this work, we also explore the possibility of using generative adversarial networks (GANs) to generate more samples and improve the accuracy of the classifier.

The rest of the paper is organized as follows: first, a review of relevant state of the art applications for story-telling, data augmentation techniques and their impact on the deep learning algorithms are reviewed in Sect. 2. Then, in Sect. 3 we explain our approach for the story-telling app and the method we used to generate new realistic samples using GANs. The experiments and results of our proposal are shown in Sect. 4. Finally, the conclusions are stated in Sect. 5 along with the statement of future research directions of this work.

## 2 Related Works

There are some works in the state of the art related to story-telling software. For instance, in [21] the authors proposed an adaptive story application. In this work, the evolution of the story must be setup by a human agent at the beginning of each session. The authors tested the approach in a clinic environment with high success. The potential value of story telling for teaching social skills to autistic children was addressed in [3]. In this proposal, the software asked the audience to select the next block of the stories. The digital stories can also be used for improve the learning process of children with autism as proposed in [13].

The reviewed works agree in that the digital stories helped the intellectually challenged to improve their learning abilities. Nonetheless, they do not consider the mood or the emotion the children is feeling. The emotions play an important role as concluded by [17]. They developed a game in which the players must express a certain emotion when asked to. In fact, the use of assistive robots is also explored in [15]. They take advantage of a robot to teach emotional reactions to children with autism.

Our approach heavily relies in the emotion recognition system. The most novel approaches take advantage of convolutional neural networks (CNNs). For instance, in [16], the authors tested different architectures reaching a top accuracy of 75.2% on the FER2013 dataset. Furthermore, in [11], the authors explored more powerful architectures reaching a 66.67% accuracy on the very same dataset. The accuracy levels reached by the models that tackle this problem are not so high as the achieved on some other issues like object recognition. It is worth noting that the emotions are shown within a context, and it is difficult to state the emotion a person is feeling just by looking at a picture of it.

As mentioned before, the data augmentation is commonly used as a method to increment the number of training samples of a learning system in order to

obtain better generalization. In our case, the different datasets are clearly unbalanced. Data augmentation is commonly used to mitigate this problem. Next, some data augmentation methods in the context of CNNs are discussed.

First, it is worth noting that the traditional methods to perform data augmentation of images include affine transformations such as shearing, cropping, reflection, rotation and other modifications that slightly change the geometry of the samples. Furthermore, some other transformations like contrast addition or subtraction, histogram equalization, white balancing and sharpening are also applied [10].

The data augmentation process likely leads to an increased performance. For instance, in [7], the authors cropped the input image and flipped them horizontally also to generate more samples. The authors altered the intensities of the RGB channels too. In total, the training data was augmented by a factor of 2048. As a result, it helped the proposal to achieve the best results so far in the ImageNet [4] challenge. In [22], the authors tested the effect of several single data augmentation technique and each of them lead to an increased accuracy. Furthermore, the authors of [20] stated some of the best practices to take the most advantage of CNNs in which is remarked the role of the data augmentation.

As mentioned, the data augmentation has many benefits. Nonetheless, this process works as long as the semantic information of the sample is not modified. For instance, apply vertical flipping to samples that depict persons is not suitable as it is very unlike to find upside-down persons on the wild.

Finally, it is worth noting that the idea of using GANs to generate samples and perform data augmentation was previously tested in several areas. For instance, in [1] the authors used a GAN to generate new samples and train better classifiers. The authors of [14] propose the utilization of a style transfer GAN to generate samples. In [9] they propose a novel GAN architecture focused in generate data augmentation. Finally, [19] states that the utilization of a GAN to generate samples boosted the performance on different scenarios. In these works, the results are polarized. Some of the approaches achieved a remarkable improvement in the accuracy of the model, whilst others did not experiment any enhancement at all. We can conclude that the ability of the GAN to create visually realistic samples is the key to improve the accuracy of the classifiers.

It is also worth noting that we did not find any work in the state if the art that utilizes a GAN to improve the performance of an emotion classification convolutional neural network.

### 3 Approach

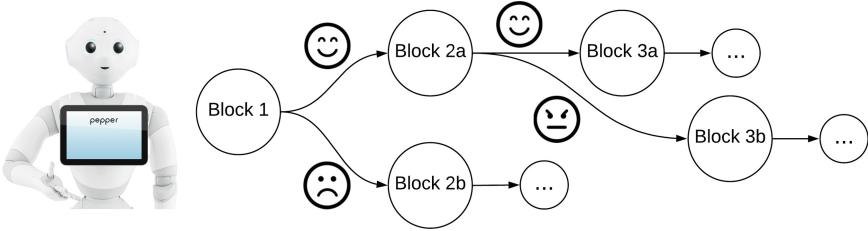
The aim of this work is to develop a social robot which is able to modify the evolution of a story by using emotion recognition capabilities. As mentioned before, a CNN is in charge of the emotion recognition capabilities. Nonetheless, as stated in the Sect. 2, the models of the state of the art still have room for improvement.

To boost the accuracy of an emotion recognition convolutional neural network we intend to use samples generated by a generative adversarial network. To do

so, we chose AffectNet [12], which is a state of the art dataset for emotion recognition. The samples were preprocessed so that the faces are all aligned the same way and undesirable artifacts removed. Then, the samples of each emotion were used for training a GAN that translates from neutral to the target emotion. Finally, these GAN models are used to generate a range of samples which are mixed with the original dataset. The resultant mix of original and generated samples are finally used to train a CNN for emotion recognition. Each piece of the proposal is thoroughly explained in the following subsections.

### 3.1 Story-Telling App for Social Robots with Emotion Recognition Capabilities

One of the main contributions of this work is the story-telling robot with emotion recognition capabilities which is able to modify the evolution of a story based on the emotions that the audience is feeling. The robot of choice is a Pepper robot. This is a social robot with high expressiveness capabilities. Its appearance is enough human-like to be appealing whilst not being unsettling. It features a range of different leds in the face which are intended to show emotions. It also has two arms, which are used to boost the emotional capabilities. The images of the audience are captured by the built-in camera.



**Fig. 1.** Story-telling app for social robots with emotion recognition capabilities

As shown in Fig. 1, a story is structured like a graph. Each node of the graph is a block of the story. The robot is monitoring the emotions of the audience throughout a block by capturing images with its camera and analyzing the emotions. When it reaches the end of a block, the emotions during the block are averaged and the most present is used to select the next block. The system considers 7 different emotions, as explained in Sect. 3.5. For instance, if the story intends to focus on happy emotions and a block triggers angry or fear, the next block would try and change that. Note that the same story block might cause different emotions to different persons.

### 3.2 Emotion Recognition Dataset

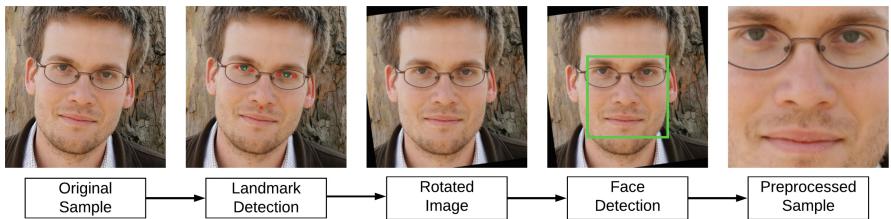
In order to create a deep learning-based emotion recognition, it is required a huge amount of labeled data. As mentioned before, the dataset of choice was

AffectNet. Some other datasets like the Extended Cohn-Kanade Dataset [8], FER Dataset [5] or EmotioNet [2] were considered. However, we discarded them because the images were in grayscale, yielded low resolution or the subjects were overacting.

AffectNet consist of 450,000 images of faces. The average image resolution is  $425 \times 425$ . The samples come from images and videos in the wild, so the depicted facial expressions are triggered by a genuine emotion. The dataset is distributed among the following categories: neutral, happy, sad, surprise, fear, disgust, and anger. The number of samples is highly unbalanced. We use different subsets of this dataset for the experiments.

### 3.3 Image Preprocessing

The images of AffectNet and real environments does not only include the face of the subjects but the full head and some background too. In addition, heads show a range of different orientations. These two factors could hinder the learning process of both GANs and CNN. Thus, to align the images so the faces yield similar orientations, we used the face landmark detection of the DLib framework. DLib is a deep learning-based system, which was trained on the i.bug Facial Point Annotations Dataset [18], which is able to detect 68 2D points corresponding to facial features distributed among the nose, eyes and mouth. We only consider the 4 points that states the start and the end of each eye. Using these points, a line is computed. Then, the rotation required to align horizontally the eyes is applied to the sample. This way, every face shares the same orientation. Finally, the face detector of the DLib framework is used to crop only the face. This way, the samples are all the same orientation and do not depict hair, background or other irrelevant features that could harm the learning process. It is worth noting that this preprocessing is applied to every sample we used either for training or testing the GANs and the CNNs. This process is shown in Fig. 2.



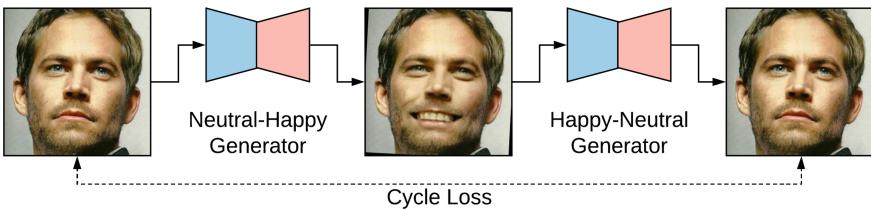
**Fig. 2.** Preprocessing steps performed to remove undesirable artifacts like hair or background and to align every face the same orientation.

### 3.4 Generation of Realistic New Samples Using GANs

Although the selected dataset contains lots of images, it is unbalanced. Several emotions contains less images than others. So, we need a method to balance

the data. To generate new samples we adopted the CycleGAN [23] approach. This particular kind of GAN is able to translate an image from a source domain to a different target domain and it does not require aligned samples and ground truth. Other traditional GANs would generate a sample from a noise vector. Each value of the input noise vector corresponds to a feature of the target domain. Other fully convolutional approaches require the same sample in both source and target domain to be trained. Nonetheless, CycleGAN is able to take as input any image of the source domain and any image of the target domain as output, and the system will learn the translation mechanism.

The CycleGAN approach is based on the minimization of the cycle loss. Actually, this implementation features two different generator networks. The first generator transforms from the source domain to the target domain, and the second generator do the inverse operation taking as an input the output of the first generator. Finally, the loss is computed between the input image and the output image of the second generator. This way, if this approach is trained to translate neutral faces to happy faces, the first generator would take the input neutral face and would transform it to a generated happy face. This generated happy face would be transformed to a generated neutral face by the second generator. Finally, the architecture would compute the loss between both actual and generated neutral face to try and minimize it. This part of the architecture is depicted in the Fig. 3. The discriminator part of the GAN takes as input pairs of real and generated images of the target domain. As a result, this architecture is able to generate visually appealing and realistic versions of the input sample in the target domain.



**Fig. 3.** Generator part of the CycleGAN approach.

In this work, we take advantage of this method to generate new samples depicting each of the target emotions from neutral faces. It is worth mentioning that we trained a different model for each target emotion. Furthermore, we chose to translate from neutral to each emotion because the neutral samples are the category with the most number of samples in any dataset, and because this emotion can not be overacted.

### 3.5 Emotion Recognition Using CNNs

To classify the images we adopted a ResNet-50 [6] architecture. This method is a state of the art deep CNN that achieves better performance than a human in

the ImageNet challenge. This network takes advantage of the residual concept. This is based on connecting the feature maps of a certain level with the output of the subsequent convolutional block. As a result, it is capable of decreasing the vanishing gradient problem and to enable deeper networks.

We adopted the architecture as is with a minor modification. The number of neurons of the last fully connected layer was modified to match the number of emotions considered in each experiment. The final model is intended to recognize the following emotions: neutral, happy, sad, surprise, fear, disgust, and anger.

## 4 Experimentation and Discussion

This work intends to demonstrate the influence of GAN generated samples. Thus, we next describe a range of experiments we carried out to benchmark this.

### 4.1 Generated Samples

We trained a GAN for the emotions happy, surprise and disgust. We chose these emotions over the rest because surprise and disgust are by far the category with the least representation in the datasets for emotion recognition in the state of the art. The happy category usually yields the most number of samples, so we used it as a benchmark.

As mentioned earlier, we trained the CycleGAN with the AffectNet dataset. A preprocessing step was applied to each image as described in Sect. 3.3. For the training, 2000 images were randomly selected from the dataset and the architecture was trained for 200 epochs. We monitored the cycle loss and the discriminator loss, and stopped the training when the GAN reached the Nash equilibrium. Finally, the results were visually inspected to validate that the generated images are realistic enough. This setup was used for each emotion, thus generating three different models that takes as an input neutral faces and generates samples of each emotion. The Fig. 4 shows some results of these models. As shown, the generated images are convincing enough.

### 4.2 Classification Accuracy

As mentioned in Sect. 3.5, we adopted the ResNet-50 architecture. We trained the architecture with different training data setups, but the test split remained the same across all experiments. A particular sample only pertains to a split and cannot be found on the rest. The remaining neutral faces of the dataset that do not appear in the training, validation or test splits are used as a source for the GAN in order to perform data augmentation. The original source neutral faces are only used once, so only one generated face across all the data setup corresponds to an original neutral face. The loss function was *categorical crossentropy*, which was optimized by Adam. The initial learning rate was of 0.01.

The results are shown in Table 1. In the first experiment, 10500 original samples were used for training. A 10% of which were used for validation. In this



**Fig. 4.** Different samples generated by the CycleGAN. The first column is the source domain (neutral) and the rest are the target domains (happy, surprise and disgust). Note that the faces were blended back to the original source image for visualization purposes only.

experiment the dataset was balanced and no data augmentation was applied. The model achieved a 49.7% accuracy. The error was evenly distributed among all the categories except happy. The happy category reached a 40% more of accuracy than the rest of the classes. This experiment is used as a baseline.

The same trend can be seen in the second experiment. In this case, 2500 samples per class were involved and no data augmentation was applied. This model achieved a test accuracy of 53.0%. Despite increasing the number of samples, only a marginal improvement was achieved.

The third experiment also achieved a similar result. In this experiment, 4500 original samples were involved for every category but for surprise and disgust. For the categories surprise and disgust only had 2500 original samples and 2000 generated samples. As mentioned earlier, surprise and disgust are the least represented in the emotion recognition datasets so we intend to evaluate the impact of using a GAN to generate new samples. As result shown, the influence of the augmented samples is not noticeable. In fact, the accuracy of the augmented categories do not allow a definitive conclusion as one class improved and the other worsen. The accuracy of this model is of 53.1%.

**Table 1.** Results over the test set for the first experiment which involved only original samples.

Accuracy (%)	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise	Mean acc.
2k original	39.2	46.8	46.0	84.6	49.4	38.8	42.8	49.7
3k original	41.0	51.0	51.0	89.6	47.8	53.0	38.0	53.0
2.5k original + 2k generated	50.6	44.8	43.2	85.8	55.4	46.0	46.0	53.1

An additional experiment was carried out to test the definitive influence of generated samples, leaving out the categories that are not augmented. In this experiment, we only involved the categories happy, disgust and surprise. We trained and evaluated two models: first, we took 2000 original samples and then, we took the same 2000 original samples and appended an additional 2000 generated samples. The results were 84% and 82% accuracy, respectively. Thus, leading again to the conclusion that the addition of GAN generated samples to the training dataset does not affect the final performance.

In the light of the experiments, it can be concluded that using GANs to perform data augmentation do not pose a noticeable impact over the accuracy of a CNNs for emotion recognition.

Besides that, it can also be concluded that the category most easy to classify is happy. It achieved a superior performance compared to the other categories in every experiment. This is because the visual features of a happy expression are very distinguishable from the rest, whilst the other classes yield so much subtle features.

It is worth noting that the final model integrated in the story-telling app for social robots was the one obtained in the third experiment because it was the best performer.

## 5 Conclusions, Limitations and Future Work

In this paper, a story-telling app for social robots with emotion recognition capabilities is proposed. A convolutional neural network is in charge of performing the emotion recognition. We explored the influence of involving generated, realistic samples in the training step. Despite the GAN in charge of doing so was able to generate good quality images, their inclusion in the training split did not improved the accuracy at all. The best performer model was trained on original data mixed with generated samples, and achieved an accuracy of 53.1%.

The GAN struggle to generate realistic samples of some faces as shown in Fig. 5. We observed the GAN tends to be error-prone on those samples that are less represented in the training dataset. In our case, afro-american people and people wearing glasses. In any case, the inclusion of generated samples in the training set barely affected the accuracy.

As future work we plan to test our story-telling social robot in a clinic environment with actual intellectually challenged children. In addition, we plan to



**Fig. 5.** Some visually non-appealing samples generated by the GAN. The left image depicts the neutral source domain and the left image depicts the happy version as generated by the GAN.

further improve the emotion recognition system by considering not only visual features of an instant but of a lapse of time.

Finally, a video of our system working can be seen at.<sup>1</sup>

**Acknowledgements.** This work has been supported by the Spanish Government TIN2016-76515R Grant, supported with Feder funds. This work has also been supported by a Spanish grant for PhD studies ACIF/2017/243 and FPU16/00887. Thanks to Nvidia for the generous donation of a Titan Xp and a Quadro P6000.

## References

1. Antoniou, A., Storkey, A., Edwards, H.: Augmenting image classifiers using data augmentation generative adversarial networks. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) Artificial Neural Networks and Machine Learning - ICANN 2018, pp. 594–603. Springer, Cham (2018)
2. Benitez-Quiroz, C.F., Srinivasan, R., Feng, Q., Wang, Y., Martínez, A.M.: EmotionNet challenge: recognition of facial expressions of emotion in the wild. CoRR. <http://arxiv.org/abs/1703.01210> (2017)
3. Chatzara, K., Karagiannidis, C., Mavropoulou, S., Stamatis, D.: Digital storytelling for children with autism: software development and pilot application, pp. 287–300. Springer New York (2014). [https://doi.org/10.1007/978-1-4614-6501-0\\_19](https://doi.org/10.1007/978-1-4614-6501-0_19)
4. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (June 2009)
5. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hammer, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Grozea, C., Bergstra, J., Xie, J., Romaszko, L., Xu, B., Chuang, Z., Bengio, Y.: Challenges in representation learning: a report on three machine learning contests. Neural Netw. **64**, 59–63 (2015). <http://www.sciencedirect.com/science/article/pii/S0893608014002159>, special Issue on “Deep Learning of Representations”
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition supplementary materials (2016)

<sup>1</sup> <http://www.rovit.ua.es/videos/socialrobot.mp4>.

7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, vol. 1, pp. 1097–1105. NIPS 2012, Curran Associates Inc., USA (2012). <http://dl.acm.org/citation.cfm?id=2999134.2999257>
8. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended Cohn-Kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, pp. 94–101 (June 2010)
9. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, A.C.I.: Bagan: data augmentation with balancing GAN. [arXiv:1803.09655](https://arxiv.org/abs/1803.09655) (2018)
10. Mikolajczyk, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. In: 2018 International Interdisciplinary PhD Workshop (IIPhDW), pp. 117–122 (May 2018)
11. Quinn, M.-A., Grant Sivesind, G.R.: Real-time emotion recognition from facial expressions. CS229 - Stanford University. <http://cs229.stanford.edu/proj2017/final-posters/5165034.pdf>
12. Mollahosseini, A., Hassani, B., Mahoor, M.H.: AffectNet: a database for facial expression, valence, and arousal computing in the wild. CoRR (2017). [http://arxiv.org/abs/1708.03985](https://arxiv.org/abs/1708.03985)
13. Parsons, S., Guldberg, K., Porayska-Pomsta, K., Lee, R.: Digital stories as a method for evidence-based practice and knowledge co-creation in technology-enhanced learning for children with autism. Int. J. Res. Meth. Educ. **38**(3), 247–271 (2015). <https://doi.org/10.1080/1743727X.2015.1019852>
14. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. [arXiv:1712.04621](https://arxiv.org/abs/1712.04621) (2017)
15. Pinto Costa, A., Steffgen, G., Rodriguez Lera, F.J., Nazarikhoram, A., Ziafati, P.: Socially assistive robots for teaching emotional abilities to children with autism spectrum disorder. In: 3rd Workshop on Child-Robot Interaction at HRI 2017 (2017)
16. Pramerdorfer, C., Kampel, M.: Facial expression recognition using convolutional neural networks: state of the art. [arXiv:1612.02903](https://arxiv.org/abs/1612.02903) (2016)
17. Rouhi, A., Catania, F., Cosentino, G., Gelsomini, M., Spitale, M.: Emotify: emotional game for children with autism spectrum disorder based-on machine learning (February 2019)
18. Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zaferiou, S., Pantic, M.: 300 faces in-the-wild challenge: database and results. Image Vis. Comput. **47**, 3–18 (2016). <http://www.sciencedirect.com/science/article/pii/S0262885616000147>, 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge
19. dos Santos Tanaka, F.H.K., Aranha, C.: Data augmentation using GANs. CoRR (2019). [http://arxiv.org/abs/1904.09135](https://arxiv.org/abs/1904.09135)
20. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings, pp. 958–963 (August 2003)
21. Stamatis, D., Karagiannidis, C., Mavropoulou, S.: Digital storytelling for children with autism: software development and pilot application, pp. 287–300 (September 2014)
22. Taylor, L., Nitschke, G.: Improving deep learning using generic data augmentation. CoRR (2017). [http://arxiv.org/abs/1708.06020](https://arxiv.org/abs/1708.06020)
23. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242–2251 (2017)



# A Novel Inter-device Calibration for Wi-Fi-aided Indoor Localization Systems

Miguel Martínez del Horro<sup>(✉)</sup>, Cristina Romero-González,  
Luis Orozco-Barbosa, and Ismael García-Varea

Instituto de Investigación en Informática de Albacete (I3A),  
Universidad de Castilla-La Mancha, Campus Univ. s/n, 02071 Albacete, Spain  
[{Miguel.MHorno,Cristina.RGonzalez,Luis.Orozco,Ismael.Garcia}@uclm.es](mailto:{Miguel.MHorno,Cristina.RGonzalez,Luis.Orozco,Ismael.Garcia}@uclm.es)

**Abstract.** Wi-Fi-based indoor localization mechanisms have attracted many research efforts in recent years due to the widespread use of this technology. All robots in indoor scenarios use this technology to provide Internet connection for Cloud services in speech understanding or human-robot interaction. However, this technology can also be used to provide localization services based on the Received Signal Strength Indicator (RSSI). Nevertheless, the majority of the current proposed indoor localization systems spend huge amounts of time in order to set-up the system in the target environment. In addition, given that the IEEE 802.11 standards leave the RSSI computation up to the manufacturers, each device which needs to be located has to survey the wireless platform to correctly calibrate the localization system. To overcome these drawbacks, this paper presents a novel inter-device calibration procedure for new potential devices which makes use of a previous calibration carried out by a different device. The proposed calibration procedure enables an on-the-fly configuration of any new device with a negligible loss of localization accuracy.

**Keywords:** Indoor robot localization · Wi-Fi based · Inter-device calibration procedure

## 1 Introduction

The need of localization services has substantially increased over the past decade [3, 13], since it is seen as one of the main pillars towards the successful implementation of context-based applications. In particular, the localization problem has a high relevance in fields such as robotics, where the robot's position is essential for the provision of services.

Although the localization problem in robots has been effectively solved, the existing solutions tend to occasionally lose accuracy in large or crowded indoor scenarios [10]. These solutions usually depend on complementary equipment to be deployed in the environment, such as Velodyne LiDAR [12], or to be embedded

in the robot, such as cameras [9]. However, these alternatives are not universal given the wide variety of robots on the market.

In order to improve the localization accuracy of those challenging situations, the robot's localization system can be combined with a complementary localization system. This is not a fusion of localization systems, but an assistance between different systems for such situations. In addition, the complementary localization system can also be used for other purposes such as human tracking. In this way, the robot is released from the continuous location of other agents, allowing its computational power to be used to enhance the human-robot interaction.

Among all the different approaches to develop indoor localization mechanisms [16], Wi-Fi-based is attracting most of the efforts due to two main reasons: (1) Wi-Fi technology is currently present in most indoor scenarios primarily to provide Internet connection, thus, the existing wireless platform can be reused to supply localization services without the need of investing money; and (2) Wi-Fi technology is currently present in most robots and daily-used devices to give access to Cloud services. However, its use in providing localization services involves several drawbacks due to the negative impact on the signal of the inherent indoor scenarios characteristics, such as walls, furniture and people.

Wi-Fi-based indoor localization systems are usually based on the Received Signal Strength Indicator (RSSI) [8], which measures the received power level by a Wi-Fi-enabled device and an access point (AP). This metric can be used to estimate the device's position, usually using the fingerprinting technique or a range-based model. The former creates a RSSI map in several positions of the environment, which is subsequently compared with the real-time RSSI to estimate the device's position using a supervised classification algorithm such as KNN or Random Forest. The latter allows estimating the distance between these devices, which is then used to compute the device's position using a trilateration algorithm. Although fingerprinting has proven to be useful in developing accurate positioning mechanisms [2], the RSSI map has to be updated every time there is a change in the environment, thus increasing the system maintenance time. In range-based models, the actual signal propagation is described by a set of parameters which are obtained from the captured RSSI values in the environment.

In one of our previous works [6], a Wi-Fi range-model-based in-motion calibration procedure capable of properly describing the signal propagation in a given environment was proposed. This procedure creates a map representing the path loss exponent in the whole environment of the widely-used Log-Distance Path Loss Model (LDPLM) [11], which can be expressed as:

$$P_r(d) = P_r(d_0) - 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) \quad (1)$$

where  $P_r(d)$  is the RSSI in dBm at  $d$  meters,  $P_r(d_0)$  is the RSSI in dBm at  $d_0$  meters and  $n$  is the path loss exponent (unitless). This exponent indicates how the radio signal spreads in a certain scenario.

The resulting signal propagation representation of this calibration procedure characterizes the signal attenuation of the calibrated device in each position of the environment. However, since the IEEE 802.11 standards leave the RSSI computation up to the manufacturers, distinct Wi-Fi chipsets may obtain different RSSI values measured at the same point of the environment. This is called the RSSI hardware variance problem [14]. Therefore, the signal propagation representation is only valid for the device being calibrated.

In this paper, we focus on making the signal propagation representation of our previous approach from one device valid for any other Wi-Fi-enabled device. That is, we focus on the calibration of new devices using the result of previous calibration. For that purpose, a inter-device calibration procedure is designed and tested in a real-world scenario.

The rest of this paper is organized as follows. First, Sect. 2 shows some related work. In Sect. 4, the previously proposed in-motion calibration procedure, which creates the baseline signal propagation representation, is introduced. The transformation of signal propagation representation between different devices is explained in Sect. 5. Section 6 presents the results of the proposed inter-device calibration procedure. Finally, the main conclusions of this work are drawn in Sect. 7.

## 2 Related Work

The development of Wi-Fi-based indoor localization systems and, specially, its calibration have been studied in recent years. While some authors propose self-calibrated localization systems [4, 5, 15], another authors try to solve the RSSI hardware variance problem between different devices [7, 14].

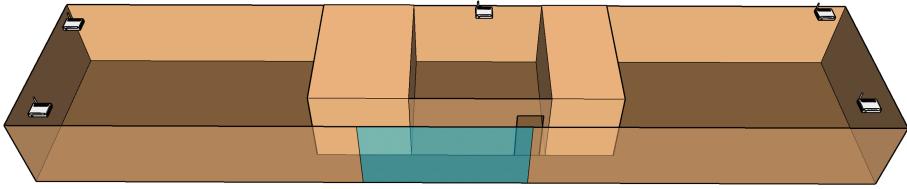
Self-calibration approaches aims to adapt the RSSI between different devices by using auxiliary devices which are continuously scanning the Wi-Fi network. Tuta et al. [15] modify the APs' firmware so that they emit Wi-Fi signal and capture the RSSI of the wireless platform at the same time. Using the captured RSSI, the path loss exponent is computed without any manual intervention. In [5], Fang et al. present some experimental results of three features against device diversity: Difference of Signal Strength (DIFF), Hyperbolic Location Fingerprinting (HLF) and Signal Strength Difference (SSD). Using these features, they are able to improve the accuracy with respect to raw RSSI. A similar approach is used in [4], where the authors compute the RSSI difference between pairs of APs instead of using the absolute RSSI in order to adjust the RSSI between different devices. Although these works prove to be able to calibrate new devices, they do not take into account the RSSI variance problem.

Works related to solving the RSSI variance problem try to determine the relationship between the RSSI values of different devices. In [7], Kim et al. apply a linear transformation to the RSSI of two devices. The calibration of the new device is performed by training a linear regression model, which thus converts the RSSI of the new device into the RSSI range of the previously calibrated device. In this way, the previous calibration can be used by the new device (both fingerprinting or range-based model approaches). Tsui et al. [14] go one step further by

introducing more complex models such as Expectation Maximization (EM) or neural networks. They train such models to convert RSSI values between devices and they compare its performance with a linear regression model. Although EM and neural networks carry out a better conversion than a linear regression model, the number of needed RSSI values increases substantially.

### 3 Test Environment Definition

We setup our experimental platform in one of the labs of the Albacete's Research Institute of Informatics, University of Castilla-La Mancha, Spain. Figure 1 shows the floor plan of the lab as well as the location of the APs (white squares) which compose the wireless platform. The lab dimensions are 28.7 m long and 6.4 m wide, i.e., its total area is 183.68 m<sup>2</sup>.

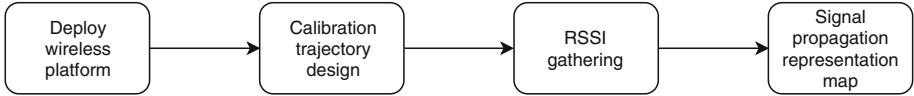


**Fig. 1.** Experimental environment layout. White squares denote the location of APs

### 4 Baseline Calibration Procedure

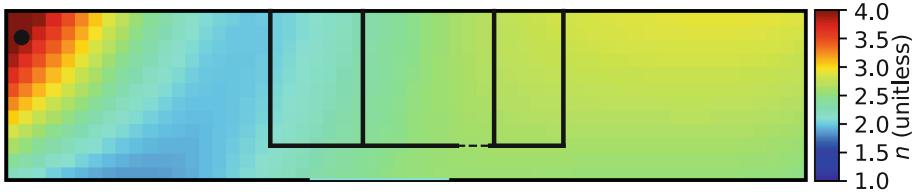
We follow the calibration procedure proposed in [6] to create the baseline Wi-Fi signal propagation representation, that is, a map representing the path loss coefficient of Eq. (1) in the whole environment. This procedure consists on four steps, as can be seen in Fig. 2. First, the target environment is equipped with a wireless platform formed by a set of Wi-Fi access points whose position is known. If the target environment has a wireless platform to provide Internet connection, it can also be used to supply localization services by obtaining the actual position of the existing access points. The next two steps are to design a trajectory that covers most of the environment (calibration trajectory) and travel it while the smartphone is capturing RSSI values (RSSI gathering). The environment's actual position where each reading is captured must be known in order for models be trained. For this purpose, an auxiliary synchronized localization system can be used. Finally, the resulting RSSI capture is used to create the signal propagation representation map.

The calibration procedure has been tested in our test environment (see Sect. 3) using a LG Nexus 5 smartphone. In Fig. 3, the signal propagation representation map of an AP (whose position is represented by a black dot) in our test



**Fig. 2.** Calibration procedure of [6]

environment is depicted. As can be seen, the path loss exponent value changes with the distance to the AP. In the vicinity of an AP, the path loss exponent value is at its highest. In Line-Of-Sight (LOS) situations, its value remains close to 2. When Non-Line-Of-Sight (NLOS) situation starts, the path loss exponent value gradually increases until reaching a value close to 3 in hard-NLOS situations.



**Fig. 3.** Signal propagation representation map of LG Nexus 5

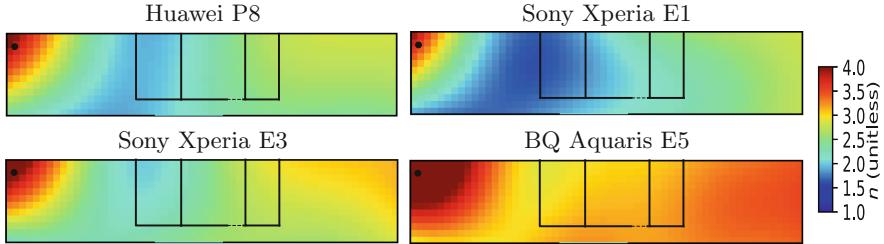
## 5 Inter-device Calibration Procedure

The proposed inter-device calibration procedure aims to calibrate a Wi-Fi-aided indoor localization system for new Wi-Fi-enabled devices based on a previous calibration of another device. The previous calibration is composed by a map representing the Wi-Fi signal propagation in the whole environment, called as baseline signal propagation representation. The goal is then that the new device being calibrated reuses the baseline signal propagation representation of the other device without performing the whole baseline calibration procedure.

### 5.1 Preliminary Study

The baseline calibration procedure has been tested in our test environment using four different smartphones: Huawei P8, Sony Xperia E1, Sony Xperia E3 and BQ Aquaris E5. The resulting signal propagation representation maps of each smartphone can be observed in Fig. 4. Comparing them with each other and with the one in Fig. 3, it can be seen that they are similar. However, some differences can be appreciated due to the RSSI hardware variance problem, even in smartphones of the same manufacturer.

Therefore, a signal propagation representation map created by a device can be used by a new device to provide localization services without the new device



**Fig. 4.** Signal propagation representation map of four smartphones

performing the entire baseline calibration procedure. This is due that how the Wi-Fi signal is propagated in a certain scenario depends upon the AP and the environment's characteristics, not upon the device receiving the signal. The only difference between distinct devices is the RSSI accuracy, i.e., the RSSI values computed by them. Therefore, converting the RSSI values of the new device into the calibrated device's RSSI values, the baseline signal propagation representation map can be reused by the new device.

## 5.2 Inter-device RSSI Value Conversion

In contrast with traditional inter-device calibration procedures, the proposed approach do not seek to convert the RSSI between devices. This is due to, although a good conversion can be achieved, a huge amount of trusty RSSI values (which can't always be obtained due to the factors affecting the Wi-Fi signal) are needed. Instead, our calibration procedure modify a parameter of LDPLM (Eq. (1)) in order to make the signal propagation representation of a device reusable for another device, regardless of the device type. This means that a signal propagation representation created by an smartphone can be reused by a robot or a tablet.

Previous works [14] on calibrating new devices based on previous calibrations show that the relationship between RSSI values of two different devices is linear. Since our approach do not seek to convert the RSSI between different devices, we propose to modify the  $P_r(d_0)$  parameter value of Eq. (1) to perform the inter-device RSSI transformation. This parameter is the RSSI at  $d_0$  meters, usually 1 meter. The baseline calibration procedure set this value to  $P_r(d_0) = -30$ , due to the RSSI accuracy of LG Nexus 5 (specified in its hardware specifications [1]) is  $[-30, -90]$ . Therefore, we propose to design a method that tunes the value of this parameter for new devices.

## 5.3 Method for Fine-Tuning the $P_r(d_0)$ Parameter Value

In order for baseline signal propagation representation map be reused for new devices, a method for fine-tuning the  $P_r(d_0)$  parameter value of the new device is designed. This method consists on three steps, as can be seen in Fig. 5. First, a

short-distance trajectory whose coordinates are known is designed in the target environment. This trajectory is then traveled while the new device is gathering the RSSI to obtain the inter-device calibration capture. Finally, using this capture, the  $P_r(d_0)$  parameter value is tuned.



**Fig. 5.** Proposed inter-device calibration procedure

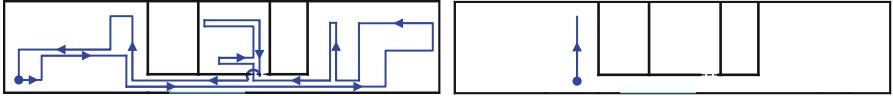
Since the positions where each RSSI reading  $P_r(d)_T$  is captured are known, the distance  $d_T$  between them and each AP belonging to the wireless platform can be computed. Furthermore, the baseline path loss exponent  $n_b$  of each reading can be obtained from the baseline signal propagation representation map. The problem is then to solve the following equation using the inter-device calibration capture:

$$P_r(d_0)_T = P_r(d)_T + 10 \cdot n_b \cdot \log_{10}\left(\frac{d_T}{d_0}\right) \quad (2)$$

## 6 Experimental Results

In this section, we show the results of the experiments carried out to evaluate the proposed inter-device calibration procedure. The primary goals of the experiments are to determine the localization accuracy of the proposed procedure and the accuracy loss with respect to the baseline calibration procedure. In order to obtain such accuracy, a particle filter particularly tailored based on our calibration approach has been implemented [6].

To reach the goals, two trajectories have been defined in our test environment, as can be seen in Fig. 6, where the starting point is represented by a dot and the followed direction is represented by the arrows. The one shown on the left is used to perform the baseline calibration procedure and is 100 m long. It is also used to determine the localization accuracy of both calibration procedures and the accuracy loss between procedures. The one shown on the right is used to carry out the proposed inter-device calibration procedure and is 4 m long. Both trajectories are traveled several times by a pedestrian holding a different smartphones at each time. The baseline calibration capture takes 10 min, while the inter-device calibration capture takes 30 s. The following smartphones are used: LG Nexus 5, Huawei P8, Sony Xperia E1, Sony Xperia E3 and BQ Aquaris E5.



**Fig. 6.** Testing trajectories: for the baseline calibration procedure (left) and for the proposed inter-device calibration procedure (right)

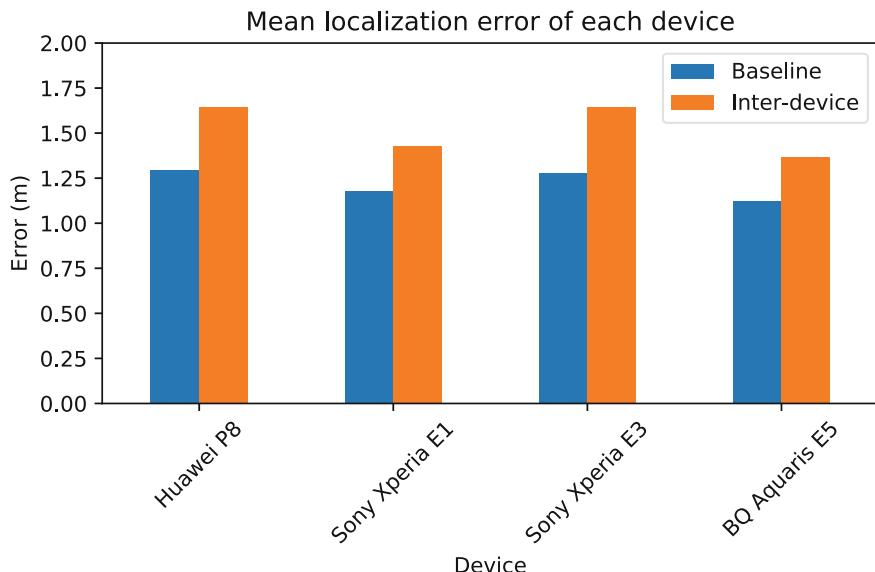
The baseline signal propagation representation map is the one resulting of carrying out the baseline calibration procedure using the LG Nexus 5 smartphone capture (see Sect. 4). Then, the  $P_r(d_0)$  parameter value is tuned for all other devices following the procedure described in Sect. 5. The resulting  $P_r(d_0)$  values can be seen in Table 1. The  $P_r(d_0)$  values of Huawei P8 and Sony Xperia E1 devices are similar to the baseline  $P_r(d_0)$  value ( $-30$  dBm). This is due to, as can be seen in Figs. 3 and 4, the baseline maps of LG Nexus 5, Huawei P8 and Sony Xperia E1 are very similar. The baseline map of BQ Aquaris E5, on his part, is the most different with respect to the LG Nexus 5 map, therefore, the  $P_r(d_0)$  value of this device is also the most different.

**Table 1.** Computed  $P_r(d_0)$  values in dBm of each device

Device	Huawei P8	Sony Xperia E1	Sony Xperia E3	BQ Aquaris E5
$P_r(d_0)$	-29.42	-28.77	-33.20	-35.70

Finally, the particle filter is executed in order to obtain the localization accuracy of each device. To do that, the Euclidean distance is used to compute the error between the position estimated by the particle filter and the actual position where the RSSI is captured (known from the captures).

The results are depicted in Fig. 7, where the mean localization error in meters of the entire capture of each device is used. The blue data, labeled as Baseline, corresponds to the executions that use the baseline calibration procedure to create the signal propagation representation map. The orange data (Inter-device) corresponds to the executions that use the inter-device calibration procedure, i.e., they use the baseline signal propagation representation map and the  $P_r(d_0)$  values of Table 1. As can be seen, the inter-device calibration procedure obtains worse accuracy than the baseline calibration procedure, approximately 20%. However, the inter-device calibration accuracy is suitable for locating devices in indoor scenarios.



**Fig. 7.** Mean localization error in m per device

## 7 Conclusions

In this paper, we have presented a novel inter-device calibration procedure for Wi-Fi-aided indoor localization mechanisms. This procedure calibrates new devices based on a previous calibration. To do that, the signal propagation representation of the previous calibration is reused and a parameter of the range-based model is adjusted for every single new device to be located.

The proposed inter-device calibration procedure has been evaluated in a real-world scenario, achieving suitable localization accuracy in indoor scenarios in a very short interval of time. In our experiments, there is a loss of accuracy of about 20%, however, the setup of new devices has been reduced from 10 min to 30 s, achieving a reduction in configuration time of 2000%. Although this procedure is evaluated using smartphones, it can be used by any combination of Wi-Fi-enabled devices, such as robots, laptops or tablets.

As a future work, we propose to test the proposed inter-device calibration procedure in our test environment using a real robot and a previous calibration carried out by a smartphone. In addition, we are interested in testing our calibration procedures in other environments.

**Acknowledgments.** This work has been partially funded by the Spanish Ministry of Economy and Competitiveness under Grant number RTI2018-098156-B-C52, and by the Regional Council of Education, Culture and Sports of Castilla-La Mancha under grant number SBPLY/17/180501/000493, supported with FEDER funds. Miguel Martínez del Horno is also funded by the Universidad de Castilla-La Mancha grant 2016/14100.

## References

1. Broadcom: BCM4339 preliminary data sheet. <http://www.mouser.com/ds/2/100/Radio%20with%20Integrated%20Bluetooth%204.1%20and%20FM%20Receive-961626.pdf>
2. Castillo-Cara, M., Lovón-Melgarejo, J., Bravo-Rocca, G., Orozco-Barbosa, L., García-Varea, I.: An empirical study of the transmission power setting for Bluetooth-based indoor localization mechanisms. *Sensors* **17**(6), 1318 (2017)
3. Chen, Z., Zou, H., Jiang, H., Zhu, Q., Soh, Y., Xie, L.: Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. *Sensors* **15**(1), 715–732 (2015)
4. Dong, F., Chen, Y., Liu, J., Ning, Q., Piao, S.: A calibration-free localization solution for handling signal strength variance. In: International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, pp. 79–90. Springer, Heidelberg (2009)
5. Fang, S.H., Wang, C.H., Chiou, S.M., Lin, P.: Calibration-free approaches for robust Wi-Fi positioning against device diversity: a performance comparison. In: 2012 IEEE 75th Vehicular Technology Conference (VTC Spring), pp. 1–5. IEEE (2012)
6. Martínez del Horno, M., García-Varea, I., Orozco Barbosa, L.: Calibration of Wi-Fi-based indoor tracking systems for Android-based smartphones. *Remote Sens.* **11**(9), 1072 (2019)
7. Kim, Y., Shin, H., Chon, Y., Cha, H.: Smartphone-based Wi-Fi tracking system exploiting the RSS peak to overcome the RSS variance problem. *Perv. Mob. Comput.* **9**(3), 406–420 (2013)
8. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **37**(6), 1067–1080 (2007)
9. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015)
10. Navarro-Serment, L.E., Mertz, C., Hebert, M.: Pedestrian detection and tracking using three-dimensional LADAR data. *Int. J. Robot. Res.* **29**(12), 1516–1528 (2010)
11. Rappaport, T.S.: Wireless Communications: Principles and Practice, 1st edn. IEEE Press, Piscataway (1996)
12. Romero-González, C., Villena, Á., González-Medina, D., Martínez-Gómez, J., Rodríguez-Ruiz, L., García-Varea, I.: InLiDa: a 3D lidar dataset for people detection and tracking in indoor environments. In: 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), VISAPP, Porto, Portugal, vol. 6, pp. 484–491 (2017)
13. Sun, G., Chen, J., Guo, W., Liu, K.: Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs. *IEEE Sig. Process. Mag.* **22**(4), 12–23 (2005)
14. Tsui, A.W., Chuang, Y.H., Chu, H.H.: Unsupervised learning for solving RSS hardware variance problem in WiFi localization. *Mob. Netw. Appl.* **14**(5), 677–691 (2009)
15. Tuta, J., Juric, M.B.: A self-adaptive model-based Wi-Fi indoor localization method. *Sensors* **16**, 2074 (2016)
16. Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., Aboutanios, E.: Recent advances in indoor localization: a survey on theoretical approaches and applications. *IEEE Commun. Surv. Tutorials* **19**(2), 1327–1346 (2016)



# A Review of Segmentation Methods for 3D Semantic Mapping

Cristina Romero-González<sup>(✉)</sup>, Jesus Martínez-Gómez,  
and Ismael García-Varea

University of Castilla-La Mancha, Campus Univ. s/n, 02071 Albacete, Spain  
[{Cristina.RGonzalez,Jesus.Martinez,Ismael.Garcia}@uclm.es](mailto:{Cristina.RGonzalez,Jesus.Martinez,Ismael.Garcia}@uclm.es)

**Abstract.** A 3D semantic map can be defined as a grid-based representation of the environment, where each bin stores a probability distribution over the possible elements to be found in it. This probability distribution can be obtained with any state-of-the-art image classifier, while the 3D position depends on the localization accuracy of the robot, the sensitivity of its RGB-D sensor, and the segmentation of the input image. In this paper, we focus on this last factor, to explore different options for image segmentation that might improve 3D maps. We will compare various approaches based on the use of 2D and 3D information to find relevant clusters of information. They will be evaluated to assess their suitability for real-time applications.

**Keywords:** Image segmentation · Object detection · Image classification · Robot vision · Deep learning

## 1 Introduction

Semantic segmentation is a pixel-wise classification of an input image, that is, this process results in both a grouping of the pixels that belong to the same object (segmentation) and the assignation of a lexical label that represents their class or category (semantic classification). Additionally, in robotic applications, these segmented predictions can be associated to 3D positions. Based on this premise, in [1] was proposed the creation of 3D semantic maps which encode the object clusters and their classes.

In the original paper, the authors took advantage of state-of-the-art image classifiers to build a 3D map of the environment with the location of the different objects. That solution used an efficient sliding window approach that caused some inaccuracies on the 3D position of the detected elements. To overcome this situation, in this paper we propose, compare and evaluate different segmentation techniques to build 3D semantic maps.

Additionally, we must remark the focus on robotic applications, which means that images will be streamed from an RGB-D device in real time. This results in several considerations:

- Each frame must be processed within a small time span. The classification with a CNN model can run in real time in a modern GPU with enough dedicated memory, and can also be performed in an adequate time with less resources. However, we rely on classic segmentation techniques, as there are not large enough datasets to train CNN models for 3D scene segmentation. Therefore, we must consider and evaluate different techniques based not only on their segmentation accuracy but also their processing time.
- The RGB-D images from a Kinect-like camera are point clouds captured from a specific viewpoint, also known as range images. In the standard generation of RGB-D images, the color information captured with a RGB camera and the depth information captured with an infrared sensor are registered to form an organized point cloud. This means that for every point in the point cloud, there is a corresponding pixel in the RGB image. If the input RGB-D image is segmented using 2D techniques, we can extract the corresponding 3D subcloud for each subimage. In a similar way, if the input RGB-D image is segmented using 3D techniques, we can extract the corresponding 2D subimage for each subcloud.

## 2 Related Work

Most computer vision tasks can be approached with Deep Learning [2] techniques. Convolutional Neural Networks (CNNs) have sufficiently proven their remarkable results in complex tasks, like image classification [3], face recognition [4] or language understanding [5].

Object localization consists in finding the position of the objects that are placed in the environment, this is mainly a computer vision problem but with special relevance to robotics applications, like object manipulation or human-robot interaction. There have been several approaches to object detection and localization with CNNs. From classification with a sliding window and a refinement of the coarse map [6], to detect regions of interest to classify [7]. Also, CNN models can be designed to predict both the lexical label of the objects and their bounding cube [8].

In the case of scene segmentation, the research has been more scarce due to the lack of large datasets annotated at pixel level. Additionally, the class categories of these datasets are usually focused on large elements (like cars or houses) and rarely have information about the type of objects that indoor robotic applications require. However, recently published datasets, like COCO [9] or ADE20K [10], have allowed the development of solutions of scene segmentation. In this area, Fully Convolutional networks [11] are the state-of-the-art, this type of networks modify part of the standard architecture of a CNN, to directly output the pixel-wise predictions.

Nonetheless, all these solutions have been designed for RGB images, while robot vision is by nature 3D. For example, object localization not only concerns the position of the objects in the image, but also their pose in the 3D world where the robot is moving. To address this situation, in [1] was proposed the building of

a 3D map with the position of the different objects in the environment by taking advantage of the accuracy of CNN classifiers and RGB-D sensors. However, that approach presents some drawbacks that have already pointed out, and for which we propose different segmentation methods in this paper.

## 2.1 3D Semantic Maps

For completeness, this section includes an overview of the 3D semantic maps presented in [1]. The underlying idea in this approach is that subimages focusing on the objects in the scene can be used for both classification with a CNN and to determine the 3D position of the objects.

Given an input RGB-D image  $\mathcal{I}$ , the first step is to divide it in a set of smaller subimages. For each RGB subimage  $I_i^{\mathcal{I}}$  we will have a corresponding 3D point cloud or subcloud  $C_i^{\mathcal{I}}$ , and vice versa. This is a seamlessly conversion because images streamed from an RGB-D sensor are organized and there is a direct relation between any point in the point cloud and its corresponding pixel in the RGB image.

After segmentation, each subimage  $I_i^{\mathcal{I}}$  is classified with a pretrained CNN model to obtain a probability distribution  $\mathcal{P}_i^{CNN}$  over the  $N$  lexical labels that the model outputs. Consequently,  $N$  will only depend on the dataset employed to pretrain the model. Meanwhile, each subcloud  $C_i^{\mathcal{I}}$  will be analyzed to calculate its centroid ( $p_i$ ) and bounding cube (defined by its minimum and maximum coordinates,  $bc_i$  and  $BC_i$ , respectively). Thus, the semantic classification process ( $f$ ) of a pair of corresponding subimage and subcloud  $(I_i^{\mathcal{I}}, C_i^{\mathcal{I}})$  results in the following function:

$$f(I_i^{\mathcal{I}}, C_i^{\mathcal{I}}) = \langle p_i, bc_i, BC_i, \mathcal{P}_i^{CNN} \rangle \quad (1)$$

This information is then used to generate the 3D semantic map, which is a grid-based representation of the environment, where each bin  $\mathcal{B}$  stores the probability distribution of the  $N$  categories of the CNN model for that part of the environment. Unlike non-grid maps, this approach allows to generalize the map representation to include any object or element that can be detected with a CNN model, independently of its 3D model or shape. However, the calculated geometrical properties are given from the viewpoint of the camera, so they must be translated to the map reference frame to update the corresponding bin. To this end, we will use the robot pose, at the time when the image was captured, to calculate the transform from camera to map. After this transformation, the coordinates are used to calculate the corresponding bin  $\mathcal{B}_j$  for the prediction centroid  $p_i$  in the semantic map.

Each bin will contain the probability distribution of the  $N$  classes  $\mathcal{P}_j^{\mathcal{B}}$ , which are assumed to be uniformly distributed at initialization. When new predictions from the CNN classifier  $\mathcal{P}_i^{CNN}$  are associated to a bin  $\mathcal{B}_j$ , its probabilities are updated considering a learning rate  $\alpha$  using the following equation:

$$\mathcal{P}_j^{\mathcal{B}} \leftarrow (1 - \alpha)\mathcal{P}_j^{\mathcal{B}} + \alpha\mathcal{P}_i^{CNN} \quad (2)$$

Finally, for each prediction the bounding cube dimensions are used to also update neighbouring points.

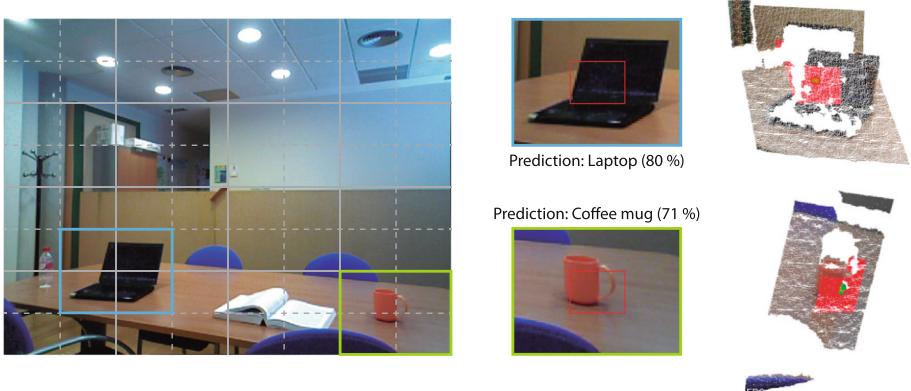
### 3 Segmentation Methods

We can take two approaches to segment the input images and generate the set of subimages to classify: to perform a 2D segmentation of the image and calculate the associated 3D subclouds, or to perform a 3D segmentation and classify the associated 2D subimages. Research in 2D segmentation is more extensive, while solutions based on 3D information can benefit from the position of the different pixels besides its intensity value.

Later, we will evaluate these segmentation methods to determine their suitability for robot vision applications.

#### 3.1 Techniques Based on 2D Information

The methods presented in this subsection use exclusively the color information of the input image to segment the scene. In this case, we opt for two almost opposite methods: (1) a basic sliding window and (2) a state-of-the-art CNN-based detector. The geometrical information of the subimages is then calculated in a similar way for both methods.

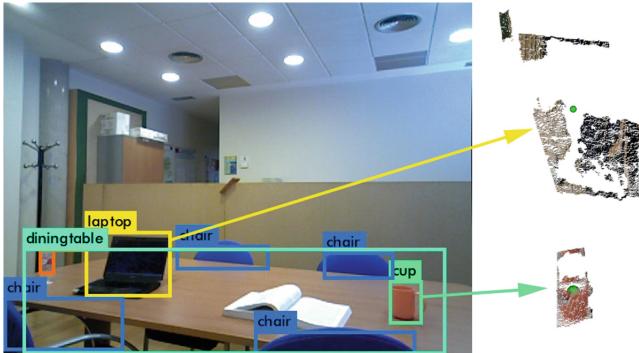


**Fig. 1.** Example of sliding window semantic segmentation using the GoogLeNet model. Sample image of size  $640 \times 480$  with a sliding window of size  $160 \times 120$  ( $4 \times 4$ ) and sampling interval of  $80 \times 60$ . The red points in the right images indicate the central points of the subimage used to calculate the centroids (green points).

- **Sliding window.** For the sake of efficiency, we can use a sliding window over the color information of the image that divides the image in  $4 \times 4$  subimages with overlapping (up to  $7 \times 7 = 49$  subimages to be classified).

Next, we calculate the subcloud centroid using only the central part of the subimage, as a CNN classifier will output a higher probability when the image depicts a single centered object. Additionally, this approach reduces the error for the centroid  $p_i$  and bounding box ( $bc_i$  and  $BC_i$ ) because it reduces the number of background points considered for their calculation. This process can be observed in Fig. 1. However, if there are not valid points in  $C_i^T$  to calculate the centroid, that subimage  $I_i^T$  will not be classified, as we are only interested in the RGB predictions if we can associate a 3D position to them. Note that this technique does not actually perform a segmentation of the image, it just reduces the original input image size to obtain specific probabilities for small objects. Consequently, this technique is expected to produce general good results in classification but with a low localization accuracy. This sliding window approach was originally proposed in [1], so it will be the baseline to compare against.

- **YOLO [8].** In this case we use a CNN pretrained to detect objects in images. You Only Look Once (YOLO) is a network that divides the input image into regions and, for each region, it predicts bounding boxes and lexical labels. In our segmentation pipeline, YOLO provides a set of subimages  $S_i^T$  along with their predicted lexical labels and their probabilities. Therefore, we can extract the corresponding subcloud  $C_i^T$  for each detected object and, with the subcloud, calculate the centroid and bounding cube (see Fig. 2).



**Fig. 2.** 3D clusters generated from YOLO detections.

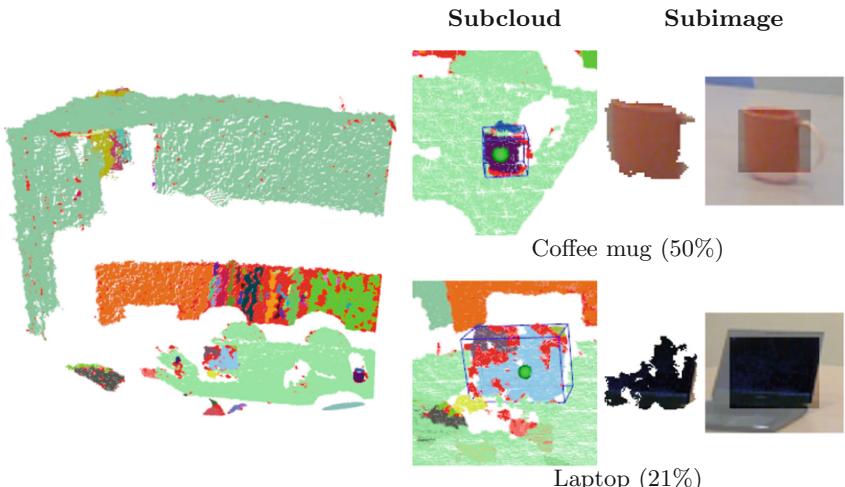
In this case, we are simultaneously performing the segmentation and semantic classification of the subimages with the same pretrained CNN model in a single step, so this option is expected to be efficient. However, YOLO is intended for localization and/or detection applications, not segmentation. Thus, the predicted 2D bounding box might result in localization errors due to background data or occlusions.

### 3.2 Techniques Based on 3D Information

The methods presented in this subsection use the geometrical information encoded in the input point cloud to determine the clusters that might contain objects of interest.

- **Region growing segmentation (RGS)** [12]. This 3D segmentation method uses a smoothness constraint to separate the different parts of a point cloud, that is, it finds smoothly connected areas in point clouds.

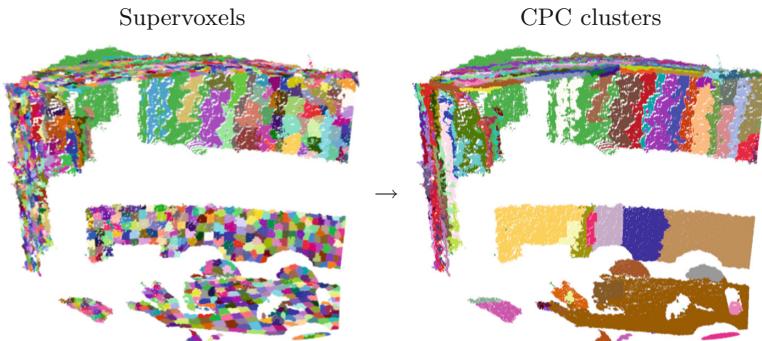
In this case, we perform a region growing segmentation over the image cloud  $C^T$ , which will result in a set of subclouds. Then, for each subcloud  $C_i^T$  we obtain its geometrical features (centroid and bounding cube) and its corresponding subimage  $I_i^T$ . In order to classify it with the CNN model,  $I_i^T$  will not contain just the pixels associated to the points in  $C_i^T$ , but also the background points needed to fill the image and a small padding (see Fig. 3).



**Fig. 3.** Example of subimage generation with region growing segmentation. The centroid (green point) and bounding cube (blue) are calculated using the cluster dimensions. The subimage is generated with color information of points from the subcloud and additional neighbouring points to fill the RGB image, then a padding is added (lighter border).

As some of the subimages that result from this segmentation process might be too small for a CNN classifier, we set a threshold to establish a minimum size. This threshold avoids the classification of images that are not informative enough and, thus, it reduces the number of subimages to be classified by frame, which will also reduce the computation time.

- **Constrained planar cuts (CPC)** [13]. This case follows the same pipeline as region growing segmentation to calculate the geometrical properties of the clusters, but using CPC segmentation instead. This algorithm over-segments the input point cloud into an adjacency graph of supervoxels [14] and uses local concavities as an indicator to determine inter-part boundaries. The initial over-segmentation of this process can also generate small images that are not adequate for classification, so we follow the same process described in the previous segmentation method, where we define a threshold to discard images with a small size. In Fig. 4, we show a sample of cloud segmented with this method.



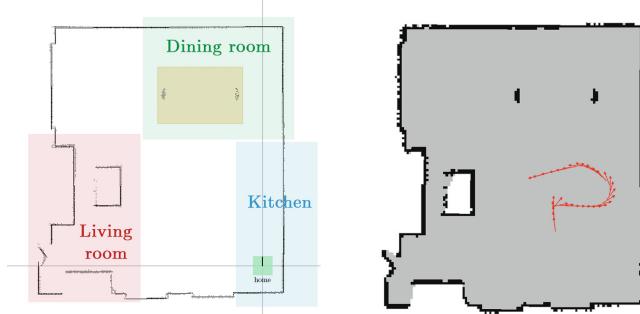
**Fig. 4.** CPC segmentation based on supervoxels adjacency.

## 4 Experiments

There are two main criteria to be considered when evaluating these semantic segmentation algorithms. The first is the time employed for both the segmentation and classification of each frame. The second is the quality of the final predictions. Thus, we are going to compare quantitatively the time and number of subimages processed with each technique, and qualitatively the semantic map generated by each method.

We have evaluated our proposal in a home-like scenario, a small room with areas corresponding to a dining room, a kitchen, and a living room. And we have placed the following objects: laptop (x2), coffee mug (x3), water bottle (x2), table (x1), chair (x4), and couch (x1), spread through different “rooms”. As robotic platform we have used a PeopleBot robot, with an ASUS Xtion PRO Live camera installed on top of it. We have recorded a sequence of 531 images with the robot following the path shown in Fig. 5, where the robot is able to visualize all the objects in the environment. This sequence is used to generate

the maps with the different methods that we want to compare, so the same images and robot poses are used to avoid any bias due to these factors. The images are processed with an nVidia GeForce GTX 970 graphics card.



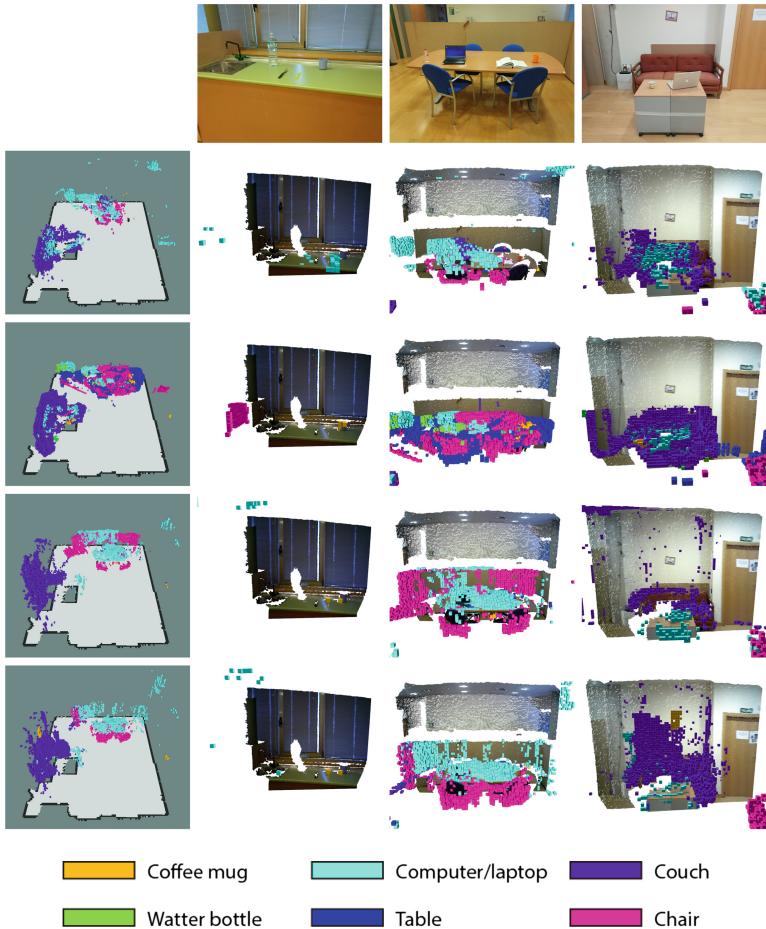
**Fig. 5.** Map with the location of the different rooms in the environment (left) and the path followed by the robot for the semantic map generation (right), which allows the robot to visualize all the objects in the different rooms.

#### 4.1 Semantic Maps Comparison

Figure 6 shows the maps generated after processing the recorded sequence with the different segmentation techniques. In this figure we can observe the final map and several captures of the different rooms to compare against their ground truth pictures.

In general, 2D segmentation methods are better to identify larger objects (like the table or the couch), meanwhile 3D methods are able to detect smaller objects (like coffee mugs). Also, the use of 3D segmentation results in a more precise update of the bins in the map, because 3D clusters allow to only update the map bins that belong to the subcloud. Otherwise, 2D subimages usually include occlusions within the bounding box that are mistakenly considered the same object. Additionally, 2D methods tend to generate false positives in the background due to the irregular shape of the objects within the bounding boxes.

3D methods seem to suffer from false positives, where large clusters are identified as only one of the objects contained in them. For example, while classifying a cluster imaging a table, the assigned label is “laptop” because it is one of the elements on the table. In this regard, the best option is YOLO, as the obtained bounding box better adjusts to the object. Also, as a detecting algorithm, YOLO produces fewer false positives than the other methods.



**Fig. 6.** Comparison of segmentation methods, from top to bottom: sliding window, YOLO, RGS, and CPC.

Finally, it is important to remark that the accuracy of 3D semantic maps also depends on both the localization error of the robot and the sensibility of the camera. Consequently, some of the inaccuracies in the final maps might be the result of these factors, instead of being segmentation errors.

## 4.2 Performance Evaluation

In Table 1 we show the segmentation and prediction (classification or detection) times for the proposed methods. Each value is the average time for all the frames in the recorded sequence. On the one hand, the methods that produce fewer clusters require less classification time. On the other hand, the segmentation itself might be too expensive to be used in real-time applications. This is the case

of RGS, which produces a small number of subimages per frame and, consequently, has the smallest classification time, but the segmentation time makes it inadvisable to run in real-time.

Among the other considered methods, YOLO clearly outperforms the use of a sliding window or CPC. Being a detection method, instead of classification, its prediction time is considerably lower and the number of subimages in each frame is also the smallest.

**Table 1.** Times for the different segmentation methods. The results are the average for all images in the test sequence.

	Time (s)			# Subimages
	Segmentation	Prediction	Total	
Sliding window	0.015	0.341	0.374	49.00
YOLO		<b>0.138</b>	<b>0.138</b>	<b>5.80</b>
RGS	5.791	<b>0.106</b>	5.908	14.66
CPC	0.499	0.267	0.781	38.79

## 5 Conclusions

We have presented different options to exploit the high accuracy of CNN-based image classifiers and detectors. We have evaluated and compared the different proposals and, in general, the best option would be to use a pretrained object detector, like YOLO, as it is the fastest alternative to detect and classify the elements in the image, and the bounding box is better adjusted to the objects. However, it still generates false positives in the map, due to background points also being updated.

There are two main drawbacks with the segmentation methods studied in this paper: 2D segmentation methods are susceptible to background noise and occlusions while 3D methods tend to over-segment the input image. So, in future works, it might be interesting to consider mixed solutions that try to cancel out these issues. For example:

- **Sliding window + CPC.** The underlying idea is to classify larger subimages than the ones obtained with 3D segmentation but to take advantage of the 3D clusters to calculate the geometric features. In this scenario, the subimages provided by the sliding window approach would be classified to obtain the lexical labels, but the centroid and bounding cube would be calculated using either the biggest cluster in the subcloud (segmented with CPC) or the cluster closest to the subcloud center.
- **YOLO + CPC.** Given the high accuracy of deep learning techniques in computer vision tasks, if the platform has enough hardware resources, it would be advisable to use YOLO for detection and classification, and CPC to calculate the geometric features.

In theory, 3D semantic maps would benefit from more accurate object detections but would require more computational resources, because the segmentation process would be more complex. Additionally, our results show that detection CNNs for RGB images offer a better segmentation of the scene to calculate 3D geometrical features. This is clear evidence that 2D semantic segmentation approaches would considerably improve the quality of generated maps. However, current solutions have some difficulty segmenting smaller objects in the scene [15], and cannot be exploited for the generation of 3D semantic maps yet.

**Acknowledgments.** This work has been partially sponsored by the Regional Council of Education, Culture and Sports of Castilla-La Mancha under grant number SBPLY/17/180501/000493, supported with Feder funds.

## References

1. Romero-González, C., Martínez-Gómez, J., García-Varea, I.: 3D semantic maps for scene segmentation. In: ROBOT 2017: Third Iberian Robotics Conference, Sevilla, Spain, pp. 603–612 (2018)
2. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
3. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105 (2012)
4. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12**(Aug), 2493–2537 (2011)
6. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
7. Girshick, R.: Fast R-CNN. In: The IEEE International Conference on Computer Vision (ICCV) (2015)
8. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
9. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Computer Vision – ECCV 2014, pp. 740–755. Springer, Heidelberg (2014)
10. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ADE20K dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2017
11. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
12. Rabbani, T., Van Den Heuvel, F., Vosselmann, G.: Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci.* **36**(5), 248–253 (2006)

13. Schoeler, M., Papon, J., Worgotter, F.: Constrained planar cuts - object partitioning for point clouds. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
14. Papon, J., Abramov, A., Schoeler, M., Wörgötter, F.: Voxel cloud connectivity segmentation - supervoxels for point clouds. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2027–2034, June 2013
15. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ADE20K dataset. *Int. J. Comput. Vis.* **127**(3), 302–321 (2019)



# Learning from the Individuals and the Crowd in Robotics and Mobile Devices

Fernando E. Casado<sup>1</sup>(✉), Dylan Lema<sup>1</sup>, Roberto Iglesias<sup>1</sup>,  
Carlos V. Regueiro<sup>2</sup>, and Senén Barro<sup>1</sup>

<sup>1</sup> CiTIUS (Centro Singular de Investigación en Tecnologías Inteligentes), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain

{fernando.estevez.casado,dylan.lema,

roberto.iglesias.rodriguez,senen.barro}@usc.es

<sup>2</sup> CITIC, Computer Architecture Group, Universidade da Coruña,  
15071 A Coruña, Spain

carlos.vazquez.regueiro@udc.es

**Abstract.** Service robots at homes or works are expected to upload data that can be used by companies to fix the controllers and improve robot behaviours. Nevertheless, this is a delicate issue that concerns data privacy. Instead, we propose an iterative process of local learning (in the robots) and global consensus (in the cloud) that still preserves the benefits of learning from the crowd but when models instead of data are uploaded to a server. This strategy is also valid for mobile phones or other devices. In fact, in order to work with a heterogeneous community of users, we have applied our strategy in a real problem with mobile phones: walking recognition. We achieved very high performances without the need of massive amounts of centralized data.

**Keywords:** Semi-supervised learning · Ensemble learning · Continuous learning · Machine learning · Intelligent systems

## 1 Introduction

The presence of an increasing number of service robots at our homes, companies, etc., is something that we all take for granted, and most likely the number of robots around us will grow in an exponential manner. Nevertheless, there is also a high chance that these robots will be connected to the cloud and they will certainly transfer data from their environments to the companies that built them. This transfer of data will be justified by the need of detecting and improve robot limitations, or to detect future advances that we will receive with our open arms and that for the companies will mean an increase in their billing. It is true that the problems that will arise when moving robots from laboratories to real environments, or the way people will react to them, are highly unpredictable. The availability of data collected from these real environments and from real

working conditions will represent an invaluable source of information to improve models and, in consequence, robot behaviour, thus benefiting robot consumers. The problem is how to perform this without putting at risk our privacy or giving away too much information. Besides, as true as it is the fact that learning from the crowds represents a fast way of improving performance in a short period of time and when each individual provides only a small portion of data, it is equally true that these consensus-based models might have to be tuned to the specificity of the hardware, sensors, users, or environments, to mention but a few examples.

In order to face these challenges, in this paper we introduce a first approach to a “glocal” learning strategy: a cyclical process that will consist on the achievement of local models, on the robots themselves, but which will be latter refined by an ensemble learning in the cloud. These global models will be then returned to the robots so that they speed up the local learning and help to quickly improve robot behaviour, at the same time that they are subdued to a new local adaptation process, after which the model will return to the cloud. “Glocal” means, in this case, a cyclical process of local learning, in the device itself, which can later be further improved at the global level, in the cloud. This cyclical process of global consensus and local adaptation can be repeated indefinitely over time. The fact that only models are moved amongst the local robots and the cloud, will help to protect, up to some extent, local privacy. It will more difficult to re-obtain data from models, but still preserving the benefit from the global interaction and exchange of models.

Obviously, this strategy will not be exclusive for robots. On the contrary, it can be applied to all the wide variety of mobile devices that we carry every day and that are connected, as it is the case of our mobile phones. In fact, in this article, we will show the performance of our approach when it is used to solve the specific task of walking recognition in mobile phones. One of the reasons why we have chosen mobile phones, instead of robots, to test our proposal is simply by the fact that it is easier to get a small community of people (and hence devices) working and sharing experiences. We are not a company selling thousands of robots from which we can get data. On the other hand, it is also easier to achieve data and local models provided by non-expert users acting normally, at everyday conditions. In opposition to the classic crowd-sourcing concept in which an heterogeneous group of individuals voluntarily take part in solving a particular task or problem, in our case we do not wish the active participation of the users. Finally, working with mobile phones is not far from robots, as somehow they can be seen as a kind of service robots. It is Vincent [1], who takes the mobile phone as an example of an ordinary device that in close dependent interaction with its user is developing into an “emotionalized social robot”: the device alone is not the social robot, it is the combination of user and mobile phone that is the social robot.

A common strategy to introduce Big Data and Deep Learning in the context of mobile devices and robots is to take the data to the cloud. However, this strategy poses several issues, such as the huge amount of data traffic that is generated and the possible privacy problems. There are also some alternatives

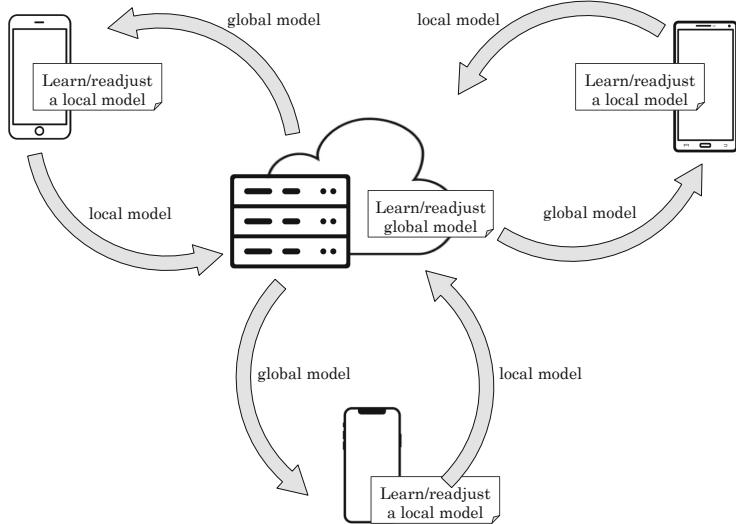
that suggest that the learning process could take place in the cloud and the learned model could be then transferred to the mobile device, which is where it is executed [2,3], or that there may even be a final adjustment of the model and adaptation of some parameters at the local level. Hybrid solutions using local and cloud computing have also been explored. For instance, Deep Neural Networks (DNNs) operate in two stages: an unsupervised pre-training and a fine supervised tuning with backpropagation algorithms. In this way, the pre-training could be carried out in the smartphone and the supervised training in the cloud [4]. However, any of these strategies would still involve moving a significant volume of potentially sensitive data and operates in a single cycle of learning without continuous adaptation.

The strategies most suitable for glocal learning are some of those suggested in the field of large-scale learning, field in which is common to deal with naturally distributed datasets [5]. However, distributed learning literature focuses on scalability, *i.e.*, storage, communication and computational costs. The proposed algorithms are typically supervised distributed methods characterized by finding an optimal solution that, once found, is not changed. This is not appropriate in realistic and non-stationary environments and can even be risky when the learning system faces evolving states – we are talking about complex tasks where the learning system sees different parts of the whole over time–.

## 2 Glocal Learning

Figure 1 shows a diagram of our generic proposal of glocal learning. As we can see, it is a cyclical architecture built from a society of devices; in the figure they are smartphones, but we could think of any other set of devices, either homogeneous or heterogeneous, including tablets, wearables or robots. In any case, each device is able to perceive its environment through its sensors and is connected to the cloud.

Iteratively, each of the devices creates and refines its own local model of the learning problem that is intended to be solved. For that, devices are continuously acquiring and storing new information through their sensors. This information is raw data, which surely must be (locally) preprocessed before being able to use it in a learning stage: noise detection, data transformation, feature selection, data normalization, instance selection, etc. Local models are sent to the cloud where a new learning stage is performed to join the local knowledge, thus obtaining a global model. The global model is then shared with all the devices in the network. In this way, each device has its own modeling of the problem and the global model and can combine that knowledge to make predictions about unseen situations. Moreover, each device can take advantage of the global model to improve the local one. New data is continuously recorded in each device and will be also used to retrain better local models or refine existing ones. Of course, an improvement of the local models will result in an improvement at the global level too. Note that, as the information available at the local level will increase progressively, it may be convenient in the preprocessing stage to select the most recent or relevant



**Fig. 1.** Diagram of the glocal proposal.

information, thus deleting unnecessary data that occupy storage memory of the device.

## 2.1 Semi-supervised Glocal Learning

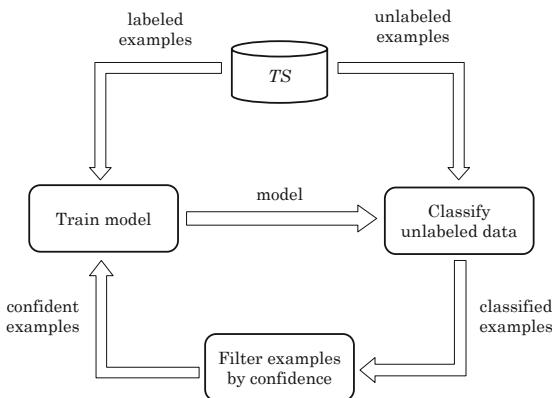
We believe that the glocal architecture that we have just described can be extended to any machine learning scenario: supervised, semi-supervised, unsupervised, reinforcement or multi-task learning. In this work, we have focused on a semi-supervised classification proposal because we realized that a semi-supervised context is quite common in many real problems in mobile devices.

We have addressed the glocal problem from a semi-supervised point of view by specifying the general glocal proposal from Sect. 2 as follows: We propose a semi-supervised scenario at the local level, in the devices, while a supervised strategy will be applied for the fusion of models at the global level.

For the supervised learning at the global level we decided to use an ensemble method, specifically we apply stacking [6]. Stacking is a technique for achieving the highest generalization accuracy which tries to induce which classifiers are reliable and which are not by using a meta-learner. Stacking is usually employed to combine models built by different inducers. It creates a meta-dataset that, instead of using the original input features, uses the probabilities of the predictions made by the local classifiers as the input attributes. The target attribute remains as in the original training set. The meta-learner we chose was a SVM, but it could be any other traditional classification algorithm.

At the local level, we have a classification problem in which each device in the network,  $d_1, d_2, \dots, d_n$ , has a set of training data  $TS_1, TS_2, \dots, TS_n$ , in which not all data is labeled. Each of these sets contains labeled data ( $L_1 \subseteq TS_1, L_2 \subseteq TS_2, \dots, L_n \subseteq TS_n$ ) and unlabeled data ( $U_1 \subseteq TS_1, U_2 \subseteq TS_2, \dots, U_n \subseteq TS_n$ ). In fact, it will be common that  $U_i \gg L_i$  for any  $i \in \{1, 2, \dots, n\}$ . For this semi-supervised scenario, we use another traditional classifier combined with a method of labeled data expansion that allows to take advantage of the unlabeled data. In particular, we train another SVM and we expand the labeled set using a technique similar to the one used in the self-training algorithm, the simplest method of semi-supervised learning and probably the earliest proposal about using unlabeled data in classification [7–9].

Self-training is a wrapper algorithm that repeatedly obtains a model using a training data set,  $TS$ , which contains labeled data,  $L \subseteq TS$ , and unlabeled data,  $U \subseteq TS$ . The method starts by training on the labeled data only ( $L$ ). In each step a part of the unlabeled points is labeled according to the current decision function; typically the decision function selects the examples for which the prediction of the classifier has a high degree of confidence. Then, the supervised method is retrained using its own predictions as additional labeled points. As the iterations progress, the size of  $U$  is reduced. The algorithm ends when there is no more data in  $U$  or an improvement in the performance of the model is not achieved. Figure 2 shows graphical representation of the self-training process.

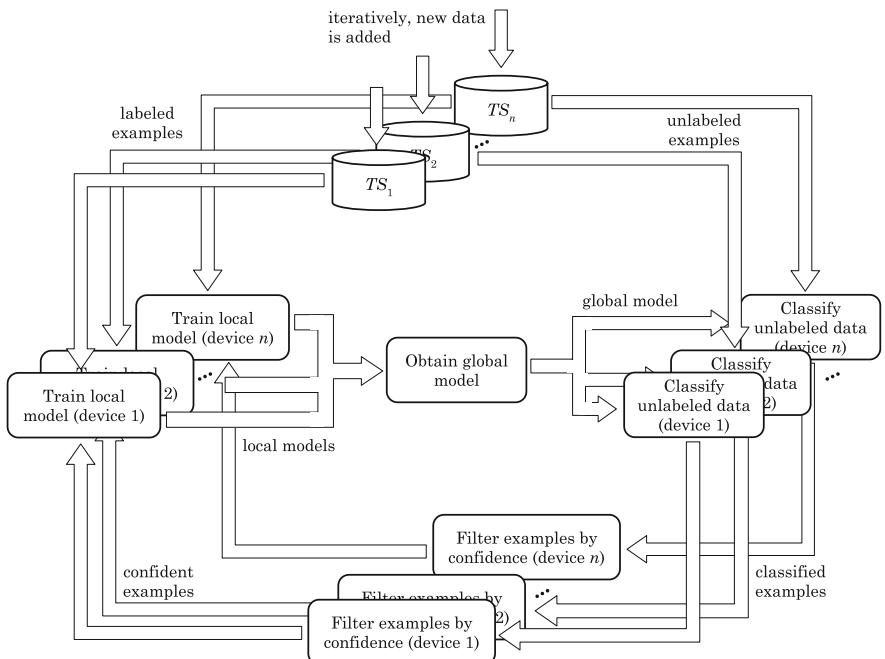


**Fig. 2.** Diagram of the self-training algorithm.

If we directly applied self-training for the local learning of our proposal, we would not be doing a glocal learning. To take advantage of the knowledge globally agreed in the cloud, our proposal uses the global model for labeling data that has no label. In each iteration, we use the latest global model to make a prediction for all the unlabeled data of each device. Then, we filter those predictions based on their degree of confidence. We define the confidence of a prediction as the probabilistic value between 0.5 and 1 given by the classifier

for the most likely class, where 1 would mean that the classifier is totally safe. We accept the prediction as the real label of the example when its confidence is equal to or greater than a threshold  $\gamma$ , whose optimal value we have empirically set at  $\gamma = 0.9$ . Low thresholds ( $\gamma < 0.8$ ) introduce a lot of noise in the training set, while very high thresholds ( $\gamma \geq 0.95$ ) allow to add very few examples in each iteration. Therefore, we consider that  $\gamma = 0.9$  is an adequate value.

Figure 3 summarizes the whole iterative process of semi-supervised glocal learning. In iteration 1, the first step is to obtain a local SVM for each device  $i$  using the labeled data  $L_i$ . Then, local models are sent to the cloud, the global model is obtained using stacking and finally it is sent back to the devices. In iteration 2, global model from iteration 1 is used to classify unlabeled instances from  $U_1, U_2, \dots, U_n$ , to expand the local data used for training. Then, a new local SVM is obtained in each device, and a new global model is agreed in the cloud. This process is repeated cyclically.



**Fig. 3.** Diagram of the semi-supervised glocal proposal.

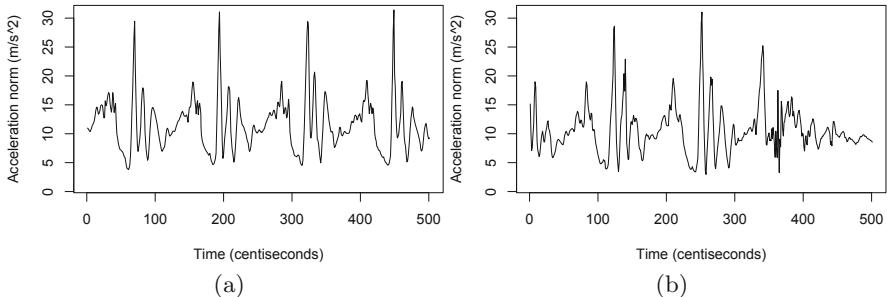
Note that in our proposal, in each iteration, the labeled data set of each device ( $L_1, L_2, \dots, L_n$ ) is expanded not only with classified data from the unlabeled sets ( $U_1, U_2, \dots, U_n$ ), but also with new labeled data recorded by the devices that was not yet available in the previous iteration. Note also that, as we use stacking to combine the local models, no restrictions are imposed in the selection of the

classifiers, both locally and globally. Despite we have chosen SVMs, any traditional learning algorithm can be used to obtain the local and global classifiers. In fact, the employed strategy could be different among the participants and the iterations. Thus, the set of local models could be homogeneous or heterogeneous.

### 3 Experimental Results

In order to evaluate the performance of our algorithm for semi-supervised glocal classification proposed in Sect. 2.1, we have selected a real problem of non-trivial solution, the detection of the walking activity on smartphones.

It is relatively easy to detect the walking activity and even count the steps when a person walks ideally with the mobile in the palm of his/her hand, facing upwards and without moving it too much. However, the situation gets much worse in real life, when the orientation of the mobile with respect to the body, as well as its location (hand, bag, pocket, ear, etc.), may change constantly as the person moves. Figure 4 shows the complexity of this problem with a real example. In this figure we can see the norm of the acceleration experienced by a mobile phone while its owner is doing two different activities. The person and the device are the same in both cases. In Fig. 4a, we can see the signal obtained when the person is walking with the mobile in the pocket. Figure 4b shows a very similar acceleration signal experienced by the mobile, but in this case when the user is standing still with the mobile in his hand, without walking, but gesticulating with the arms in a natural way.



**Fig. 4.** Norm of the acceleration experienced by a mobile phone when its owner is walking (a), and not walking, but gesticulating with the mobile in his/her hand (b).

In a previous work [10], we solved the problem in a classical way using an ensemble of traditional supervised SVMs. For that, we collected data from 77 different people walking and not walking in several conditions. Obtaining large amounts of labeled data in the context of mobile devices is not straightforward. It usually requires the active participation of a multitude of device users and possibly also of one or more domain experts. What we did in our previous work [10] to

get a ground truth was to ask the volunteers to carry a set of sensorized devices in the legs (besides their own mobile) tied with sports armbands. We will use this dataset as test dataset to measure the performance of all the models (both local and global) obtained in our new experiments, which we describe below.

In contrast to our previous work [10], in this case we assumed a more realistic scenario in which it is not possible to label the data easily. In particular, we developed an Android application that samples and logs the inertial data on mobile phones continuously, after being processed. These data are all unlabeled. Nevertheless, the app allows the user to indicate whether he/she is walking or not through a switch button in the graphical interface, but this is optional, so depending on the user's willingness to participate, there will be more or less labeled data. The app also labels autonomously some examples applying a series of heuristic rules when it comes to clearly identifiable positives or negatives (e.g., when the mobile is at rest). With this app, we collected partially labeled data from 10 different people. Participants installed our application and recorded data continuously while they were performing their usual routine. When they moved, as they did it freely and in different environments, their speed and way of movement were different (walking in corridors, in the wild, going upstairs, etc.), as it was the position of the mobile they carried (in their hand, pocket, backpack, etc.). Despite the fact that less people participated in this experiment than in the previous work (10 people instead of the original 77), in this case a lot more data was collected from each volunteer. Table 1 summarizes the data distribution in the new semi-labeled dataset and the old dataset, perfectly labeled and used for testing purposes. We can notice that the percentage of labeled data is around 50% of the training set. It is useful to have a test set where all the data is perfectly labeled and obtained from a high diversity of users (77 people). This set will be used to evaluate the good generalization of the models we obtain.

**Table 1.** Summary of the training and test sets, indicating the number of examples attending to the label.

	Walking (positive)	Not walking (negative)	Unlabeled	Total
Training set	30524	41713	74359	146596
Test set	9747	5168	0	14915

For comparison purposes, we trained a traditional supervised SVM applying 5-fold cross validation on the labeled data from the whole training dataset to establish a baseline. We also applied the self-training method (described in Sect. 2.1) to take advantage of the large amount of unlabeled data (Table 2).

Using our glocal procedure, in the first iteration all devices will collect data until they achieve a set with 40 labeled examples (the number of examples will be higher, as only some of them are labeled either by the user or our heuristic). Once a phone reaches 40 labeled instances, it learns a local model, a SVM, and sends it to the server in the cloud. In the meanwhile, it keeps collecting new data

**Table 2.** Performance of a SVM on the test set, trained using the whole training set, both in a supervised and a self-training semi-supervised way.

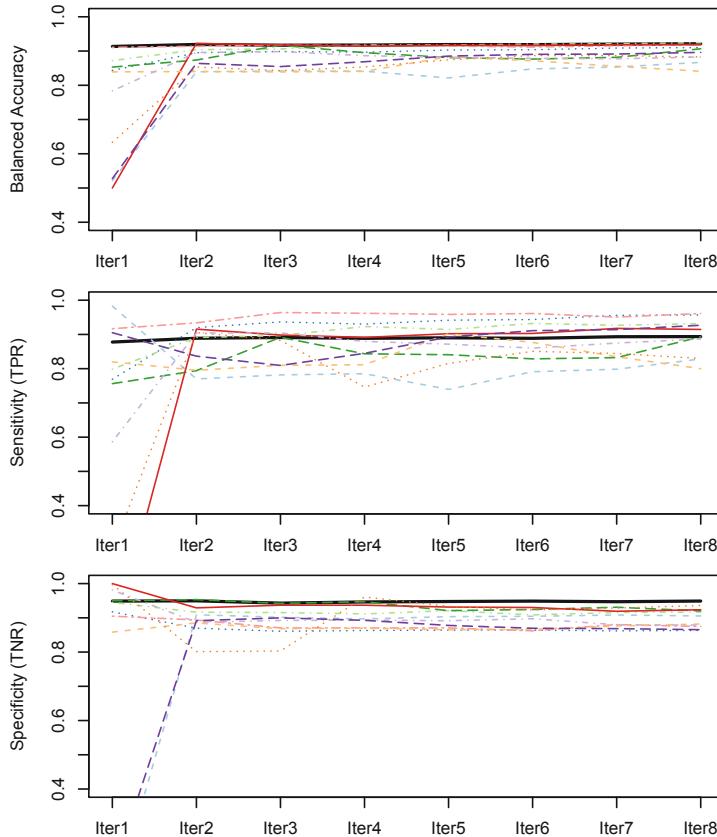
	Balanced accuracy	Sensitivity	Specificity
Supervised SVM	0.9344	0.9611	0.9077
Self-training SVM	0.9332	0.9608	0.9056

until it gathers a second set, with 40 new labeled samples. Once the devices have sent their local models to the server, a global model is obtained by using the stacking ensemble described in Sect. 2.1. Then, the global model is sent back to each of the devices, where it will classify the unlabeled data and label those examples for which the confidence of the prediction is equal or greater than 0.9. In the second iteration, the mobiles will learn the models using both the first and second data sets collected in the first and second iterations. The process will be repeated, i.e., the local models will be sent to the cloud, where a new stacking ensemble will be trained and sent back to the mobiles.

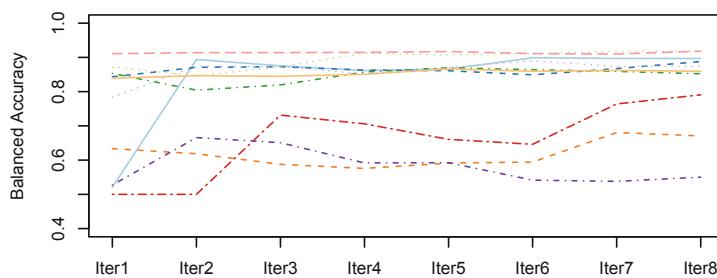
In order to carry out the supervised training of the stacking ensemble (meta-classifier at the global level), we use a portion of 30% of the training data (Table 1), available at the beginning of the process. This set includes examples from all the participants and it does not change throughout the different iterations.

Figure 5 and Table 3 show the performance of the glocal process. As we can see, the global model always exhibits a very high performance. Because in each iteration the unlabeled local data is labeled with the most recent global model, the more unlabeled data the device has, the more it will be enriched by the community’s knowledge. We can see how in the second iteration most models improve drastically. Therefore, we can affirm that the global model helps the devices to improve their performance quickly.

For comparison purposes, Fig. 6 and Table 4 show what happens without the global feedback. We can see the performance of the local models along the iterative process. The local SVMs are trained using the set formed by all the labeled data progressively obtained and accumulated throughout all the iterations. As we can see, there are users which have better models than others. This may be due to various conditions, such as the quality and diversity of the data that the application has recorded on each device, as well as how much the user has participated in the labeling of the data. Since in this case there is no global interaction and feedback, the local learning is much slower and some local performances are really low and do not improve along the process (users 8 and 10).



**Fig. 5.** Results for semi-supervised glocal learning. The thick black line corresponds to the global model. The rest of the lines are each of the 10 anonymous users.



**Fig. 6.** Results for incremental supervised local learning when there is no global feedback. Each of the lines represents one of the 10 anonymous users.

**Table 3.** Accuracies achieved using semi-supervised glocal learning for the global and local models.

	Iteration							
Model	1	2	3	4	5	6	7	8
Global	0.9133	0.9192	0.9170	0.9175	0.9189	0.9187	0.9202	0.9213
User 1	0.5212	0.8395	0.8417	0.8411	0.8215	0.8478	0.8533	0.8669
User 2	0.8433	0.8949	0.8987	0.8987	0.9029	0.9039	0.9088	0.9101
User 3	0.8719	0.9039	0.9059	0.9173	0.9177	0.9203	0.9213	0.9243
User 4	0.8530	0.8737	0.9168	0.8952	0.8809	0.8765	0.8816	0.9065
User 5	0.9108	0.9138	0.9172	0.9161	0.9145	0.9116	0.9155	0.9180
User 6	0.5000	0.9222	0.9176	0.9141	0.9168	0.9165	0.9184	0.9191
User 7	0.8391	0.8402	0.8390	0.8406	0.8835	0.8716	0.8554	0.8409
User 8	0.6336	0.8533	0.8429	0.8536	0.8744	0.8867	0.8866	0.8828
User 9	0.7837	0.8956	0.8864	0.8865	0.8816	0.8787	0.8773	0.8826
User 10	0.5273	0.8641	0.8688	0.8688	0.8852	0.8903	0.8912	0.8961

**Table 4.** Accuracies achieved using incremental supervised local learning, without global feedback.

	Iteration							
Model	1	2	3	4	5	6	7	8
User 1	0.5212	0.8938	0.8755	0.8615	0.8657	0.8990	0.8969	0.8969
User 2	0.8433	0.8710	0.8728	0.8627	0.8611	0.8490	0.8672	0.8879
User 3	0.8719	0.8458	0.8726	0.9123	0.9072	0.9114	0.9174	0.9179
User 4	0.8530	0.8042	0.8197	0.8572	0.8698	0.8641	0.8592	0.8522
User 5	0.9108	0.9137	0.9140	0.9145	0.9177	0.9110	0.9099	0.9181
User 6	0.5000	0.5000	0.7314	0.7059	0.6607	0.6461	0.7640	0.7907
User 7	0.8391	0.8466	0.8448	0.8504	0.8662	0.8594	0.8619	0.8594
User 8	0.6336	0.6185	0.5876	0.5757	0.5914	0.5943	0.6805	0.6707
User 9	0.7837	0.8744	0.8730	0.8517	0.8705	0.8898	0.8735	0.8744
User 10	0.5273	0.6659	0.6508	0.5918	0.5924	0.5415	0.5379	0.5502

## 4 Conclusions

In this paper we have proposed a novel method for semi-supervised glocal classification. Our method allows to train local models in a network of devices (mobile phones, robots...) and then share those models with the cloud, where a global model is created. This global model is sent back to the devices and helps them to label unknown data to improve the performance of their local models. We have applied our proposal to a real classification problem, the recognition of the

walking activity, and we have shown that our proposal obtains very high performances without the need of large amounts of labeled data collected at the beginning. On the contrary, glocal learning is continuous and it can cope with more realistic semi-labeled data collected without restrictions.

We believe that we have opened a very promising line of research and that a large amount of work can be done in this context. The global model obtained with stacking is a good first approximation, but it has limitations. We want to investigate new ways of merging local models in the cloud. We also want to analyze the use of the global models for instance selection in the local devices or the application of techniques such as amending [11].

**Acknowledgments.** This research has received financial support from AEI/FEDER (EU) grant number TIN2017-90135-R, as well as the *Consellería de Cultura, Educación e Ordenación Universitaria* and the European Regional Development Fund (ERDF) (accreditation 2016–2019, ED431G/01 and ED431G/08 and reference competitive group ED431C 2018/29).

## References

1. Vincent, J.: The mobile phone: an emotionalised social robot. In: Social Robots from a Human Perspective, pp. 105–115. Springer (2015)
2. Nakkiran, P., Alvarez, R., Prabhavalkar, R., Parada, C.: Compressing deep neural networks using a rank-constrained topology. In: Proceedings of the 16th Annual Conference of the International Speech Communication Association (2015)
3. Vasilyev, A.: CNN optimizations for embedded systems and FFT. Standford University Report (2015)
4. Lane, N.D., Georgiev, P.: Can deep learning revolutionize mobile sensing? In: Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications, pp. 117–122. ACM (2015)
5. Peteiro-Barral, D., Guijarro-Berdiñas, B.: A survey of methods for distributed machine learning. Prog. Artif. Intell. **2**(1), 1–11 (2013)
6. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
7. Agrawala, A.: Learning with a probabilistic teacher. IEEE Trans. Inf. Theory **16**(4), 373–379 (1970)
8. Fralick, S.: Learning to recognize patterns without a teacher. IEEE Trans. Inf. Theory **13**(1), 57–64 (1967)
9. Scudder, H.: Probability of error of some adaptive pattern-recognition machines. IEEE Trans. Inf. Theory **11**(3), 363–371 (1965)
10. Rodríguez, G., Casado, F.E., Iglesias, R., Regueiro, C.V., Nieto, A.: Robust step counting for inertial navigation with mobile phones. Sensors **18**(9), 3157 (2018)
11. Triguero, I., García, S., Herrera, F.: Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl. Inf. Syst. **42**(2), 245–284 (2015)



# Socially-Accepted Path Planning for Robot Navigation Based on Social Interaction Spaces

Araceli Vega<sup>1</sup>, Ramón Cintas<sup>1</sup>, Luis J. Manso<sup>2</sup>, Pablo Bustos<sup>1</sup>,  
and Pedro Núñez<sup>1</sup>(✉)

<sup>1</sup> RoboLab, University of Extremadura, Badajoz, Spain  
pnuntru@unex.es

<sup>2</sup> School of Engineering and Applied Science, University of Aston, Birmingham, UK

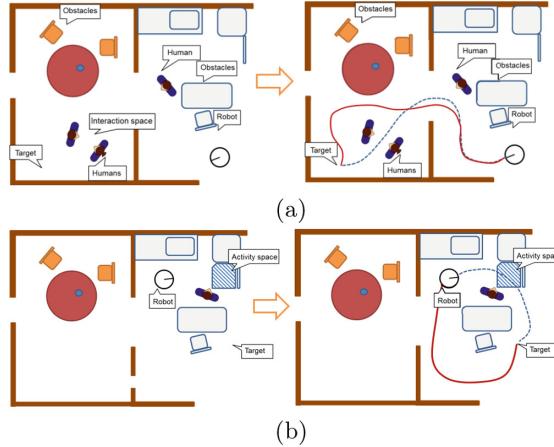
**Abstract.** Path planning is one of the most widely studied problems in robot navigation. It deals with estimating an optimal set of waypoints from an initial to a target coordinate. New generations of assistive robots should be able to compute these paths considering not only obstacles but also social conventions. This ability is commonly referred to as social navigation. This paper describes a new socially-acceptable path-planning framework where robots avoid entering areas corresponding to the personal spaces of people, but most importantly, areas related to human-human and human-object interaction. To estimate the social cost of invading personal spaces we use the concept of proxemics. To model the social cost of invading areas where interaction is happening we include the concept of object interaction space. The framework uses Dijkstra's algorithm on a uniform graph of free space where edges are weighed according to the social traversal cost of their outbound node. Experimental results demonstrate the validity of the proposal to plan socially-accepted paths.

**Keywords:** Social navigation · Path-planning · Dijkstra

## 1 Introduction

The interest that research in social robotics has drawn in the last decade is remarkable, especially in human-populated environments such as museums and hospitals. Working in these scenarios is challenging, as people's behaviour changes frequently and their state is difficult to predict over time. To make these robots able to work seamlessly in these environments, they must act considering social conventions, including those related to navigation.

Traditionally, navigation has been approached by solving three main problems: (i) where is the robot in the world, that is, the localization problem; (ii) how the world around the robot is and how it is built; and (iii) how robot plans an optimal path, which is usually known as the path-planning problem. All these problems have been arousing interest for decades, and many solutions have been



**Fig. 1.** Two different everyday scenarios: (a) left: original scenario where two people are interacting each other; right: red path is the only one accepted by people according to social rules; and (b) left: the interaction between the human and the fridge blocks the path; right: red path is socially-accepted.

presented in simple as well as complex environments. However, in human environments, it is more difficult and novel to find optimal solutions. This is the particular case of algorithms that plan socially-accepted paths for robots.

Social navigation is expected to become an increasingly important skill in the next generation of social robots [1]. During recent years many works have been proposed to make robot navigation algorithms consider social aspects [2]. Figure 1 shows two different scenarios where the robot plans a path to the target in environments with people. In Fig. 1a, the robot has several options to reach the destination, but only one is the most accepted (highlighted in red). Similarly, Fig. 1b shows another scenario where robot has different possible routes, but only one is the most appropriate in social terms (also illustrated in red).

To estimate the best social route from the robot to the target pose, this work proposes using the concept of social mapping [3]. Unlike classical path planning approaches, the proposal described in this paper adds social information on top of the free-space graph previously built in order to build a social map. To this end, the system associates different personal spaces (intimate, social and public) to humans and groups of people in the environment. In the same way, the algorithm associates different activity spaces to objects with which people can interact (known as *Space Affordances* [4]). These different interaction spaces modify the free space graph, penalizing the cost of traversing some areas when planning the path to the target [5].

The **main contributions** of this paper are: (i) the definition of a new framework for planning social paths in human and interactive environments; (ii) the description of a novel social path-planning algorithm based on social interaction spaces that uses information from people and objects; (iii) an adaptive method-

ology to penalize paths depending of the level of confidence between the robot and people.

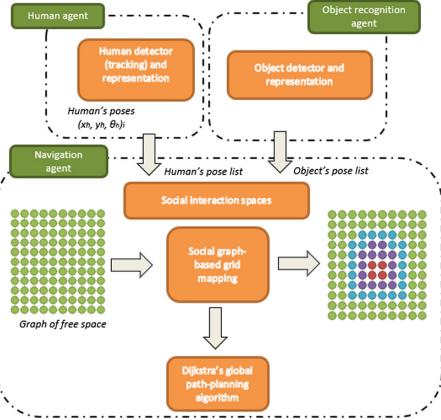
This paper is organized as follows: Sect. 2 provides a discussion of previous works related to robot navigation in environment with people. Section 3 presents an overview of the proposed social navigation architecture, including the definition of the social interaction spaces. Section 4 describes the socially-accepted path-planning algorithm presented in this paper. In Sect. 5, the experimental results are outlined. The conclusions and future works of the approach are summarized in Sect. 6.

## 2 Related Work

Path-planning in human environments is a complex problem that has aroused great interest in recent years. The way in which a robot navigates in these environments must not only consider task constraints, such as minimizing the distance traveled to the target, but also social rules, such as keeping a comfortable distance from humans [6]. Most works use proxemics (*i.e.*, the relationship between distances and the type of interaction) in order to plan a socially-accepted path [3, 7, 8]. These works typically define regions in which robot's navigation is forbidden. Other authors use the term affordances of objects and/or activity spaces, and try to prevent robots from navigating near them creating regions where navigation is forbidden [4, 9]. The definition of these regions that allow a more social navigation is what is usually called *social mapping*, which extends classical concepts such as *metric and/or semantic mapping* by defining these social interaction spaces. The proposal described in this paper uses a framework to perceive social interaction spaces and build a social map of the environment.

Classical methods need global path planners in order to choose the best route from the robot to the target and then, they apply social conventions and constraints to modify this path. Classical global path planners use a spatial representation of the robot's surrounding, so they require a map of the environment. Numerous path-planning algorithms have been proposed in the literature, from classical Dijkstra or A\* algorithms to other more complex systems. An interesting review of path planning algorithms was written by LaValle et al. [14].

How autonomous robots move in human environments has a strongly effect on the perceived intelligence [10]. A path that explicitly takes into account the human presence in the environment must address situations such as not passing between two people talking or avoid getting out of the field of view of the people, with the possibility of scaring them unnecessarily. Social navigation started being extensively studied in the last years and several methods have been proposed since then. On one hand, some authors propose models of social rules by using cost functions [11, 12]. A typical solution is to add social conventions and/or social constraints. In [11], for instance, the authors use a classical A\* path planner in conjunction with social conventions, like to pass a person on the right. Other work such as [12] use potential fields and a proxemics model. On the other hand, several authors use human intentions in order to model the social



**Fig. 2.** Overview of the proposal.

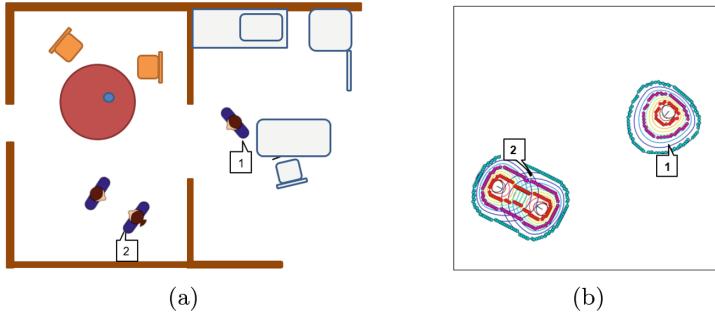
navigation [13]. Recently, in the work presented in [6], the concept of interaction spaces and their use to define social paths is introduced. This same concept is described in this article, but also adding the spaces of interaction between people and objects in the environment. The proposal uses the classical Dijkstra's algorithm, where weights of the graphs are modified in order to take into account the social map of the environment.

### 3 Social Interaction Spaces in Real Environments

This section describes the framework for planning socially-acceptable paths in human and interactive environments. In order to compute paths in these real scenarios, it is necessary to create a social map of the environment. For this reason, the robot's perception system needs to: (i) detect and model people (position and orientation); (ii) model their social space of interaction; (iii) group people in case they are engaged in interaction, modelling the space they need to do such interaction; and (iv) detect objects and model their space of interaction according to their *Space Affordances*. Figure 2 shows an outline of the proposed system. The proposed framework uses the CORTEX cognitive architecture for communication between perception agents and the robot navigation system (see [16] for a detailed description of the architecture). Next, the framework is described in details.

#### 3.1 Social Spaces of Interaction

The proposal for human-aware social navigation uses the model described in [4]. In this model, the presence of people generate regions where navigation is forbidden or penalized. Let  $H_n = \{h_1, h_2 \dots h_n\}$  be a set of  $n$  humans detected by the people perception system, where  $h_i = (x, y, \theta)$  is the pose of the  $i$ -th



**Fig. 3.** (a) People in a simulated environment; (b) asymmetric Gaussian associated to person ‘1’ and clustering of the group of two people labeled as ‘2’ in Fig. 3a.

human in the environment<sup>1</sup>. To model the personal space of each individual  $h_i$  an asymmetric 2-D Gaussian curve  $g_i(x, y)$  is used, as described in [4].

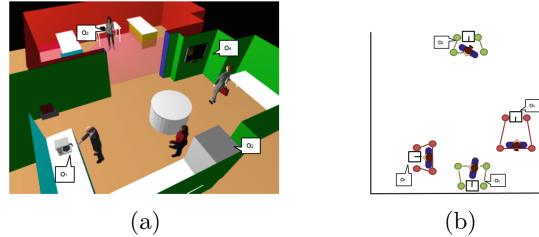
Once people have been detected, the algorithm clusters humans in the environment according their distances by performing a Gaussian Mixture [4]. The personal space function  $g_i(h)$  of each individual  $i$  in the environment is summed and a Global Space function  $G(p)$  is built. From this function, a contour  $J_i$  is established as a function of the density threshold  $\phi$ . Finally, the contours of these forbidden regions are defined by a set of  $k$  polygonal chain (*i.e.*, polyline)  $L_k = \{l_1, \dots, l_k\}$ , where  $k$  is the number of regions detected by the algorithm. The curve  $l_i$  is described as  $l_i = \{a_1, \dots, a_m\}$ , being  $a_i = (x, y)_i$  the vertices of the curve, which are located in the contour of the region  $J$ .

According to [8] it is possible to classify the space around a person into four zones, depending on social interaction: public, social, personal and intimate zones. Each human  $h_i$  present in the environment will have three associated spaces: the intimate space, defined by the polyline  $L_k^{intimate}$ ; the personal space, defined by  $L_k^{personal}$ ; and the social space, delimited by  $L_k^{social}$ , each of them being larger than the previous one, as it was introduced in [8]. The public zone will be the remaining free space. These contours, which are created by choosing different values of the density threshold  $\phi$ , can be seen in the Fig. 3: in color red is shown the intimate space, in purple the personal one and as blue color the social space.

### 3.2 Space Affordances and Activity Spaces

The concept of *Space Affordances* refers to areas where humans usually perform particular activities [9]. In interactive scenarios, these spaces are related to objects with which people often interact, for example, the space near a poster or a coffee machine. These spaces are called *Activity spaces* when people interact with objects.

<sup>1</sup> The actual detection of humans is out of the scope of the paper. In the experiments carried out it was performed by the Human agent of the CORTEX architecture.



**Fig. 4.** Examples of social interaction spaces: (a) People interacting with different objects in the environment; and (b) the corresponding space affordances representation generated by the algorithm.

Let  $O_n = \{o_1, \dots, o_n\}$  be the set of  $n$  objects with which humans interact in the environment. Each object  $o_k$  stores the interaction space  $i_{o_k}$  as an attribute, which is associated to the space required to interact with this object, and also its pose  $p_{o_k} = (x, y, \theta_k)$ ,

$$o_k = (p_{o_k}, i_{o_k})$$

Different objects in the environments have different interaction spaces. For instance, when using a coffee machine, a smaller space is needed in comparison to when reading a poster because it can be done from a farther distance. Next, the *Space Affordance*  $A_{o_k}$  is defined for each object  $o_k \in O_n$ . In this paper, the shape of these spaces has been modeled as an symmetrical trapezoid with height  $a_h$  and widths  $(a_{w1}, a_{w2})$ , as described in [4].

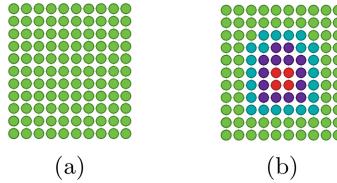
The *Activity Space*  $A_{o_k}$  is modeled by a polyline described by four vertices  $v_a$  that will be used to delimit forbidden areas for navigation. Finally, the set  $L_o = \{A_{o_1}, \dots, A_{o_n}\}$  describes the set of polylines used by the navigation algorithm for defining forbidden navigation areas. In Fig. 4a four humans in different poses and four objects are shown (a coffee machine, a fridge, a phone, and a pin board).

## 4 Socially-Acceptable Path-Planning Algorithm

This section describes the social path-planning algorithm. The space is represented by a uniform graph where obstacle-free nodes have a constant finite traversal cost and non-free nodes have an infinite one.

#### 4.1 Graph-Based Grid Mapping

Space is represented by a graph  $G(N, E)$  of  $n$  nodes, regularly distributed in the environment. Each node  $n_i$  has two parameters: availability,  $a_n$ , and cost,  $c_n$ . The availability of a node is a boolean variable whose value is 1 if the space is free, 0 otherwise. The cost,  $c_i$ , indicates the traversal cost of a node, i.e., what it takes for the robot to visit that node. Initially, all nodes have the same cost 1. Figure 5a shows an original free-space graph in which all nodes have the same cost and availability (as there are no obstacles in the area depicted).



**Fig. 5.** Graph-based grid mapping: (a) original free-space graph; and (b) final free-space graph, after including the social interaction space associated to a person.

The classical Dijkstra algorithm is employed for determining of the shortest path between an initial position and a target to which the robot must travel. Given a node of origin, the algorithm calculates the cost from origin to the target node taking into account the cost of the nodes. The cost of a path is the sum of the cost of the nodes that compose it.

## 4.2 Social Graph-Based Grid Mapping

The free space graph is modified to include the social spaces of interaction: firstly, those associated with the interaction between one person and another -or groups of people-, and secondly, those associated with the interaction between people and objects.

**Personal Space Mapping.** Being  $A$  the matrix formed by the availability of each node of the free space graph and  $C$  the matrix formed by the costs and considering the set of polygonal curves defined bellow,  $L_k^{intimate}$ ,  $L_k^{personal}$  and  $L_k^{social}$ , this paper present the modification of the cost and availability of the nodes of the graph according to these interaction spaces.

In first place, considering only the intimate space around the person  $h_i$ , for each polyline  $l_i^{intimate}$  is defined a polygon  $P_i^{intimate}$  formed by the points of the polyline. The availability  $a_i$  of all the nodes  $N_i \in N$  contained in the space formed by  $P_o^{intimate}$  is set to occupied,  $a_i = occupied$ . This means that the robot will not be able to invade this space, as it would disturb the person. For personal and social spaces, the availability of the nodes of the graph will not be modified, but its cost will be changed.

Considering the personal space around the human  $h_i$ , for each polyline  $l_i^{personal}$  a polygon  $P_i^{personal}$  has been defined. The cost  $c_i$  of all the nodes  $n_i \in N$ , contained in the space formed by  $P_p^{personal}$  will be modified and set to  $c_i = 4.0$ . In the same manner, for the social space, a polygon  $P_p^{social}$  is defined for each polyline  $l_i^{social}$ . All the nodes  $N_i \in N$  contained in the space formed by  $P_i^{social}$  will have cost  $c_i = 2.0$ . The public space will be the rest of the graph whose costs remain unchanged. Figure 5b show the final free-space graph, where the costs of nodes are modified according to the social spaces of interaction.

Intimate areas forbidden for navigation. Personal and social spaces are available, but their costs are higher, being personal spaces more expensive than social

spaces. This way, when the robot plans the shortest path, it will move away from the person. The social and personal spaces are not considered occupied so if the robot does not have enough space to navigate, for example in a corridor, it won't be blocked, but it will navigate through the social space, even if its cost is higher. If the robot does not have another alternative, it will cross the personal space, but it will never cross the intimate one.

**Space Affordances of Objects.** This same technique has been used for *Space Affordances*. Let  $L_o = \{A_{o_1}, \dots, A_{o_n}\}$  be the set of polylines that describe the defined *Space Affordances*. For each  $A_{o_i}$  the polygon  $P_i^{aff}$  is formed. The nodes of the free space graph  $N_i \in N$  contained in  $P_i^{aff}$  are modified in order to set its cost to  $c_i = 1.5$ . In this way, the *Space Affordances* have less weight in the graph than the social space of the person, so if the robot have to go through one of them, it will go through the *Space Affordance*.

## 5 Experimental Results

The software has been written in C++. The tests have been performed on a PC with an Intel Core i5 2.4GHz processor with 4Gb of DDR3 RAM and GNU/Linux Ubuntu 16.10. In order to assess the effectiveness of the proposed navigation approach, the methodology has been evaluated accordingly to the following metrics: (i) average minimum distance to a human during navigation,  $d_{min}$ ; (ii) distance traveled,  $d_t$ ; (iii) navigation time,  $\tau$ ; (iv) cumulative heading changes,  $CHC$ ; and (v) personal space intrusions,  $\Psi$ . These metrics have been already established by the scientific community (see [17, 18]).

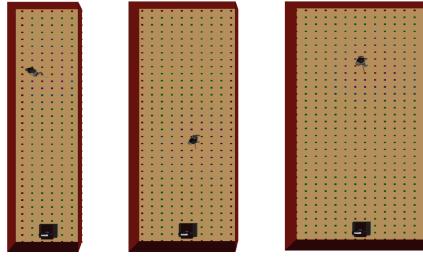
### 5.1 Navigation with Interaction Spaces

To evaluate the performance of the navigation algorithm, several simulations have been performed in three different environments rooms using a simulated robot. The widths of the rooms used were 2, 3 and 4m. Figure 6 depicts the used scenarios. The robot had to navigate from the position  $x = 0\text{ m}$ ,  $y = 0\text{ m}$  to  $x = 8.5\text{ m}$ ,  $y = 0\text{ m}$ , through those scenarios in which a person was located in random positions.

The results obtained for the simulations of rooms 2, 3 and 4 meters wide can be found in the tables 1, 2 and 3, respectively.

### 5.2 Interactive Scenario with Space Affordances

A rectangular simulated environment with a whiteboard has been used to test the effectiveness of the Space Affordance algorithm. The simulated environment is shown in Fig. 7a. The object has been placed in the position  $x = 2\text{ m}$ ,  $y = 4.5\text{ m}$  with  $a_s = 3\text{ m}$  in order to create a *Space Affordance* which the robot has to avoid, if it is being used as an *Activity Space*.



**Fig. 6.** Scenarios used in the second experiment: (a–c) 2 m, 3 m and 4 m wides, respectively.

**Table 1.** Navigation results for 2 m wide room considering interaction spaces

Navigation with social behaviour		Navigation without social behaviour	
Parameter	Obtained value ( $\sigma$ )	Parameter	Obtained value ( $\sigma$ )
$d_t$ (m)	9.71 (0.56)	$d_t$ (m)	9.10 (0.07)
$\tau(s)$	40.39 (11.83)	$\tau(s)$	35.54 (3.72)
$CHC$	1.49 (0.58)	$CHC$	0.79 (0.17)
$d_{min}$ Person (m)	1.13 (0.16)	$d_{min}$ Person (m)	0.47 (0.2)
$\Psi$ (Intimate) (%)	0.0 (0.0)	$\Psi$ (Intimate) (%)	1.58 (2.17)
$\Psi$ (Personal) (%)	4.43 (7.38)	$\Psi$ (Personal) (%)	5.55 (4.23)
$\Psi$ (Social) (%)	16.46 (11.46)	$\Psi$ (Social) (%)	17.26 (13.28)
$\Psi$ (Public) (%)	79.10 (13.92)	$\Psi$ (Public) (%)	75.60 (11.16)

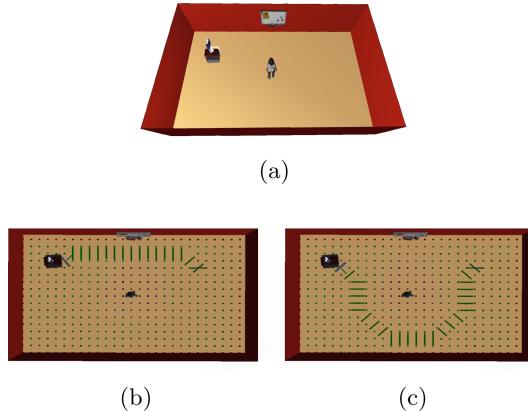
**Table 2.** Navigation results for 3 m wide room considering interaction spaces

Navigation with social behaviour		Navigation without social behaviour	
Parameter	Obtained value ( $\sigma$ )	Parameter	Obtained value ( $\sigma$ )
$d_t$ (m)	10.02 (0.49)	$d_t$ (m)	9.50 (0.16)
$\tau(s)$	31.39 (3.64)	$\tau(s)$	29.95 (2.75)
$CHC$	0.88 (0.18)	$CHC$	0.68 (0.16)
$d_{min}$ Person (m)	1.62 (0.28)	$d_{min}$ Person (m)	0.70 (0.38)
$\Psi$ (Intimate) (%)	0.0 (0.0)	$\Psi$ (Intimate) (%)	1.06 (2.13)
$\Psi$ (Personal) (%)	0.0 (0.0)	$\Psi$ (Personal) (%)	8.15 (4.58)
$\Psi$ (Social) (%)	3.29 (5.41)	$\Psi$ (Social) (%)	14.95 (6.97)
$\Psi$ (Public) (%)	96.70 (5.41)	$\Psi$ (Public) (%)	75.83 (6.34)

A single human, placed in front of the object in the position  $x = 2$  m,  $y = 2$  m, has been used for this test. The robot has had to navigate from the position  $x = -0.8$  m,  $y = 3$  m to  $x = 4.5$  m,  $y = 3$  m, avoiding the *Activity Space*.

**Table 3.** Navigation results for 4 m wide room considering interaction spaces

Navigation with social behaviour		Navigation without social behaviour	
Parameter	Obtained value ( $\sigma$ )	Parameter	Obtained value ( $\sigma$ )
$d_t$ (m)	10.81 (0.55)	$d_t$ (m)	9.97 (0.12)
$\tau(s)$	45.65 (19.24)	$\tau(s)$	36.64 (2.28)
$CHC$	1.27 (0.51)	$CHC$	0.86 (0.13)
$d_{min}$ Person (m)	1.76 (0.18)	$d_{min}$ Person (m)	0.80 (0.59)
$\Psi$ (Intimate) (%)	0.0 (0.0)	$\Psi$ (Intimate) (%)	0.74 (1.48)
$\Psi$ (Personal) (%)	0.0 (0.0)	$\Psi$ (Personal) (%)	2.97 (2.52)
$\Psi$ (Social) (%)	2.017 (3.37)	$\Psi$ (Social) (%)	5.55 (4.84)
$\Psi$ (Public) (%)	97.98 (3.37)	$\Psi$ (Public) (%)	90.74 (7.73)

**Fig. 7.** Interactive scenario described for the test and navigation results with and without *Space Affordances*: (a) original scenario; (b) Navigation without Space Affordance; and (c) Navigation with Space Affordance

The same test has been carried out with and without *Space Affordances*. The comparison between the different paths the robot took can be seen in Fig. 7b, and Fig. 7c where the paths planned have been highlighted. It can be noticed that, in the first case, the robot interrupts the human in the performance of its activity.

Table 4 shows the results of navigation with and without *Space Affordances*, obtained for each of the metrics used: average minimum distance to a human during navigation,  $d_{min}$ ; distance traveled,  $d_t$ ; navigation time,  $\tau$ ; cumulative heading changes,  $CHC$  and personal space intrusions,  $\Psi$ . It is also indicated whether the activity performed by the human has been interrupted or not.

**Table 4.** Navigation results with space affordances

Navigation with space affordances		Navigation without space affordances	
Parameter	Value ( $\sigma$ )	Parameter	Value ( $\sigma$ )
$d_t$ (m)	8.76 m	$d_t$	5.18 m
$\tau$	64.1 s	$\tau$	33.84 s
$CHC$	1.47 (0.11)	$CHC$	0.21 (0.05)
$d_{min} Person$ (m)	0.78 (0.007)	$d_{min} Person$ (m)	1.10 (0.005)
$\Psi$ (Intimate) (%)	0.0 (0.0)	$\Psi$ (Intimate) (%)	0.0 (0.0)
$\Psi$ (Personal)(%)	0.0 (0.0)	$\Psi$ (Personal)(%)	0 (0.0)
$\Psi$ (Social)(%)	15.46 (0.6)	$\Psi$ (Social)(%)	12.54 (0.57)
$\Psi$ (Public)(%)	84.53 (0.6)	$\Psi$ (Public)(%)	87.44 (0.9)
<i>Interruption</i> (Y/N)	N	<i>Interruption</i> (Y/N)	Y

## 6 Conclusions and Future Works

This article presents an extension of an algorithm for planning socially-accepted paths in human environments. The algorithm is based on the well-known Djisktra's algorithm, where the original free space graph is modified according to the social interaction spaces. This article takes into account the personal spaces in an interaction between people, and also the spaces between people and the objects with which they interact. The metrics used to validate the proposal demonstrate that the planned paths have a socially-accepted behavior.

Although the results demonstrate the validity of the proposal, in future works the use of a real robot and questionnaires is considered in order to gather information on the acceptability of the planned paths.

**Acknowledgment.** This work has been partially supported by the National project RTI2018-099522-B-C42, by the Extremaduran Government projects GR15120, IB18056 and by the FEDER project 0043-EUROAGE-4-E (Interreg V-A Portugal-Spain - POCTEP).

## References

1. Gomez, J., Mavridis, N., Garrido, S.: Social path planning: generic human-robot interaction framework for robotic navigation tasks. In: Workshop on Cognitive Robotics Systems: Replicating Human Actions and Activities at IEEE/RSJ International Conference on Robots and Systems (2013)
2. Lichtenhaler, C., Peters, A., Griffiths, S., Kirsch, A.: Social navigation - identifying robot navigation patterns in a path crossing scenario. In: Lecture Notes in Computer Science, vol. 8239, pp. 84–93 (2013)
3. Charalampous, K., Kostavelis, I., Gasteratos, A.: Robot navigation in large-scale social maps: an action recognition approach. Expert Syst. Appl. **66**, 261–273 (2016)

4. Vega, A., Manso, L., Bustos, P., Núñez, P., Macharet, D.: Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances. *Pattern Recogn. Lett.* **1**, 72–84 (2019)
5. Foux, G., Heymann, M., Bruckstein, A.: Two-dimensional robot navigation among unknown stationary polygonal obstacles. *IEEE Trans. Robot. Autom.* **9**, 96–102 (1993)
6. Weihua, C., Tie, Z., Yanbiao, Z.: Mobile robot path planning based on social interaction space in social environment. *Int. J. Adv. Rob. Syst.* **1**, 1–10 (2018)
7. Kruse, T., Pandey, A., Alami, R., Kirsch, A.: Human-aware robot navigation: a survey. *Robot. Auton. Syst.* **61**(12), 1726–1743 (2013)
8. Rios-Martinez, J., Spalanzani, A., Laugier, C.: From proxemics theory to socially-aware navigation: a survey. *Int. J. Soc. Robot.* **7**(2), 137–153 (2015)
9. Rios-Martinez, J.: Socially-aware robot navigation: combining risk assessment and social conventions. Ph.d. Inria, France (2013)
10. Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T., Christensen, H.I.: Navigation for human robot interaction tasks. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 1894–1899 (2004)
11. Kirby, R., Simmons, R., Forlizzi, J.: COMPANION: a constraint-optimizing method for person-acceptable navigation. In: IEEE International Symposium on Robot and Human Interactive Communication, pp. 607–612 (2009)
12. Tranberg Hansen, S., Svenstrup, M., Andersen, H.J., Bak, T.: Adaptive human aware navigation based on motion pattern analysis. In: IEEE International Symposium on Robot and Human Interactive Communication, pp. 927–932 (2009)
13. Photchara, R., Mae, Y., Ohara, K., Kojima, M., Arai, T.: Social navigation model based on human intention analysis using face orientation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. vol. 1, pp. 1682–1688 (2010)
14. LaValle, S.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
15. Lam, C., Chou, C., Chiang, K., Fu, C.: Human-centered robot navigation - towards a harmoniously human-robot coexisting environment. *IEEE Trans. Rob.* **27**(1), 99–112 (2011)
16. Bustos, P., Manso, L., Bandera, A., Bandera, J.P., García-Varea, I., Martínez-Gómez, J.: The CORTEX cognitive robotics architecture: use cases. *Cogn. Syst. Res.* **55**, 107–123 (2019)
17. Okal, B., Arras, K.: Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning. In: IEEE International Conference on Robotics and Automation, pp. 2889 – 2895 (2016)
18. Kostavelis, I.: Robot behavioral mapping: a representation that consolidates the human-robot coexistence. *Robot. Autom. Eng.* **1**, 1–3 (2017)



# A Modelling and Formalisation Tool for Use Case Design in Social Autonomous Robotics

Alba Gragera<sup>(✉)</sup>, Alba María García, and Fernando Fernández

Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain  
[agragera@pa.uc3m.es](mailto:agragera@pa.uc3m.es), [100346091@alumnos.uc3m.es](mailto:100346091@alumnos.uc3m.es), [ffernand@inf.uc3m.es](mailto:ffernand@inf.uc3m.es)

**Abstract.** The modelling of use cases in Social Robotics and its formalisation in a control architecture is a complex and time consuming task, which can be seen as a knowledge engineering process. This manuscript focuses on the description of a novel tool created to simplify the process of modelling these interactions through the definition of such use cases as nominal behaviours with exogenous events. The control architecture of the robotic platform used is based on Automated Planning, allowing robots to deal with stochastic environments. The tool comprises two main objectives: to ease the knowledge engineering process to model social interactions and to generate Planning Domain Description Language code (PDDL) as a formalisation that can be injected into a control architecture. We model different use cases and test the PDDL formalisation by using the control architecture. The test carried out shows evidence of good results, encouraging further research of this type of tools to provide autonomy and facility to experts and non-experts in the field.

**Keywords:** Autonomous robotics · Automated planning · Use case modelling

## 1 Introduction

Social Robotics [13] is a branch of robotics where machines are in close contact with people and must be able to make decisions and change their course of action depending on the environment. Since the expression *make decisions* is introduced, we discard robots for remote control and those ones with just reactive behaviours. For that reason, we are focused on Social Autonomous Robotics, which have to deal autonomously with people and uncertain environments, where every action can be interrupted by an external event. To manage that, Automated Planning (AP) [5] is applied to implement the deliberation process, where Classical Planning is known to achieve great results in a wide range of domains. By using this approach, AP is able to generate a sequence of actions to be carried out by the robot in order to solve the problem or meet a certain goal.

In order to do this, a control architecture and a formal definition language are required. In this work, PELEA [1] (Planning, Execution and Learning Architecture) and PDDL 2.1 [3] (Planning Domain Definition Language) are assumed.

Current research about Social Robotics through Classical Planning and replanning already exists [7, 21]. However, applying it presents a major challenge since we have to identify and specify all possible situations that could occur, specially in robots with sophisticated behaviours. This implementation involves extracting knowledge from experts (for instance, therapists or clinicians), who determine how the interaction flow between the robot and the person has to be, i.e. the specific use case that must be accomplished. Modelling the use case is a time consuming endeavour based on a knowledge engineering process [12], which needs cooperation between these experts and engineers to ensure that the request of the expert will be fulfilled in the robot performance. Related to this, developing PDDL code and the process behind that usually is a bottleneck aimed to be carried out by professional experts in the Automated Planning field.

The main purpose addressed in this work is the development of a modelling tool that enables the robot use case definition in a effortless way. The application is based on an interface where experts can introduce information graphically while it displays the interaction workflow on the screen. This tool is restricted to make sure that only suitable information can be inserted, but it is complete enough to design complex behaviours for Social Robotics. We have found previous works in this field [6], where challenges while modelling Classical Planning domains for Social Robotics are identified. However, they do not mention how to model those features in PDDL code or how to handle them at a high-level. Therefore, we continue research and also build a formalisation that maps the conceptual model graphically introduced to the formal definition language. As a consequence, the main idea of this work is to provide a modelling tool where domain experts not only can introduce the desired workflow that must be followed by the robot, but also generate the corresponding code suitable for the proper system operation, all of this without further engineer support. Thanks to this, we hope to bring formalisation closer to domain experts and to provide them autonomy, saving time and efforts on both sides.

This manuscript focuses on the description of the developed tool and the main concepts that it involves. In the next section we present the background of this work. Next, Sects. 3 and 4 explain the concepts behind our work, providing a use case in complex Social Robotics environments. Finally, the evaluation performed and the main conclusions are detailed.

## 2 Background

In this section we present the main research fields that compose our work, which includes Automated Planning, Social Autonomous Robotics and control architectures, showing how they can take advantage of their combination. We also mention the state of the art about similar tools currently used.

### 2.1 Automated Planning

The application of Automated Planning consists in the use of a problem-solver to find a sequence of actions that allows to transit from the initial state to a

state where all goals are met. As we mentioned, Classical Planning [4] is used in this work. The main characteristic of those models is that they need complete information about the world, which means that it is necessary to model every fact that can take place and have some impact in the execution.

Based on this, a planning task is a tuple  $P = (S, A, I, G)$ , where  $S$  is set of states or possible situations which can occur during the execution of the plan, while  $A$  is the set of actions that allow to transit from one state to another one. Actions are deterministic operators defined through a declarative language and composed of preconditions and effects. These are sets of facts grouped by conjunctions or disjunctions which establish the requirements that must be true to perform an action and the impact of carrying it out.  $I \subset S$  defines the initial state and  $G \subset S$  specifies the goal or conditions that the plan must meet. As a result, a sequence of actions called plan is obtained, formally defined as  $\pi = (a_1, a_2, \dots, a_n)$ , by which the agent is able to achieve the established goal. This plan is calculated at the beginning considering that no event is going to interrupt the regular workflow, what we call nominal behaviour. However, real world is unpredictable, so a several range of unexpected events may take place. To deal with that, one of the main advantages of the Automated Planning is the ability to re-plan, in such a way that the planner finds a new plan  $\pi'$  which handles the current situation and continues reaching the purposed goal.

The typical way to represent this model is through the domain and the problem files in PDDL code, where the domain contains a list of possible actions, and the problem includes the specification of all those predicates or facts which are true in the initial state, along with a set of goals that must be achieved.

## 2.2 Social Autonomous Robotics

Taking the previous idea, the concept of robots with reactive behaviours can be switched for autonomous robots that can make decisions and change their actions according to the context. This approach is especially common in Social Robotics, where these robots have to interact socially with humans in a fluent and natural way. Achieving that is still a challenge [20] addressed in the study of human-robot interaction (HRI). The main issue is that Social Robotics should have a wide range of behaviours depending on the situation, which makes machines with sophisticated operating modes.

To achieve that, several approaches related to Artificial Intelligence have been addressed. Some works rely on simulating the human way of learning by using symbolic models implemented through networks and layers [15]. However, these kind of models are hard to program and highly domain dependent, making them difficult to reuse and understand. In the literature also appears solutions which are designed with Finite State Machines (FSM) [14], an easier solution only in those robots with simple behaviour due to the complexity of identifying and connecting all possible state transitions.

For those cases where the application of FSM is a challenging task, Automated Planning is a field addressed in previous works with great success [2, 18], since the task of finding a possible sequence of actions that allow to transit to

the goal state and its control is solved thanks to the use of a problem solver and a monitoring architecture. Therefore, this approach provides a straightforward way to code the desired behaviour on the robotic platform.

### 2.3 Planning, Execution and Learning Architecture

Since robots have deliberative behaviours, a control architecture is needed to handle and supervise the performance of the robot. This layer is between the machine and the problem solver and it is used to translate the external information sensed by the robot, which is received at a low level. This information provides data about the state of the world and the current situation, which should fit with the expected state according to the plan  $\pi$ . To verify if both information match, a translation between low and high level has to be carried out. If any problem is found, then a new plan  $\pi'$  is calculated. Otherwise, the plan will continue with the expected execution. An example of such architecture is PELEA, the control architecture used in this work. It contains three main modules, which are synchronised to ensure the consistency of the system. These are EXECUTION, MONITORING and DECISION SUPPORT, following the operational mode shown in Fig. 1, which is executed in a loop until the goals are met.

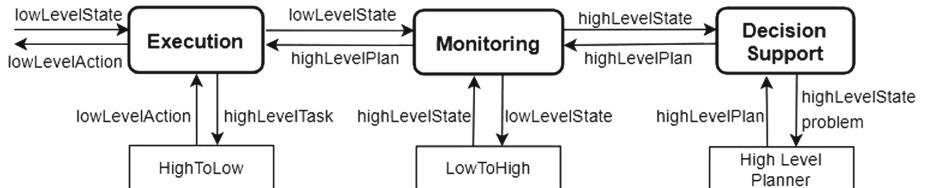


Fig. 1. PELEA architecture

Regarding to minor modules, HIGHToLOW and LOWToHIGH, they receive a catalogue of actions in both levels, high and low, where the mapping between them are specified. These files are written *ad hoc* and are domain dependent, needing to be changed according to the desired execution. About the High Level Planner, this architecture also contains a plan-solver called Metric-FF [10], an extension of the FF (Fast-Forward) [9] domain independent planner adapted to handle with numerical variables. The planner input is the domain and the problem developed in PDDL code.

### 2.4 Similar Tools

Besides the number of PDDL editors, several tools to make the process of developing PDDL code easier already exist. The first one is GIPO [19], where a planning domain definition can be made graphically through an object-centric perspective. On this application, each object is defined with an abstract state

machine, specifying the different states and actions that involves the object. The predicates can be created by associating the objects with their properties, while the PDDL actions are generated combining these state diagrams. Similar to this, VLEPPO [8] allows to create problems and domains by using an interface, where definition of predicates and actions are made explicitly by defining an ontology which contains the classes and relations. From this information to obtain the predicates and actions in PDDL code is possible.

These two tools make a translation to PDDL, similarly to our purpose. However, none of them allow to define workflows or to deal this Social Robotics issues, features that we want to exploit in our work.

### 3 Modelling Autonomous Robotic Tasks

Since AP problems attempt to find a sequence of tasks to achieve a goal, they can be defined as state transition systems which represent that flow of actions. Then, the logical approach to model it in a graphic interface is using a state graph. Before entering in the details of this model, we have to clarify some concepts.

#### 3.1 The Nominal Flow

The *nominal flow* is the state graph of a model without exogenous events, where we assume that no incident nor stochastic outcomes will happen. This normal behaviour is modelled through states and actions connecting them. States are composed by predicates and functions, which are classified in different categories: persistent, they represent facts that, once added, should not be deleted; static/dynamic, they may or may not change its state during the execution of the plan; and internal/sensed, depending on whether its value is obtained from the sensors of the robot or it is an internal value. Every predicate will be marked by choosing some of these characteristics, depending on the function that it has in its domain.

The result of modelling nominal flow is a directed graph  $S = (S, A)$ , where  $S$  is the set of vertex representing the states, which contains several predicates that will be true or false by the time that the state describes.  $A$  is the set of edges that represent actions, since they allow to transit from one state to another.

#### 3.2 Identifying Exogenous Events

The environment is represented along with the nominal workflow, where exogenous events can occur. Exogenous events are those that may cause interruption or failure. They are excluded from the nominal flow because they can happen at any time and none of the actions of the normal behaviour can produce them. Therefore, these events will generate an unexpected state that must be fixed in order to return to nominal schedule. Then, we have to take into account minor workflows for each interruption to recover it. What it is usually done in Classical Planning is to go back to the state in which the failure happened and to continue

with the execution from that point. But following our approach, the main idea is to deal with replanning at a high level and return to a determinate state of the workflow, providing coherence to the robot performance.

### 3.3 Defining *Islands*

The states of the *nominal flow* can be grouped to form an *island*. The reason of it is that every social interaction has a number of steps that are usually executed in order, like to greet each other, to start talking, to do some activity and finally the goodbye. In this way, we define *islands* as a group of actions that determine the different stages of the human interaction.

Related to this, a landmark [11] state in planning is a state which must be transited to go from the initial state to any goal state. In addition, we use the concept of landmark to identify returning states from exogenous events. The user can set any state like the beginning of an *island* and this point will act as a landmark, a state where the execution must return when it recovers the nominal flow. In case of having several islands, the plan will get back to the start of the last island where it was executing. We decided to introduce this option to provide coherence to the robot performance.

### 3.4 Representing Sequential Connections and Loops

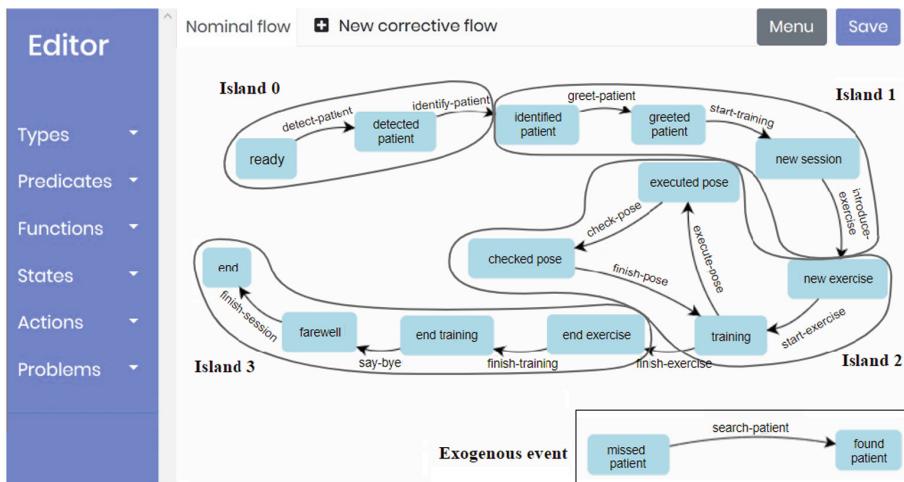
In the nominal flow we can also find some elements that compose it, which are important for modeling and formalization.

- **Sequential connections.** It is the simplest way to connect two or more states. An edge  $a \in A$  connects two vertices  $s_1, s_2 \in S$  if the first state goes before the next in terms of social interaction. In other words, state  $s_1$  is the precondition of action  $a$ . Since a state contains several predicates, that means that all predicates inside will be preconditions of the action  $a$ . About the effects, as a way of generalisation and to simplify the model all effects are included in the actions themselves.
- **Loops.** Social behaviour can also present loops in such a way that an action can be repeated several times until a condition is satisfied. In this sense, a loop in the diagram can be seen as a *for* or a *while* statement. This difference lies in the type of predicate that is part of the condition, namely, internal or sensed. It will be similar to *for* statement when a internal predicate takes part. In that case we usually have a numerical variable used as a counter, which is initialized in the problem and is increased or decreased along the cycles, provided that it is different that the exit condition. Otherwise, the loop will be analogous to a *while* statement if the predicate is sensed. Similarly, the variable must meet a condition to leave the loop, but it is not modified in a explicit way, but sensed from the exterior. In order to simplify the formalization, when a loop appears in the domain, users must mark a checkbox specifying what is the action that allows to leave the loop.

## 4 Formalizing Complex Social Robotics Use Cases

To provide a clear example about the concepts explained and the utility of our system, a NAO THERAPIST [16] interaction workflow is presented. NAO THERAPIST is a project that represents a support tool for therapists, based on a humanoid robot whose potential is to help children in physical rehabilitation therapies. Through this example, we show a complex flow composed by nominal behaviour containing loops and *islands* along with an exogenous event, as is depicted in Fig. 2. The interaction between the robot and the patient is to follow a number of steps, including rehabilitation exercises based on pose executions<sup>1</sup>. This workflow has been model specifying the characteristics of each state through predicates and connecting them with the actions that will build the domain.

The objective of this section is to describe how these flows can be formalized at high level. Such formalization requires the use of PDDL code along functions, disjunctions and control predicates based on the input model. Since users only have to decide how to model it graphically and define islands, exogenous events and so on, the automatic generation of PDDL operates as a black box for them, adding extra code to ensure the proper performance.



**Fig. 2.** Use case for NAO Therapist modelled in the editing tool (the original image has been modified to provide a clearer example in the paper)

<sup>1</sup> Different videos of NaoTherapist can be seen in <https://www.youtube.com/user/NAOTherapist>.

#### 4.1 Formalizing the Nominal Flow

Looking at the example depicted in Fig. 2, the state `identified-patient` is a landmark since it has been identified as the beginning of an *island* by the user, so the plan will have to return at that point in case of failure or interruption. Getting back to that point means that the next action to be executed again in case of interruption will be to greet the person. Then, the action may be executed even if an error exists. We manage that by introducing an *or* statement in preconditions. It is also included in the effects that we are located in the first island. We remember that all these flags will be inserted automatically during the PDDL code generation. Figure 3 shows the resulting action.

```
(:action greet-patient
  :parameters (?r - robot ?p - patient)
  :precondition (and (not (emergency_situation ?r ?p))
                      (detected_patient ?r ?p)
                      (identified_patient ?r ?p)
                      (or (can_continue)(landmark)))
  :effect     (and (greeted_patient ?r ?p)
                      (can_continue)
                      (not (landmark))
                      (assign (island_number) 1))
)
```

**Fig. 3.** Code for greet-patient action

#### 4.2 Formalizing Exogenous Events

If the patient leaves the room, the robot has to detect that the person is missing. That state is not contemplated in the nominal behaviour, so it triggers the exogenous event (`not (detected_patient ?r ?p)`) that must be fixed by the action `search-patient`, shown in Fig. 4.

```
(:action search-patient
  :parameters (?r - robot ?p - patient)
  :precondition (and (not (can_continue))
                      (not (detected_patient ?r ?p)))
  :effect     (and (detected_patient ?r ?p)
                      (clear_states))
)
```

**Fig. 4.** Code for search-patient action

As effect of this corrective action, the predicate `clear-states` is inserted, activating an action automatically generated: `restore-from-one`. This action is responsible for deleting predicates added as effect of intermediate actions. This is the only way to return at the beginning of the island. Otherwise, the facts

added by these actions would still be true and there would be no need to execute them again, reason why the replanning problem would go back to the exit point where the event occurred. In this point the importance of the predicate definition comes up. All predicates that the user considers as persistent must be marked like that during the predicate definition.

An example of this action can be seen in Fig. 5. As many actions of this type will be generated as number of islands have been marked. Depending on the island where we were located when the exogenous event occur, the appropriate action will be activated to delete all predicates until the beginning of the *island*. These *island* definitions provides consistency to the robot performance, making it act more natural by controlling the error recovery in this way.

```
(:action restore-from-one
:parameters (?e - exercise ?r - robot ?p - patient ?s - session)
:precondition (and (clear-states)
                   (= (island-number 1))
:effect      (and (not (introduced_exercise ?e))
                   (not (robot_is_training ?r))
                   (not (greeted_patient ?r ?p))
                   (not (started_session ?r ?p ?s))
                   (landmark))
)
```

**Fig. 5.** Code for restore action

### 4.3 Formalizing Loops

To ensure the proper execution of the poses loop, it needs an exit condition defined as the number of repetitions and an internal predicate simulating a counter, which will increase in each cycle to reach the desired number of poses. According to that, we have a state in which two things can occur: the counter is lower than the exit condition or it is equal to it. The user has to model both possible options, where the former corresponds to the action that keeps running the loop (Fig. 6) and the last with the action that leaves it.

```
(:action execute-pose
:parameters (?pos - pose ?e - exercise ?pleft ?pright - posture)
:precondition (and (training_exercise ?e)
                   (pose_exercise ?pos ?e)
                   (pose_postures ?pos ?pleft ?pright)
                   (< (pose-counter)(pose-number ?e))
                   (can_continue))
:effect      (executed_pose ?pos)
)
```

**Fig. 6.** Execute pose action

The predicate (`pose-number ?e`) contains how many poses that must be performed in that exercise, while (`pose-counter`) is the current number of poses

already done. This counter is increased at the exit of the loop, in **finish-pose** action (Fig. 7). As effect of this action it is also important to delete the poses executed once they are finished, in order to ensure that the user could repeat the same pose or that there is no extra poses executed in case to return to the beginning of the island. The exit condition would be formalized in the action which leaves the loop, **finish-exercise**, through the predicate ( $=$ (pose-counter)(pose-number ?e)).

```
(:action finish-pose
  :parameters (?pos - pose ?e - exercise ?pleft ?pright - posture)
  :precondition (and (training_exercise ?e)
    (pose_postures ?pos ?pleft ?pright)
    (executed_pose ?pos)
    (correct_pose ?pos)
    (can_continue))
  :effect (and (increase (pose_counter) 1)
    (not (executed_pose ?pos)))
    (not (correct_pose ?pos)))
)
```

**Fig. 7.** Finish pose action

## 5 Graphic Editor and Evaluation

The editor handles all the process of modelling the use cases. It contains three main sections: lateral menu, graph area and management buttons. The lateral menu allows the user to edit the models completely, so that s/he can see in real-time all the changes on the graph area. Since the model may consist of several flows in the case exogenous events are added, the user can switch the view in the upper tabs to the desired one. Finally, the management buttons were created to save the model. An example of the interface is shown in Fig. 2.

An evaluation process has been carried out in order to test the correct performance of the developed system. We use the use case about NAOTherapist explained in the previous section. The generated PDDL code was introduced in the control architecture PELEA. For that purpose we developed two unequivocal *highToLow* and *lowToHigh* files where the high level action is the same as the low level action. We also deploy a ROS (Robot Operating System) [17] layer on PELEA to simulate sending messages between the system and the robot.

Each high level action is sent in its respective low level, waiting until the response of the robot. If the answer is successful, next action is transmitted. After **introduce-exercise**, we simulate that the patient is lost. In that moment the corrective action is applied as expected. In this case, the replanning time is 0.01 seconds, the same value as the computing time of the initial plan. It is known that this time may change in case of harder domains. However, search space is not larger since social robots behaviours are modelled as a workflow. Therefore, AP presents a good approach for that kind of tasks, simplifying the challenge of software developing for social robots to model complex behaviours.

## 6 Conclusions and Future Work

Applying Automated Planning to robotics has been extensively tested in several domains with great success. This work focuses particularly in social robots, which have to deal autonomously with people in dynamic and stochastic environments. One of the most time-consuming processes about Automated Planning is to develop PDDL code, specially in this field, where a great knowledge engineering process is required beforehand. In order to make this task easier, we have developed a tool where modelling an interaction workflow is a simple task. We also go a step further and we define a model to automatically formalize the diagram created to PDDL code, providing autonomy to those who are domain experts but are not developers. We find encouraging results in this high level approach and in the developed system, which is not only useful to model Social Robotics behaviour, but also to define several range of Automated Planning models, such us the ones used in planning competitions.

In future work, we are planning to extend the tool in order to include *highToLow* and *lowToHigh* files generation. This would simplify the development and integration of use cases by automating the whole process of getting the system ready when a new domain or use case is created, avoiding to write *ad hoc* files. PELEA already has an online server, so joining it with our system would allow any person who has a robot anywhere to create and to implement smoothly his/her own use cases.

**Acknowledgements.** This research has been partially supported by research projects founded by Ministerio de Ciencia, Innovación y Universidades (RTI2018-099522-B-C43) and Ministerio de Economía y Competitividad (TIN2015-65686-C5-1-R).

## References

1. Alcázar, V., et al.: PELEA: planning, learning and execution architecture. In: Proceedings of the 28th Workshop of the UK Planning and Scheduling (2010)
2. Cashmore, M., et al.: Rosplan: Planning in the robot operating system. In: Proceedings of the 25th International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, 7–11 June, pp. 333–341 (2015)
3. Fox, M., Long, D.: PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.* **20**, 61–124 (2003)
4. Geffner, H., Bonet, B.: A concise introduction to models and methods for automated planning. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, chap. 2, pp. 15–34. Morgan & Claypool Publishers (2013)
5. Ghallab, M., Nau, D.S., Traverso, P.: Automated Planning - Theory and Practice. Elsevier, Amsterdam (2004)
6. González, J.C., Fernández, F., García-Olaya, A., Fuentetaja, R.: On the application of classical planning to real social robotic tasks. In: Proceedings of the 5th Workshop on Planning and Robotics (PlanRob), ICAPS Conference, Pittsburgh, Pennsylvania, USA, pp. 38–47, June 2017
7. González, J.C., Pulido, J.C., Fernández, F.: A three-layer planning architecture for the autonomous control of rehabilitation therapies based on social robots. *Cogn. Syst. Res.* **43**, 232–249 (2017)

8. Hatzi, O., Vrakas, D., Bassiliades, N., Anagnostopoulos, D., Vlahavas, I.P.: A visual programming system for automated problem solving. *Expert Syst. Appl.* **37**(6), 4611–4625 (2010)
9. Hoffmann, J.: FF: the fast-forward planning system. *AI Mag.* **22**(3), 57–62 (2001)
10. Hoffmann, J.: The metric-FF planning system: translating “ignoring delete lists” to numeric state variables. *J. Artif. Intell. Res.* **20**, 291–341 (2003)
11. Hoffmann, J., Porteous, J., Sebastia, L.: Ordered landmarks in planning. *J. Artif. Intell. Res.* **22**, 215–278 (2004)
12. Kambhampati, S.: Model-lite planning for the web age masses: the challenges of planning with incomplete and evolving domain models. In: Proceedings of the Twenty-Second Conference on Artificial Intelligence AAAI, 22–26 July, Vancouver, British Columbia, Canada, pp. 1601–1605 (2007)
13. Leite, I., Martinho, C., Paiva, A.: Social robots for long-term interaction: a survey. *Int. J. Soc. Robot.* **5**(2), 291–308 (2013)
14. Lera, F.J.R., Olivera, V.M., González, M.Á.C., Rico, F.M.: HiMoP: a three-component architecture to create more human-acceptable social-assistive robots. *Cogn. Process.* **19**(2), 233–244 (2018)
15. Prenzel, O., Feuser, J., Graser, A.: Rehabilitation robot in intelligent home environment-software architecture and implementation of a distributed system. In: 9th International Conference on Rehabilitation Robotics, ICORR. pp. 530–535. IEEE (2005)
16. Pulido, J.C., González, J.C., Suárez-Mejías, C., Bandera, A., Bustos, P., Fernández, F.: Evaluating the child-robot interaction of the naotherapist platform in pediatric rehabilitation. *Int. J. Soc. Robot.* **9**, 343–358 (2017)
17. Quigley, M., et al.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software, Kobe, Japan, vol. 3 (2009)
18. Rajan, K., Py, F.: T-REX: partitioned inference for AUV mission control. In: Further Advances in Unmanned Marine Vehicles, pp. 171–199 (2012)
19. Simpson, R.M., Kitchin, D.E., McCluskey, T.L.: Planning domain definition using GIPO. *Knowl. Eng. Rev.* **22**(2), 117–134 (2007)
20. Tapus, A., Mataric, M.J., Scassellati, B.: Socially assistive robotics [grand challenges of robotics]. *IEEE Robot. Automat. Mag.* **14**(1), 35–42 (2007)
21. Vaquero, T.S., Mohamed, S.C., Nejat, G., Beck, J.C.: The implementation of a planning and scheduling architecture for multiple robots assisting multiple users in a retirement home setting. In: AAAI Workshop, Austin, Texas, USA, January 2015



# People Detection and Tracking Using an On-Board Drone Camera

Cristian Cifuentes-García<sup>(✉)</sup>, Daniel González-Medina,  
and Ismael García-Varea

University of Castilla-La Mancha, Campus Univ. s/n, 02071 Albacete, Spain  
*Cristian.Cifuentes@alu.uclm.es, {Daniel.Gonzalez,Ismael.Garcia}@uclm.es*

**Abstract.** The problem of people detection and tracking in unmanned ground vehicles has been studied in depth in computer vision and autonomous robotics research communities. Different well-known proposals have already been proposed to solve the problem of people detection and tracking using machine vision algorithms. However, for unmanned aerial vehicles, it is still a subject of research today. The lack of high-quality sensors and on-board cameras and the capability to process the data collected in real-time makes it difficult to achieve optimal solutions in real-time. In this work, we propose to use machine vision algorithms to process in real-time the images collected by the camera of a drone and subsequently performing the detection and tracking of people who are located in the environment. The proposal was experimentally evaluated comparing different semantic segmentation techniques. Finally, to validate the proposal, a real scenario was created and carried out, which consisted of detecting and tracking people with a drone autonomously in a controlled environment.

**Keywords:** People detection and tracking · Semantic segmentation · Computer vision · Robotics · Drone

## 1 Introduction

Most drones, from the most advanced to the most basic, incorporate a camera that allows them to record videos or take images of what they are capturing or viewing in real-time.

Drones are flying vehicles that, due to their versatility and aerodynamics, enable people to reach places that they are unable to reach by their means. Therefore, they can be of great help in locating and tracking people in real situations, for example, to perform or facilitate rescue operations in areas of difficult accessibility, to record video documentaries for high-risk sports or challenges, or in any other situations that involves a certain risk to people's lives.

In any of these situations, it is necessary to perform a processing of the image obtained by the drone camera to allow us to segment the scene and extract the elements of interest, in this case, people.

More specifically, this segmentation consists of identifying which pixels of the captured images belong to the different objects or elements of the scene. That it is usually done by identifying basic geometric shapes such as planes, spheres, and cylinders. They can also be obtained by analyzing surface normals to determine which points belong to the same object as shown in [10]. Once the image has been segmented, each segment is classified by assigning a label that represents its semantic category (house, person, dog, cat, bicycle, etc.). This whole process is called semantic segmentation.

In this work, we will use semantic segmentation techniques to locate people in images and track them by using a drone. The tracking will be carried out by sending specific command movements to the drone so that it autonomously follows the person according to the information extracted from the images captured with its camera.

More specifically, we will use and compare pre-trained RGB image classification models such as a linear SVM that uses images processed with the Histogram of Oriented Gradients [2] (SVM-HOG), the Google MobileNet-SSD [7, 12] trained using the framework Caffe [6], and the YOLOv2Tiny model trained using YOLO (You Only Look Once) [11]. Concretely, we will use the implementation of these image segmentation algorithms provided by the well-known and widely used OpenCV library [1]. We will also make use of the ROS robotic development environment [9], which provides us with tools to interact with real and simulated drones, specifically with the AR Drone 2.0 that we have used in our experiments.

The experimental results that we have carried out in a real scenario demonstrate that we can build a real-time people detection and tracking system using a low-cost drone (and in turn, a low-cost camera) establishing a trade-off between efficiency and effectiveness.

The rest of the paper is organized as follows; Sect. 2 will introduce the previous studies or research that have been done regarding the subject of drones and artificial vision algorithms. Next, in Sect. 3, we will present in detail the system that has been developed. Then, in Sect. 4, we will describe the experiments that have been carried out along with the results we have obtained to validate our system. Finally, in Sect. 5, we will highlight the main conclusions are drawn and some proposals for future work are depicted.

## 2 Related Work

People detection has been intensively studied so far in the computer vision community, primarily to improve the speed and accuracy of well-known detection algorithms. In 2004, Viola and Jones [13] proposed a technique of using Haar wavelets to detect faces in real-time. In 2005, Dalal and Triggs [2] proposed a technique in which people were obtained based on the histogram of gradients oriented as contrast obtained in the image. In 2012, R. Benenson [5] proposed a technique that consisted instead of evaluating a single model in several layers, evaluated a set of models using the approximation approach. In 2013, De Smedt [3] proposed a solution where person detection was optimized using parallel GPU and CPU as a single hybrid system. Naseer [8] describes a system

for tracking people using a quadrocopter. They used two cameras, one to determine the 3D position of the drone-based on markers on the ceiling, while the second camera was a depth camera, with which they detected the person in 3D. The image from the depth camera is warped based on the calculated 3D position. They make use of a wireless connected computer to perform demanding computational resources needed to steer the UAV, which also requires an online learning stage. In 2015, Floris De Smedt proposed a solution for tracking people with a UAV [4] using Aggregate Channel Feature (ACF) detector. And to compensate for frames without detections, they used a particle tracker based on color information.

In our system, the UAV is completely autonomous, using an external computer for image processing. This is because the drone that has been used in this work does not present room to embed any processing device. To perform the people detection and tracking, a cheap onboard RGB camera is used, using three different pre-trained classifiers. This also means that we do not require any previous learning stage, and therefore the need to capture and store a large and costly training dataset. Besides, our system contemplates the possibility of detection and monitoring of several people, using, in this case, a tracking algorithm based on centroids to follow the first person detected. With all this, our system is not restricted to simulated environments and can be deployed in real scenarios.

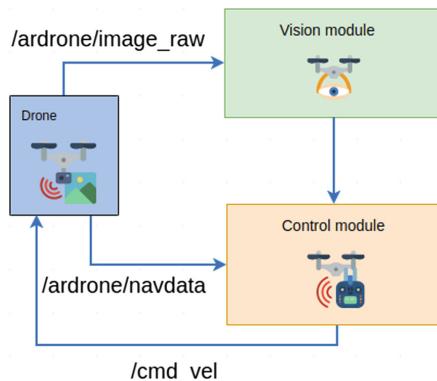
### 3 People Detection and Tracking System

The system has been designed to detect and track people through semantic segmentation algorithms using images obtained by a drone camera. Its functionality consists of obtaining the image of the drone's camera in real-time, processing it with one of the image classification algorithms above mentioned, namely SVM-HOG, MobileNet-SSD or YOLOv2Tiny. Then, depending on the detection and location of the person within the image, a set of movements are sent to the drone to center the person in the image, therefore, tracking that person as they move throughout the environment. In Fig. 1 the structure of the developed system depicts both vision and control.

In the following sections, the different components of our system are described in detail. First, the robotic development environment that we will use to interact with the drone is presented, as well as the movements that have to be sent to the drone to conveniently track a detected person. Next, the different modules that compose the system developed are also explained.

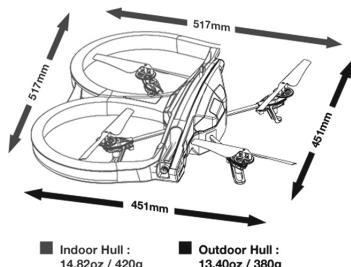
#### 3.1 Robotic Drone Platform

For the development of the system we are going to use the AR Drone 2.0 of the Parrot company (see Fig. 2), whose technical specifications are summarized in Table 1. One of the advantages of using this drone is that a ROS package



**Fig. 1.** Diagram of interaction between the drone and the vision and control modules

is provided. This package is called `ardrone_autonomy`<sup>1</sup>, and it is based on the official AR Drone SDK version 2.0.1, which has been developed at the Autonomy Lab of Simon Fraser University by Mani Monajjemi and other contributors. Among other functionalities, this package provides many useful ROS topics to easily interface and interact with the drone. In Table 2 the main ROS topics that have been used in the system developed are described.



**Fig. 2.** AR Drone 2.0 dimensions with indoor hull and with outdoor hull.

### 3.2 Vision Module

In the vision module, the image captured by the drone camera is obtained in real-time and converted from ROS Image type to OpenCV Image format. After which, image processing is required to detect the different elements present in the image, and finally classify them as a person or not. Then, depending on the location of the detected person in the image, the confidence value of the

<sup>1</sup> [http://wiki.ros.org/ardrone\\_autonomy](http://wiki.ros.org/ardrone_autonomy).

**Table 1.** AR Drone 2.0 sensors specifications

Sensors	Characteristics
Front camera	Images of $640 \times 360$ resolution and frame rate of 30 FPS
Gyroscope and magnetometer	Allows to calculate the orientation of the drone
Pressure and ultrasonic sensor	Allows to measure the altitude of the drone

**Table 2.** ROS topics to communicate with the AR Drone 2.0

Topic	Description
/ardrone/navdata	Provides information of the navigation and state of drone
/ardrone/image_raw	Provides the raw image of front camera of the drone
/cmd_vel	Allows to send movement commands to the drone

detection process (returned by the classification algorithm), and the state of the people detection process (see Fig. 3), a series of movements are sent to the control module to move the drone accordingly in order to track a potential person.

As already mentioned above we use and compare three different semantic segmentation models: SVM-HOG, MobileNet-SSD, and YOLOv2Tiny, to classify an image as containing or not a person, along with its confidence value (probability of being a person) and the corresponding bounding box containing that person. Let give some details about these classification algorithms:

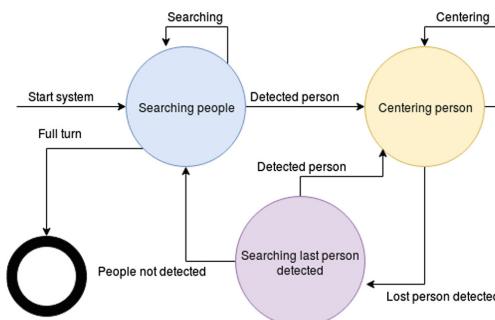
- SVM-HOG: This model, proposed by Dalal and Triggs [2] uses the histogram of oriented gradients (HOG) descriptors as features in a linear support vector machine (SVM) for object detection. In our case, the model has been previously trained using a large pedestrian dataset, so that it classifies an input image as person/no-person with its corresponding confidence value and bounding box.
- Google MobileNet-SSD [12] is a semantic segmentation model that has been pre-trained with the Caffe framework. This model returns a vector of 1000 categories assigning a confidence value to each of them. In our case, we will focus on the person category. The internal structure of MobileNet-SSD is a convolutional neuronal network.
- Concerning the YOLO classifier, we are going to specifically use your YOLOv2-Tiny pre-trained model [11]. This model is faster than the basic model but also yields worse detection results. This classifier, similarly as MobileNet-SSD, internally uses a convolutional neural network that segments an image into regions and classifies them into different semantic categories. Once again we focus only on the category person.

It is important to note that all the classifiers we have used return the detection of a person along with a confidence value. This value corresponds to the certainty that the algorithm grants to an element classified with a specific label that corresponds to that label, in our case a person. This will allow us, varying

the confidence value, to establish a compromise between efficiency and effectiveness for each algorithm. In the experimentation section, we will present results for different confidence values.

Regardless of the semantic segmentation algorithm used, the people detection process is implemented following the finite state machine presented in Fig. 3. The process starts in the “Searching people” state where the drone rotates to the left intending to detect a person. If after a complete turn no person has been detected, the process terminates and the drone is landed. However, if a person is detected, the system changes to the state “Centering person”, where depending on the location of the person in the image, the situations can arise. These situations along with the associated movements the drone must carry out to center the person in the image are shown in Table 3. If the current person is lost after performing a movement trying to center them on the image, the process is changed to the “Searching last person detected” state. In this state, the drone will be ordered to make a reverse movement to the one previously made to detect and follow again the last lost person. Once this reverse movement has been made, if the previously lost person is detected again, then process changes to the “Centering person” state. But, if the previously person is not detected again, the process changes to the “Searching person” state and the process starts a new detection.

In the case where several people are detected simultaneously, a centroid-based tracking algorithm has been implemented. In this case, the first person detected will be the candidate to be followed by the drone. If during the tracking process the current person is lost, for whatever reason, the algorithm chooses the next person detected as a candidate, and so on until there are no more candidates detected, in which case a new person detection process will be started.



**Fig. 3.** Diagram of states through which the detection of a person can happen.

### 3.3 Control Module

In the control module, we have all the information regarding the drone, such as the state (Laying, Landing, Flying), altitude, battery status, odometry and the

speed of their engines. Based on these values and the movements ordered by the vision module (see Table 3), a set of actions are sent to the drone so that it moves intending to center the person in the image, and as a consequence tracking the detected person. All these actions are carried out autonomously.

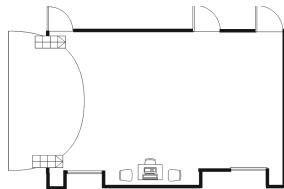
**Table 3.** Actions sent to the drone to center the detected person in the image

Position of the person	Action
Left/right	Rotate or move to the left/right
Above/below	Increase/decrease the altitude
Near/far	Go away/come closer

## 4 Experiments and Results

To validate our proposal two different experiments are carried out. First, a comparison of the three different semantic segmentation algorithms are performed to define the best classification model for our purpose, establishing a trade-off between efficiency and effectiveness. Then, a real use case is defined to verify the performance of the best algorithm according to the previous experiment in a real environment.

The environment we have used to evaluate our proposal consists of a large empty room of  $60 \times 30$  m so that the drone can move comfortably. The map of the environment is shown in Fig. 4.



**Fig. 4.** Experiment setup: 2D map of the environment.

### 4.1 Comparison of Semantic Segmentation Algorithms

The test set used to compare the three semantic segmentation algorithms consists of a video sequence composed of 1000 images, which has been recorded with the real drone commanded manually. We also tested the algorithms using two confidence values, 0.0 and 0.6.

Table 5 shows the classification results obtained with SVM-HOG, MobileNet-SSD and YOLOv2Tiny, using confidence values of 0.0 (top) and 0.6 (bottom). These results are presented as a confusion matrix (see Table 4) to facilitate the calculation of the metrics that will allow us to evaluate the goodness of these algorithms, which can be quantitatively compared using the following metrics:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \quad \text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}; \quad \text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

These metrics are shown in the Table 6, for the two confidence values considered, 0.0 and 0.6. The frames per second (FPS) that each algorithm can process are also shown. In Fig. 5, the corresponding Precision-Recall curve is plotted. Besides, a visual example of how each algorithm detects a person in a specific test frame is presented in Fig. 6.

**Table 4.** Confusion matrix example

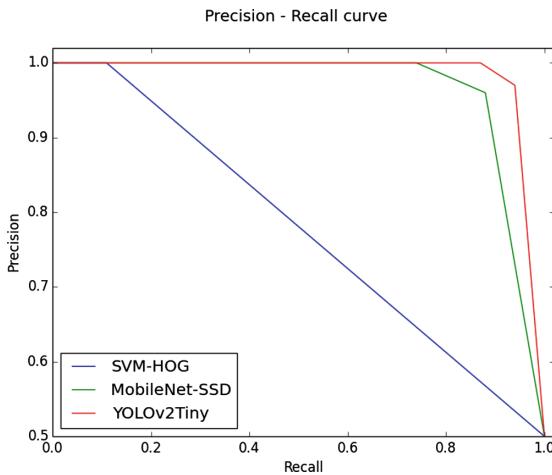
		Real	
		Person	No person
Prediction	Person	TP	FP
	No Person	FN	TN

**Table 5.** Classification results for confidence values of 0.0 (top) and 0.6 (bottom)

	SVM-HOG		MobileNet-SSD		YOLOv2Tiny	
	Person	No person	Person	No person	Person	No person
Person	76	0	616	23	657	20
No person	641	283	81	280	41	282
Person	61	0	527	2	624	3
No person	656	283	188	283	90	283

These metrics serve to determine how accurate the classifier is concerning the test set that has been used, in this case a video composed of 1000 frames. All these metrics except FPS, are measured in the range [0,1] with 1 being the ideal value and 0 the least. Concerning the FPR, the opposite occurs, with an ideal value of 0.

The higher the value of the Precision and the Recall, the higher the value of the F1-score and the Accuracy. If the F1-score and the Accuracy have a value close to 1, we can say that it is a good classifier.



**Fig. 5.** Precision-recall curve for the three algorithms and conf. values of 0.0 and 0.6



**Fig. 6.** Detections obtained with the same frame in the video test with the algorithms SVM-HOG (left), MobileNet-SSD (center) and YOLOv2Tiny (right)

**Table 6.** Metric figures for confidence values of 0.0 (top) and 0.6 (bottom)

Algorithm	Precision	Recall	F1 score	Accuracy	FPR	FPS
SVM-HOG	1.00	0.11	0.19	0.36	0.000	54.22
MobileNet-SSD	0.96	0.88	0.92	0.90	0.035	28.45
YOLOv2Tiny	0.97	0.94	0.96	0.94	0.029	20.55
SVM-HOG	1.00	0.09	0.16	0.34	0.000	54.22
MobileNet-SSD	1.00	0.74	0.85	0.81	0.003	28.45
YOLOv2Tiny	1.00	0.87	0.93	0.91	0.004	20.55

From these figures, we can observe that when increasing the level of confidence, we obtain a better Precision, that is, the higher true positives. Nevertheless, the Recall is lower, as the number of detections is reduced, as well as the possible false positives, causing the F1-score, Accuracy and FPR also lower.

The algorithm that obtained the highest F1-score and Accuracy is YOLOv2Tiny, followed by MobileNet-SSD and SVM-HOG, so we can conclude

that the classifier that has obtained the best results concerning our test set is YOLOv2Tiny.

Regarding the FPS, as expected the best value is obtained when using SVM-HOG because the drone camera can capture up 30 FPS. The results show that YOLOv2Tiny is not able to process all images in real-time provided with the camera because the algorithm only process 21 FPS approximately.

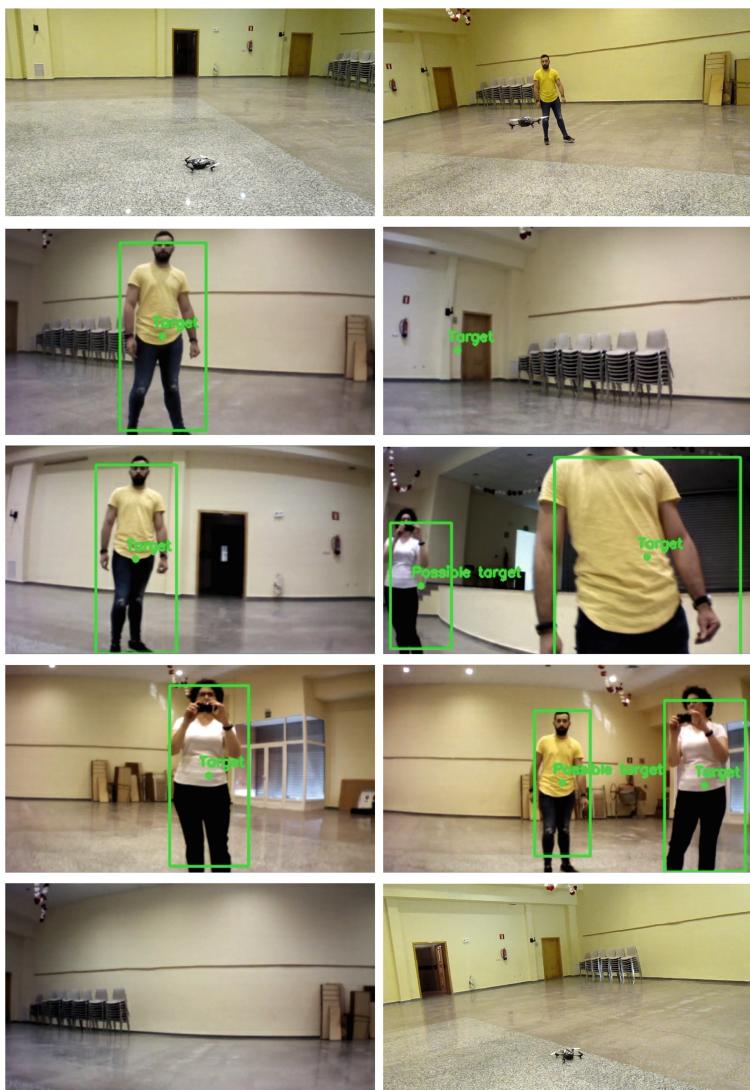
In light of the results obtained we have to establish a compromise between efficiency and effectiveness to choose the algorithm to be used in the next experiment to be carried out. Therefore we will use MobileNet-SSD with a confidence value of 0.5 because we can guarantee acceptable results without causing loss of information that would cause the algorithm not to work as good as expected.

## 4.2 A Real Use Case: Flying the Drone to Detect and Track People

According to the results of the previous experiment, and trying to establish a trade-off between efficiency and effectiveness, the best algorithm to be used in a real experiment is MobileNet-SSD. The use case consists in taking off the drone to test how the detection algorithm with a confidence value of 0.5 behaves in a real environment, with the aim of checking if our system is capable of detecting and tracking people in a real-time scenario.

In Fig. 7 the complete sequence of actions carried out with the drone are shown. These sequence of actions are:

1. The drone is placed in the middle of the environment and starts taking off.
2. Whilst the drone is flying, a person appears in the image, that will be detected and tracked by the drone.
3. The person moves around the environment to be tracked by the drone.
4. Suddenly the person disappears from the image with the aim the drone will rotate trying to find it, once it has lost (or failed to pick up) the person.
5. The drone must rotate until the person is detected again and continue tracking. The person moves left and right, forward and backward to check that the developed movement control is working properly.
6. A new person appears in the environment. The person being followed by the drone will approach the new person in the environment. The drone is able to detect both people. Once they are detected, the person being followed disappears from the image, and the drone starts to follow the new detected person.
7. The new detected person moves around the environment until she meet the person previously detected in the middle of the environment.
8. Both people will come together to be detected by the drone, and once both people are detected, the person being followed will disappear from the image, so that the drone begins to follow the initial person again.
9. Now both previously detected people disappear from the image so that the drone is not able to detect anyone and starts again to rotate with the aim of detecting a new person to follow.
10. The drone makes a complete turn rotating to the left, and after not detecting anybody, it lands finishing the flight test.



**Fig. 7.** Different situations through which the detection and tracking experiment occurs in a real flight.

As we can see in Fig. 7 all actions are carried out properly by the proposed system in a real scenario. In <https://youtu.be/q7Ms6-wNmS8> you can find a recorded video with the use case carried out.

## 5 Conclusions and Future Work

In this paper, we have proposed, implemented and tested a system based on semantic segmentation algorithms that can detect and track people using the information provided by an image captured by a drone. Besides, we have been able to implement a system that obtains adequate results for people detection and tracking using a low-cost device and pre-trained classification algorithms.

As a future piece of work, we want to extend this study to more sophisticated drones that provide higher quality images, so that we can check whether the semantic segmentation algorithms used are susceptible to be used in more complex systems. We also intend to test the proposed system in a more realistic real rescue situation. In the same way, we intend to extend the study to more and new semantic segmentation algorithms and more real drones.

**Acknowledgments.** This work has been partially sponsored by the Regional Council of Education, Culture and Sports of Castilla-La Mancha under grant number SBPLY/17/180501/000493, supported with Feder funds.

## References

1. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893, June 2005. Accessed 12 Jan 2019
3. De Smedt, F., Van Beeck, K., Tuytelaars, T., Goedemé, T.: Pedestrian detection at warp speed: exceeding 500 detections per second. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 622–628, June 2013
4. De Smedt, F., Hulens, D., Goedemé, T.: On-board real-time tracking of pedestrians on a UAV. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–8 (2015)
5. Gool, L.V., Mathias, M., Timofte, R., Benenson, R.: Pedestrian detection at 100 frames per second. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2903–2910. IEEE Computer Society, June 2012
6. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv preprint [arXiv:1408.5093](https://arxiv.org/abs/1408.5093) (2014)
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C.Y., Berg, A.C.: SSD: single shot multibox detector. In: ECCV (2016)
8. Naseer, T., Sturm, J., Cremers, D.: FollowMe: person following and gesture recognition with a quadrocopter. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 624–630 (2013)
9. 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, p. 5 (2009)
10. Rabbani, T., Van Den Heuvel, F., Vosselmann, G.: Segmentation of point clouds using smoothness constraint. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. **36**(5), 248–253 (2006)

11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
12. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
13. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vis.* **57**(2), 137–154 (2004)



# Design of a Robotic as a Service Platform to Perform Rehabilitation Therapies

Adrián Gallego, José Carlos Pulido<sup>(✉)</sup>, José Carlos González,  
and Fernando Fernández

Computer Science and Engineering Department, Universidad Carlos III de Madrid,  
Leganés, Spain

kugoad@gmail.com, {jcpulido, josgonza, ffernand}@inf.uc3m.es

**Abstract.** The NAOTherapist architecture is a robotic platform which aims to help patients with Cerebral Palsy and Obstetric Brachial Palsy to improve their condition, through the autonomous interaction provided by a social robot. This work is a Proof of Concept to evaluate if the NAOTherapist Architecture can be distributed into several computers, to get the benefits from Cloud Computing and Robotics as a Service. This work focuses on the enhancement of the NAOTherapist architecture, in which some of the components that compose the architecture have been integrated into a cloud server. This new architecture allows to protect the source code from being copied and to share the resources charged with the intelligence of the platform. The distributed architecture introduce new problems as the communication delay caused by the communication over the network. Then, this problems has been evaluated both technically and surveying the patients involved in the experiments performed. These evaluations have shown promising results, confirming that it is possible to deploy NAOTherapist as a Robotics as a Service.

**Keywords:** Robotic as a Service · Cloud Computing · Socially assistive robotics · Neurorehabilitation

## 1 Introduction

Robotics as a Service can be seen as the union of service-oriented architectures and Cloud Computing with software for robotic interfaces. During the last thirteen years, it has been a raise in service-oriented architectures and Cloud Computing models in which major computing companies started to support this new paradigm [1]. This paradigm offers new solutions, for the users, to problems that previously had a high computational cost and implied the need of expensive resources. In addition, Robotics as a Service architecture has been widely adopted due to the new commercial possibilities that brings to the Robotic companies, as it allows a major control over the software they deliver to their users.

According to the definition of Cloud Computing stated in NIST publication [5] definition, a Robot as a Service platform must have the following features:

1. *It must be an on demand service.* This must offer the resources that the users need and give every user what they ask for but neither more nor less. This can provide the users the capacity to perform expensive software operations without the need of them all to buy the whole resources.
2. *Broad network access.* The users shall be allowed to connect to the service from different locations and with heterogeneous devices.
3. *Resource pooling.* It consist on assigning resources to some users until they stop using them and then reassigning that resources to other users that need them. From the economical point of view this can help companies to buy less resources and just reuse them sharing this resources among their users, but appearing to their users as if they have always a whole resource assigned to them.
4. *Rapid elasticity.* It consist as showing up to the users as if they belong an infinite amount of resources. This allow the users to perform operations and computations that need an expensive hardware to be performed.
5. *Measured service.* Resource service can be monitored and controlled. This can help companies to monetize the application in function of what they used.

Robotics as a Service can also improve the development productivity, this is because in a Robot as a Service architecture the software is located on the cloud, on computers which have similar features, then it is not necessary to develop a different version for the different platforms and operating systems that the users may use, but just for the platform and operating system which runs on the cloud computer. This can help to focus the development on fixing problems and developing new features rather than on extending the software to other platforms.

Some cloud-based robotic approaches deal with different communication protocols to guarantee data protection and computing models related to the architecture distribution [3]. New trends of cloud robotics point to service and assistive robots. Paralleling the computation workload is important in order to solve bottlenecks when interacting in real-world environments [8]. In particular, currently there are few works for cloud-based rehabilitation robots. The main motivation of them is to manage therapy sessions remotely and to monitor the rehabilitation process through the Internet [4].

The NAOTherapist architecture is a Robotic Platform which aims to help pediatric patients with Cerebral Palsy and Obstetric Brachial Palsy to improve the state of their condition [2]. NAOTherapist uses a 3D sensor and a NAO robot to autonomously drive upper-limb rehabilitation sessions to motivate and monitor the evolution of the patients. Artificial intelligence is the core aspect to perform deliberative task reasoning and achieve enough autonomy to carry out these rehabilitation sessions. This process relies on an external computer to distribute the computational workload. Previous works confirmed that patients enjoyed the robotic therapeutic sessions more when the social robot was involved in the interaction [6,7]. The main reason for the NAOTherapist project to become a Robotics as a Service is to improve the distribution capabilities and to keep control over the features that the users are allowed to use on the platform.

To date, the NAOTherapist architecture run in a complex architecture that needs two computers, one running Linux and another running Windows, to run one instance of the platform, which is valid for just one patient at a time [2]. With the emergence of cloud computing technologies, the need of using these techniques to improve the resource usage and to protect the code arises. The objective of this work is to enhance the implementation of the NAOTherapist architecture as a Robotic as a Service (RaaS), by applying Cloud Computing techniques, and to evaluate this new architecture, in order to know if it still provide an enjoyable communication with the patients, despite of the delay caused by the communication over the network. The work presented in this manuscript is in an early stage of development, some aspects such data protection and a deep comparison with other potential computing models are part of a near future work.

## 2 Methods

This section explains the requirements for this new NAOTherapist architecture as well as the possible problems that will emerge from this architecture.

### 2.1 Design Requirements

This work is a proof of concept to evaluate if it is possible and feasible to implement the NAOTherapist Platform as a RaaS to get the benefits from Cloud Computing. Then, the new architecture has to fulfill the following requirements:

- *Requirement 1. It is necessary to protect as much code as possible from being copied.*
  - In order to maximize the amount of protected code it would be optimal to execute as many components as possible on the cloud. However there are some restrictions due to the hardware components. For instance, the controller components NuitrackComp and NaoComp can only operate in local as they are strongly related to the hardware. In addition some of the components shall modify the way in which the connections are established. This is because in the connection between two components it may be necessary to invert the role of each component in the communication, to provide firewall traversal or to discover the server IP.
- *Requirement 2. The resources charged with the intelligence of the platform must be shared among different client computers.*
  - In order to remove the need of having two separate computers, it is necessary to be able to locate the intelligence of the platform on a Cloud Computer which will be connected by the client computers, and whose resources will be shared among the different computers. This way the expensive computations will move to the Cloud Computer.

As it is a proof of concept, it is out of the scope of this work to be compared with other robotic platforms, as it is the case of ROS. Moreover, for this first iteration it will not be taken into account the security of the communications, however, as the platform deals with sensitive data, it is important for the next iterations of this work to ensure data protection and compliance with the laws in force.

## 2.2 Design Problems

This new RaaS architecture will introduce new problems due to the communication over the network. Then, the design has to minimize these problems as much as possible:

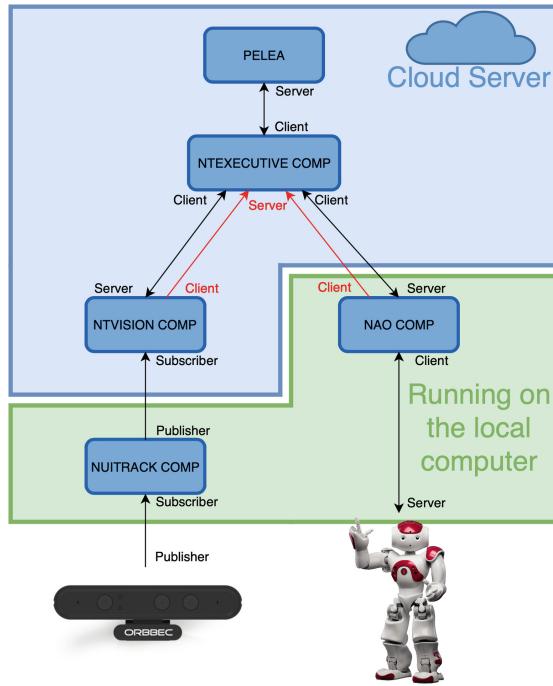
- *Problem 1. There can be a delay in the communication among components due to a limited bandwidth and the data transmission transfer time.*
  - In order to solve this problem it is necessary to analyze the delay due to transfer time and to evaluate if this delay is long enough to make the architecture useless for working in the cloud. It is also interesting to compare the time required to perform request to servers in different places in the world and to calculate where is the best place to locate this servers.
- *Problem 2. There can be limited resources in the cloud computer.*
  - As a result of having a finite amount of resources, it is necessary to analyze how many instances of NaoTherapist can run in one cloud computer, compare it with the resources available on that computer, and evaluate if the number of instances per computer worth's paying for that resources.

## 2.3 NAOTherapist as a Robot Service in the Cloud

The NAOTherapist Platform has been distributed across several computers, in order to get the benefits from Cloud Computing and the from a RaaS platform. With the objective of meeting the requirements stated in the Sect. 2.1, different implementation of the NAOTherapist architecture has been implemented and evaluated, however, in this work it is explained just the architecture which offers betters results, according to the already stated requirements.

The NAOTherapist architecture which offered the best results consisted on a computer running on the cloud, where the intelligence of the platform was located, and a computer running locally, where the components connected to the hardware, as the Kinect component and the NAO controller component, ran.

Figure 1 shows the different components of the implemented architecture to move the NAOTherapist architecture, previously running completely in just one local computer, to a distributed architecture, which allows several computer to connect to the intelligence of the platform, and to share these cloud resources among different computers.



**Fig. 1.** PELEA, NTEXecutiveComp and NTVisionComp running on DigitalOcean

The components running on the cloud are *PELEA*, the component in charge of performing the automated planning, deciding the next action to perform according to the current state of the robot; *NTEXecutiveComp*, the main controller of the application, this component is the responsible of distributing the tasks among the different components; and *NTVisionComp*, which is the component that translates from the current articulation coordinates received by *NuitrackComp* and compares this information with the expected articulation coordination received by *PELEA*. All these components belong to the original NAOTherapist architecture [2]. The changes on them allow their execution in local or in distributed mode depending only on their configuration.

The component running on the computer are the local computer are those which are closely related with the hardware and as consequence are constrained to run on the same computer where the software is located. These components are, *NuitrackComp*, which is the component responsible of reading images from the Kinect camera and of converting them into a set of articulation coordinates to send through the network, this conversion into coordinates is important because sending full images over the network can saturate the bandwidth, decreasing the speed of communication of the component over the network; and *NAOComp*, which is the component that receives the next action to be performed by the NAO and tell him to execute it.

In the original NAOTherapist architecture, the one that was not distributed over the network expected the components, *PELEA*, *NTVisionComp* and *NAO-Comp* to perform always as servers. Their function was just to process the information received by the other components and perform the appropriate action according to their state. That communication schema was based in two main premises:

1. All the components know the IP address and Port on which the server components they connect with are running.
2. There is no firewall filtering the communication among the different components.

This was not true anymore as a consequence of distributing the NAOTherapist architecture on several computers over the network. The first premise was null-able in the case that one of the components which perform as a server needed to be executed on a computer with dynamic IP address, as it is the case of the *NAOComp* component, which runs on a local computer with an unknown dynamic IP address. The second premise was null-able because even if it was possible for us to set the firewall on the cloud computers, as they belong to us, we were not able to decide whether or not there was a firewall running on the local computer, and though we could not set the firewall rules. Then, it was necessary to create a new communication schema not depending on these two premises.

The result of the new communication schema can be observed in Fig. 1. The arrows are directed from the component which initializes the connection, the one that acts as a client, to the component which listens for a connection, the one that acts as a server. While the black arrows represent the communications already present in the previous communication schema, the red arrows represent the new connections introduced in this new implementation of the NAOTherapist architecture. The main goal of these two new connections is to allow the main controller, *NTExecutiveComp*, to be notified, by *NTVisionComp* and *NAOComp*, with the information about the IP and Port on which they are running. This is important because the *NAOComp*, runs always on a computer with unknown IP address, then this notification is important. In the case of *NTVisionComp*, this new connection is only relevant in the case it is desired to distribute the components over different network computer with the objective of reduce the work performed by every cloud computer.

Moreover, the new red connections allow to encapsulate their correspondent black connections over the same session. This is important because it allows the connection to be constrained by the firewall rules on the components running on the cloud, the rules that can be managed by us. This is done because when the client of the red connection performs the first connection, it opens a session, and every request performed by the encapsulated black connection are treated by the client firewall as communication belonging to the previously established red connection. This way the firewall problem is surpassed.

### 3 Evaluation

The Distributed NAOTherapist has been evaluated in terms of performance, and compared with the previous architecture where all the components ran on a local computer, aiming to prove that it is reliable to distribute the different components in different computers on the cloud. It is important to notice that the communication times are expected to grow, as consequence of placing the computers far from the others, however, the objective of this evaluation to prove that new introduced delay is short enough as to be acceptable in a Social Interaction context.

The experiments have been carried out on two computers, a local and a cloud computer. The features of these computers are the following:

1. *Local Computer.* The local computer uses Ubuntu 16.04 as operating system. The computer hardware is composed by 8 GB RAM memory and 1 Intel Core 2 6300. This computer is physically located at Madrid (Spain).
2. *Cloud Computer.* The cloud computer uses Ubuntu 16.04. The computer hardware is composed by 2 GB RAM memory and 1 Intel(R) Xeon(R) CPU E5-2650L v3 @ 1.80 GHz. This computer is physically located at Frankfurt (Germany).

#### 3.1 Architecture Performance

This section compares the time that takes to communicate for the components *NAOComp* and *NTExecutiveComp*, and *NuitrackComp* and *NTVisionComp*. First, it is evaluated the communication time when the components run on the same computer, and later, when they communicate over the network. Only these two communication are evaluated, as these two are the only ones which are strictly constrained to perform over the network, in the new Distributed NAOThrerapist architecture.

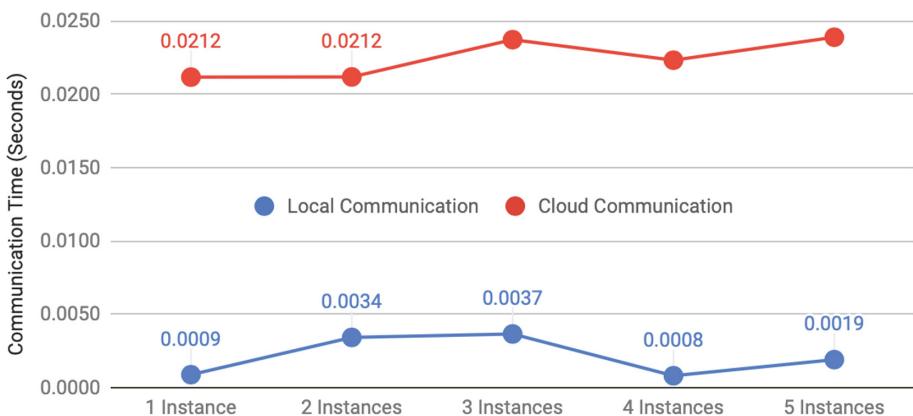
In Fig. 2, it is compared the *Local Communication* and *Cloud Communication* between *NTExecutiveComp* and *NAOComp*. It is observable that the communication time is about ten times higher, which means that it can delay a lot the whole process, however, 0.04 s is an acceptable communication time for the *NAOComp* component. This increment of time is explained because now the communications are constrained by the network *Round Trip Time*, which is much higher than the *NAOComp* processing time.

In Fig. 3, it is shown the comparison of the *Local Communication* and *Cloud Communication* between *NuitrackComp* and *NTVisionComp*. It can be observed that the *Cloud Communication* time is 0.02 s above the *Local Communication*. This extra delay is half the average *Round Trip Time*, that is, because the communication of these two components is unidirectional, and though only the time for sending a message from *NuitrackComp* to *NTVisionComp* is measured. In this case, the communication between the computer is higher than the *NTVisionComp* processing time, which implies that the communication time is constrained



**Fig. 2.** Communication Time Graph between NTExecutiveComp and NaoComp

by the *Round Trip Time*. It can be also observed, that the *Cloud Communication* is very unstable, it is probably caused because the communication across the network can be very variable.



**Fig. 3.** Communication Time Graph between NuitrackComp and NTVisionComp

After this evaluation it is possible to state that in the NAOTherapist architecture the communications are time constrained by the *Round Trip Time*.

### 3.2 Evaluation with Users

The new NAOTherapist architecture has been evaluated with a group of children in the Joan Miró school, in Leganes. The children, who represented the target patients of the platform, were coursing the first year of primary school and their

ages ranged between 6 and 8 years old. During the evaluation, the children were surveyed to know if the delays introduced due to the communication over the network were significant in the context of the social interaction between patients and platform.

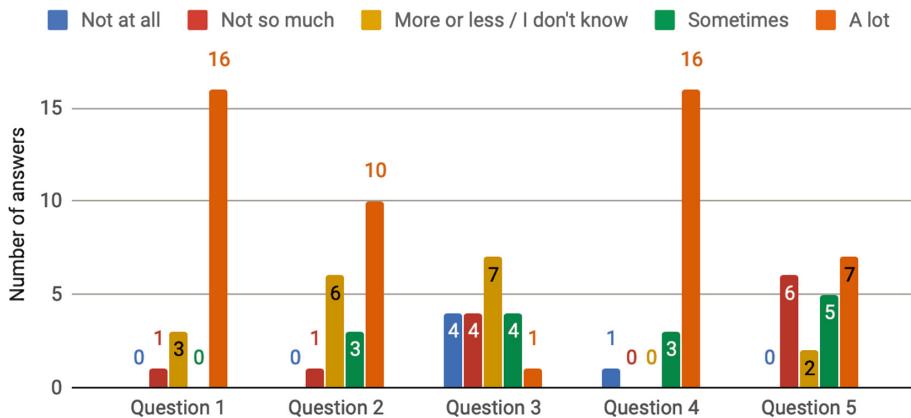
The evaluations, have been evaluated with groups of children, in which one of the children took the *Main* role in the communication with the system and the other just took a *Support* role. The *Main* or lead children were chosen randomly at the beginning of every session and the children were not aware about who was the leading one. In this evaluation, 50 children have been evaluated, 20 of them taking a *Main* role and 30 of them taking a *Support* role. In this work, only the results for the children with *Main* role are shown, as they are the only ones who relevant for this evaluation of the architecture.

After the evaluation the children were surveyed with five questions to know if they understood the exercises, if they enjoyed the session and whether the network delay was acceptable in the context of social iteration. The questions were the following ones:

1. *Question 1 (Did you understand how to work with the robot?).* This question aims to evaluate if the children understood what is the objective of the robot and what they had to do in the interaction with the robot. It is important because if they do not understand the exercises, then the interaction could be unpleasant. This question can also determine whether should be improved the introductory speech in which the Nao explains the exercises to perform. The possible answers to this question ordered from worst to best are: *Not at all, Not so much, More or less, Sometimes* and *A lot*.
2. *Question 2 (Did you manage to get the robot postures done?).* This question aims to evaluate whether the children had problems performing the exercises. It is important because if they have problems performing the different positions they may feel frustrated, which can lead to negative answers. The possible answers to this question ordered from worst to best are: *Not at all, Not so much, More or less, Sometimes* and *A lot*.
3. *Question 3 (Do you think the game was slow?).* This question aim to evaluate how they perceive the interaction with the robot either fast or slow. It is important because it helps to determine if the children notice the delay in the communication. The possible answers to this question ordered from worst to best are: *A lot, Sometimes, I don't know, Not so much* and *Not at all*.
4. *Question 4 (Did robot green eyes color help you to imitate the robot?).* The eyes of the robot turn slowly to green when the children are performing right and they turn white when the children are performing wrong. Then, this question tries to evaluate if the application refresh delay is imperceptible for the children despite of the communication delay. The possible answers to this question ordered from worst to best are: *Not at all, Not so much, I don't know, Sometimes* and *A lot*.
5. *Question 5 (Do you think the robot performed fast?).* This question is exactly the same that *Question 3* but negating its predicated. The objective of this question is determine if the children are providing coherent answers to the

questions, if the answers are coherent the answers should be an inversion of *Question 3*. The possible answers to this question ordered from worst to best are: *Not at all*, *Not so much*, *I don't know*, *Sometimes* and *A lot*.

Figure 4 shows that the Questions 1, 2 and 4 have a trend in which the best answer, *A lot*, is the most voted and the worst answer, *Not at all*, is almost not voted at all. It can be explained as a result of most of the children understanding the exercises to perform with the robot and the positive feedback given by the robot on every exercise. Moreover, the communication delay was low enough to allow them to benefit from an immediate feedback. Then these questions have the best answers.



**Fig. 4.** Survey answers frequency graph (main children)

In contrast, the results of Questions 3 and 5 (see Fig. 4) do not have a preferred value and they have almost inverted values. The fact that Question 3 is the inversion of Question 5 explains why the results of Question 5 are exactly the inverted values. In addition, a lot of children mark the game and the robot as slow, and also there is plenty of children who marked *I don't know* as answer because the NaoTherapist Game interaction is planned to be slow, as they are designed for therapies. However, this does not mean that the communication time between the different components is slow.

Moreover, in this evaluation, it has been also evaluated the communication delay between the components which communicate over the network from a technical point of view to know the communication delay in a close to real environment, and to compare it with the results obtained in the previous section.

Table 1 shows the *Average*, *Max*, *Min* and *Standard Deviation* of the communication time for NTVisionComp-NUITrackComp and NTExecutiveComp-NAOComp of all the request in every session performed during the school evaluation. On one hand, for the NTVisionComp-NUITrackComp communication,

comparing the obtained results with those of Sect. 3 it is possible to appreciate a rise on the average time from 0.021 s to 0.035 s due to the speed of the elementary school network, and a very high rise on the standard deviation from 0.0009 to 0.02 due to the stability of the network. On the other hand, for the NTExecutiveComp-NAOComp communication, it is possible to appreciate a rise on the average time from 0.041 s to 0.068 s due to the speed of the elementary school network, and a very high rise on the standard deviation from 0.001 to 0.029 due to the stability of the network.

**Table 1.** Communication Time Graph between distributed components during school evaluation

Global data	NTExecutiveComp-NAOComp	NTVisionComp-NUITrackComp
Average	0.068 s	0.035 s
Max.	0.227 s	0.181 s
Min.	0.039 s	0.020 s
Std. deviation	0.029 s	0.020 s

## 4 Conclusion

The evaluation results show that it is possible to implement the NaoTherapist architecture as a RaaS platform, as the communication delay are constrained by the network Round Trip Time and they are not too high in the context of social interaction. Moreover, the survey to the children shows that they were able to benefit from the feedback given by the robot, which is a feature that needs to perform fast in order to offer any benefit. The architecture shall be executed totally or partially on the cloud. Then in this iteration of the project it has been implemented and evaluated the possible distributions of the NaoTherapist architecture in order to know the advantages and disadvantages of each distribution.

The results obtained from the evaluation have been mostly promising. Then, we can state that it is possible to implement and effective NAOTherapist RaaS, then the following iterations will focus in improve the usage of this platform by improving the Configuration Interfaces and the management of the data of this Distributed architecture. It is also important to improve the security of the system, as it will deal with personal clinical data.

The different distributed architectures implemented for this iteration of the project have been: PELEA on the cloud, PELEA and NTExecutive on the cloud, and PELEA, NTVisionComp on the cloud, being the first one the less interesting for the project and the last the most interesting, in terms of code protection and having the control over most of the components.

The evaluation of the different distributed architectures showed that the communication across the network was constrained by the *Round Trip Time*. Then,

the time required for sending a message was more significant than the processing time, except for the communication with PELEA. However, the *Round Trip Time* is small enough to allow the NaoTherapist architecture to work as a *Real Time* application from the user experience point of view.

In conclusion, as every distributed architecture meets the minimum requirements to work properly, it has been proven that it is possible to run the NaoTherapist architecture as a Robot as a Service. In addition, the distributed architecture which runs more components on the cloud is preferred, so the *PELEA*, *NTExecutiveComp* and *NTVisionComp on the Cloud* architecture is the one which is chosen to implement the distributed NaoTherapist architecture.

Finally, it is also important to remark that it is a proof of concept and as future work it is important to implement the security in the platform connections, to comply with the data protection laws in force for sensitive data.

**Acknowledgements.** This work is partially funded by grant RTI2018-099522-B-C43 by MICINN and TIN2015-65686-C5-1-R by MINECO. We also want to thank the Joan Miró school of Leganés for their assistance with the evaluations, to the teachers and the management team for their support, and specially to all the children who kindly participated in the evaluation.

## References

1. Chen, Y., Du, Z., García-Acosta, M.: Robot as a service in cloud computing. In: 2010 Fifth IEEE International Symposium on Service Oriented System Engineering, pp. 151–158, June 2010
2. González, J.C., Pulido, J.C., Fernández, F.: A three-layer planning architecture for the autonomous control of rehabilitation therapies based on social robots. Cogn. Syst. Res. (CSR) **43**, 232–249 (2017)
3. Hu, G., Tay, W.P., Wen, Y.: Cloud robotics: architecture, challenges and applications. IEEE Netw. **26**(3), 21–28 (2012)
4. Li, H.J., Song, A.G.: Architectural design of a cloud robotic system for upper-limb rehabilitation with multimodal interaction. J. Comput. Sci. Technol. **32**(2), 258–268 (2017)
5. Mell, P., Grance, T.: The NIST definition of cloud computing. Technical report 800-145, National Institute of Standards and Technology (NIST), September 2011
6. Pulido, J.C., Suárez Mejías, C., Gonzalez Dorado, J.C., Duenas Ruiz, A., Ferrand Ferri, P., Martínez Sahuquillo, M.E., Ruiz De Vargas, C.E., Infante-Cossio, P., Parra Calderon, C.L., Fernandez, F.: A socially assistive robotic platform for upper-limb rehabilitation: a longitudinal study with pediatric patients. IEEE Robot. Autom. Mag. **26**(2), 1 (2019)
7. Pulido, J.C., González, J.C., Suárez-Mejías, C., Bandera, A., Bustos, P., Fernández, F.: Evaluating the child-robot interaction of the NAOTherapist platform in pediatric rehabilitation. Int. J. Soc. Robot. **9**, 1–16 (2017)
8. Saha, O., Dasgupta, P.: A comprehensive survey of recent trends in cloud robotics architectures and applications. Robotics **7**(3), 47 (2018)

# Author Index

## A

- Acevedo, Jose Joaquin, 37  
Acosta, José Ángel, 72  
Albuquerque, Daniel F., 367  
Alimardani, Maryam, 96  
Álvarez, David, 184  
Amaro, Pedro, 537  
Amat, Josep, 485  
Aranda, Joan, 497  
Armada, Manuel, 137, 148  
Azuar, David, 599

## B

- Bagheri, Azam, 237  
Bagheri, Elahe, 237  
Bandera, Antonio, 575  
Bandera, Juan P., 575  
Baptista, Rui, 392  
Barro, Senén, 632  
Becerra, Jose A., 250  
Bellas, Francisco, 250  
Biundini, Iago Z., 513  
Blanco, Andrea, 477  
Brandenburg, Samuel, 199  
Braun, João, 339  
Brito, Thadeu, 339  
Brull, Asier, 405  
Bustos, Pablo, 644

## C

- Cabanes, Itziar, 405  
Casado, Fernando E., 632  
Casals, Alicia, 485, 497  
Castanheira, José, 379

- Castellano-Quero, Manuel, 159  
Castillejo-Calle, Alejandro, 37  
Castro-Vargas, John-Alejandro, 551  
Catalán, José M., 477  
Cazorla, Miguel, 563, 599  
Cebollada, Sergio, 351  
Cifuentes-García, Cristian, 668  
Cintas, Ramón, 644  
Contreras, Aldo Francisco, 417  
Corrales, Juan Antonio, 171  
Costa, Paulo, 60, 339  
Cunha, Márcio, 60  
Curado, Francisco, 379

## D

- Dalmasso, Marc, 84  
Dalmedico, Nicolas, 327  
de Arruda, Lúcia Valéria Ramos, 327  
de Azevedo Fernandes, Lucas, 429  
de la Fuente, Eusebio, 453  
de Oliveira, André Schneider, 279, 303, 327  
Deibe, Alvaro, 315  
Dias, André, 122  
Díez, Jorge A., 477  
Dogru, Sedat, 392  
dos Reis, Alexandre Soares, 465  
Duarte, Abel J., 465  
Duarte, Fernando F., 263  
Duro, Richard J., 250

## E

- Escalona, Felix, 563, 599  
Esteban, Pablo G., 237

**F**

- Fernández, Fernando, 656, 681  
 Fernández, Roemi, 137, 148  
 Fernández-Madrigal, Juan-Antonio, 159  
 Ferre, Manuel, 417  
 Ferreira, João Filipe, 199  
 Ferreira, Paulo, 465  
 Flores, María, 351  
 Funk, Eugen, 291

**G**

- Gallego, Adrián, 681  
 Gallud, Guillermo, 599  
 Galvan, Marta, 15  
 Gandarias, Juan M., 441  
 García, Alba María, 656  
 García, Álvaro Muñoz, 453  
 García, Ángel, 417  
 García, José V., 477  
 García-Aracil, Nicolás M., 477  
 García-Cerezo, Alfonso, 109  
 García-Cerezo, Alfonso J., 159, 441  
 Garcia-Garcia, Alberto, 551  
 Garcia-Rodriguez, Jose, 551  
 García-Varea, Ismael, 610, 620, 668  
 Garrell, Anaís, 84  
 Gaspar, José António, 3  
 Gielen, Elien, 465  
 Gil, Óscar, 213  
 Gines, Jonathan, 587  
 Gómez Eguílez, Augusto, 48  
 Gómez-de-Gabriel, Jesús M., 441  
 Gomez-Donoso, Francisco, 563, 599  
 Gonçalves, Edgar, 379  
 Gonçalves, Edgar S., 367  
 González, José Carlos, 681  
 González-de-Santos, Pablo, 137, 148  
 González-Medina, Daniel, 668  
 Gorrotxategi, Aitor, 405  
 Gragera, Alba, 656  
 Guedes, Pedro, 465  
 Guerrero-Higueras, Ángel Manuel, 587

**H**

- Hernansanz, Albert, 485  
 Hidalgo-Paniagua, Alejandro, 575  
 Honorio, Leonardo M., 513

**I**

- Iglesias, Roberto, 632

**J**

- Jiménez, Pablo, 84  
 Justo, Jorge, 465

**K**

- Kalempa, Vivian Cremer, 279

**L**

- Lambrecht, Jens, 291  
 Lau, Nuno, 263, 537  
 Leitão, Paulo, 429  
 Lema, Dylan, 632  
 Lima, José, 60, 339  
 Lima, José Luis, 429  
 Lledó, Luis D., 477  
 Lobato, Emilio, 477  
 López, Miguel A., 524  
 Losada-Pita, Javier, 315  
 Loureiro, Gabriel, 122

**M**

- Machado, Pedro, 199  
 Malheiro, Benedita, 465  
 Mandow, Anthony, 109  
 Manso, Luís J., 644  
 Marcato, Andre L. M., 513  
 Marinero, Juan Carlos Fraile, 453  
 Marins, Guilherme M., 513  
 Marques, Lino, 392  
 Martínez del Horno, Miguel, 610  
 Martínez, María Alcázar, 109  
 Martínez-de Dios, José Ramiro, 48  
 Martínez-Gómez, Jesus, 620  
 Martinez-Gonzalez, Pablo, 551  
 Martín-Rico, Francisco, 587  
 Martins, Alfredo, 122  
 Matellán-Olivera, Vicente, 587  
 Mateo, Carlos M., 171  
 Maza, Ivan, 37  
 McGinnity, T. M., 199  
 Melo, Aurelio G., 513  
 Mezouar, Youcef, 171  
 Millan-Alcaide, Andrés, 575  
 Millan-Romera, Jose A., 37  
 Monteiro, Nuno Barroso, 3  
 Moreira, Thiago Fernandes Moya, 60  
 Moreno, Luis, 184  
 Muñoz, Victor, 453  
 Muñoz-Ramírez, Antonio J., 441

**N**

- Nakano, Alberto, 339  
 Nakano, Alberto Yoshiro, 429  
 Navas, Eduardo, 137, 148  
 Nekoo, Saeed Rafee, 72  
 Neves Júnior, Flávio, 327  
 Nuevo-Gallardo, Cristina, 524  
 Núñez, Pedro, 644

**O**

- Ollero, Aníbal, 37, 48, 72  
Oprea, Sergiu, 551  
Orjales, Felix, 315  
Orozco-Barbosa, Luis, 610

**P**

- Palar, Piatan Sfair, 327  
Pasternak, Marcel, 465  
Pastor, Francisco, 441  
Payá, Luis, 351  
Paz-Lopez, Alejandro, 315  
Pedrosa, Eurico, 379  
Pedrosa, Eurico F., 367  
Pereira, Ana, 339  
Pereira, Artur, 263  
Perez-Leon, Hector, 37  
Piardi, Luis, 303, 339  
Pinto, Milena F., 513  
Pont, David, 417  
Prieto-Arranz, Javier, 524  
Pulido, José Carlos, 681

**Q**

- Qurashi, Sonia, 96

**R**

- Ramos, Julio Endress, 327  
Ravindranath, Vignesh Padubidri, 28  
Regueiro, Carlos V., 632  
Reinoso, Óscar, 351  
Reis, Luís P., 263  
Reis, Luís Paulo, 537  
Repiso, Ely, 15  
Ribeiro, Cristina, 465  
Rivas, Irene, 453  
Roa, Máximo A., 184  
Rodríguez-Gómez, Juan Pablo, 48  
Rodriguez-Larrad, Ana, 405  
Rodríguez-Lera, Francisco Javier, 587  
Roesler, Oliver, 28, 225  
Rohrich, Ronnier Frates, 303  
Román, Vicente, 351  
Romero, Alejandro, 250

- Romero-González, Cristina, 610, 620  
Ruiz, Ricardo, 417

**S**

- Sabater, Jose María, 453  
Sáez, Francisco Javier, 417  
Samper, José Luis, 417  
Sánchez, Miguel Ángel, 417  
Sanfeliu, Alberto, 15, 84, 213  
Santos, Higor Barbosa, 327  
Sayols, Narcís, 485  
Schneider, Tobias, 465  
Sepúlveda, Delia, 137, 148  
Shinde, Pranjali, 199  
Sierra, Juan Felipe García, 587  
Silva, Manuel F., 465  
Silva, Tiago, 537  
Simões, David, 537  
Slongo, Juliano Scholz, 327  
Soares, Luís, 122  
Sooäär, Vaido, 465

**T**

- Teixeira, Francisco C., 367  
Teixeira, Marco Antônio Simões, 279, 303, 327  
Tejado, Inés, 524  
Terres, Vinicius Vargas, 327  
Tomé, Ana, 379  
Torres-Camara, Jose Miguel, 563  
Toscano-Moreno, Manuel, 109  
Traver, José Emilio, 524  
Turiel, Javier P., 453

**V**

- Vanderborgth, Bram, 237  
Vega, Araceli, 644  
Vinagre, Blas M., 524  
Vinagre, Manuel, 497

**W**

- Wopereis, Ko, 465

**Z**

- Zubizarreta, Asier, 405