HABILITATION À DIRIGER DES RECHERCHES

présentée à

L'UNIVERSITÉ DE RENNES 1

SPÉCIALITÉ: (1) AUTOMATIQUE (2) TRAITEMENT DU SIGNAL

par

Paolo Robuffo Giordano

Contributions to shared control and coordination of single and multiple robots

---

Date de soutenance prevue le 15/01/2016

Jury

Dr. Michèle Basseville	Directeur de recherche CNRS, IRISA, Rennes, France	Président
Dr. Simon Lacroix	Directeur de recherche CNRS, LAAS, Toulouse, France	Rapporteur
Dr. Isabelle Fantoni	Directeur de recherche CNRS, Heudiasyc, Compiègne, France	Rapporteur
Prof. Volkan Isler	Associate Professor, University of Minnesota, USA	Rapporteur
Dr. François Chaumette	Directeur de recherche INRIA, Rennes, France	Examinateur
Prof. Alessandro De Luca	Full Professor, Sapienza University of Rome, Italy	Examinateur
Prof. Domenico Prattichizzo	Full Professor, University of Siena, Italy	Examinateur

Contents

Part I

Extended Curriculum Vitae

Chapter I

Extended Curriculum Vitae

I.1 Personal data

Name: Paolo
Surname: Robuffo Giordano
Date and Place of Birth: January, 30th, 1977, Rome, Italy
Citizenship: Italian

Campus Universitaire de Beaulieu
35042 Rennes cedex, France
Tel: +33 (0)2 99842545
Fax: +33 (0)2 99847171
Email: prg@irisa.fr
Website: www.irisa.fr/lagadic/team/Paolo.Robuffo_Giordano.html

Current position: CNRS Researcher (CR1) in the Lagadic team at Irisa and Inria Rennes Bretagne Atlantique, Rennes, France since December 2012

I.2 Education

- 2016: Habilitation à Diriger des Recheches (HDR), Université de Rennes 1, France
  - Thesis title: "Contributions to shared control and coordination of single and multiple robots", submitted on September 12th, 2015, **to be defended on January 15th, 2016**
- 2004-2008: Ph.D. in Systems Engineering (Dottorato di ricerca in Ingegneria dei Sistemi), University of Rome "La Sapienza", Rome, Italy
  - Supervisor: Prof. Alessandro De Luca
  - Thesis title: "Visual Estimation and Control of Robot Manipulating Systems", defended on April 2nd, 2008

- 1995-2001: M.Sc. in Computer Science Engineering (Laurea in Ingegneria Informatica) University of Rome "La Sapienza", Rome, Italy
  - Supervisor: Prof. Alessandro De Luca
  - Thesis title: "Posizionamento e navigazione tra ostacoli di un robot mobile via feedback visuale", defended on November 7th, 2001

I.3 Professional Experience

Academic:

- Dec 2012 - today: **CNRS Researcher** (CR1) in the Lagadic team at Irisa and Inria Rennes Bretagne Atlantique, Rennes, France
- Oct 2008-Dec 2012: **Project Leader** (**Senior Research Scientist** – Head of the Human-Robot Interaction group) at the Max Planck Institute for Biological Cybernetics, Bülthoff Department, Tübingen, Germany
- Nov 2007-Oct 2008: **Research Scientist** at the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany
- Jan 2007-Jul 2007: Visiting PhD Student at the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany
- Nov 2004-Oct 2007: PhD Student Scholarship in Systems Engineering, Dipartimento di Informatica e Sistemistica, University of Rome "La Sapienza", Italy (advisor: Prof. A. De Luca)
- Jan 2002-Jun 2002: Research assistant, University of Rome "La Sapienza", Italy

Industrial:

- Sep 2003-Oct 2004: Software engineer. Datamat Spa, Rome, Italy
- Jul 2002-Sep 2003: Control engineer. ELV Spa, Colleferro, Italy

I.4 Supervision and Mentoring

Postdocs

- 2015-2016: Dr. Nicolò Pedemonte (Ph.D. in 2014 from Université de Laval, Quebec City, Canada)
- 2015-2016: Dr. Riccardo Spica (Ph.D. in 2015 from University of Rennes 1, France). Common publications: [J23, J17, C61, C60, C55, C54, C53, C52, C51, C39, W18, W17]
- 2010-2012: Dr. Antonio Franchi (Ph.D. in 2010 from University of Rome "La Sapienza", Rome, Italy). Common publications: [J10, J11, J12, J14, J15, J16, J18, J21, J24, J25, J27, C64, C63, C57, C56, C52, C50, C48, C45, C44, C43, C41, C39, C38, C37, C36, C33, C32, C31, C30, C29, C28, C27, C25, C24]

- 2010-2012: Dr. Hyoung Il Son (Ph.D. in 2010 from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea). Common publications: [J16, J14, J10, C38, C29, C27, C25, C24]
- 2010-2012: Dr. Suguna Subramanian (Ph.D. in 2010 from University of Tübingen, Germany)

Engineers

- 2015-2016: Thomas Bellavoir advised at 90% with Fabien Spindler, INRIA/IRISA, Rennes, France

Ongoing Ph.D. Students

- 2015-2018: Firas Abj Farraj, advised at 100%, topic: "Shared Control Architectures for Visual Servoing Tasks"
- 2015-2018: Muhammad Usman, advised at 100%, topic: "Robust Vision-Based Navigation for Quadrotor UAVs"
- 2015-2018: Bryan Penin, advised at 60% with Dr. F. Chaumette, INRIA/IRISA, Rennes, France, topic: "Model predictive visual servoing for UAVs"
- 2014-2017: Fabrizio Schiano, advised at 60% with Dr. F. Chaumette, INRIA/IRISA, Rennes, France, topic: "Collective control with onboard sensors for multiple quadrotor UAVs". Common publications: [C64]
- 2013-2016: Suman Raj Bista, advised at 60% with Dr. F. Chaumette, INRIA/IRISA, Rennes, France, topic: "Visual navigation of a humanoid robot". Common publications: [J26, C65]
- 2013-2016: Nicolas Cazy, advised at 60% with Dr. F. Chaumette, INRIA/IRISA, Rennes, France, topic: "Model predictive visual servoing of a humanoid robot". Common publications: [C58, C59]

Past Ph.D. Students

- 2012-2015: Riccardo Spica, advised at 60% with Dr. F. Chaumette, INRIA/IRISA, Rennes, France. Thesis title "Contributions to Active Visual Estimation and Control of Robotic Systems", University of Rennes 1, France, defended on **December 2015**. Common publications: [J23, J17, C61, C60, C55, C54, C53, C52, C51, C39, W18, W17]
- 2010-2014 Markus Ryll, advised at 90% together with Prof. Dr.-Ing. F. Allgöwer, University of Stuttgart, Germany.
  Thesis title "A Novel Overactuated Quadrotor UAV", University of Stuttgart, Germany, defended in **February 2015**. Common publications: [J19, J12, J11, C52, C46, C35, W16]
- 2010-2014: Volker Grabe advised at 90% together with Prof. Dr. Andreas Schilling (University of Tübingen) and Prof. Dr. Heinrich H. Bülthoff (Max Planck Institute for Biological Cybernetics). Thesis title "Towards Robust Visual-Controlled Flight of Single and Multiple UAVs in GPS-Denied Indoor Environments", University of Tübingen, Germany, defended on **March 2014**. Common publications: [J20, J11, C49, C48, C40, C34, C23]

- 2010-2014 Carlo Masone, advised at 50% together with Dr. A. Franchi (Max Planck Institute for Biological Cybernetics) and Prof. Dr.-Ing. F. Allgöwer, University of Stuttgart, Germany.
  Thesis title "Planning and Control for Robotic Tasks with a Human-in-the-Loop", University of Stuttgart, Germany, defended on **July 2014**. Common publications: [J25, J11, C56, C42, C41, C31, C26, C19, C18]

- 2009-2013: Florian Soyka, advised at 60% with Prof. Dr. H. H. Bülthoff (Max Planck Institute for Biological Cybernetics) and Prof. Dr. Uwe Ilg (University of Tübingen).
  Thesis title: "A Cybernetic Approach to Self-Motion Perception", University of Tübingen, Germany, defended on **January 2013**. Common publications: [J8, J6, C15, W10, W9]

M.Sc. Students

- 2012: Thomas Nestmeyer, with Dr. A. Franchi (Max Planck Institute for Biological Cybernetics) and Prof. Dr. W. Rosenstiel, University of Tübingen, Germany. Common publications: [J27]

- 2012: Riccardo Spica, with Dr. A. Franchi (Max Planck Institute for Biological Cybernetics) and Prof. G. Oriolo, University of Rome "La Sapienza", Italy

- 2012: Martin Riedel, with Dr. A. Franchi (Max Planck Institute for Biological Cybernetics) and Prof. Dr. W. Rosenstiel, University of Tübingen, Germany. Common publications: [C48, C38]

- 2011: Maria Liebsch, with Prof. Dr. W. Rosenstiel, University of Tübingen, Germany

- 2011: Johannes Lächele, with Prof. Dr. W. Rosenstiel, University of Tübingen, Germany. Common publications: [C43, C21]

- 2010 : Hendrik Deusch, with Prof. A. Kistner, University of Stuttgart, Germany. Common publications: [C21]

- 2010: Volker Grabe, with Prof. Dr. W. Rosenstiel, University of Tübingen, Germany

- 2010: Carlo Masone, with Prof. G. Oriolo, University of Rome "La Sapienza", Italy

- 2007: Francesca Pettine, with Prof. G. Oriolo, University of Rome "La Sapienza", Italy

- 2007: Massimo Ferri, with Prof. G. Oriolo, University of Rome "La Sapienza", Italy. Common publications: [C10]

- 2006: Roberto Onesti, with Prof. A. De Luca, University of Rome "La Sapienza", Italy

I.5 Teaching

- May 2015: Lecturer of the module: "Analysis and Control of Multi-Robot Systems" in "Elective in Robotics", Department of Computer and System Sciences, University of Rome "La Sapienza", http://www.dis.uniroma1.it/~venditt/eir/

- May 2014: Lecturer of the module: "Analysis and Control of Multi-Robot Systems" in "Elective in Robotics", Department of Computer and System Sciences, University of Rome "La Sapienza", http://www.dis.uniroma1.it/~venditt/eir/

- May 2013: Lecturer of the module: "Analysis and Control of Multi-Robot Systems" in the course "Elective in Robotics", Department of Computer and System Sciences, University of Rome "La Sapienza", http://www.dis.uniroma1.it/~venditt/eir/
- March 2012 - July 2012: Lecturer of the Class "Analysis and Control of Multi-agent Systems" together with Dr. D. Zelazo at the Institute for Systems Theory and Automatic Control, University of Stuttgart
  http://www.ist.uni-stuttgart.de/education/courses/Multi-Agent2012/
- May 2012: Lecturer of the module: "Analysis and Control of Multi-Robot Systems" in "Elective in Robotics", Department of Computer and System Sciences, University of Rome "La Sapienza", http://www.dis.uniroma1.it/~venditt/eir/
- March 2011 - July 2011: Lecturer of the Class "Analysis and Control of Multi-agent Systems" together with Dr. D. Zelazo at the Institute for Systems Theory and Automatic Control, University of Stuttgart
  http://www.ist.uni-stuttgart.de/education/courses/Multi-Agent2011/

I.6 International Collaborations

- Prof. F. Allgöwer, Full Professor at the University of Stuttgart, Germany
  http://www.ist.uni-stuttgart.de/allgower/allgower.shtml.
  Topic: multi-agent distributed estimation and control.
  Results: [C37] and the courses held during March 2011-July 2011 and March 2012-July 2012 at the University of Stuttgart
- Prof. D. Zelazo, Assistant Professor at Technion, Israel http://zelazo.net.technion.ac.il/.
  Topic: multi-agent distributed estimation and control.
  Results: [J18, C64, C63, C57, C37], and the organization of the Invited Session "Rigidity Theory for Problems in Multi-Agent Coordination" at the 54-th IEEE Conf. on Decision and Control (CDC 2015)
- Prof. D.J. Lee, Associate Professor at Seoul National University, Korea
  http://inrol.snu.ac.kr/.
  Topic: bilateral teleoperation of multiple robots.
  Results: [J16, C29, C27, C24]
- Prof. D. Scaramuzza, Assistant Professor at the University of Zürich, Switzerland
  http://rpg.ifi.uzh.ch/people_scaramuzza.html.
  Topic: fully-onboard visual-based navigation of quadrotor UAVs.
  Results: [J20]
- Prof. S. Haddadin, Full Professor at the Leibniz Universität Hannover, Germany
  http://www.irt.uni-hannover.de/haddadin.html and A. Peer, Full Professor at the University of West England, Bristol, UK http://www.brl.ac.uk/brlpeople.aspx.
  Co-organization of the Special Issue on "Autonomous Physical Human-Robot Interaction" in the *International Journal of Robotics Research*, 2011.
  Results: [J9]

- Prof. C. Secchi, Assistant Professor at the University of Modena and Reggio Emilia, Italy
  http://personale.unimore.it/rubrica/dettaglio/csecchi.
  Topic: bilateral teleoperation of multiple robots and passivity-based control.
  Results: [J15, J12, J10, C45, C36, C33, C30, C28, C25] and the 2012 ICRA Workshop "Haptic Teleoperation of Mobile Robots: Theory, Applications and Perspectives"

- Prof. G. Antonelli, Associate Professor at the University of Cassino, Italy
  http://www.docente.unicas.it/gianluca_antonelli.
  Topic: adaptive control for flying robots.
  Results: [J21, C50, C47]

- Prof. R. Mahony, Full Professor at the Australian National University, Australia
  http://people.cecs.anu.edu.au/user/3843.
  Co-organization of the Symposium on Flying Robots at The International Symposium on Robotics Research (*ISRR 2015*)

I.7 Awards

- 2015: Best Associate Editor for ICRA 2015
- 2011: JVRC 2011 WOW-Factor competition, "A Novel Framework for Closed-Loop Robotic Motion Simulation", 2011 Joint Virtual Reality Conference
- 2009 : Best Video Award for the video submission "Rollin Justin - Mobile Platform with Variable Base", International Conference on Robotics and Automation (ICRA) 2009
- 2009: Best italian Ph.D. Thesis on Robotics and Control in 2008

I.8 Research Projects and External Funding

- 2015–2018: H2020 project "RoMaNS – Robotic Manipulation for Nuclear Sort and Segregation". Role: principal investigator at CNRS. Funding: 806k€
- 2014–2017: ANR (French National Research Agency) Young Researcher Project "SenseFly – Sensor-Based Flying Multi-Robot System". Role: principal investigator. Funding: 365k€
- 2014–2017: Rennes Metropole Young Researcher Grant. Role: principal investigator. Funding: 40k€
- 2011–2014: FP7 EU STREP myCopter – Enabling Technologies for Personal Aerial Vehicles (http://www.mycopter.eu). Role: proposal writing, responsible for the development of control and motion cueing algorithms for simulating motion of a vehicle on a robotic motion simulator (the MPI CyberMotion simulator)
- 2009–2012: FP7 EU STREP Supra – Simulation of Upset Recovery in Aviation (http://www.supra.aero). Role: use of system theory and state estimation tools for modeling the human's self-motion perception

- 2006–2008: FP6 EU STREP Phriends – Physical Human-Robot Interaction: Dependability and Safety (http://www.phriends.eu). Role: exploitation of redundancy in collision detection and reaction strategies on the 7R DLR/KUKA arm
- 2005–2008: FP6 EU STREP CyberWalk – Enabling Unconstrained Omni-Directional Walking in Virtual Worlds (http://www.cyberwalk-project.org). Role: responsible for the motion control and overall control architecture of the CyberWalk platform

I.9 Patents

- 2014 : H. H. Bülthoff and P. Robuffo Giordano, "Teleoperation Method And Human Robot Interface For Remote Control of a Machine by a Human Operator", US 8634969

I.10 Workshop Organization

- 2015: co-organizer of the "Eighth Workshop on Human-Friendly Robotics", Technische Universität München (TUM), Germany. http://www.hri.ei.tum.de/hfr2015/home
- 2015: Invited Session "Rigidity Theory for Problems in Multi-Agent Coordination" at the 54-th IEEE Conf. on Decision and Control (CDC 2015), www.cdc2015.ctrl.titech.ac.jp, together by D. Zelazo and A. Franchi
- 2015: Symposium on Flying Robots at The International Symposium on Robotics Research (*ISRR 2015*), www.isrr-2015.org, together with Prof. R. Mahony, Australia National University
- 2014: co-organizer of the "Seventh Workshop on Human-Friendly Robotics", Institute of Scuola Superiore Sant'anna, Pisa, Italy. http://sssa.bioroboticsinstitute.it/workshops/hr2014
- 2014: co-organizer of the Workshop "On the centrality of decentralization in multi-robot systems: holy grail or false idol? " at the *2014 International Conference on Robotics and Automation*, Hong Kong, China. http://homepages.laas.fr/afranchi/events/icra2014mrs/w/Home.html
- 2013: co-organizer of the "Sixth Workshop on Human-Friendly Robotics", University of Rome "La Sapienza", Italy. http://hfr13.wordpress.com/
- 2013: co-organizer of the Workshop "Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration" at the *2013 International Conference on Robotics and Automation* (ICRA 2013), Karlsruhe, Germany. http://icra2013mrs.tuebingen.mpg.de/index.php?id=38
- 2012: co-organizer of the "Fifth Workshop on Human-Friendly Robotics", University of Brussels, Belgium. http://hfr2012.wordpress.com/
- 2012 : co-organizer of the Workshop "Haptic Teleoperation of Mobile Robots: Theory, Applications and Perspectives" at the *2012 International Conference on Robotics and Automation* (ICRA 2012), St. Paul, USA. http://www.arscontrol.org/events/61-icra2012htmr

- 2011: co-organizer of the "Fourth Workshop on Human-Friendly Robotics", University of Twente, Netherlands. http://hfr2011.wordpress.com/
- 2011: co-organizer of the Workshop "Towards Autonomous Physical Human-Robot Interaction" at the *2011 International Conference on Robotics and Automation* (ICRA 2011), Shanghai, China. http://www.robotic.dlr.de/288
- 2010: local organizer of the "Third Workshop on Human-Friendly Robotics", Max Planck Institute for Biological Cybernetics, Germany. http://hfr2010.wordpress.com/
- 2009: "Second Workshop on Human-Friendly Robotics", Italian Institute of Technology, Italy. http://hfr2009.wordpress.com/

I.11 Professional Service

- 2015: Workshop and Tutorial Session Chair for IROS 2016, Daejeon, Korea www.iros2016.org
- 2014-2015: Reviewer for the EU FP7 IP Project SHERPA http://www.sherpa-project.eu/
- 2014: Reviewer for the SIR 2014 program (Scientific Independence of young Researchers), Italian Ministry for Education, University and Research
- 2013: Reviewer for evaluating research proposals of the Swiss National Science Foundation (SNSF)
- 2012: Reviewer for evaluating research proposals of the Swiss National Science Foundation (SNSF)
- 2011: External reader for the doctoral thesis of M. Di Rocco, "A multi-robot system: from design to intelligent cooperation", advisor Prof. S. Panzieri, University of Roma Tre, Italy

I.12 Editorial Service and Committees

Editor

- Robotics: Science and Systems, Area Chair (RSS 2014, RSS 2015) http://www.roboticsconference.org/
- Guest Editor for the Special Issue on "Autonomous Physical Human-Robot Interaction" in the *International Journal of Robotics Research*, 2011, with Dr. Sami Haddadin (Institute of Robotics and Mechatronics, German Aerospace Center, Germany) and Dr. Angelika Peer (Institute of Automatic Control Engineering, Technische Universität München, Germany) http://ijr.sagepub.com/content/31/13.toc?etoc

Associate Editor

- IEEE Transactions on Robotics (2012–2016) http://www.ieee-ras.org/publications/t-ro

- IEEE International Conference on Robotics and Automation (ICRA 2012, ICRA 2013, ICRA 2014, ICRA 2015, ICRA 2016)
- IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)

Program Committee

- International Symposium on Distributed Autonomous Robotic Systems (DARS 2014)
- Robotics: Science and Systems (RSS 2010, RSS 2011, RSS 2013)
- International Joint Conference on Artificial Intelligence (IJCAI 2013)

Reviewer

- Journals: IEEE Transactions on Robotics, The International Journal of Robotics Research, IEEE Robotics and Automation Magazine, IEEE Transactions on Automatic Control, IEEE/ASME Transactions on Mechatronics, Mechatronics, IEEE Transactions on Control Systems Technology, Advanced Robotics, Autonomous Robots, International Journal of Robotics and Automation, Journal of Real-Time Image Processing
- Conferences: IEEE International Conference on Robotics and Automation, IEEE/RSJ International Conference on Intelligent Robots and Systems, Robotics: Science and Systems, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, IEEE Conference on Decision and Control, American Control Conference, European Control Conference, International Joint Conference on Artificial Intelligence, SIAM Conference on Control and Its Applications

I.13 Dissemination

- CyberWalk Project: Press releases at BBC news (2008), Focus Online (2008), National Geographic (2008), Spectrum Magazine (2010)
- MPI CyberMotion Simulator: Press releases at BBC Focus Magazine (2010), Wired UK (2010), Spectrum Magazine (2010) and YouTube video (560k views), GizModo (2010), EnGadget (2010), ABC Television (2011)
- Human-Robot Interaction: Press releases at Die Rheinpfalz (German newspaper, 2011), Die Zeit (German newspaper 2011), Schwäbisches Tagblatt (German newspaper 2011), Deutschlandradio (German radio 2011), TG2 Dossier (Italian public tv 2011)
- Quadrotor UAVs: Press release at Spectrum Magazine (2013)

I.14 Summary of Publications and Invited Talks

Category	Published	Under Review	Total
Refereed Journal Publications	20	7	27
Refereed Conference Publications	63	2	65
Conference Talks and Workshops	18	0	18
Invited Talks and Seminars	29	0	29

Journals: **6** The International Journal of Robotics Research (1 in a *special issue*), **4** IEEE Transaction on Robotics, **2** IEEE Transactions on Control Systems Technology, **2** ACM Transactions on Applied Perception, **2** Experimental Brain Research, **1** Robotica (in a *special issue*), **1** IEEE Robotics and Automation Magazine (1 in a *special issue*), **1** IEEE Transactions on Systems, Man, and Cybernetics, part B, **1** IEEE/ASME Transactions on Mechatronics (in a *special issue*)

Conferences: **28** IEEE International Conference on Robotics and Automation (ICRA), **12** IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (**2** Invited), **3** IEEE Conference on Decision and Control (**1** Invited), **2** Robotics: Science and Systems Conference (RSS), **1** World Haptic Conference (WHC), **1** Eurohaptics, **1** International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), **1** European Conference on Mobile Robots (ECMR), **1** European Control Conference (ECC)

I.15 Bibliometric indicators

Google scholar profile: http://scholar.google.it/citations?user=dpayTBUAAAAJ, H-index: 23 (January 2016)

I.16 Publications and Invited Talks

Submitted Journal Publications

[**J27**] T. Nestmeyer, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi. Decentralized Simultaneous Multi-target Exploration using a Connected Network of Multiple Robots. *submitted to Autonomous Robots*, 2015.

[**J26**] S. Bista, P. Robuffo Giordano, and F. Chaumette. Appearance-based Indoor Navigation by IBVS using Line Segments. *submitted to the IEEE Robotics and Automation Letters*, 2015.

[**J25**] C. Masone, P. Robuffo Giordano, and A. Franchi. A Framework for Shared Planning and Control with Integral Haptic Feedback. *submitted to The International Journal of Robotics Research*, 2015.

[**J24**] A. Franchi and P. Robuffo Giordano. Online Leader Selection for Improved Collective Tracking and Formation Maintenance. *submitted to the IEEE Transactions on Control of Network Systems*, 2015.

[**J23**] R. Spica, P. Robuffo Giordano, and F. Chaumette. Bridging Visual Control and Active Perception via a Large Projector Operator. *submitted to the IEEE Transactions on Robotics*, 2015.

[**J22**] F. Flacco, P. Robuffo Giordano, and A. De Luca. A Reverse Priority Approach to Smooth Task Transitions. *submitted to the IEEE Transactions on Robotics*, 2015.

[**J21**] G. Antonelli, E. Cataldi, F. Arrichiello, P. Robuffo Giordano, S. Chiaverini, and A. Franchi. Adaptive Trajectory Tracking for Quadrotors MAVs in Presence of Parameter Uncertainties and External Disturbances. *submitted to the IEEE/ASME Transactions on Mechatronics*, 2015.

Refereed Journal Publications

[**J20**] V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. Robuffo Giordano. Nonlinear Ego-Motion Estimation from Optical Flow for Online Control of a Quadrotor UAV. *The International Journal of Robotics Research*, 34(8):1114–1135, 2015.

[**J19**] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. A Novel Overactuated Quadrotor UAV: Modeling, Control and Experimental Validation. *IEEE Transactions on Control Systems Technology*, 23(2):540–556, 2015.

[**J18**] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Decentralized Rigidity Maintenance Control with Range Measurements for Multi-Robot Systems. *The International Journal of Robotics Research*, 34(1):105–128, 2015.

[**J17**] R. Spica, P. Robuffo Giordano, and F. Chaumette. Active Structure from Motion: Application to Point, Sphere and Cylinder. *IEEE Transactions on Robotics*, 30(6):1499–1513, 2014.

[**J16**] D. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano. Semi-Autonomous Haptic Teleoperation Control Architecture of Multiple Unmanned Aerial Vehicles. *IEEE/ASME Transactions on Mechatronics,* **Focused Section on Aerospace Mechatronics**, 18(4):1334–1345, 2013.

[**J15**] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. A Passivity-Based Decentralized Strategy for Generalized Connectivity Maintenance. *The International Journal of Robotics Research*, 32(3):299–323, 2013.

[**J14**] H. I. Son, A. Franchi, L. L. Chuang, J. Kim, H. H. Bülthoff, and P. Robuffo Giordano. Human-Centered Design and Evaluation of Haptic Cueing for Teleoperation of Multiple Mobile Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 43(2):597–609, 2013.

[**J13**] A. De Luca, R. Mattone, P. Robuffo Giordano, H. Ulbrich, M. Schwaiger, M. Van den Bergh, E. Koller-Meier, and L. Van Gool. Motion Control of the CyberCarpet Platform. *IEEE Transactions on Control Systems Technology*, 21(2):410–427, 2013.

[**J12**] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Shared Control: Balancing Autonomy and Human Assistance with a Group of Quadrotor UAVs. *IEEE Robotics and Automation Magazine,* **Special Issue on Aerial Robotics and the Quadrotor Platform**, 19(3):57–68, 2012.

[**J11**] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Modeling and Control of UAV Bearing-Formations with Bilateral High-Level Steering. *The International Journal of Robotics Research,* **Special Issue on 3D Exploration, Mapping and Surveillance**, 31(12):1504–1525, 2012.

[**J10**] A. Franchi, C. Secchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of Groups of Mobile Robots with Time-Varying Topology. *IEEE Transactions on Robotics*, 28(5):1019–1033, 2012.

[**J9**] S. Haddadin, A. Peer, and P. Robuffo Giordano. Guest Editorial: **Special Issue on Autonomous Physical Human-Robot Interaction**. *The International Journal of Robotics Research*, 31(13):1529–1530, 2012.

[**J8**] F. Soyka, P. Robuffo Giordano, M. Barnett-Cowan, and H. H. Bülthoff. Modeling direction discrimination thresholds for yaw rotations around an earth-vertical axis for arbitrary motion profiles. *Experimental Brain Research*, 220(1):89–99, 2012.

[J7] J. L. Souman, P. Robuffo Giordano, M. Schwaiger, I. Frissen, T. Thümmel, H. Ulbrich, A. De Luca, H. H. Bülthoff, and M. O. Ernst. CyberWalk: Enabling unconstrained omnidirectional walking through virtual environments. *ACM Transactions on Applied Perception*, 8(4), 2011.

[J6] F. Soyka, P. Robuffo Giordano, K. Beykirch, and H. H. Bülthoff. Predicting Direction Detection Thresholds for Arbitrary Translational Acceleration Profiles in the Horizontal Plane. *Experimental Brain Research*, 209(1):95–107, 2011.

[J5] P. Robuffo Giordano and M. Vendittelli. Shortest paths to obstacles for a polygonal Dubins car. *IEEE Transactions on Robotics*, 25(5):1184–1191, 2009.

[J4] J. L. Souman, P. Robuffo Giordano, I. Frissen, A. De Luca, and M. O. Ernst. Making virtual walking real: perceptual evaluation of a new treadmill control algorithm. *ACM Transactions on Applied Perception*, 7(2), 2009.

[J3] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Feature Depth Observation for Image-based Visual Servoing: Theory and Experiments. *The International Journal of Robotics Research*, 27(10):1093–1116, 2008.

[J2] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Image-based visual servoing schemes for nonholonomic mobile manipulators. *Robotica*, 25(2):131–145, 2007.

[J1] P. Robuffo Giordano, M. Vendittelli, J.-P. Laumond, and P. Souères. Nonholonomic distance to polygonal obstacles for a car-like robot of polygonal shape. *IEEE Transactions on Robotics*, 22(5):1040–1047, 2006.

Submitted Conference Publications with Proceedings

[C65] F. Schiano, A. Franchi, D. Zelazo, and P. Robuffo Giordano. A Rigidity-Based Decentralized Bearing Formation Controller for Groups of Quadrotor UAVs. In *submitted to the 2016 IEEE Int. Conf. on Robotics and Automation (ICRA 2016)*, 2016.

[C64] S. R. Bista, P. Robuffo Giordano, and F. Chaumette. Appearance based Indoor Navigation by IBVS using Mutual Information. In *submitted to the 2016 IEEE Int. Conf. on Robotics and Automation (ICRA 2016)*, 2016.

Refereed Conference Publications with Proceedings

[C63] D. Zelazo, P. Robuffo Giordano, and A. Franchi. Bearing-Only Formation Control Using an $SE(2)$ Rigidity Theory. In *2015 IEEE Conf. on Decision and Control (CDC 2015)*, 2015.

[C62] Omar Tahri, Paolo Robuffo Giordano, and Youcef Mezouar. Rotation Free Active Vision. In *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2015)*, 2015.

[C61] P. Robuffo Giordano, R. Spica, and F. Chaumette. Learning the Shape of Image Moments for Optimal 3D Structure Estimation. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015.

[C60] R. Spica, P. Robuffo Giordano, and F. Chaumette. Plane Estimation by Active Vision from Point Features and Image Moments. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015.

[C59] N. Cazy, P.-B. Wieber, P. Robuffo Giordano, and F. Chaumette. Visual Servoing when visual information is missing: experimental comparison of Visual Feature Prediction Schemes. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015.

[C58] N. Cazy, C. Dune, P.-B. Wieber, P. Robuffo Giordano, and F. Chaumette. Pose Error Correction For Visual Features Prediction. In *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2014)*, 2014.

[C57] D. Zelazo, A. Franchi, and P. Robuffo Giordano. Rigidity Theory in $SE(2)$ for Unscaled Relative Position Estimation using only Bearing Measurements. In *Proc. of the 2014 European Control Conference (ECC 2014)*, 2014.

[C56] C. Masone, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi. Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C55] R. Spica, P. Robuffo Giordano, and F. Chaumette. Coupling Image-Based Visual Servoing with Active Structure from Motion. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C54] R. Spica, P. Robuffo Giordano, and F. Chaumette. Active Structure from Motion for Spherical and Cylindrical Targets. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C53] P. Robuffo Giordano, R. Spica, and F. Chaumette. An Active Strategy for Plane Detection and Estimation for a Monocular Camera. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C52] R. Spica, P. Robuffo Giordano, M. Ryll, H. H. Bülthoff, and A. Franchi. An Open-Source Hardware/-Software Architecture for Quadrotor UAVs. In *2nd Workshop on Research, Education and Development of Unmanned Aerial System*, 2013.

[C51] R. Spica and P. Robuffo Giordano. A Framework for Active Estimation: Application to Structure from Motion. In *Proc. of the 2013 IEEE Conf. on Decision and Control (CDC 2013)*, 2013.

[C50] G. Antonelli, E. Cataldi, P. Robuffo Giordano, S. Chiaverini, and A. Franchi. Experimental validation of a new adaptive control scheme for quadrotors MAVs. In *Proc. of the 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2013)*, 2013.

[C49] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. A Comparison of Scale Estimation Schemes for a Quadrotor UAV based on Optical Flow and IMU Measurements. In *Proc. of the 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2013)*, 2013.

[C48] V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi. The TeleKyb Framework for a Modular and Extendible ROS-based Quadrotor Control. In *Proc. of the 6th European Conf. on Mobile Robots (ECMR 2013)*, 2013.

[C47] G. Antonelli, F. Arrichiello, S. Chiaverini, and P. Robuffo Giordano. Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances. In *Proc. of the 2013 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM 2013)*, 2013.

[C46] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. First Flight Tests for a Quadrotor UAV with Tilting Propellers. In *Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*, 2013.

[C45] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Control of the Degree of Connectivity in Multiple Mobile-robot Teleoperation. In *Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*, 2013.

[C44] A. Franchi and P. Robuffo Giordano. Decentralized Control of Parallel Rigid Formations with Direction Constraints and Bearing Measurements. In *Proc. of the 51st IEEE Conference on Decision and Control (CDC 2012)*, pages 5310–5317, Maui, HI, Dec. 2012.

[C43] J. Lächele, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. SwarmSimX: Real-time Simulation Environment for Multi-robot Systems. In *Proc. of the 3rd Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2012)*, Tsukuba, Japan, Nov. 2012.

[C42] A. Nesti, C. Masone, M. Barnett-Cowan, P. Robuffo Giordano, H. H. Bülthoff, and P. Pretto. Roll rate thresholds and perceived realism in driving simulation. In *Proc. of the Driving Simulation Conference 2012 Europe (DSC 2012)*, 2012.

[C41] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Interactive Planning of Persistent Trajectories for Human-Assisted Navigation of Mobile Robots. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, pages 2641–2648, Vilamoura, Portugal, Oct. 2012.

[C40] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. Robust Optical-Flow Based Self-Motion Estimation for a Quadrotor UAV. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Portugal, Oct. 2012.

[C39] R. Spica, A. Franchi, G. Oriolo, H. H. Bülthoff, and P. Robuffo Giordano. Aerial Grasping of a Moving Target with a Quadrotor UAV. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, pages 4985–4992, Vilamoura, Portugal, Oct. 2012.

[C38] M. Riedel, A. Franchi, H. H. Bülthoff, P. Robuffo Giordano, and H. I. Son. Experiments on Intercontinental Haptic Control of Multiple UAVs. In *Proc. of the 12th International Conference on Intelligent Autonomous System (IAS 2012)*, pages 227–238, Jeju Island, Korea, Jun. 2012.

[C37] D. Zelazo, A. Franchi, F. Allgöwer, H. H. Bülthoff, and P. Robuffo Giordano. Rigidity Maintenance Control for Multi-Robot Systems. In *Proc. of the 2012 Robotics: Science and Systems Conference (RSS 2012)*, Sydney, Australia, Jul. 2012.

[C36] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of a Group of UAVs with Communication Delays and Switching Topology. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 4307–4314, St. Paul, MN, May 2012.

[C35] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Modeling and Control of a Quadrotor UAV with Tilting Propellers. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 4606–4613, St. Paul, MN, May 2012.

[C34] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 491–497, St. Paul, MN, May 2012.

[C33] C. Secchi, P. Robuffo Giordano, and A. Franchi. Bilateral Teleoperation of Groups of Mobile Robots with Time-Varying Topology. In *Automatica.it 2011*, Pisa, Italy, Sep. 2011.

[C32] A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Distributed Online Leader Selection in the Bilateral Teleoperation of Multiple UAVs. In *Proc. of the 50th IEEE Conference on Decision and Control,* **Invited paper** *(CDC 2011)*, pages 3559–3565, Orlando, FL, Dec. 2011.

[C31] A. Franchi, C. Masone, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of Multiple UAVs with Decentralized Bearing-only Formation Control. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* **Invited paper** *(IROS 2011)*, pages 2215–2222, San Francisco, CA, Sep. 2011.

[C30] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. Experiments of Passivity-Based Bilateral Aerial Teleoperation of a Group of UAVs with Decentralized Velocity Synchronization. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* **Invited paper** *(IROS 2011)*, pages 163–170, San Francisco, CA, Sep. 2011.

[C29] H. I. Son, L. L. Chuang, A. Franchi, J. Kim, D. J. Lee, S. W. Lee, H. H. Bülthoff, and P. Robuffo Giordano. Measuring an Operator's Maneuverability Performance in the Haptic Teleoperation of Multiple Robots. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2011)*, pages 3039–3046, San Francisco, CA, Sep. 2011.

[C28] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. Bilateral Teleoperation of Groups of UAVs with Decentralized Connectivity Maintenance. In *Proc. of the 2011 Robotics: Science and Systems Conference (RSS 2011)*, Los Angeles, CA, Jun. 2011.

[C27] H. I. Son and J. Kim and L. Chuang and A. Franchi and P. Robuffo Giordano and D. Lee and H. H. Bülthoff. An evaluation of haptic cues on the tele-operator's perceptual awareness of multiple UAVs' environments. In *Proc. of the 2011 IEEE World Haptics Conference (WHC 2011)*, pages 149–154, Istanbul, Turkey, Jun. 2011.

[C26] C. Masone, P. Robuffo Giordano, and H. H. Bülthoff. Mechanical Design and Control of the new 7-DOF CyberMotion Simulator. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 4935–4942, Shanghai, China, May 2011.

[C25] A. Franchi, P. Robuffo Giordano, C. Secchi, H. I. Son, and H. H. Bülthoff. A Passivity-Based Decentralized Approach for the Bilateral Teleoperation of a Group of UAVs with Switching Topology. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 898–905, Shanghai, China, May 2011.

[C24] D. Lee, A. Franchi, P. Robuffo Giordano, H. I. Son, and H. H. Bülthoff. Haptic Teleoperation of Multiple Unmanned Aerial Vehicles over the Internet. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 1341–1347, Shanghai, China, May 2011.

[C23] V. Grabe, P. Pretto, P. Robuffo Giordano, and H. H. Bülthoff. Influence of Display Type on Drivers' Performance in a motion-based Driving Simulator. In *Proc. of the Driving Simulation Conference 2010 Europe (DSC 2010)*, 2010.

[C22] S. M. C. Alaimo, L. Pollini, A. Magazzù, J.-P. Bresciani, P. Robuffo Giordano, M. Innocenti, and H. H. Bülthoff. Preliminary Evaluation of a Haptic Aiding Concept for Remotely Piloted Vehicles. In *Proc. of EuroHaptics 2010*, 2010.

[C21] P. Robuffo Giordano, H. Deusch, J. Lächele, and H. H. Bülthoff. Visual-Vestibular Feedback for Enhanced Situational Awareness in Teleoperation of UAVs. In *Proc. of the AHS International 66th Annual Forum & Technological Display*, 2010.

[C20] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Kinematic Control of Nonholonomic Mobile Manipulators in the Presence of Steering Wheels. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 1792–1798, 2010.

[C19] P. Robuffo Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini, and H. H. Bülthoff. A Novel Framework for Closed-Loop Robotic Motion Simulation - Part I: Inverse Kinematics Design. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 3876–3883, 2010.

[C18] P. Robuffo Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini, and H. H. Bülthoff. A Novel Framework for Closed-Loop Robotic Motion Simulation - Part II: Motion Cueing Design and Experimental Validation. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 3896–3903, 2010.

[C17] A. De Luca, R. Mattone, P. Robuffo Giordano, and H. H. Bülthoff. Control Design and Experimental Evaluation of the 2D CyberWalk Platform. In *Proc. of the 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2009)*, pages 5051–5058, 2009.

[C16] M. Niccolini, L. Pollini, M. Innocenti, P. Robuffo Giordano, H. Teufel, and H. H. Bülthoff. Towards Real-Time Aircraft Simulation with the MPI Motion Simulator. In *Proc. of the 2009 AIAA Modeling and Simulation Technologies Conference*, 2009.

[C15] F. Soyka, H. Teufel, K. Beykirch, P. Robuffo Giordano, J. Butler, F. M. Nieuwenhuizen, and H. H. Bülthoff. Does jerk have to be considered in linear motion simulation? In *Proc. of the 2009 AIAA Modeling and Simulation Technologies Conference*, 2009.

[C14] C. Borst, T. Wimböck, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. Robuffo Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schäffer, and G. Hirzinger. Rollin' Justin – Mobile Platform with Variable Base. In *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation,* **Best Video Award** *(ICRA 2009)*, pages 1597–1598, 2009.

[C13] M. Fuchs, C. Borst, P. Robuffo Giordano, A. Baumann, E. Krämer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze, R. Burger, F. Schmidt, T. Wimböck, and G. Hirzinger. Rollin' Justin – Design Considerations and Realization of a Mobile Platform for a Humanoid Upper Body. In *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation (ICRA 2009)*, pages 4131–4137, 2009.

[C12] P. Robuffo Giordano, M. Fuchs, A. Albu-Schäffer, and G. Hirzinger. On the Kinematic Modeling and Control of a Mobile Platform Equipped with Steering Wheels and Movable Legs. In *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation (ICRA 2009)*, pages 4080–4087, 2009.

[C11] P. Robuffo Giordano, A. Stemmer, K. Arbter, and A. Albu-Schäffer. Robotic assembly of complex planar parts: An experimental evaluation. In *Proc. of the 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2008)*, pages 3775–3782, 2008.

[C10] A. De Luca, M. Ferri, G. Oriolo, and P. Robuffo Giordano. Visual Servoing with Exploitation of Redundancy: An Experimental Study. In *Proc. of the 2008 IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*, pages 3231–3237, 2008.

[C9] P. Robuffo Giordano, A. De Luca, and G. Oriolo. 3D Structure Identification from Image Moments. In *Proc. of the 2008 IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*, pages 93–100, 2008.

[C8] A. De Luca, G. Oriolo, and P. Robuffo Giordano. On-Line Estimation of Feature Depth for Image-Based Visual Servoing Schemes. In *Proc. of the 2007 IEEE Int. Conf. on Robotics and Automation (ICRA 2007)*, pages 2823–2828, 2007.

[C7] A. De Luca, R. Mattone, and P. Robuffo Giordano. Acceleration-level control of the CyberCarpet. In *Proc. of the 2007 IEEE Int. Conf. on Robotics and Automation (ICRA 2007)*, pages 2330–2335, 2007.

[C6] A. De Luca, R. Mattone, and P. Robuffo Giordano. Feedback/Feedforward Schemes for Motion Control of the CyberCarpet. In *Proc. of the 8th Int. IFAC Symposium on Robot Control (SYROCO 2006)*, 2006.

[C5] P. Robuffo Giordano and M. Vendittelli. The Minimum-Time Crashing Problem for the Dubin's Car. In *Proc. of the 8th Int. IFAC Symposium on Robot Control (SYROCO 2006)*, 2006.

[C4] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Kinematic Modeling and Redundancy Resolution for Nonholonomic Mobile Manipulators. In *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation (ICRA 2006)*, pages 1867–1873, 2006.

[C3] A. De Luca, R. Mattone, and P. Robuffo Giordano. The motion control problem for the CyberCarpet. In *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation (ICRA 2006)*, pages 3532–3537, 2006.

[C2] A. De Luca, G. Oriolo, L. Paone, P. Robuffo Giordano, and M. Vendittelli. Visual-based planning and control for nonholonomic mobile robots. In *Proc. of the 10th IEEE Mediterranean Conference on Control and Automation (MED 2002)*, 2002.

[C1] A. De Luca, G. Oriolo, L. Paone, and P. Robuffo Giordano. Experiments in visual feedback control of a wheeled mobile robot. In *Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation (ICRA 2002)*, pages 2073–2078, 2002.

Misc

[M1] P. Robuffo Giordano. Interaktionen zwischen Menschen und autonomen Robotern: Human-Robot Interaction by Shared Autonomy. *Jahrbuch der Max-Planck-Gesellschaft*, 2012.

Theses

[TH1] P. Robuffo Giordano. *Visual Estimation and Control of Robot Manipulating Systems.* PhD thesis, Università di Roma "La Sapienza", 2008.

Conference Talks and Workshops

[W18] R. Spica, P. Robuffo Giordano, and F. Chaumette. A Framework for Coupling Visual Control and Active Structure from Motion. In *ICRA 2015 Workshop: Scaling Up Active Perception*, Seattle, USA, May 2015.

[**W17**] R. Spica, G. Claudio, F. Spindler, and P. Robuffo Giordano. Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms. In *ICRA 2014 Workshop: MATLAB/Simulink for Robotics Education and Research*, Hong Kong, China, July 2014.

[**W16**] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Overactuation in UAVs for Enhanced Aerial Manipulation: A Novel Quadrotor Concept with Tilting. In *6th Int. Work. on Human-Friendly Robotics (HFR 2013)*, Rome, Italy, September 2013.

[**W15**] T. Nestmeyer, P. Robuffo Giordano, and A. Franchi. Human-assisted Parallel Multi-target Visiting in a Connected Topology. In *6th Int. Work. on Human-Friendly Robotics (HFR 2013)*, Rome, Italy, September 2013.

[**W14**] J. Lächele, M. Riedel, P. Robuffo Giordano, and A. Franchi. SwarmSimX and TeleKyb: Two ROS-integrated Software Frameworks for Single- and Multi-Robot Applications. In *Int. Work. on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration at ICRA 2013*, Karlsruhe, Germany, May 2013.

[**W13**] T. Nestmeyer, M. Riedel, J. Lächele, S. Hartmann, F. Botschen, P. Robuffo Giordano, and A. Franchi. Interactive Demo: Haptic Remote Control of Multiple UAVs with Autonomous Cohesive Behavior. In *Int. Work. on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration at ICRA 2013*, Karlsruhe, Germany, May 2013.

[**W12**] T. Nestmeyer, P. Robuffo Giordano, and A. Franchi. Multi-target Simultaneous Exploration with Continual Connectivity. In *2th International Workshop on Crossing the Reality Gap: From Single to Multi- to Many Robot Systems at ICRA 2013*, Karlsruhe, Germany, May 2013.

[**W11**] P. Robuffo Giordano. Shared Control of Multi-Robot Systems: Passivity, Decentralization, and Connectivity Maintenance. In *Workshop on Haptic Teleoperation of Mobile Robots: Theory, Applications and Perspectives at ICRA 2012*, St. Paul, MN, USA, May 2012.

[**W10**] F. Soyka, M. Barnett Cowan, P. Robuffo Giordano, and H. H. Bülthoff. Translations are processed slower than rotations: reaction times for self-motion stimuli predicted by vestibular organ dynamics. In *27th Bárány Society Meeting*, Uppsala, Sweden, June 2012.

[**W9**] F. Soyka, M. Barnett Cowan, P. Robuffo Giordano, and H. H. Bülthoff. Temporal processing of self-motion: Translations are processed slower than rotations. In *13th International Multisensory Research Forum (IMRF 2012)*, Oxford, UK, June 2012.

[**W8**] M. Riedel, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Intercontinental Haptic Control and Advanced Supervisory Interfaces for Groups of Multiple UAVs. In *5th Int. Work. on Human-Friendly Robotics (HFR 2012)*, Bruxelles, Belgium, Oct. 2012.

[**W7**] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Shared Trajectory Planning for Human-in-the-loop Navigation of Mobile Robots in Cluttered Environments. In *5th Int. Work. on Human-Friendly Robotics (HFR 2012)*, Bruxelles, Belgium, Oct. 2012.

[**W6**] A. Franchi, C. Masone, and P. Robuffo Giordano. A Synergetic High-level/Reactive Planning Framework with Application to Human-Assisted Navigation. In *2012 IEEE IROS Work. on Real-time Motion Planning: Online, Reactive, and in Real-time (IROS 2012)*, Vilamoura, Portugal, Oct. 2012.

[**W5**] P. Robuffo Giordano. Towards Aerial Telerobotics: Enabling human operators to bilaterally control single/multiple UAVs for accomplishing remote tasks. In *euRobotics Forum: UAV Workshop*, Västeras, Sweden, April 2011.

[**W4**] P. Robuffo Giordano. Decentralized Bilateral Aerial Teleoperation of Multiple UAVs - Part II: a Bottom-Up Perspective. In *RSS 2011 Work. on 3D Exploration, Mapping, and Surveillance with Aerial Robots (RSS 2011)*, Los Angeles, CA, Jun. 2011.

[**W3**] P. Robuffo Giordano, A. Franchi, H. I. Son, C. Secchi, D. Lee, and H. H. Bülthoff. Towards Bilateral Teleoperation of Multi-Robot Systems (hfr 2010). In *3rd Int. Work. on Human-Friendly Robotics*, Tübingen, Germany, Oct. 2010.

[**W2**] P. Robuffo Giordano. Providing vestibular cues to a human operator for a new generation of human-machine interfaces. In *2nd Int. Work. on Human-Friendly Robotics (HFR 2009)*, 2009.

[**W1**] P. Robuffo Giordano, J. L. Souman, R. Mattone, A. De Luca, M. O. Ernst, and H. H. Bülthoff. The CyberWalk Platform: Human-Machine Interaction Enabling Unconstrained Walking through VR. In *1st Int. Work. on Human-Friendly Robotics (HFR 2008)*, 2008.

Invited Talks and Seminars

[**T29**] P. Robuffo Giordano. **Invited Talk**: Control of Quadrotors in Unstructured Environments. *RSS 2015 Symposium: Frontiers of Robotics*, April 2015.

[**T28**] P. Robuffo Giordano. **Invited Talk**: An Overview on Collective Coordination of Multiple Robots. *Computer Science and Telecommunications Department, Ecole Normale Supèrieure, Rennes, France*, October 2014.

[**T27**] P. Robuffo Giordano. **Invited Talk**: Collective Control, State Estimation and Human Interaction for Teams of Mobile Robots. *ACROSS Workshop on Cooperative Systems, Dubrovnik, Croatia*, September 2014.

[**T26**] P. Robuffo Giordano. **Invited Talk**: Collective Control, State Estimation and Human Interaction for Teams of Mobile Robots. *Symposium "The Very Best of Robotics Today", Department of Computer Science, Rice University, Houston, USA*, April 2014.

[**T25**] P. Robuffo Giordano. **Invited Talk**: Vision-Based Bilateral Shared Control and Autonomous Flight for Single and Multiple Quadrotor UAVs. *IEEE/RSJ IROS Int. Work. on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments*, November 2013.

[**T24**] P. Robuffo Giordano. **Invited Talk**: Overactuation, state estimation and human interaction for quadrotor UAVs. *UAV French day, Ensam, Paris, France*, June 2013.

[**T23**] P. Robuffo Giordano. **Invited Talk**: An overview of recent results in the bilateral shared control of multiple mobile robots. *ISIR, Université Pierre et Marie Curie, Paris, France*, June 2013.

[**T22**] P. Robuffo Giordano. **Invited Seminar**: A Framework For Nonlinear Active Estimation With Applications To Structure From Motion. *DIAG, University of Rome "La Sapienza", Rome, Italy*, May 2013.

[**T21**] P. Robuffo Giordano. **Invited Talk**: An overview of recent results in the bilateral shared control of multiple mobile robots. *INRIA, Grenoble, France*, April 2013.

[**T20**] P. Robuffo Giordano. **Invited Talk**: Passive and Decentralized Shared Control of Multi-Robot Systems. *I3S, Sophia-Antipolis, France*, February 2013.

[**T19**] P. Robuffo Giordano. **Invited Talk**: Passive and Decentralized Shared Control of Multi-Robot Systems. *Australian National University, Canberra, Australia*, July 2012.

[**T18**] P. Robuffo Giordano. **Invited Talk**: Multi-UAV Bilateral Shared Control and Decentralization. *Summer School Airobots: Aerial Service Robotics, Eidgenössische Technische Hochschule, Zürich, Switzerland*, July 2012.

[**T17**] P. Robuffo Giordano. **Invited Seminar**: Passive and Decentralized Shared Control of Multi-Robot Systems. *Università di Modena e Reggio Emilia, Reggio Emilia, Italy*, June 2012.

[**T16**] P. Robuffo Giordano. **Invited Talk**: Shared control of remote multi-robot aerial systems: state-of-the-art and future perspectives. *"UAVs in perspective" Symposium, Università di Bologna, Bologna, Italy*, May 2012.

[**T15**] P. Robuffo Giordano. **Invited Talk**: Passive and Decentralized Shared Control of Multi-Robot Systems. *Boston University, Boston, MA, USA*, May 2012.

[T14] P. Robuffo Giordano. **Invited Talk**: Shared Control of Remote Multi-Robot Systems: State-of-the-art and Future Perspectives. *IRISA/INRIA, Rennes, France*, March 2012.

[T13] P. Robuffo Giordano. **Invited Talk**: Shared Control of Remote Multi-Robot Systems: State-of-the-art and Future Perspectives. *University of Würzburg–Institute of Computer Science, Würzburg, Germany*, March 2012.

[T12] P. Robuffo Giordano. **Invited Seminar**: Bilateral Teleoperation of Multi-Robot Systems: Passivity, Decentralization, and Connectivity Maintenance. *Oberseminar Intelligent Autonomous Systems Group, Technische Universität Darmstadt, Germany*, December 2011.

[T11] P. Robuffo Giordano. **Invited Seminar**: Bilateral Teleoperation of Groups of Mobile Robots. *SIRSLab, Dipartimento di Ingegneria dell'Informazione Università di Siena, Siena, Italy*, November 2011.

[T10] P. Robuffo Giordano. **Invited Seminar**: An Introduction to Passivity and port-Hamiltonian Systems. *Robotics Group, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Roma, Italy*, December 2010.

[T9] P. Robuffo Giordano. **Invited Talk**: Towards Bilateral Teleoperation of Multi-Robot Systems. *Autonomous Intelligent Systems Lab, University of Freiburg, Germany*, November 2010.

[T8] P. Robuffo Giordano. **Invited Talk**: Realizing a Closed-Loop Robotic Motion Simulator. *GRASP Lab: University of Pennsylvania, Philadelphia, USA*, May 2010.

[T7] P. Robuffo Giordano. **Invited Talk**: Human-Machine Interfaces for VR and Real World Applications. *Autonomous Systems Laboratory, Eidgenössische Technische Hochschule Zürich (ETH), Switzerland*, April 2010.

[T6] P. Robuffo Giordano. **Invited Talk**: Towards a New Generation of Multisensory Human-Machine Interfaces for VR and Real-World Applications. *Kolloquium Technische Kybernetik, Institut für Systemtheorie und Regelungstechnik, Stuttgart, Germany*, January 2010.

[T5] P. Robuffo Giordano. **Invited Seminar**: Introduction to the state estimation for dynamical systems. *Max Planck Institute for Biological Cybernetics, Tübingen, Germany*, November 2009.

[T4] P. Robuffo Giordano. **Invited Talk**: Modeling and Control of a Wheeled Mobile Platform with Movable Legs. *Institute for Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany*, October 2008.

[T3] P. Robuffo Giordano. **Invited Talk**: A control theory approach for modeling and control of robotic systems. *Max Planck Institute for Biological Cybernetics, Tübingen, Germany*, May 2008.

[T2] P. Robuffo Giordano. **Invited Seminar**: Introduction to Visual Servoing: Basic and Advanced Methods. *Institute for Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany*, May 2007.

[T1] P. Robuffo Giordano. **Invited Seminar**: Shortest Path Synthesis for a Reeds-Shepp Car. *Robotics Group, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Roma, Italy*, June 2005.

Chapter II

Summary of Early Research Activities

The chapter summarizes the research activities carried out during my Ph.D. at the University of Rome "La Sapienza" (October 2004–November 2007), my first PostDoc at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR, November 2007– October 2008), and the short visits paid at the Max Planck Institute for Biological Cybernetics (MPI) during this period.

During the early stages of my career I gave contributions in **robot motion planning and control**, **visual/force control of redundant fixed- and mobile-manipulators**, **3D structure estimation from motion**, and **locomotion interfaces for Virtual Reality (VR)**. Brief summaries of these works, partly done within the EU FP6 STREP Phriends and EU FP6 STREP CyberWalk international research projects, are detailed in the next Sect. II.1, Sect. II.2 and Sect. II.3.

II.1 Planning and Control for Mobile Robots and Fixed/Mobile Manipulators

During my PhD I studied the problem of controlling the motion of Wheeled Mobile Robots (WMRs) and Nonholonomic Mobile Manipulators (NMMs), that is, robotic structures with restricted instantaneous mobility but still possessing full (but not instantaneous) controllability of their configuration space. The mobility restriction of wheeled robots is caused by their actuation system: the constraint of wheels perfectly rolling without slipping introduces a set of so-called "nonholonomic" constraints on the robot base velocities [Murray et al., 1994]. As a consequence, motion planning and control for these systems is considerably more challenging than for normal fully actuated robotic structures (such as robot manipulators), and has received large attention by the control and robotics communities. Within this context, I gave contributions in the following areas:

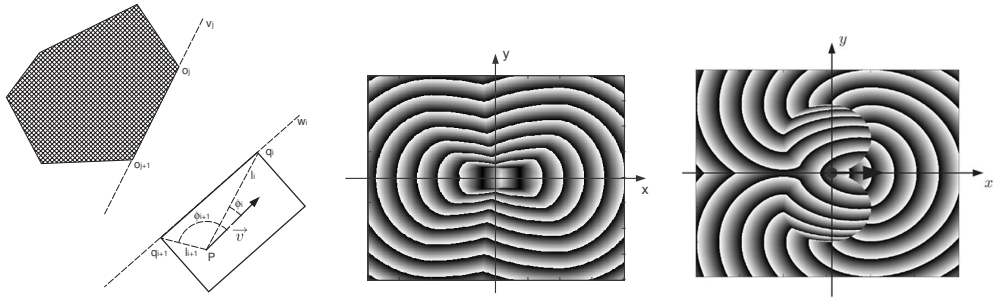


FIGURE II.1: Left: some relevant definition of the robot-obstacle distance. Center and Right: isodistance curves for a Reeds-Shepp car and a Dubins' car, respectively

II.1.1 Shortest Paths to Obstacles for WMRs

In the context of planning among obstacles, I studied the problem of defining a proper notion of "distance to obstacles" for WMRs with a *nontrivial polygonal shape*, that is, a suitable metric able to capture the nonholonomy of WMRs. In fact, knowledge of this distance is particularly relevant in the context of motion planners relying on obstacle distance computation, like skeletonization, potential field methods, and sampling-based motion planners. More in general, knowledge of this distance has strong implications whenever one needs to evaluate the traveled distance of a (polygonal) WMR to goal locations or from surrounding obstacles. As a trivial example: for a car-like robot, obstacles in the forward direction are "closer" than obstacles in the side direction because of the well-known impossibility for cars to instantaneously travel sideways. In [J1, C5, J5], I have analyzed this problem borrowing from tools of optimal control, and found a closed-form expression for the sought distance to polygonal obstacles. Figure II.1(a) shows some relevant definitions, and Figs. II.1(b–c) illustrates the 'isodistance' curves in the plane for a *polygonal WMR* able to travel in forward/backward motion (Reeds-Shepp car), and only in forward motion (Dubins' car).

II.1.2 Modeling and Control of Redundant NMMs

Extending my previous works on WMRs, I then addressed the issues of kinematic modeling and control of a (at that time) novel class of robotic systems, i.e., the so-called Nonholonomic Mobile Manipulators (NMMs). These robotic systems are made of articulated arms mounted on a wheeled mobile platform. Since this mechanical arrangement combines the dexterity of the arm with the workspace extension of the mobile platform, it is clearly appealing for many applications, and in particular for service robotics. In [C4], I addressed the problem of modeling and controlling NMMs by finding ways to exploit the internal redundancy of the system, arising both from the manipulator kinematics and from the platform mobility. This allowed to describe NMMs by means of a unified kinematic model automatically taking into account the platform nonhlonomic constraints. Adopting this modeling approach, one can then address all the classical problems considered for standard redundant manipulators (study of singularities and their avoidance, augmentation of tasks and their prioritization, optimization of performance criteria, cyclicity of repeated motions, etc.). Figure II.2(a) illustrates the schematics of a standard NMM (platform with unicycle kinematics equipped with a 3R spatial arm). This modeling and control framework for NMMs was also applied to the case of *visual control* tasks, that is, when a camera mounted on the end-effector arm is the source of "position feedback" for the robot [J2]. Again, it was possible to exploit

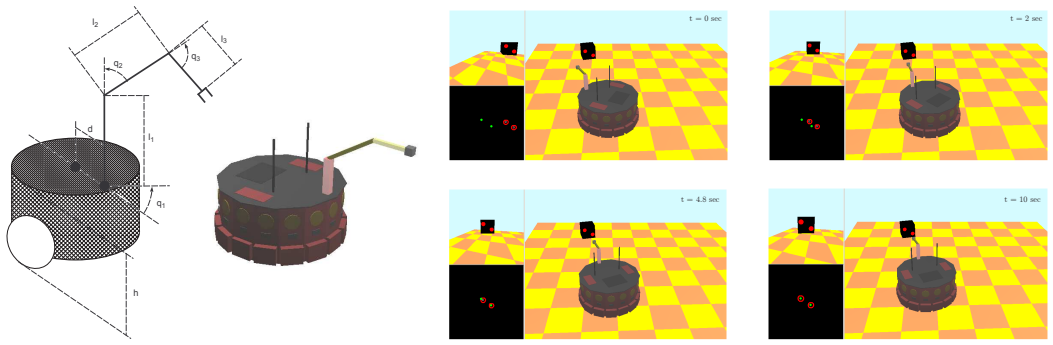


FIGURE II.2: Left: schematics of a NMM with a 3R spatial manipulator. Right: snapshots of a simulation involving visual-based navigation for a NMM

the NMM unified model to find general control laws within the so-called *image-based visual servoing* framework [Chaumette and Hutchinson, 2006, Chaumette and Hutchinson, 2007], see Fig. II.2(b).

This control design was further extended to the case of modeling and control of a mobile platform actuated by *four independent steering wheels* and with *movable (inward/outward) legs* [C12, C13, C14], see Figs. II.3(a–d). This mobile platform was developed for the two-arm humanoid robot Justin of the German Institute of Robotics and Mechatronics (DLR). The purpose of this platform was to provide full planar mobility to the Justin robot so as to allow the execution of complex dual handed manipulation tasks with increased (possibly infinite) workspace capabilities, and the ability to reconfigure online the platform shape (footprint) in order to adapt to different environments and stability requirements. The video showing the Justin robot and its mobile platform in action won the **Best video award** at the International Conference on Robotics and Automation (ICRA) 2009 [C14], and is available at `http://youtu.be/ZPwpGpMoAxs`.

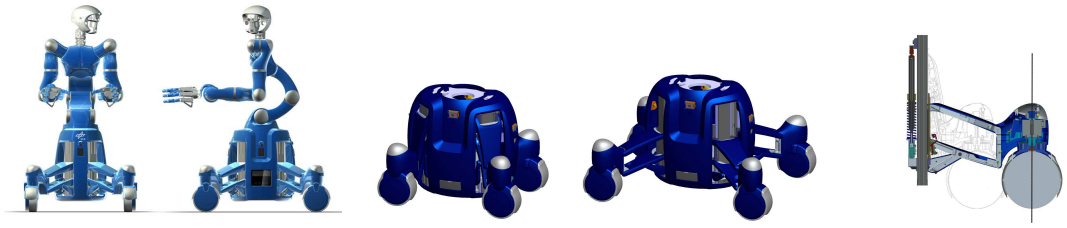


FIGURE II.3: Snapshots of the Justin robot on its 4-steering wheel mobile platform. The platform legs can also be extended/retracted in order to increase/decrease the footprint of the platform

II.2 Robotics and Vision

Broadly speaking, robotic vision lies at the intersection of two complementary fields: computer vision and image processing, and motion control from visual feedback. In fact, once the relevant visual information is extracted from the scene, the problem of exploiting it so as to reach a location, or grasp an item, still remains to be tackled. In the context of Robotic Vision, I focused on Structure from Motion (SfM) strategies for recovering online the unobservable 3D structure of the scene, on visual-based navigation schemes for WMRs and NMMs, and on visual/force assembly tasks for the KUKA/DLR LWR 7-dof manipulator.

FIGURE II.4: Snapshots of simulations involving recovering of 3D structure from motion for point features, arbitrary planar objects, and spherical objects

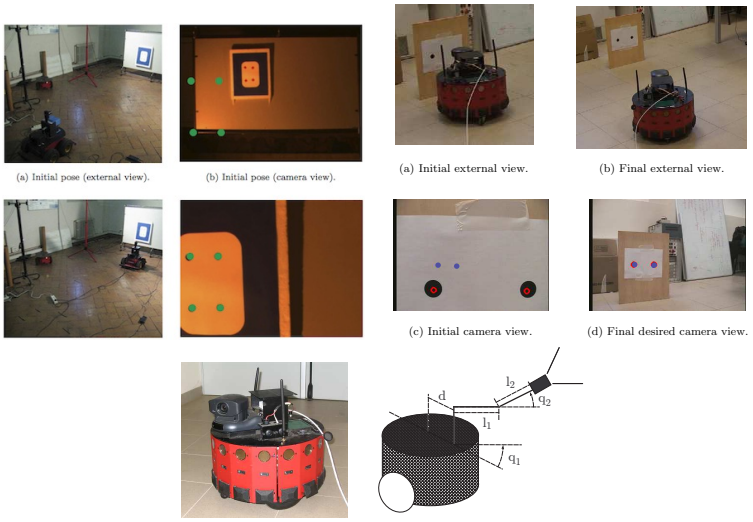


FIGURE II.5: Top: Visual navigation of a mobile robot exploiting the internal 3D estimation algorithm to recover the missing depths of the feature points. Bottom: A mobile robot equipped with a pan-tilt camera (seen a a 2-dof manipulator) performing visual-navigation tasks

II.2.1 3D Structure Estimation and WMR Visual Navigation

The problem of recovering the scene 3D structure from the motion of a robotic system equipped with a camera onboard has been addressed in [C8, C9, J3], by proposing a nonlinear estimation framework able to recover the missing 3D information during the robot/camera motion. The analysis gave an explicit characterization of the *observability conditions* for the estimation task, that is, the set of camera motions actually allowing recovery of the 3D structure. The theory was initially developed for feature points on the image plane (Fig. II.4(a)), and then extended to the more general case of image moments of arbitrary shaped objects (Fig. II.4(b)). Being able to recover the scene structure online, one can then devise control laws based on pure visual feedback (and internal 3D structure estimation) by following the *Image-based Visual Servoing* framework. This allowed to control the motion of a robotic system when performing some navigation task such as, e.g., reaching a goal location. In [J3, C10], I presented an experimental implementation of these controllers on a mobile robots with a fixed camera onboard (Fig. II.5(a–b)), and a mobile robot equipped with a pan-tilt camera unit (Fig. II.5(c)).

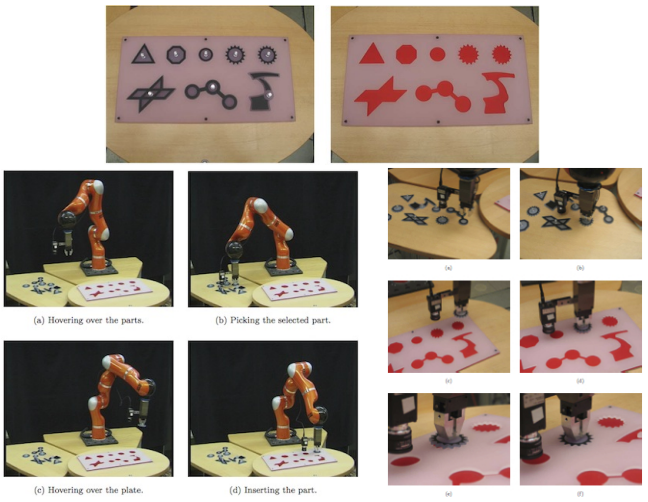


FIGURE II.6: Details of the automatic visual insertion strategy for the 7-dof LWR robot of KUKA/DLR. Top: snapshots of the 8 parts and plate used for the assembly task. Left: an illustration of the assembly procedure. Right: details on the insertion phase with the robot being controlled with impedance control schemes

II.2.2 Automatic Assembly based on Force/Visual Feedback

These results on visual estimation and control of mobile robots were also exploited in [C11] for a more industrial-oriented application developed together with KUKA GmbH during my first visit at DLR, Germany, in 2007. The goal of the application was to combine visual and force-torque feedback for automatic assembly of complex planar parts with a lightweight DLR/KUKA 7-DOF LWR robot manipulator. Figure II.6(a) shows the planar parts and the plate used for the assembly task. Note that the tolerance of the parts w.r.t. the corresponding holes in the plate was less than 1 mm, thus making the insertion phase not a trivial task. Figure II.6(b) shows four snapshots taken during the execution of the insertion sequence. The robot, equipped with a camera on its end-effector, first looks for a part on the table, then moves to the plate and finds the location of the corresponding hole, then approaches the hole by using an image-based visual serving law, and finally inserts the part into the hole exploiting impedance-control laws. Figure II.6(c) shows additional details of the picking and insertion phases. In this context, I took advantage of my previous results on image-based visual servoing and 3D estimation techniques, to achieve fine positioning of the robot manipulator during the approaching phase to the parts/holes. This application was shown live in several fairs (ICRA2007, Automatica 2008, 2010, RSS 2008) and is one of the top demos in the KUKA/DLR portfolio for the light-weight robot.

Finally, besides these theoretical and more applied results, in 2008 I also received the "**Award for Best Italian PhD Thesis in Automatic Control**" because of my contributions to visual and motion control of fixed/mobile robots.

II.3 VR applications

Apart from modeling and control of autonomous robots, I also worked on several aspects of Human-Machine Interaction during the end of my PhD and my early PostDoc period. In particular, I have been responsible for the motion control of the CyberWalk platform, a large size 2D omni-directional platform that allows unconstrained locomotion possibilities to a walking user for VR exploration, see Fig. II.7.

FIGURE II.7: Left: architecture of the CyberWalk platform. Center: a snapshot of the CyberWalk platform during its construction. Right: a picture of a user walking on the platform

The CyberWalk platform was the outcome of the STREP project "CyberWalk" funded by the European Union and resulted in the construction of the largest 2D locomotion platform in the world: a human user, wearing a Head Mounted Display (HMD), was allowed to freely walk on the platform planar surface while a controller was compensating for his/her intentional motion, so as to keep her/him always close to the platform center while limiting the perceptual effects due to actuation commands. In [J13, C3, C6], I studied the modeling and control problem for an earlier prototype of the CyberWalk platform, the "ball-array" CyberCarpet. This prototype had a different actuation design which resulted in a nonholonomic behavior for the combined platform/user system. Later on, I addressed in [C17, J4, J7] the modeling and control problem for the final CyberWalk platform where I presented both a full experimental evaluation of the platform/controller behavior with walking users, and a psychophysical evaluation of the human perceptual perspective during a walking task. The CyberWalk project was a world-wide recognized success as attested by the many press releases it received, for instance on BBC News, Focus Online and National Geographic.

Part II

Summary of Scientific Achievements

Chapter III

Introduction

ROBOTICS is rapidly entering a new era where robots will be part of our daily life by changing the way we behave as well as many aspects of our society, similarly to what has happened with the digital revolution of early 2000. As a measure of this fast pace, it is worth considering that, according to official statistics, the total number of robots populating the world has grown from 6.5 millions in 2007 up to 18.0 millions in 2011 [Robotics2020, 2014]. In the near future, robots are expected to perform various complex tasks in a more and more autonomous way, relieving human beings from dull or boring duties, serve as human companions in households, perform a variety of service tasks such as housekeeping, baby-sitting, elderly assistance, and surveillance and security maintenance. In this respect, the official *Strategic Research Agenda (SRA) for Robotics in Europe* [Robotics2020, 2014], issued by the European Robotics Technology Platform (EUROP, `www.robotics-platform.eu`), foresees that:

> Robotics is likely to be a pivotal element when targeting social challenges such as the ageing population, the creation and retention of high-quality, socially inclusive employment, external and internal security threats and dealing with economic disparity arising from the recent and future EU enlargements. Therefore, European society stands to benefit greatly from a leadership position of its robotics industry.

Similarly, outside Europe, one can find the recent *"A Roadmap for U.S. Robotics"* [Robotics in the United States of America, 2013] stating:

> Robotics technology holds the potential to transform the future of the country and is expected to become as ubiquitous over the next decades as computer technology is today.

A necessary requirement towards the success of this vision lies in the flexibility and dexterity of the envisaged robotic structures: to be successful, future robots should be able to perform a *variety* of tasks and not be designed for a *single* task, i.e., they must be *flexible*. A good way to summarize this view is perhaps the following: "Present-day robots are made for the purpose of repeating several tasks thousands of times. Future robots, on the other hand, will have to perform thousands of tasks several times" [World Robotics, 2011].
Additionally, future robotic systems will necessarily need the ability to interface themselves with other

artificial/living entities for reasons ranging from mere co-existence, to direct co-operation, or to full exploitation. An example is the growing interest towards topics related to *Human-Robot Interaction* [Thrun, 2004, Goodrich and Schultz, 2007], i.e., roughly speaking, how to implement a natural, seamless, safe and effective co-operation between robots and humans when performing common tasks. Another example is the emerging field of *cloud robotics* [Waibel et al., 2011] which aims at studying ways to exploit the Internet (i.e., the *cloud*) to let the robots share information across themselves and to dig into the huge database of online-stored data.

Along similar lines, but in a more general context, large attention of the public and funding agencies, as well as large research efforts have been devoted towards the to so-called *"Cyber-Physical Systems"* (CPS). Quoting Wikipedia [Wikipedia, 2013], a cyber-physical system (CPS) is "a system of collaborating computational elements controlling physical entities". CPS envisage the use of a network of computational and physical resources spread over large areas and able to, e.g., collectively monitor the environment and act upon it. Examples of CPS are: micro- and nano-scale physical materials, cooperating medical devices and systems, powergrids, intelligent ground/marine/air vehicles and transportation systems[1]. Within robotics, instances of CPS can be found in all those contexts involving coordination of multiple (mobile) robots for environmental monitoring, patrolling, exploration, and data collection.

The future of robotics possesses, therefore, a great and widely recognized potential. At the same time, the challenges to turn this potential into reality are large and still very far to be solved. Just to mention a few, future robots will most likely need to operate in highly unstructured environments for long periods, to gather, fuse and interpret information from different sources (different sensors and/or spatially/temporally separated), to safely co-exist with humans and effectively co-operate with them, to finally possess a large degree of autonomy for taking complex decisions in dynamic situations.

With respect to these challenges and difficulties, the approach taken in this Thesis can be briefly summarized as follows: on the one hand, empower robots with a *large degree of autonomy* for allowing them to effectively operate in non-trivial environments (e.g., outside completely defined factory settings). On the other hand, include *human operators* in the loop for having them in (partial and bilateral) control of some aspects of the overall robot behavior.
One could perhaps argue that these two goals are somehow in conflict since higher robot autonomy should imply lower (or absence of) human intervention. However, we believe the research direction undertaken in this Thesis is well motivated since:

- as explained, despite the many advancements in robot autonomy, complex and high-level cognitive-based decisions are still out of reach. In most applications involving tasks in unstructured environments, uncertainty, and interaction with the physical word, human assistance is still necessary, and will probably be for the next decades. On the other hand, robots are extremely capable at autonomously executing specific and repetitive tasks, with great speed and precision, and at operating in dangerous/remote environments, while humans possess unmatched cognitive capabilities and world awareness which allow them to take complex and quick decisions;

- the cooperation between humans and robots is often an implicit constraint of the robotic task itself. Consider for instance the case of assistive robots supporting injured patients during their physical recovery, or human augmentation devices. As we will see in the next Chapters, another example of an application in which the interaction of the robot with the user is the primary objective are

[1] Good overviews can be found in [Lee, 2008, Shi et al., 2011].

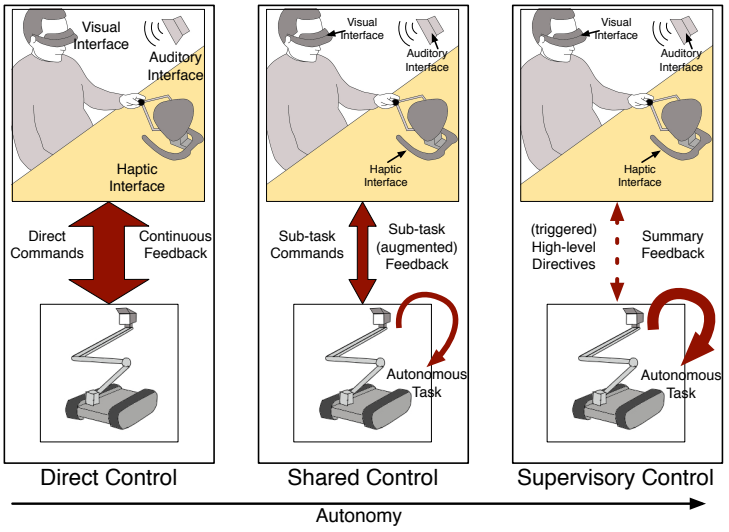


FIGURE III.1: Different control architectures for shared execution of tasks. Direct control (left), shared control (center), supervisory control (right)

motion simulators, i.e., robotic platforms designed for providing motion cues to the user in order to simulate the motion of a vehicle for training or teleoperation purposes;

- finally, safety regulations can require the presence at all times of a person in charge of supervising and, if necessary, take direct control of the robotic workers. For example, this is a common requirement in applications with UAVs, in particular when flying in civil airspace such as over urban or populated areas.

Summarizing, the general goal of the various contributions presented in this Thesis is to provide an effective way for a human operator to be in 'partial' control of semi-autonomous single or multiple mobile robots (in particular quadrotor UAVs) with, additionally, the possibility for the operator to receive suitable feedback cues informative of the behavior of the remote robots. A common theme will be the issue of where to "draw the line" between robot autonomy and human intervention. Obviously, there is no general answer, and any design choice depends on the particular task at hand and/or on the technological/algorithmic possibilities of the robotic system under consideration. A broad characterisation of the several existing possibilities is given in Fig. III.1 where three classical paradigms for teleoperating a remote robot are shown: *direct control* (almost no robot autonomy), *supervisory control* (almost full robot autonomy), and *shared control* (a mix between the two) [Niemeyer et al., 2008]. With this classification in mind, most of the following contributions can then be considered as instances of the *shared control* paradigm that will be declined throughout the Thesis in a number of different contexts and application scenarios (for instance, by considering single or multiple mobile robots, or different kinds of sensory cues for the human operator). The next Section provides a more detailed overview of the main contributions described in the following Chapters.

III.1 Overview of the Main Contributions

Figure III.2 provides a quick and, in our opinion, effective overview of the several topics addressed in the following Chapters, that is, the multifaceted problem of implementing shared control architectures

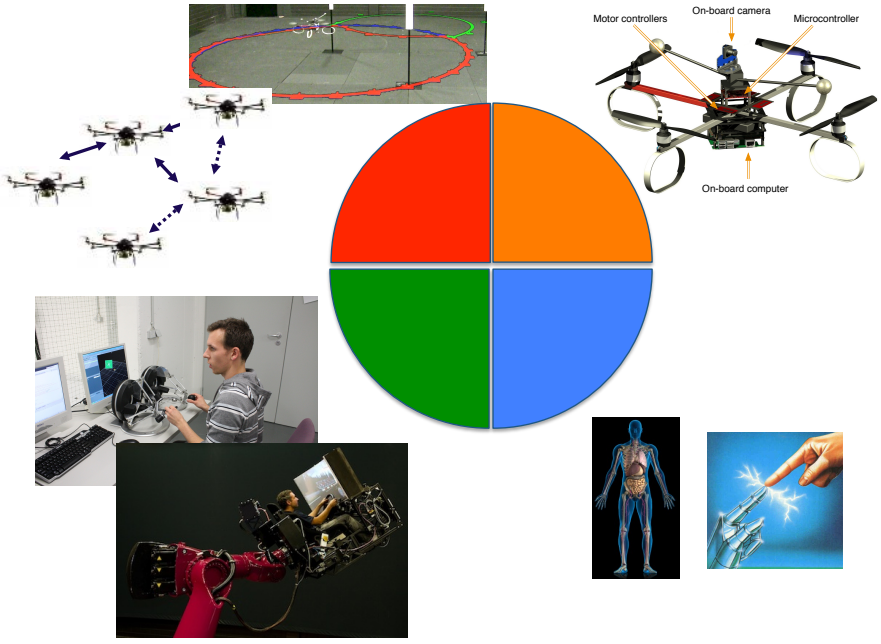


FIGURE III.2: A pictorial overview of the several topics addressed in the Thesis: multi-robot coordination, bilateral teleoperation, flight control for quadrotor UAVs, motion planning for mobile robots, users' evaluations

for *single* and *multiple* mobile robots (with a particular focus on quadrotor UAVs) with the goal of providing the human operator with the possibility of (*i*) easily controlling the motion of single/multiple robots for performing remote tasks in a semi-autonomous way and, at the same time, (*ii*) receiving a multi-sensory feedback informative of the task execution performance and of the remote environment. This research plan is clearly quite general and spans challenges from very diverse fields (e.g., teleoperation, motion/flight control of single and multiple robots, state estimation for single and multiple robots, user evaluations). In this respect, the decomposition given by the next Chapters attempts to group (to the best possible extent) the individual contributions topic-wise in order to provide the reader with a clear and comprehensive view of the several activities reported in the Thesis. In particular:

- Chapter IV will discuss a shared control architecture for single mobile robots with a special focus on *visual/vestibular* sensory feedback for the human operator. Here, the term 'vestibular feedback' refers to the possibility of stimulating the human sense of self-motion via a motion platform (or motion simulator) that carries the operator and physically displaces her/him through space. While classical applications of motion simulators involve training of airline pilots, we will revisit their use as a means to provide vestibular (self-motion) cues to a human operator in control of remote mobile robot, and we will discuss a user evaluation obtained by remotely piloting a quadrotor UAV;

- Chapter V will instead focus on the more classical case of shared control for single/multiple mobile robots (quadrotor UAVs) with *visual/force* sensory feedback for the operator. The Chapter will first address the case of shared control for a single mobile robot where an extension able to take into account the future consequences of the (temporally and spatially) local commands of the user will be detailed. Then, the case of shared control of *multiple* mobile robots will be considered, by proposing a number of different possibilities for resolving the large redundancy between the few operator commands and the many DOFs of the robot group, together with a suitable force feedback

for the operator meant to inform about the status of the group. A user evaluation of some of the proposed architectures will also be discussed;

- Chapter VI will then present contributions at a more algorithmic level, by discussing some of the technical challenges (and results) faced when dealing with the problem of controlling the formation of multiple mobile robots with onboard sensing/communication capabilities. This Chapter can then be seen as a technical complement to some of the material reported in Chapter V;
- Chapter VII will illustrate some advancements in the mechanical design and control of a novel overactuated quadrotor UAV (thus, by spanning topics such as dynamical modeling of flying robots, mechatronic prototyping and flight testing). The purpose of this novel quadrotor design is to overcome the limitations of standard quadrotors (in particular, their underactuation) in order to achieve full controllability of the quadrotor 6-DOF pose in space (and thus facilitate any task involving navigation and/or physical interaction with the environment);
- Chapter VIII will consider the issue of designing (and implementing) *active* perception schemes for robots equipped with onboard cameras (as the UAVs employed in the previous Chapters) with the aim of obtaining an accurate and 'fast' estimation of the surrounding scene thanks to an *online* optimization of the camera motion. A number of active strategies will be discussed for several geometric primitives (points, 3D objects, planar scenes) and setups (manipulator arms with eye-in-hand cameras and quadrotor UAVs with onboard camera/IMU). Furthermore, the issue of coupling an active perception strategy with the concurrent execution of a visual task will also be considered;
- Chapter IX will finally conclude the Thesis, discuss open points and draw some future perspectives worth of investigation.

Chapter IV

Shared Control with Visual-Vestibular Feedback

This Chapter will discuss an example of shared control for mobile robots with *visual-vestibular* feedback for the human operator, while the *visual-force* feeedback case will be addressed in the next Chapter V. The goal, in both cases, is to allow a human operator to control the motion of a mobile robot while receiving a feedback informative of its motion status and/or of some environmental conditions (e.g., presence of obstacles or vicinity to goal locations). However, contrarily to the (more classical) visual-

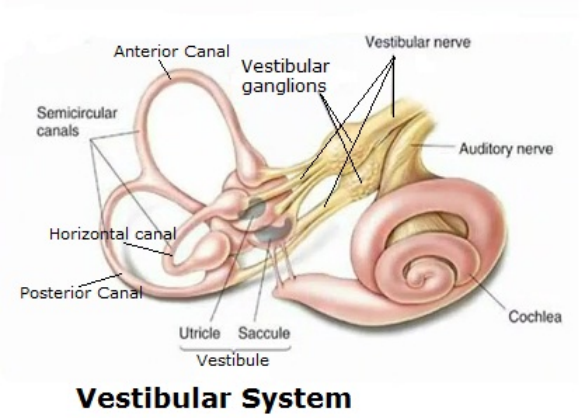


FIGURE IV.1: A pictorial representation of the human vestibular system in the inner ear.

force case, use of visual-vestibular feedback represents a much less common choice of sensory feedback for interfacing a human operator with a remote mobile robot. Specifically, the design choice here is to provide the user with a *direct feeling* of the robot motion by stimulating the "motion receptors" present in the human's inner hear (the so-called vestibular system, see Fig. IV.1). This is achieved by placing the human user on an actuated mechanical device (a *motion simulator*) which physically displaces the user for producing the sought motion perception. The use of actuated mechanical devices for providing vestibular feedback has a long history in the field of motion simulation for training purposes (e.g., in the aviation domain [Burki-Cohen et al., 2001]). The next sections will discuss how this idea can be extended to the case of *teleoperation* of remote mobile robots.

IV.1 Overview

The idea behind the use of visual-vestibular feedback for shared control of remote mobile robots is depicted in Fig. IV.2. In a nutshell, the human user is located on an actuated mechanical device (the

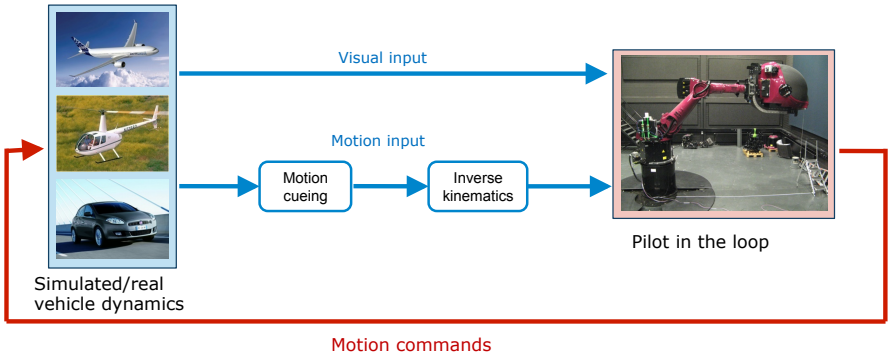


FIGURE IV.2: A schematic representation of the visual-vestibular shared control architecture.

motion simulator) and acts on an input device (e.g., a joystick) for giving motion commands to a simulated or real mobile robot. An onboard view from the robot (again, simulated or acquired by onboard cameras) is then displayed to the human user on a screen mounted on the robot or on a Head Mounted Display (HMD) worn by the user. At the same time, the actual robot motion is sensed, processed (via the so-called *motion cueing* and *inverse kinematics* blocks in the scheme of Fig. IV.2) and replayed on the motion simulator, thus closing the loop. The main challenge, here, lies in the *motion cueing* and *inverse kinematics* blocks since some pre-filtering is usually required to make the 'ideal' vehicle motion compatible with the limited workspace of the chosen motion platform. Indeed, the motion range of a platform fixed to the ground is in general too limited to reproduce 1:1 *any* vehicle trajectory such as, e.g., a car or an airplane constantly accelerating over time. To this end, a *Motion Cueing* block is exploited for transforming the input motion profile into a Cartesian trajectory compatible with the platform workspace, but still inducing a *realistic* motion perception onto the user. A classical example is the well-known tilt-coordination algorithm [Groen and Bles, 2004] that exploits the gravity vector in the user's body frame to simulate presence of a sustained linear acceleration. This filtered Cartesian trajectory must then be fed to an *Inverse Kinematics* algorithm for its actual realization on the motion simulator. Here, the difficulty lies in the fact that the classical structure of a motion cueing algorithm does not allow to take explicitly into account all the motion simulator constraints expressed at the joint level. Moreover, the filtered trajectory is completely arbitrary (in terms of geometric path) and unknown in advance — it eventually depends on the unpredictable inputs of the user to the vehicle. Hence, the sought inverse kinematics must be able to realize at best and in real-time a Cartesian trajectory that 1) is geometrically unpredictable and unknown in advance, and 2) may violate the simulator mechanical constraints (e.g., max. acc or max. vel of its joints) over time.

Historically, use of motion simulators dates back to 1965 where *Gough-Stewart* platforms (also known as *Stewart platform* or *hexapods* [Stewart, 1965]) started to be employed in the field of aeronautics at NASA in the 70's [Conrad and Schmidt, 1971, Dieudonne et al., 1972]. Indeed, a motion simulator clearly allows to reproduce unexpected and risky situations in a controlled, safe and repeatable manner. A growing application field is also the entertainment industry, where robotic motion platforms are used as roller-coasters in amusement parks. Motion simulators are also important tools in the automotive industry,

for testing new car dynamics beforehand, or for evaluating novel driving assistive technologies [Siuikat, 2005, Robuffo Giordano et al., 2010]. Other applications involve also simulation of helicopters [Beykirch et al., 2007], motorcycles [Ferrazzin et al., 1999, Avizzano et al., 2000, Nehaoua et al., 2011], sailing boats [Avizzano et al., 2010] and even space flights [Heindl et al., 2006].

As explained, most existing mechanical designs are based on fully actuated hexapods (i.e., Stewart platforms, see Fig. IV.3(a)). This design indeed grants high motion bandwidth, mechanical robustness and (relatively) low price. However, hexapods also suffer from several shortcomings, such as their limited workspace, the impossibility to achieve large linear and angular displacements and rates because of the the closed chain nature of their actuation system, and the presence of cross-couplings between the actuated axes (e.g., presence of spurious rotations when commanding a translational motion). As a possible alternative, the idea of exploiting industrial robot manipulators as motion platforms has drawn some attention in the scientific community [Heindl et al., 2005, Bellmann et al., 2007, Pollini et al., 2008]. Indeed, a serial 6-DOF industrial manipulator offers higher dexterity, larger motion envelope, the possibility to realize any end-effector posture within the workspace, and the ability to displace heavy loads (up to 1000 [kg]) with large accelerations and velocities (see Fig. IV.3(b)). One can then attach a cabin carrying the user to the robot end-effector, and take advantage of the large dexterity envelope to move the cabin along complex coordinated trajectories and attitudes (e.g., even placing the cabin upside down).

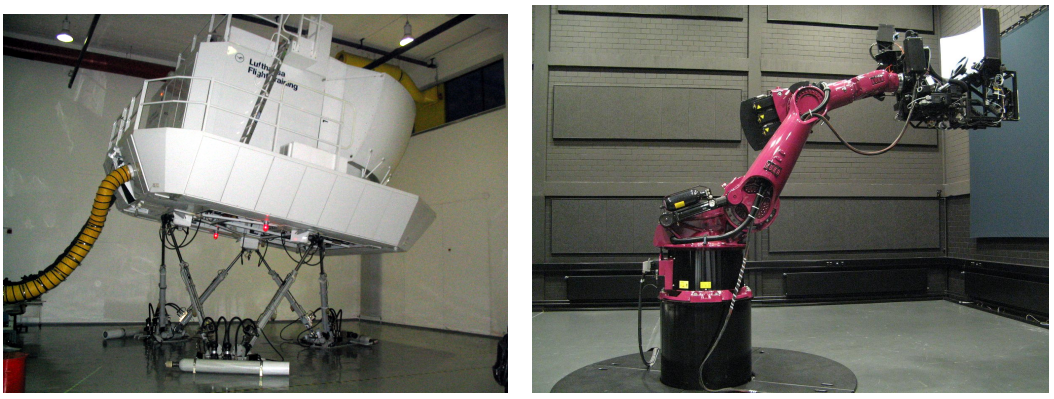


FIGURE IV.3: Left: a Stewart platform, also known as hexapod, often exploited as motion simulator for airline pilot training. Right: the CyberMotion simulator, a 6-DOF industrial manipulator arm with a cabin mounted on its end-effector.

In the following, this latter possibility will be considered for implementing a motion simulator for visual-vestibular feedback. In particular, we will discuss how to exploit the *CyberMotion Simulator* developed by the the Max Planck Institute for Biological Cybernetics in Tübingen, Germany (Fig. IV.3(b)) as a possible robotic platform for providing motion cues. The CyberMotion simulator consists of a 6-DOF anthropomorphic robot arm based on the commercial KUKA Robocoaster [Robocoaster,] and adapted and customized to be used as a tool for motion simulation. The main algorithmic challenges (briefly discussed in the following sections) needed to turn it into a versatile motion simulation are: (*i*) the design of a suitable *inverse kinematics* scheme able to cope with an unpredictable and arbitrary desired cabin motion, generated online as a function of the (unpredictable) user inputs to the controlled vehicle, and (*ii*) the design of a *motion cueing* block tailored to the specific motion envelope of a serial manipulator.

IV.2 Motion Cueing Design

IV.2.1 Brief overview on washout filters

As explained, the goal of a motion cueing algorithm is to 'filter' the ideal motion of the controlled vehicle to make it compatible with the limited workspace of a motion platform fixed to the ground, while still inducing (as much as possible) the same *motion perception* onto the user. This is a fundamental but difficult problem that has been studied since the early '70 [Schmidt and Conrad, 1970] until nowadays [Huang and Fu, 2006], and keeps drawing the attention of many researchers. Perhaps the most widespread motion cueing algorithm is the so-called *classical washout filter*, i.e., a combination of washout filters for reproducing high-frequency motions, and tilt-coordination algorithms for reproducing low-frequency motions [Nahon and Reid, 1990, Grant and Reid, 1997].

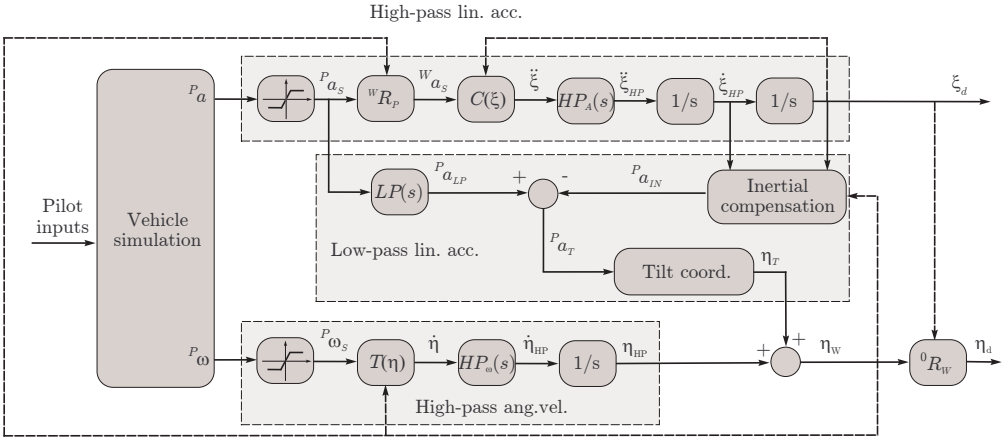


FIGURE IV.4: A schematic representation of motion cueing algorithm implemented on the CyberMotion simulator.

With the help of Fig. IV.4, the classical washout filter can be summarized as follows: the vehicle motion, specified in terms of linear accelerations and angular velocities[1], is split into high-frequency and low-frequency components. The high-frequency component is reproduced by actually moving the motion platform, since this component will in general generate small (thus feasible) displacements. On the other hand, the low-frequency component, comprehensive of sustained linear accelerations, is not achieved by physically accelerating the platform, but by exploiting the local gravity vector as a source of 'persistent' acceleration. Indeed, by properly orienting the gravity vector in the cabin frame, one can reproduce the illusion of persistently accelerating in a given direction. This method, of course, has several limitations: for instance, it cannot reproduce sustained accelerations larger than 1 [g], it introduces rotational motion artifacts during the coordination phase, and it reduces the amount of gravity acceleration perceived by the user because of the tilting. Still, in most cases the classical washout framework represents the only viable option as motion cueing algorithm.

Many variants of this framework have been proposed over the years, especially in the context of flight simulators. Just to mention a few, some attempts have been done in the direction of adaptive algorithms [Parrish et al., 1975, Ariel and Sivan, 1984, Nahon et al., 1992], and optimal control approaches [Sivan et al., 1982, Telban et al., 2002]. A thorough comparison can be found in [Reid and Nahon, 1985, Reid and

[1] These are thought to be the motion states sensed by humans [Greig, 1988].

Nahon, 1986a, Reid and Nahon, 1986b]. Despite the big variety of solutions, it is worth noting that many authors still indicate the classical washout filter as one of the most effective algorithm in terms of design simplicity, easiness of tunability, and human perception fidelity [Nahon and Reid, 1990, Nehaoua et al., 2006b, Nehaoua et al., 2006a].

IV.2.2 Implementation on the CyberMotion simulator

Compared to a Stewart platform, the motion envelope of the CyberMotion Simulator is closer to a cylinder rather than a rectangular box in Cartesian space. For instance, by exploiting the rotation of the first vertical joint, one can obtain large lateral displacements along circular trajectories, considerably larger than any achievable linear trajectory. This motivates to design the high-pass filters for the linear acceleration in cylindrical coordinates, similarly to what proposed for the spherical washout filter [Wentink et al., 2005] implemented in the Desdemona motion simulator developed at TNO [Bles and Groen, 2009]. The idea is to keep linear forward and upwards motions unchanged, and replace linear lateral motions with circular motions (i.e., moving on the surface of a vertical cylinder).

Before proceeding to briefly illustrate the concept, it is useful to define some quantities of interest. With reference to Fig. IV.5, let $\mathcal{F}_0 : \{O; \vec{X}_0, \vec{Y}_0, \vec{Z}_0\}$ be a world reference frame fixed to the robot

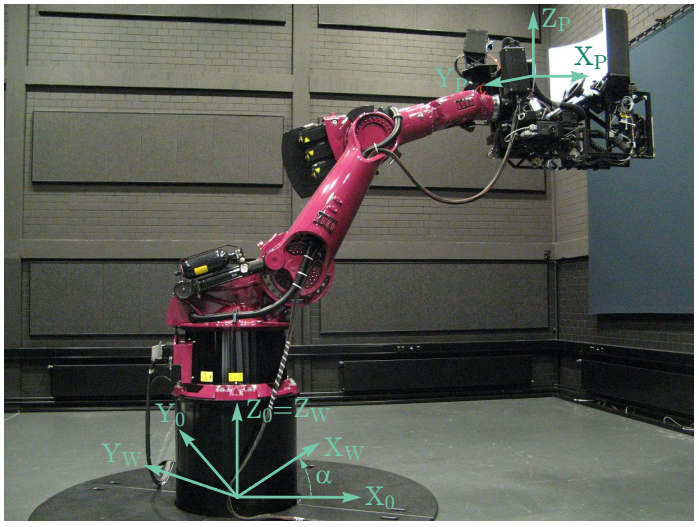


FIGURE IV.5: The CyberMotion simulator and several reference frames of interest.

base, with $\vec{Z}_0$ pointing upwards and $(\vec{X}_0, \vec{Y}_0)$ spanning the horizontal plane. A moving reference frame $\mathcal{F}_P : \{O_P; \vec{X}_P, \vec{Y}_P, \vec{Z}_P\}$ is attached to the the pilot's head (supposed fixed to the cabin) and has its axes aligned with the pilot's forward/left/upward direction, respectively. Let also $p = [x\ y\ z]^T \in \mathbb{R}^3$ represent the coordinates of O_P in $\mathcal{F}_0$. The Cartesian coordinates p can be transformed into cylindrical ones $\xi = [R\ \alpha\ z]$ as

$$\begin{cases} R &= \sqrt{x^2 + y^2} \\ \alpha &= \text{atan2}(y,\, x) \\ z &= z \end{cases},$$

with associated a third moving frame $\mathcal{F}_W : \{O_W; \vec{X}_W, \vec{Y}_W, \vec{Z}_W\}$, denoted as the *washout frame*, with $O_W \equiv O$, $\vec{Z}_W \equiv \vec{Z}_0$, and $\vec{X}_W$ rotated of angle α w.r.t. $\vec{X}_O$ (therefore, axis $\vec{X}_W$ always points towards the current angular position α on the cylinder). Finally, ${}^W R_P$ represents the rotation matrix from frame

$\mathcal{F}_W$ to frame $\mathcal{F}_P$, $\eta = [\rho\ \theta\ \psi]^T \in \mathbb{R}^3$ the usual set of roll-pitch-yaw Euler angles parameterizing ${}^W R_P$, g is the gravity vector, and g the scalar value of the gravity acceleration.

IV.2.2.1 High-pass channel

The input to this filter is ${}^P a$, the linear acceleration of the vehicle expressed in the frame $\mathcal{F}_P$ and without gravity components, i.e., ${}^P a = {}^P f + {}^P g$ where ${}^P f$ is the specific force acting on the vehicle[2]. By following the classical washout framework, ${}^P a$ is first scaled and limited to obtain ${}^P a_S$ which is then expressed into $\mathcal{F}_W$ as ${}^W a_S$. These Cartesian accelerations are then transformed into cylindrical ones as

$$\ddot{\xi} = \begin{bmatrix} \ddot{R} \\ \ddot{\alpha} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \dfrac{1}{R} & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^W a_S = C(\xi) {}^W a_S \tag{IV.1}$$

Linear forward accelerations in $\mathcal{F}_W$ correspond to radial accelerations $\ddot{R}$, and lateral accelerations in $\mathcal{F}_W$ correspond to angular accelerations $\ddot{\alpha} R$.

The accelerations $\ddot{\xi}$ are then high-pass filtered through the transfer function $HP_A(s)$ to yield the high-pass component of the linear motion $\ddot{\xi}_{HP}$, which is subsequently double-integrated into the desired platform displacement $\xi_{HP} = \xi_d$. Many choices are possible for $HP_A(s)$. The typical one is to take

$$HP_A(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \cdot \frac{s}{s + \omega_b},$$

where the natural frequency ω_n, the damping ratio ζ, and the break frequency ω_b are in general different for each component of $\ddot{\xi}$. This choice ensures the *washout* characteristics of the filter, i.e., the fact that, at steady state and for constant inputs, the platform displacement $\xi_{HP} = \ddot{\xi}/s^2$ goes back to the initial position.

The angular high-pass channel (bottom block in Fig. IV.4) is designed as in the classical washout scheme. In short, the input angular velocity ${}^P\omega$ expressed in $\mathcal{F}_P$ is first scaled and limited, then transformed into the corresponding Euler rate $\dot{\eta}$ which is high-pass filtered through $HP_\omega(s)$ to obtain the high-pass component of the angular velocity $\dot{\eta}_{HP}$. This is finally integrated into the corresponding angular displacement η_{HP}. Here, $HP_\omega(s)$ can be chosen as a second-order high-pass filter

$$HP_\omega(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

to ensure the washout effect.

IV.2.2.2 Low-pass channel and tilt coordination

The purpose of this step is to orient ${}^P g$ so as to simulate presence of sustained linear accelerations. This is achieved by matching the low-pass components of ${}^P a$ with the corresponding components of ${}^P g$ through the so-called *tilt coordination* algorithm. This idea can also be extended to compensate for the undesired inertial accelerations due to the choice of working in cylindrical coordinates. Indeed, the benefits of

[2] We recall that the specific force ${}^P f$ is defined as ${}^P a - {}^P g$, so that during free fall (${}^P a = {}^P g$) it is ${}^P f = {}^P g - {}^P g = 0$.

moving along a horizontal circular trajectory instead of a straight line (i.e., increased robot workspace) come at the price of introducing spurious centripetal and Coriolis accelerations during the motion. These disturbances can be largely attenuated by using the gravity vector ${}^{P}g$ to filter them out.

With, again, reference to the scheme in Fig. IV.4: the scaled linear acceleration ${}^{P}a_{S}$ is first low-pass filtered through $LP(s)$ into ${}^{P}a_{LP}$. The filter $LP(s)$ is taken as $LP(s) = 1 - HP_A(s)$ in order to obtain a perfect coordination between the high-pass and low-pass channels. At this point, the standard tilt coordination algorithm would compute the required cabin orientation η_T to match the first two components of ${}^{P}g$ with $-{}^{P}a_{LP}$. By imposing ${}^{P}R_{W}\,{}^{W}g = -{}^{P}a_{LP}$, one obtains

$$\begin{cases} \rho_T &= \arcsin \dfrac{{}^{P}a_{LP_y}}{\mathrm{g}\cos\theta_T} \\ \theta_T &= -\arcsin \dfrac{{}^{P}a_{LP_x}}{\mathrm{g}} \\ \psi_T &= 0 \end{cases} .$$

These angles are typically rate limited to avoid a strong rotational cueing on the user.

The centrifugal and Coriolis accelerations due to the cylindrical motion in $\mathcal{F}_W$ are ${}^{W}a_{IN} = [-R\dot{\theta}^2 \;\; 2\dot{R}\dot{\theta} \;\; 0]^T$. These can be transformed into the cabin frame ${}^{P}a_{IN} = {}^{P}R_{W}\,{}^{W}a_{IN}$, and subtracted from ${}^{P}a_{LP}$ to obtain the final acceleration vector ${}^{P}a_{T} = {}^{P}a_{LP} - {}^{P}a_{IN}$ to be sent to the tilt coordination algorithm. The corresponding η_T is then added to angular displacement η_{HP} from the high-pass angular velocity channel to yield the sought cabin orientation $\eta_W = \eta_T + \eta_{HP}$. The final step is to transform η_W into the corresponding world frame quantity. By recalling the definition of $\mathcal{F}_W$, this can be easily achieved by adding the angular displacement α to the third (yaw) component of η_W, thus obtaining the desired cabin orientation input to the robot inverse kinematics $\eta_d = \eta_W + [0\;0\;\alpha]^T$.

IV.3 Inverse Kinematics Design

Once the desired (and filtered) cabin motion $(\xi_d(t),\, \eta_d(t))$ has been determined by the motion cueing step, one needs to feed it to an Inverse Kinematic block for its actual realization on the robot arm. Like most industrial robot arms, the motion capabilities of the CyberMotion Simulator are subject to several constraints such as limited joint range, velocity and acceleration

$$\begin{cases} \text{a)} \quad \forall i, \forall t \geq 0, \quad q_{i,min} \leq q_i(t) \leq q_{i,max} \\ \text{b)} \quad \forall i, \forall t \geq 0, \quad |\dot{q}_i(t)| \leq \dot{q}_{i,max} \\ \text{c)} \quad \forall i, \forall t \geq 0, \quad |\ddot{q}_i(t)| \leq \ddot{q}_{i,max} \end{cases} . \tag{IV.2}$$

Taking then $r = [\xi^T \; \eta^T]^T \in \mathrm{R}^6$ as the *task variables* to be controlled (with $\xi = [R\;\alpha\;z]$ and $\eta = [\rho\;\theta\;\psi]^T \in \mathbb{R}^3$), one can easily derive the mapping (differential kinematics) between $\dot{q}$ and $\dot{r}$

$$\dot{r} = J(q)\dot{q}, \tag{IV.3}$$

with $J(q)$ being the 6×6 task Jacobian. The inverse kinematics problem can then be (classically) formulated as that of *realizing a given reference task trajectory* $r_d(t)$ *output of the motion cueing algorithm* given the constraints (IV.2) and the differential mapping (IV.3) (with its associated 'singularities'). Since the robot directly accepts joint velocities $\dot{q}$ as inputs, the problem is naturally tackled at the kinematic

FIGURE IV.6: Left: a screenshot of the 3D environment displayed to the user on the onboard projection screen. Right: a bird's eye view of the Monza track

level, by exploiting the classical concepts of *kinematic inversion* and *kinematic control* see, e.g., [Chiacchio et al., 1991].

Being $r_d(t)$ *not* known in advance, one cannot resort to offline methods to modify the path or the associated timing law so as to cope with the constraints. Therefore, one needs an online solution able to realize *online* the best feasible approximation of $r_d(t)$: whenever realization of $r_d(t)$ is compatible with all the constraints, then the robot should track it exactly. For instance, proximity to a joint limit should not degrade the robot motion until strictly necessary (as opposite to the effect of many potential field methods). On the other hand, whenever a constraint is violated, the robot should move as best as it can, i.e., by minimizing the Euclidean norm of the tracking error $\|e(t)\| = \|r(t) - r_d(t)\|$.

The approach taken for coping with these issues was to resort to a (suitable) combination of Task Priority (TP) inversion scheme [Chiaverini, 1997] (flexibility in dealing with singularities), online scaling of the commanded joint velocity $\dot{q}$ for coping (online) with maximum velocity and acceleration constraints at the joint level, and suitable exploitation of the theory of bang-bang optimal control [Lewis, 1995] for staying within the limited joint range by stopping "at the very last moment" any joint motion violating the allowed range.

IV.4 Evaluation of the Motion Control Framework

The proposed Motion Cueing and Inverse Kinematics algorithms have been first validated in experiments involving a human user piloting a *simulated* race car. This step has allowed to verify the overall soundness of the approach and to perform a proper tuning of the various parameters involved in the algorithms. Then, the same framework has been ported to the case of shared control of a real flying robot (a quadrotor UAV) piloted by a human user with a visual-vestibular feedback informative of the motion status of the UAV. In this case, a detailed study involving a performance evaluation on 12 test subjects has also been performed, with the aim of assessing the possible benefits of the proposed visual-vestibular feedback for facilitating remote control of a flying robot.

IV.4.1 Piloting a simulated race car

Figure IV.6(a–c) show a picture of the setup with the human pilot performing a maneuver (Fig. IV.6(a)), the (virtual) visualization of the car cockpit displayed to the user on the onboard screen (Fig. IV.6(b)), and a top view of the track chosen for the experiment (Fig. IV.6(c) — the virtual track of Monza [Monza racetrack,], the Italian official track of the Formula 1 world championship). The complex and sudden motions experienced during this lap clearly constitute a solid benchmark for the proposed approach. The

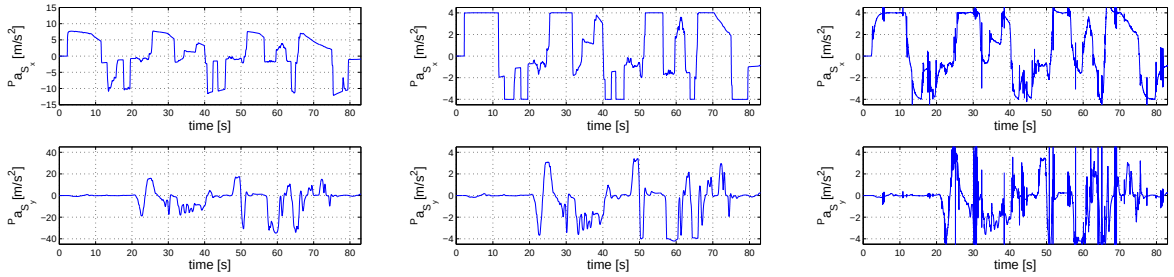


FIGURE IV.7: Top: the forward (top) and lateral (bottom) linear accelerations ${}^{P}a$ of the car during the lap. Middle: the scaled accelerations ${}^{P}a_{S}$. Bottom: the actual specific force felt on the cabin as a consequence of the robot motion. Note the substantial agreement between plots (b) and (c). This confirms the good performance of our whole architecture, i.e., the motion cueing algorithm combined with the inverse kinematics of *Part I*

whole experiment can also be appreciated in the online video at http://youtu.be/oMLarxR-q08 where an external and onboard (camera-car) viewpoints are shown.

Figures IV.7(a–c) show the forward (top plots) and lateral (bottom plots) accelerations during the whole lap. In particular, Fig. IV.7(a) depicts the accelerations ${}^{P}a$, direct output of the car dynamical model. One can notice how the car can achieve quite large accelerations: the peak values are 8 [m/s^2] as forward acceleration, -12 [m/s^2] during braking, and about 34 [m/s^2] during the roughest turns. Of course, no tilt coordination algorithm could reproduce such motions. After the scaling and saturation, we obtain the profiles shown in Fig. IV.7(b) which are more feasible for the CyberMotion simulator, and must be compared with Fig. IV.7(c) where the actual specific force of the cabin is reported. Apart from some unavoidable noise, it is nevertheless possible to notice a substantial agreement among the two plots, thus confirming the good performance of our simulator also in reproducing such complex motions.

IV.4.2 Piloting a real flying robot

This experiment extends the results of the previous Section by considering human subjects in control of a *real* remote flying robot with visual-vestibular feedback. The goal was to assess the effectiveness of different combinations of this feedback in improving the performance during the flying task execution.

IV.4.2.1 Description of the Experiment

As controlled robot, the quadrotor UAV shown in Fig. IV.8 was employed: a small-scale commercial quadrotor, the *MikroKopter L4-ME* model made by the company HiSystems GmbH, equipped with an onboard camera. This small-scale quadrotor is meant to represent the qualitative behavior of a generic rotorcraft vehicle, e.g., possibility to hover or to perform sideways movements.

As for the system architecture, Fig. IV.9 shows the experimental setup: starting from the left, the quadrotor receives through a wireless link its 6-DOFs pose from an external motion capture system (Vicon). This is both used internally for implementing a collision avoidance algorithm, and forwarded via the wireless link to the CyberMotion Simulator that uses this data for determining the motion feedback provided to the operator. The quadrotor is also connected to a Visualization PC to which it

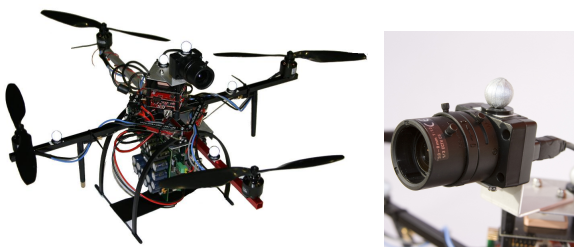


FIGURE IV.8: The UAV used in our experiments instrumented with reflective markers to be tracked by our Vicon system

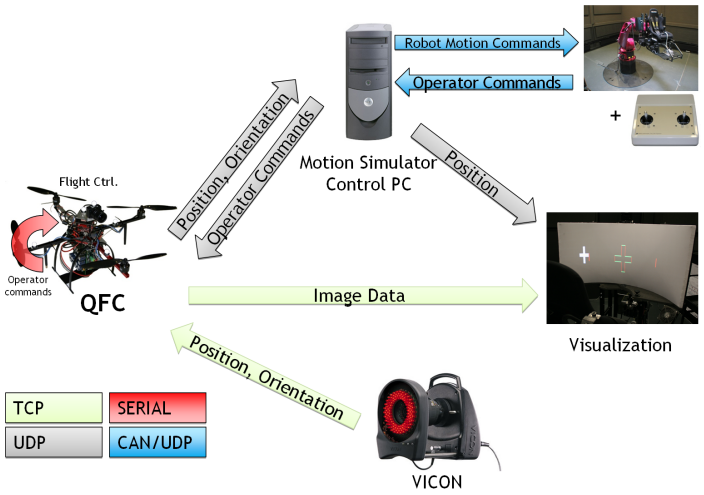


FIGURE IV.9: Network diagram of the experiment setup

sends the onboard camera images at a rate between 20 and 25 fps. These are then displayed to the operator on the screen mounted on the motion simulator cabin.

The control task was a planar hovering task in which the test subjects had to regulate the quadcopter position from a starting location $P_0 = (0,\ 0,\ -2.5)$ m to a desired goal location $P_1 = (2.5,\ -2.5,\ -2.5)$ m in the world frame E (Fig. IV.10). The task was considered completed when they could stay within a box

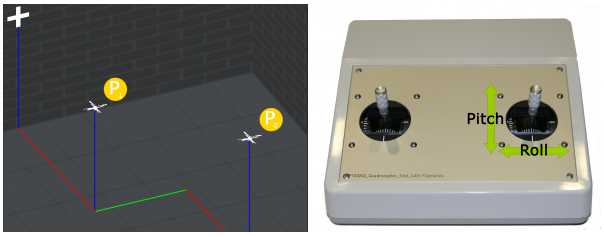


FIGURE IV.10: Overview of the starting position P_0 and the target position P_1 in the simulated tracking lab

B centered at P_1 with edges of 0.03 m for 3 s. In order to simplify the control task, the available quadcopter inputs $(\tau,\ \phi,\ \theta,\ \dot{\psi})$ (the total thrust, roll, pitch, and yaw angles) were reduced to the pair $(\phi,\ \theta)$ roll/pitch commands by implementing on the quadrotor a standard height and yaw controller. Therefore, the test subjects could only control the planar forward and lateral accelerations of the quadcopter by acting on the right stick of their input device, see Fig. IV.10(b).

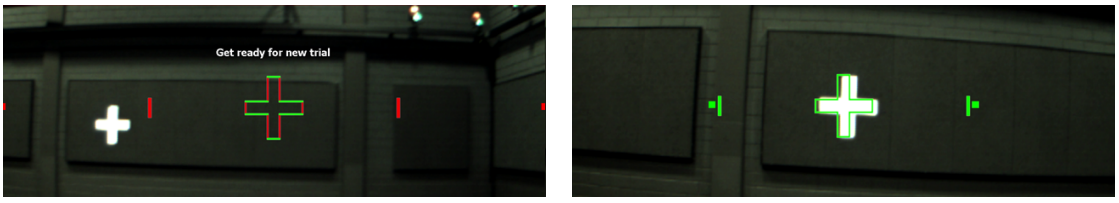


FIGURE IV.11: Camera views relative the the starting (Fig. (a)) and goal (Fig. (b)) locations in the simulated environment. Figures (c) and (d) show the same views from the real quadcopter camera

Figures IV.11 illustrate the visual feedback given to the operator during each trial. A white vertical cross on the facing wall in the experimental hall acts as external visual landmark to be compared with a synthetic wireframe cross superimposed on the camera images. This wireframe cross is chosen so that, when at P_1, it perfectly coincides with the contour of the white cross on the wall (Fig. IV.11(b)). To further help in judging the forward distance to the target (the depth), another visual cue was added. Two dots, one on the right side and one on the left side of the camera image, were moved along a horizontal line passing through the image center. The horizontal position of the right dot was proportional to the relative depth between the quadcopter and the target location, so that the smaller the depth, the closer the dot to the image center. The left dot was moved symmetrically from the left side of the image. Two small vertical bars indicated the required location of the two dots when at P_1 and were used by the subjects as a direct measure of the depth error from the target location. Finally, all these artificial cues had a red color when outside of the box B, and a green color when inside B. This color cue was useful for judging when the quadcopter entered the target box.

As for the vestibular feedback, three different conditions were tested: (*i*) normal motion (*grav*), (*ii*) artificial motion (*aff*), and (*iii*) no motion (*vis*). In the condition *grav*, the CyberMotion simulator was actuated so as to reproduce the forward and lateral components of gravity in the body frame B as a consequence of the quadcopter rotation in the world frame. In this way, the subjects could use this motion cue to infer the quadcopter orientation (and thus its world acceleration) besides what they could visually perceive from the camera images.

In condition *aff*, the CyberMotion simulator was used to reproduce an artificial acceleration proportional to the Cartesian vector from the current position to the target location P_1. As opposite to the previous case, this motion cue was meant to inform the subjects on where to steer in order to complete the task rather than on their current motion status.

Finally, in the last condition *vis*, the subjects fulfilled the task by relying on the *sole* visual feedback.

IV.4.2.2 Results

The performance of the subjects in remotely piloting the UAV was evaluated by considering the average total time needed to complete the task and the average control effort (roll and pitch commands) sent to the UAV during the trials. The differences in performance were analyzed by resorting to the standard statistical tools (mixed model ANOVA [McLean et al., 1991]). The quantity $\sqrt{s_\phi^2(t) + s_\theta^2(t)}$ averaged over the whole trial was taken as a measure of the control effort spent by the subjects.

Table IV.2 and Fig. IV.12(a) report the average time over all trials and subjects for the different conditions, while Table IV.1 and Fig. IV.12(b) show the results for the completion time.

	grav	*aff*	*vis*	*overall*
simulation	0.08498	0.1093	0.1096	0.1013
real	0.1306	0.1474	0.1509	0.1428

TABLE IV.1: Average control effort over all trials and subjects in the different conditions

	grav	*aff*	*vis*	*overall*
simulation	60.384	66.65	44.75	56.94
real	67.67	71.51	66.68	68.56

TABLE IV.2: Average completion time over all trials and subjects in the different conditions

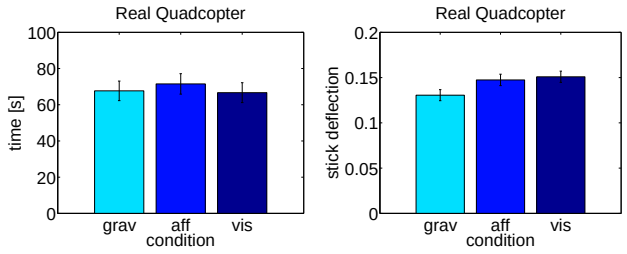


FIGURE IV.12: Overall average time (a) and control effort (b) per condition

By referring to the detailed analysis reported in [C21], it was possible to show that the amount of input commands was correlated with the tested conditions. In particular, in the *grav* condition the subjects had more 'gentle' control actions compared to the *aff* and *vis* conditions. This result could be explained by the fact that, in the *grav* condition, the subjects could directly feel the UAV motion and thus were more careful in applying strong or erratic commands as if they were piloting from onboard. This phenomenon, denoted as *shared fate* in [Hing and Oh, 2009], is a consequence of the improved situational awareness of the test subjects in the *grav* condition. Often pilots report that, by feeling the accelerations they are exposed to, they can become aware of dangerous loads and stresses in the aircraft structure and avoid too risky maneuvers. Of course, lack of this information in a remote control scenario may be dangerous for the UAV safety and ultimately lead to unnecessary losses that could be avoided by providing the correct motion cues to the ground operator.

IV.5 Extensions

In this last last Section, a short overview of some additional activities related to what discussed so far will be given. In particular, we will illustrate an improved mechanical design (and inverse kinematics scheme) for the CyberMotion simulator, as well as a modeling approach for determining the humans' perceptual thresholds to motion stimuli (threshold that can then be used for the design/tuning of the motion cueing algorithm implemented on the CyberMotion simulator).

IV.5.1 Novel Mechanical Design for the CyberMotion Simulator

The initial mechanical design of the CyberMotion simulator (a 6-dof serial manipulator with a *static* cabin attached to it, see Fig. IV.3(b)) has been later on improved by making use of an *actuated* and closed cabin. Figure IV.13(a) shows the new cabin design mounted on the robot arm, while Figs. IV.13(b–c)

give some details on the actuation system and on the cabin internal space. Essentially, the new cabin

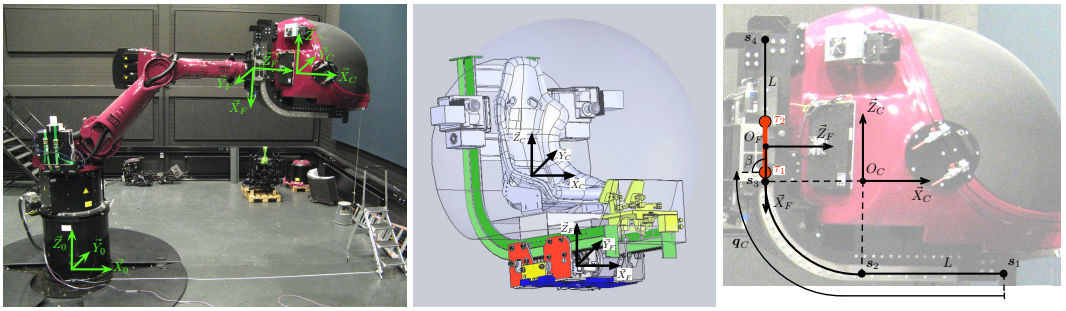


FIGURE IV.13: Side view of CyberMotion Simulator with the actuated cabin

consists of two main parts, a gondola and an actuation system: the gondola is a $1.6 \times 1.8 \times 1.9$ m closed shell that includes the seat (and the pilot). The closed shell design allows to attenuate undesired sensory cues from the external environment, thus improving the simulation experience. The front part of the gondola also serves as a large screen for visualization purposes (with a larger field of view compared to the previous design).

The actuation system consists of a linear/circular/linear rail on which an actuated flange (attached to the robot end-effector) is allowed to 'slide'. Figure IV.14 reports a sequence of snapshots that illustrate the cabin motion while sliding on the rail from end to end.

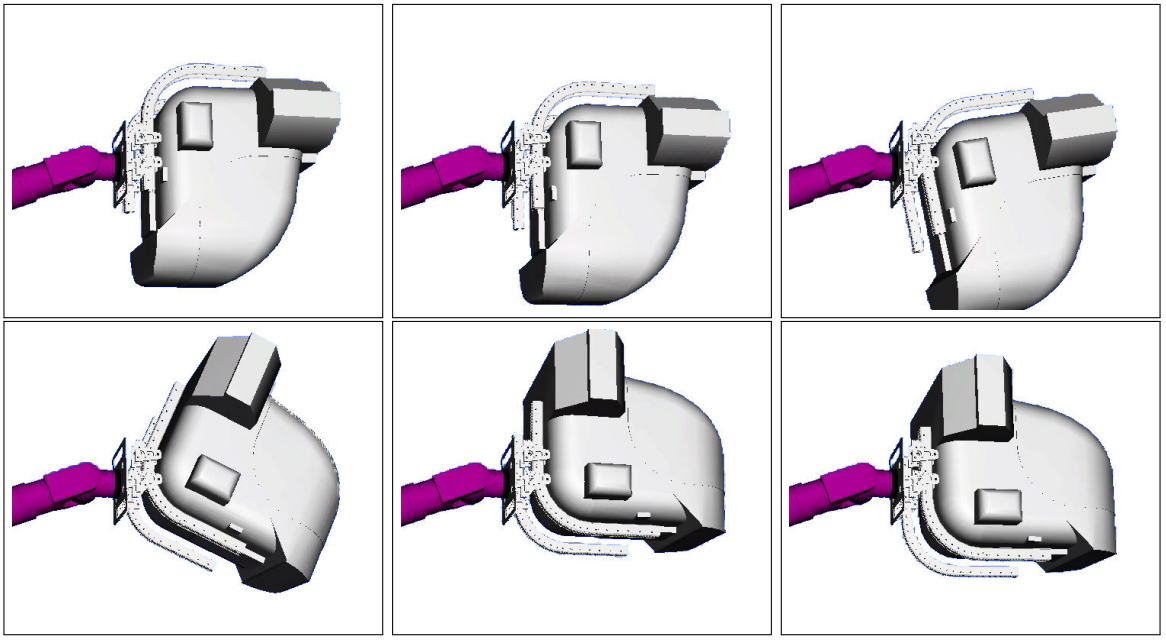


FIGURE IV.14: Simulation 1 - snapshots.

Therefore, thanks to the new cabin design, the CyberMotion Simulator gains an extra degree of freedom for executing a motion task (it becomes a 7-dof *redundant* robot manipulator). It is then of course possible to exploit this redundancy in the Inverse Kinematics scheme for improving the execution of the desired trajectory $r_d(t)$ output of the Motion Cueing block. Among the many possible choices, we opted for the optimization of a criterium attempting to maximize both the (classical) manipulability measure and the distance from joint limits. To this end, the following cost function (introduced in [Nelson and

Khosla, 1995]) was considered

$$H(q) = \left(1 - e^{-k\prod_{i=1}^{n} \frac{(q_i - q_{i,min})(q_{i,max} - q_i)}{(q_{i,max} - q_{i,min})^2}}\right) \sqrt{\det(J(q)J^T(q))}, \tag{IV.4}$$

where k is a design parameter and n is the number of joints. By maximizing $H(q)$, the robot will avoid as much as possible singularities and joint limits, with k tuning the relative weights among these two requirements.

This comparative simulation allows to appreciate the optimization action on $H(q)$: the goal is to realize a constant linear deceleration ${}^{C}a = [-7\ 0\ 0]^T$ [m/s^2] in $\mathcal{F}_C$ (the frame attached to the cabin, see Fig. IV.13(a)). Figure IV.15 shows the robot motion for the cases of *non-actuated* cabin (left column) and *actuated* cabin (right column), respectively, with the current direction of the fifth robot link (blue dashed line) and its limits (solid red line) depicted as well. By looking at the left column (fixed cabin), one can note how the cabin moves backward and rotates until hitting the fifth joint limit, while in the right column (actuated cabin) the joint limit is never reached thanks to the optimization of $H(q)$ (note how the cabin rotates during the maneuver). Figure IV.16 shows the behavior of the actual cabin accelerations ${}^{C}a$ for the fixed case (solid lines) and the actuated case (dashed lines). It is evident how the actuated case allows for a far better reproduction of the desired acceleration, while in the fixed case a significant distortion is present because the joint limit is not avoided (and, thus, the task $r_d(t)$ is not fully executed).

Finally, the same experiment described in Sect. IV.4.1 has been re-run on the new setup. Figure IV.17 shows a snapshot from inside the cabin.

IV.5.2 Modeling of human perception thresholds

The goal of the motion cueing algorithm of Sect. IV.2 is to transform a physical motion (linear acceleration and angular velocity) into a different motion which is compatible with the motion simulator capabilities, but it still induces as much as possible the same motion perception onto the user. The algorithm consists of several steps, including filtering, scaling, rate limiting and saturations. The several parameters involved in the process are usually determined by experience and trial and error. It is however possible to take a more analytical approach and attempt to *model* parts of the human motion perception in order to have some guidelines for the choice of these parameters.

A step in this direction has been taken in [J6, J8] where a model able to predict perceptual direction thresholds of human subjects exposed to arbitrary linear/angular motions has been proposed. Essentially, the vestibular system in the human inner ear includes two organs — the otoliths able to sense linear accelerations (thus, acting as an accelerometer) and the semicircular canals able to sense angular velocities (thus, acting as a gyroscope). Figure IV.18 gives a pictorial view of the vestibular system.

The study of the vestibular system has a long history in the neuroscience community, with works starting from the '60s until nowadays [Young and Meiry, 1968, Fernandez and Goldberg, 1976, Hosman and van der Vaart, 1978, Benson et al., 1986, Zaichik et al., 1999, Heerspink et al., 2005]. While the process of motion perception is quite complex and involves many steps (physical response to a stimulus, cognitive interpretation, and so on), it is nevertheless interesting to note that often simple models can be exploited

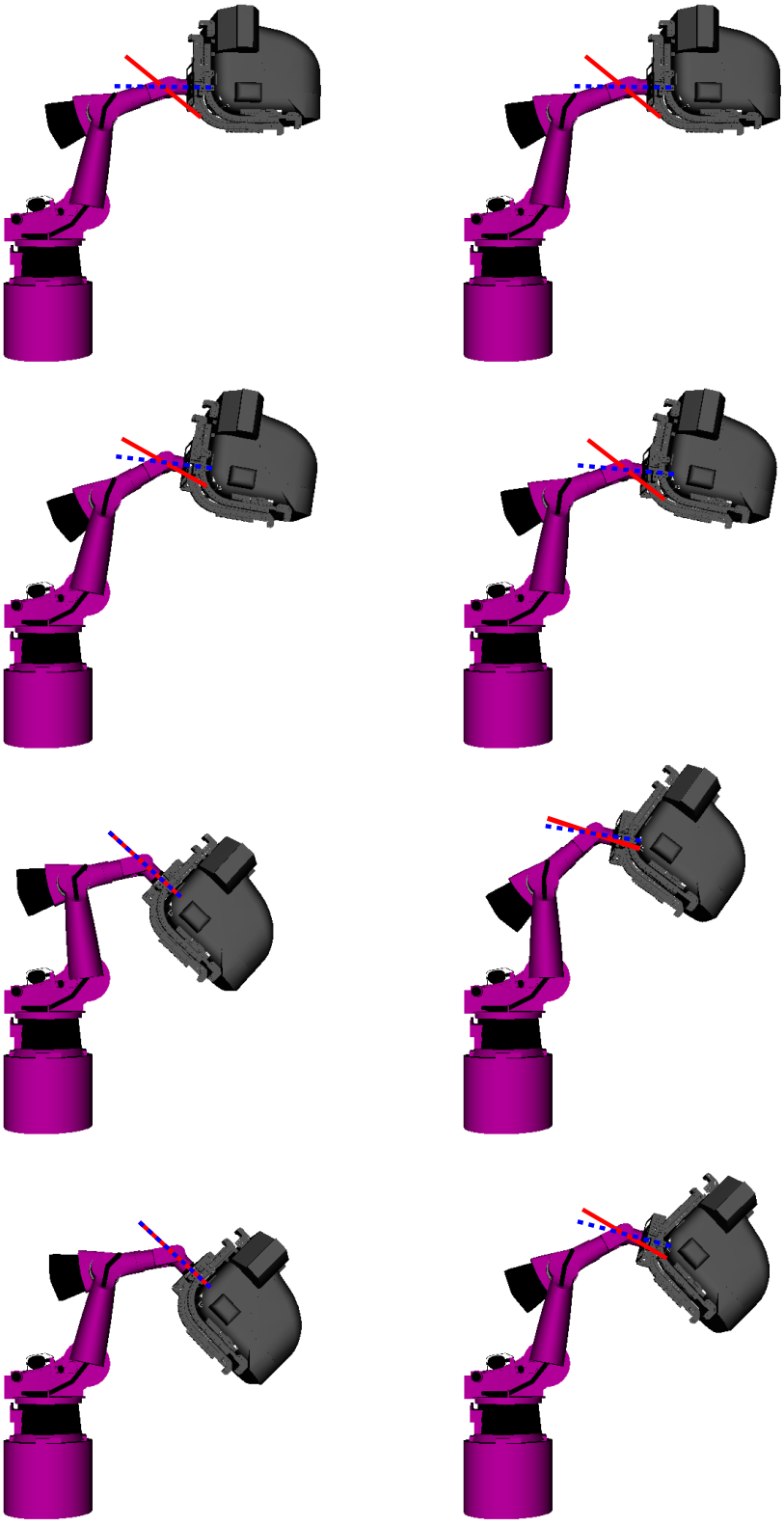


FIGURE IV.15: Snapshots of the robot during the execution of the task, without using the 7-th joint (left column: a,c,e,g) and using it (right column: b,d,f,h). The red line indicates the current orientation of the 5-th joint and the dashed blue line indicates its limit.

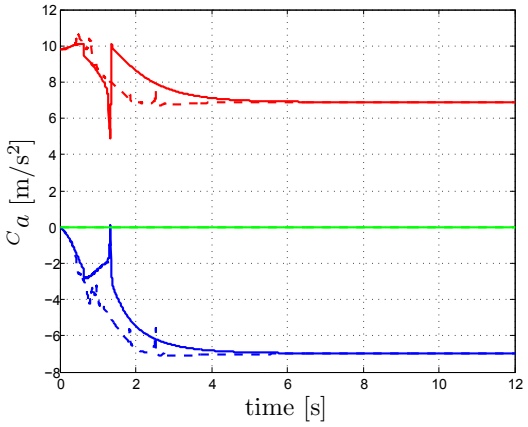


FIGURE IV.16: Simulated acceleration ${}^C a$ in $\mathcal{F}_C$. Solid lines refer to the fixed cabin case, dashed lines refer to the actuated cabin case. Blue, green and red lines refer to accelerations along X_C, Y_C and Z_C, respectively

FIGURE IV.17: Experiment - snapshot from a camera inside the cabin.

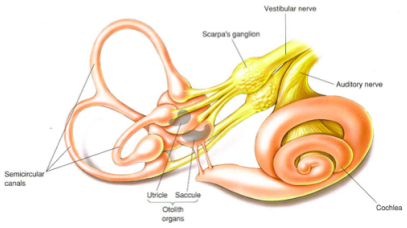


FIGURE IV.18: Experiment - snapshot from a camera inside the cabin.

to explain the underlying mechanisms in good detail. This approach was taken in [J6, J8] where a linear transfer function mapping input stimuli (linear accelerations or angular velocities) to the neuron firing rate of the otholiths and semicircular canals (the information processed by the brain) was employed. In particular, it was possible to show that the otolith firing rate can be modeled as

$$H_{oth}(s) = K\frac{(1+\tau_N s)}{(1+\tau_1 s)(1+\tau_2 s)}$$

and that a linear motion is *detected* whenever the output of this transfer function exceeds a given threshold (taken as 1 by a proper choice of gain K). Similarly, the firing rate response of the semicircular canals

could be modeled as

$$H_{sen}(s) = K\frac{s(1+\tau_N s)}{(1+\tau_1 s)(1+\tau_2 s)}$$

with the same considerations as above. The parameters (K, τ_1, τ_2) can be determined by running psychophysical studies on human subjects to different motion stimuli, and by analyzing their detection response.

By using these simple, yet effective, models it is possible, for instance, to predict whether a given motion profile will be detectable or not, or how much one should amplify/modify a trajectory in order to make it under-/over-threshold. As an example, Fig. IV.19(a) shows two input motion profiles (solid lines) which are at threshold level (black line) or barely under threshold level (grey line). Figure IV.19(b), on the other hand, shows how much one should amplify a given motion profile (left) in order to make it detectable (right).

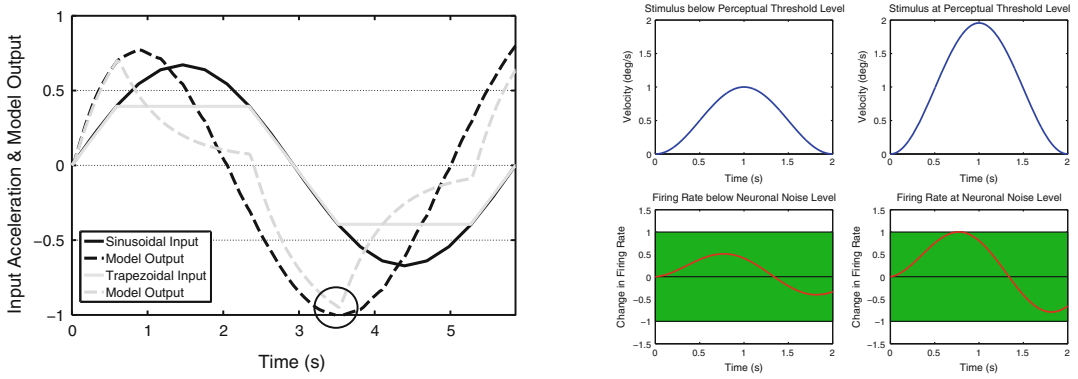


FIGURE IV.19: Example of an input profile (solid black curve), for which the amplitude is at threshold level. The corresponding output of the transfer function model has a maximum absolute value of one (black circle). The solid gray input curve is just below threshold level, because the output does not reach a maximum value of one

It is indeed worth noting that, because of the 'system dynamics' associated to the motion perception process, the *transient* part of a motion stimulus can be dominant w.r.t. its steady-state component, and thus one cannot rely on a simple proportional amplification factor for any kind of trajectory (as often assumed in prior literature). These findings can then be exploited within a motion cueing algorithm for, e.g., adaptively tuning the saturation thresholds of the input linear accelerations/angular velocities so as to make sure a given motion stimulus is correctly perceived (or not perceived at all).

IV.6 Discussion

IV.6.1 Summary

This first Chapter has addressed the shared control paradigm for single robots by resorting to a (non-classical) combination of visual and *vestibular* feedback, that is, the possibility to stimulate the user self-motion perception (his sensed linear/angular accelerations) in addition to providing him with a visual stimulus. This combination represents a non-standard possibility w.r.t. the more consolidated visual-force paradigm (which will be thoroughly addressed in the next Chapter V): in order to implement this shared control paradigm, an initial, but fundamental, part of this work has been the design

and testing of an online *motion cueing* and *inverse kinematics* schemes tailored to a 6-dof large-scale industrial serial manipulator arm — the *CyberMotion* simulator. The motion cueing (responsible for generating a trajectory feasible for the robot motion envelope but still representative of the intended motion cue) and inverse kinematics have been extended from standard state-of-the-art methods such as the classical washout and tilt coordination algorithms (adapted to the almost cylindrical workspace of the CyberMotion simulator) and the Task Priority approach with velocity and acceleration bounds.

The overall motion control framework has been validated in a number of experiments with human subjects: by simulating a race car driving a lap on the Monza track, and by controlling the motion of a (real) remote quadrotor UAV equipped with an onboard camera. The former case study allowed to verify the good performance of the control framework in replicating on the user a large spectrum of linear/angular acceleration (such as those one can experience onboard a race car). The latter case study, on the other hand, was meant to prove the effectiveness of providing a user with a combination of visual and *vestibular* feedback in remotely piloting a real robot. Indeed, the results of the user study indicates that a combination of visual and vestibular feedback is statistically helpful in reducing the control effort while piloting the remote robot (a result also known as *shared fate*).

Finally, two extensions have been presented: an improved inverse kinematics scheme taking advantage of the a mechanical upgrade of the CyberMotion simulator (the addition of a 7-th actuated axis), and a series of psychophysical studies meant to model motion detection thresholds in human subjects when exposed to linear or angular motions). These threshold modeling, indeed, can serve as a more systematic way to tune or optimize any motion cueing algorithm where, typically, all parameters are chosen by experience or trial and error procedures.

IV.6.2 Perspectives

Despite the good results discussed in this Chapter, several extensions and improvements are of course possible. For instance, from an algorithmic point of view, the inverse kinematic design described in Sect. IV.3 could be improved by relying on the recent advancements in the field of control and optimization for redundant robots, see, e.g., [Saab et al., 2013, Escande et al., 2014, Flacco and De Luca, 2013b, Flacco and De Luca, 2013a, Flacco and De Luca, 2014]. In particular, one could leverage the unified optimization framework proposed in these work for handling all the several constraints of Sect. IV.3 at once in a more principled way. Along the same lines, another important improvement would be to merge the motion cueing and inverse kinematics algorithms as the solution of a single optimization problem (rather than as the sequence of two somehow independent problems): how to filter a desired motion profile into a trajectory compatible with the motion simulator kinematics and still inducing, as much as possible, the same motion perception. Finally, the use of models of the human perceptual system (such as those briefly discussed in Sect. IV.5.2) should be exploited for (*i*) tuning the various parameters of the motion control software and for (*ii*) defining suitable metrics for deciding when a filtered trajectory is 'representative enough' of the original one from a perceptual point of view.

In this respect, one should aim for the conceptual scheme depicted in Fig. IV.20: exploiting a suitable modeling of the human perceptual system (such as the one presented in [Wentink et al., 2006]), it could be possible to compare the *output* of such system when excited with the *intended* vehicle motion and the *filtered* one (because of both the motion cueing and inverse kinematics). The corresponding *error signal* can then drive a numerical optimization routine aimed at optimizing the various parameters for zeroing

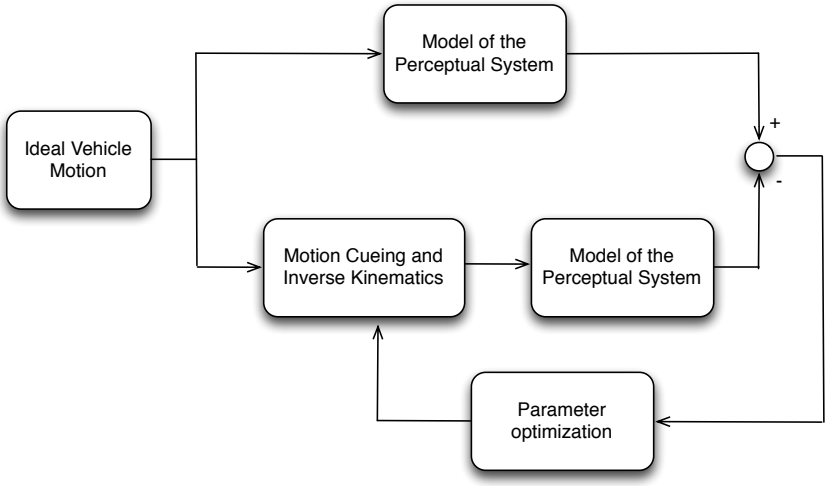


FIGURE IV.20: A conceptual scheme for optimizing the parameters of a motion cueing algorithm: by comparing the outputs of a suitable model of the human perceptual system, one could aim at minimizing the error between the 'real' perception (i.e., what the human would experience when exposed to the actual vehicle motion) and the 'simulated' perception (i.e., what the human feels when on board the motion simulator).

the error signal (i.e., resulting in the same motion perception for ideal and filtered vehicle trajectory). While conceptually simple, solving this problem is quite complex in practice because of several issues such as availability of a reliable model of the human perception system, the definition of meaningful *output* signal for the model (and corresponding error function), and the possibility to optimize an arbitrarily nonlinear (and possibly non-smooth) cost function via numerical routines and at runtime. The Motion Perception and Simulation group[3] at the Max Planck Institute for Biological Cybernetics, Tübingen, Germany, is currently devoting large research efforts in this direction.

IV.6.3 Main references

The work presented in this Chapter is based on the following material:

- Ph.D. Thesis of Florian Soyka "A Cybernetic Approach to Self-Motion Perception", January 2013, University of Tübingen, Germany
- Ph.D. Thesis of Carlo Masone "Planning and Control for Robotic Tasks with a Human-in-the-Loop", July 2014, University of Stuttgart, Germany
- F. Soyka, P. Robuffo Giordano, M. Barnett-Cowan, and H. H. Bülthoff. Modeling direction discrimination thresholds for yaw rotations around an earth-vertical axis for arbitrary motion profiles. *Experimental Brain Research*, 220(1):89–99, 2012. (ref [J8])
- F. Soyka, P. Robuffo Giordano, K. Beykirch, and H. H. Bülthoff. Predicting Direction Detection Thresholds for Arbitrary Translational Acceleration Profiles in the Horizontal Plane. *Experimental Brain Research*, 209(1):95–107, 2011. (ref [J6])

[3] http://tinyurl.com/kynmac2

- C. Masone, P. Robuffo Giordano, and H. H. Bülthoff. Mechanical Design and Control of the new 7-DOF CyberMotion Simulator. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 4935–4942, Shanghai, China, May 2011. (ref [C26])

- P. Robuffo Giordano, H. Deusch, J. Lächele, and H. H. Bülthoff. Visual-Vestibular Feedback for Enhanced Situational Awareness in Teleoperation of UAVs. In *Proc. of the AHS International 66th Annual Forum & Technological Display*, 2010. (ref [C21])

- P. Robuffo Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini, and H. H. Bülthoff. A Novel Framework for Closed-Loop Robotic Motion Simulation - Part I: Inverse Kinematics Design. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 3876–3883, 2010. (ref [C19])

- P. Robuffo Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini, and H. H. Bülthoff. A Novel Framework for Closed-Loop Robotic Motion Simulation - Part II: Motion Cueing Design and Experimental Validation. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 3896–3903, 2010. (ref [C18])

- M. Niccolini, L. Pollini, M. Innocenti, P. Robuffo Giordano, H. Teufel, and H. H. Bülthoff. Towards Real-Time Aircraft Simulation with the MPI Motion Simulator. In *Proc. of the 2009 AIAA Modeling and Simulation Technologies Conference*, 2009. (ref [C16])

- F. Soyka, H. Teufel, K. Beykirch, P. Robuffo Giordano, J. Butler, F. M. Nieuwenhuizen, and H. H. Bülthoff. Does jerk have to be considered in linear motion simulation? In *Proc. of the 2009 AIAA Modeling and Simulation Technologies Conference*, 2009. (ref [C15])

Finally, the US patent N. 8,634,969, available at `http://www.google.com/patents/US8634969`

- H. H. Bülthoff and P. Robuffo Giordano, "Teleoperation method and human robot interface for remote control of a machine by a human operator", January 2014

has also been published using parts of the work presented above.

IV.6.4 Videos

- Experiments of driving a lap on a simulated race car with the CyberMotion simulator (Sect. IV.4.1): `https://youtu.be/oMLarxR-q08`

- Experiments of visual-vestibular teleoperation of a real flying robot (a quadrotor UAV) with the CyberMotion simulator (Sect. IV.4.2): `https://youtu.be/APnu3ydfOnY`

Chapter V

Shared Control with Visual-Force Feedback

This Chapter will discuss several shared control architectures for *single* and *multiple* mobile robots exploiting the more standard choice of visual/force combination for providing a feedback to the human operator, that is, following (and extending) the well-known bilateral force feedback paradigm which is a classical topic in the robotics literature [Hokayem and Spong, 2006, Nuno et al., 2011]. This choice can then be seen as a complement to what presented in the previous chapter in exploiting a different sensory channel besides vision (the force/haptic one) for providing the human operator with some meaningful information about the status of the remote mobile robot(s).

V.1 Overview of bilateral teleoperation with force feedback

V.1.1 The standard case

In a force-feedback bilateral teleoperation scheme, a human operator physically acts on an actuated *master* device (e.g., an actuated joystick or a robot manipulator) for controlling its position and/or velocity by applying the needed forces. The master position/velocity is transmitted via a communication channel towards a remote *slave* device (e.g., a remote robot in physical contact with the environment) which mimics the master motion and exchanges forces with the local environment. These forces are then reflected back (still via the communication channel) towards the master device which displays them to the human operator (in addition to a visual feedback streamed from the remote site). Figure V.1 shows

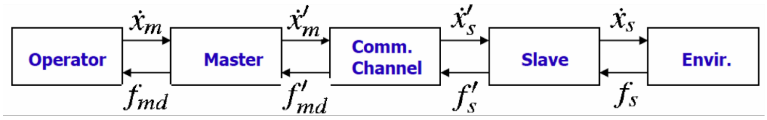


FIGURE V.1: A schematic view of a classical bilateral teleoperation loop

a simple schematic view of a classical bilateral teleoperation architecture, while Fig. V.2 gives a pictorial illustration in the specific case of a *slave* consisting of multiple quadrotor UAVs commanded by a single human operator (later discussed in Sect. V.3).

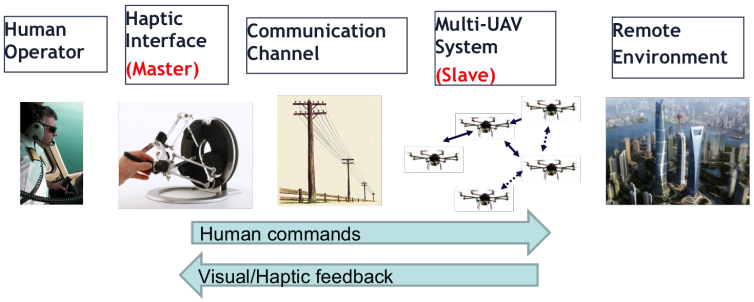


FIGURE V.2: Conceptual architecture of a bilateral teleoperation of multiple mobile (flying) robots

While an exhaustive treatment of all the various control schemes and subtleties of a bilateral force feedback architecture is of course not possible here[1], we will now briefly discuss the main issues concerning the stability of the composition illustrated in Fig. V.1. The main (and typical) control problem faced when implementing a bilateral force-feedback teleoperation is how to guarantee a *stable* interaction between master and slave while ensuring the largest possible *transparency* of the interaction itself (e.g., the ability to faithfully replicate the master motion on the slave side, and the forces exchanged between the slave and the environment on the master side). With reference to Fig. V.1, the *communication channel* is one typical source of instabilities because of the possible delays, discretization, quantization, packet losses and other non-idealities introduced by the communication medium. Another possible source of instabilities is the unknown model of the human operator and of the environment (the two ends of the chain in Fig. V.1). Indeed, the remote slave will typically interact with some unknown environment for which a model cannot be assumed available (e.g., a model of how the environment dynamically exchanges forces with the slave robot). Analogously, the human operator will exchange forces with the master in a 'unpredictable' way (or hard to model), thus inducing an additional degree of uncertainty in the loop.

Early works on bilateral teleoperation attempted to *robustify* the master/slave loop by assuming simple linear models for human and environment (e.g., linear mass/spring damper systems) [Hokayem and Spong, 2006]. However, these approaches can nowadays be considered as surpassed by a number of methods exploiting energetic (passivity) considerations, which are instead able to deal with a large class of nonlinear and even time-varying models for the human, environment and communication medium [Nuno et al., 2011]. Roughly speaking, the idea behind this class of algorithms is to exploit the strong 'energetic' properties of mechanical systems (i.e., the typical master/slave robots as well as a large class of 'environments') by resorting to *passivity* arguments [Sepulchre et al., 1997]. In energetic terms, a passive system cannot produce *energy* on its own but it can just store (and possibly re-use) any energy obtained from the 'outside world', i.e., through the 'power port' $(u(t), y(t))$, with the lower-bounded function $V(x(t))$ representing the internal stored energy. The advantages of taking a passivity (energetic) perspective in analyzing the overall behavior of a teleoperation system is mainly due to the facts that (i) passivity can be easily proven for a large class of nonlinear (and, to some degree, uncertain) dynamical systems (such as, e.g., mechanical systems with parameter uncertainties) and (ii) proper combinations of passive systems remain passive despite their (possibly complex) interconnection. This is in particular the case of the standard teleoperation architecture of Fig. V.1 where, if the individual components (master, communication channel, slave and environment) are passive, their connection remains a passive (and therefore input/output stable) system upon which a human operator can act [Nuno et al., 2011]. Since the master and slave are typically standard mechanical systems (and thereby inherently passive), and, in most cases, the environment can also be modeled as a passive component (e.g., a wall, or any

[1]The interested reader is anyway referred to [Hokayem and Spong, 2006, Nuno et al., 2011] for a comprehensive overview.

other surface with which the slave may be in contact), the only common source of non-passivity (i.e., of internal energy generation) is the communication channel due to delays, discretization and quantization effects in transmitting information from the master to the slave and back.

A number of approaches has been proposed in the literature for *passifying* the communication channel, i.e., for compensating for its possible non-passive behavior via suitable control actions at the master and/or slave sides. In the subsequent works, the possible stability issues arising from non-idealities of the communication channel (as well as other sources of non-passive behaviors) will be tackled by exploiting passivity arguments and, in particular, by relying on either the Passive Set-Point Modulation (PSPM) algorithm introdced by Lee et al. [Lee and Huang, 2010], or by making use of the concept of *energy tanks* applied to the teleoperation case as detailed in, e.g., [Franken et al., 2011]. Both algorithms share a common goal: exploit an *online* estimation of the energetic exchanges between the master and the slave sides for then suitably acting/modulating/reshaping the exchanged signals whenever a possible passivity violation is detected (in an integral sense).

V.1.2 Teleoperation of mobile robots

We conclude this introductory part with some considerations about the specific issues encountered when implementing a teleoperation channel between a standard master side with a limited workspace (e.g., a joystick upon which a human operator acts) and a single (or groups of) *mobile* robot(s).

V.1.2.1 Kinematic dissimilarity

A first important difference is the kinematic dissimilarity between the *bounded* master workspace and the (in principle) *unbounded* slave one. In a classical teleoperation setting one can easily design a $1:1$ mapping between the two *bounded* workspaces of the master and slave slides even though these can have different geometries or extents. This allows to directly couple the master and slave position or velocity via some suitable scaling/rotation mappings, that is, master and slave are linked at the same *differential* level (position/velocity of the master with the position/velocity of the slave). However, when considering single (or groups of) *mobile* robots as slave side one cannot (in general) rely on a similar scheme and must, instead, look for couplings at non-homogeneous differential levels, for instance by linking the master position with the slave velocity (as in a car driving metaphor). While this modification is quite intuitive and easily accepted by a human operator, it poses some technical issues from the passivity point of view in the teloperation stability. In short, while a mechanical system is known to be passive w.r.t. to the *force/position* input/output pair, the same does not hold for other output functions (such as, e.g, the master position). When coupling master and slave at the same differential level (as in standard teleoperation), one essentially leverages the intrinsic passivity of mechanical systems by making sure that the master/slave velocity (or their integrals) and forces are properly coupled. However, when attempting to couple the master position with the the slave velocity, the passivity argument unfortunately does not hold any more. This is a well-known technical difficulty in producing stable teleoperation architectures for mobile robots, and several variations or workarounds have been proposed to tackle this issue. For instance, one can resort to a *modified* 'position-like' quantity, also known as r-variable [Chopra et al., 2008, Lee and Xu, 2011], to be used as master output for controlling the slave velocity. This quantity is a suitable combination of master position *and* velocity and it is possible to prove that its use (in place of the standard master velocity) does not threaten the overall teleoperation passivity by a proper choice

of the control gains[2]. Another possibility is to rely on passifying actions such as the above-mentioned PSPM algorithm (for first-order master/slaves) or the tank machinery as done, for instance, in [C36] (and better discussed later on).

V.1.2.2 Fabricated environmental forces

Another difference when teleoperating mobile robots (and, even more so, flying robots as it will be the case in the next sections) is the typical absence of any substantial force exchanged with the environment: while a classical slave robot is typically in touch with the environment and can then sense *real* interaction forces reflected back on the human operator, in most contexts a mobile robot must keep away from obstacles/walls or other hard contacts with the environment for a safe (teleoperated) navigation. Therefore, no real forces are sensed at the slave side, and one must then *fabricate* virtual interaction forces representative of the slave operation to be reflected back on the master side. Figure V.3 shows

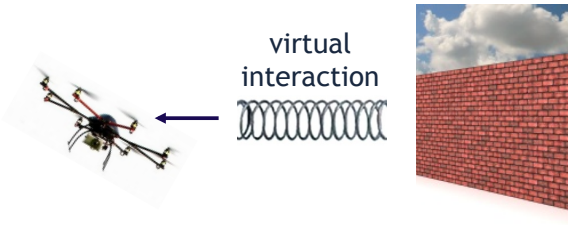


FIGURE V.3: Since actual contact between a flying robot and the environment (obstacles) is in general avoided, fictitious environmental forces (e.g., repulsive actions) must be implemented for a safe navigation and for providing a feedback to the human operator.

an example for a flying robot in proximity of a wall: as direct contact with the wall must be avoided (for the robot safety), one can, for instance, exploit proximity sensors for detecting the distance to the obstacle and producing a virtual (spring-like) force informative of the surrounding environment. As this (or other) force are *fabricated*, one clearly has large freedom in coming up with force cues representative of (even quite abstract) operational states of the slave side. However, the downside is that any force cue must (*i*) clearly comply with any stability/passivity constraint[3], and (*ii*) should still provide a meaningful information to the human operator (i.e., it must be clearly and quickly interpretable during the mobile robot teleoperation).

As a final note, it is also worth noting that some researchers have instead chosen to reflect back the real environmental forces experienced by mobile (flying) slave robots taken as, e.g., the forces needed to support their flight or to resist to atmospheric drag/turbulences. On the same page, the use of *admittance*-like master devices has been explored for providing a human operator with the possibility of directly commanding a desired force to be applied to the slave while receiving, as feedback, an information based on the master position. This approach is basically the dual of the (far more classical) *impedance* one where the operator acts on the master position/velocity and receives as feedback the force exerted by the master device. Each approach has its own pros/cons depending on the particular application and slave robot. Some discussion can be found in [Hou et al., 2013, Hou and Mahony, 2014] and references therein[4].

[2] However, this comes at the price of a degraded transparency of the bilateral teleoperation since the human operator can now command a desired slave velocity only at *steady-state* with the master fixed at some reference position.

[3] Although this issue can often be tackled by adopting passifying algorithms such as the PSPM or the tank machinery.

[4] All the schemes presented hereafter will be based on the *impedance* approach.

V.1.2.3 Large slave redundancy

Finally, a last point worth of attention when designing bilateral teleoperation schemes for *multiple* mobile robots (e.g., a group of UAVs) is the typical *large* discrepancy between the master controlled dofs (in the range of $3-6$) and slave controlled dofs (in the range of $6N$ for N robots composing the slave side). Figure V.4 depicts an illustrative example. Because of the slave redundancy w.r.t. the master dofs,

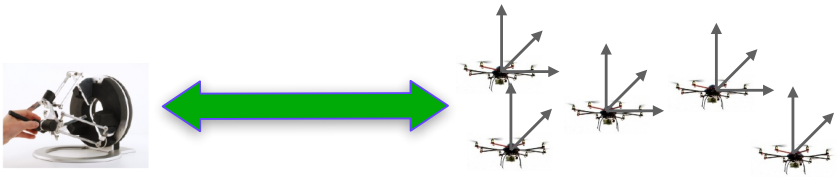


FIGURE V.4: A single human operator acting on a device with 3–6 free DOFs must control a slave side made of N robots with $\sim 6N$ DOFs. This large redundancy must be resolved autonomously by the slave side for not overloading the human operator.

clearly there exists an infinity of possible mappings between master commands and slave motions. For instance, one could exploit the master to physically guide a single robot in the group (the leader) with the other ones following it via some formation control laws, or to control the barycenter or other integral quantities associated to the robot formation. Again, a large freedom is left to the designer for devising meaningful ways (depending on the particular task/application) for coupling the limited master dofs with the 'many' slave dofs. The slave redundancy also leaves the possibility for implementing many 'internal' motions compatible with the master commands and aimed at realizing secondary tasks of interest (e.g., optimal positioning of the robots w.r.t. some criterium, concurrent exploration while following the master navigation commands, and so on). Section V.3 will report many examples in this context.

V.2 Bilateral Teleoperation of a Single Mobile Robot

V.2.1 Overview

Teleoperation of a *single* mobile robot is, to some extent, a well-established field: in general, the human operator commands the *current*[5] desired state of the robotic system (e.g., the current desired position, velocity, and acceleration), and the robot executes the command while exploiting its local autonomy in order to, e.g., avoid obstacles or fulfil other secondary tasks. The loop is then closed by rendering on the haptic feedback a force proportional to the mismatch between commanded and executed motion in order to increase the operator's situational awareness. In the recent literature, this paradigm has been successfully applied to many cases, for instance in [Lee and Spong, 2005] a passivity-based approach to bilaterally teleoperate a group of holonomic/non-holonomic ground robots is presented, and in [Rodríguez-Seda et al., 2010] bilateral teleoperation of a group of UAVs is realized by directly coupling the position of the haptic device to the position of the formation centroid. This solution does not take into account the *kinematic dissimilarity* between the haptic device and the slave mobile robots (bounded vs. unbounded workspace), which, on the contrary, is explicitly considered in [C24, J16]. Similar approaches have also been seen considered in [Lam et al., 2006b, Lam et al., 2006a].

[5] i.e., the state at the current time or in the very next future.

With respect to these prior works, the framework presented in the following attempts to extend this state-of-the-art by allowing the human operator commanding a *future trajectory* for the robot (rather than an instantaneous state of motion) with the concurrent possibility, for the robot, to deform *online* the commanded trajectory in order to meet additional local goals (e.g., obstacle avoidance or closeness to some points of interest). The idea, thus, is to shift the amount of human intervention by granting the operator with the role of "trajectory supervisor" (a higher-level role) and delegate the robot with more autonomy for attempting to facilitate the operator task. As for the force feedback component, the natural choice is to provide an *integral* feedback informative of the mismatch along the commanded/executed *future trajectory* rather than based on an instantaneous discrepancy between, e.g., commanded/executed robot velocity. The human can then be informed (via a force cue) about the *future* consequences of his actions (over some finite future time window), for instance by feeling a repulsive force if the commanded trajectory deformation will eventually lead to an obstacle collision at some future time. Figure V.5 gives

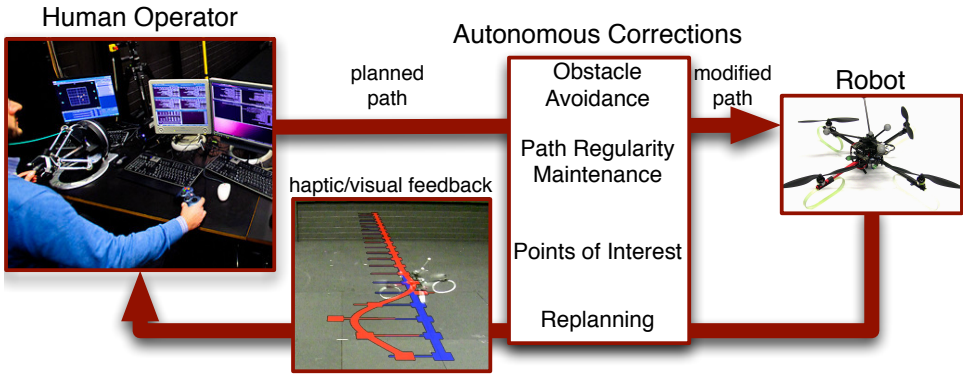


FIGURE V.5: A force-feedback is computed on the the base of the mismatch between the path planned by the human and the one corrected by the robot. A visual feedback using a virtual representation of the environment or an augmented video stream can be used to show the path to the operator.

a block-scheme representation of the proposed architecture: a human operator acts on a force-feedback device (the master) for commanding a *desired trajectory* to be followed by a mobile robot (a quadrotor UAV). These commands can take the form of, e.g., a trajectory translation, scaling, rotation, or any other meaningful geometrical manipulation of some intended path. The robot receives the desired trajectory but it has still the ability to locally modify it via an *autonomous corrector* algorithm whose role is to adjust the commanded trajectory in order to comply with any local requirements/constraints that should be tackled by the robot itself (i.e., obstacle avoidance, maintenance of path regularity properties, closeness to points of interests, and so on). As the commanded trajectory may be deformed by the robot own autonomy, a force feedback signal is then generated out of the *integral* mismatch between the commanded and actual trajectory that is then displayed back to the human operator, thus closing the teleoperation loop. Because of these reasons, the proposed framework is then denoted as *shared planning* in the following sections, as opposite to the more classical shared *instantaneous* interfacing between human operator and robot. The next Sections give some additional details about the proposed shared planning architecture, with a full treatment can be found in [J25, C56, C41].

V.2.2 Shared Planning Architecture

V.2.2.1 Robot

The scenario considered for the design of a shared planning framework consists of a single mobile robot meant to travel along a desired path. The robot is modelled as a rigid body in space and its position in the environment is expressed with respect to an inertial frame denoted with $\mathcal{F}_\mathcal{W} : \{O_\mathcal{W}; \vec{X}_\mathcal{W}; \vec{Y}_\mathcal{W}; \vec{Z}_\mathcal{W}\}$, where $O_\mathcal{W}$ indicates the origin of the frame and $\vec{X}_\mathcal{W}$, $\vec{Y}_\mathcal{W}$, $\vec{Z}_\mathcal{W}$ are its coordinate axes.

The robot is simply assumed to possess a characteristic point (output), whose position in $\mathcal{F}_\mathcal{W}$ is denoted as $\boldsymbol{p}_r$, that is capable of traveling with a non-zero speed along any sufficiently smooth regular path. This requirement is easily achievable for flat systems if the trajectory is sufficiently smooth , see e.g [Faiz et al., 2001], as it will be the case of the quadrotor robot taken as experimental platform for testing the architecture.

V.2.2.2 Path

In the proposed shared planning framework, paths are represented as B-splines (see [Biagiotti and Melchiorri, 2008]), i.e., as a linear combinations of a certain number of suitable basis functions. The use of B-splines is motivated by their generality and versatility, since they can describe or approximate arbitrary functions. Moreover, the relation between the parameters of a B-spline and the shape of the corresponding path can be easily managed and exploited to create an intuitive interface for a human operator.

The family of planar B-spline curves considered in this context is described by the function

$$\boldsymbol{\gamma} : \mathbb{R}^{2n} \times S \to \mathbb{R}^2, \tag{V.1}$$

where $S \subset \mathbb{R}$ is a compact set and $\mathbb{R}^{2n}$ is the parameterization domain. A B-spline curve of this family is a function

$$\boldsymbol{\gamma}(\boldsymbol{x}, \cdot) : S \to \mathbb{R}^2, \quad s \mapsto \boldsymbol{\gamma}(\boldsymbol{x}, s) \tag{V.2}$$

that is parameterized by the vector of coplanar control points $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1^T & \cdots & \boldsymbol{x}_n^T \end{pmatrix}^T \in \mathbb{R}^{2n}$. According to this notation, $\boldsymbol{\gamma}(\boldsymbol{x}, s) \in \mathbb{R}^2$ is a single point of the B-spline curve, i.e., the point obtained by evaluating the function $\boldsymbol{\gamma}(\boldsymbol{x}, \cdot)$ in $s \in S$. Finally, the path corresponding to the B-spline curve $\boldsymbol{\gamma}(\boldsymbol{x}, \cdot)$ is

$$\boldsymbol{\gamma}_S(\boldsymbol{x}) = \{\boldsymbol{\gamma}(\boldsymbol{x}, s) \in \mathbb{R}^2 \,|\, s \in S\}, \tag{V.3}$$

i.e., the set of points obtained by varying the coordinate s within S. Therefore, according to (V.3) the control points $\boldsymbol{x}$ parameterizing the B-spline define the shape of the path $\boldsymbol{\gamma}_S(\boldsymbol{x})$. Figure V.6 shows an example of path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ given by six control points ($n = 6$).

Because of the requirement of exact path-following for the robot model, one important condition is to guarantee $\boldsymbol{\gamma}(\boldsymbol{x}, \cdot)$ to be at least a $\mathcal{C}^k$ function ($k > 0$) with respect to both the parameters $\boldsymbol{x}$ and the coordinate s. This requirement can be easily satisfied by choosing a sufficiently high degree for the B-spline. However, another issue needs to be taken into account concerning the well known concepts of *singularity* and *regularity* of parametric curves (see [Stone and DeRose, 1989, Manocha and Canny,

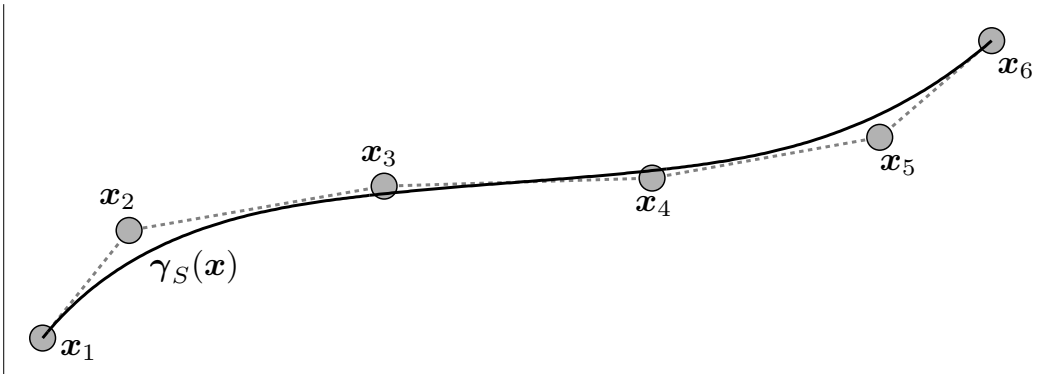


FIGURE V.6: Example of B-spline path.

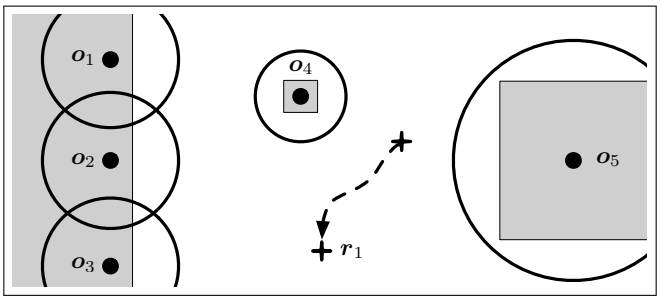


FIGURE V.7: Example of environment. The obstacles (grey) are approximated by several obstacle balls of different radii. A single point of interest, $\boldsymbol{r}_1$, is moving in the environment.

1992]). Namely, a singularity is a point in which the tangent space of the path, i.e., the directions where the motion is allowed, vanishes. Therefore, besides requiring sufficient smoothness, one also needs to guarantee that $\boldsymbol{\gamma}(\boldsymbol{x}, \cdot)$ remains regular in order to avoid a vanishing direction of motion.

V.2.2.3 Environment

The environment where the task takes place is assumed populated by static *obstacles* to be avoided and *points of interest* to be reached. Obstacles are modeled as a finite set of $n_{\mathcal{O}} \geq 0$ balls with fixed radius. The position of the center of an obstacle ball in $\mathcal{F}_{\mathcal{W}}$ is denoted as $\boldsymbol{o} \in \mathbb{R}^2$, and the vector with all the centers is indicated with $\mathcal{O} \in \mathbb{R}^{2 \times n_{\mathcal{O}}}$. The path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ is considered as collision free if it lies outside the obstacle balls. This formulation is quite generic, because any obstacle can be approximated by several balls of various radii (see example of Fig. V.7).

The $n_{\mathcal{R}} \geq 0$ points of interest (PoIs (Points of Interest)) represent important locations for the task. For example, they could be fixed stations that allow to exchange data within a limited range, victims and critical locations in search and rescue applications or moving objects to be monitored. The position of a generic PoI in $\mathcal{F}_{\mathcal{W}}$ is indicated as $\boldsymbol{r} \in \mathbb{R}^2$ and the vector of all PoIs is denoted as $\mathcal{R} \in \mathbb{R}^{2 \times n_{\mathcal{R}}}$. Contrarily to the obstacles, PoIs are not necessarily static (see Fig. V.7): they could be added or removed by the human operator during the task execution or they can be dynamically generated by an external algorithm such as a dynamic routing strategy [Bullo et al., 2011].

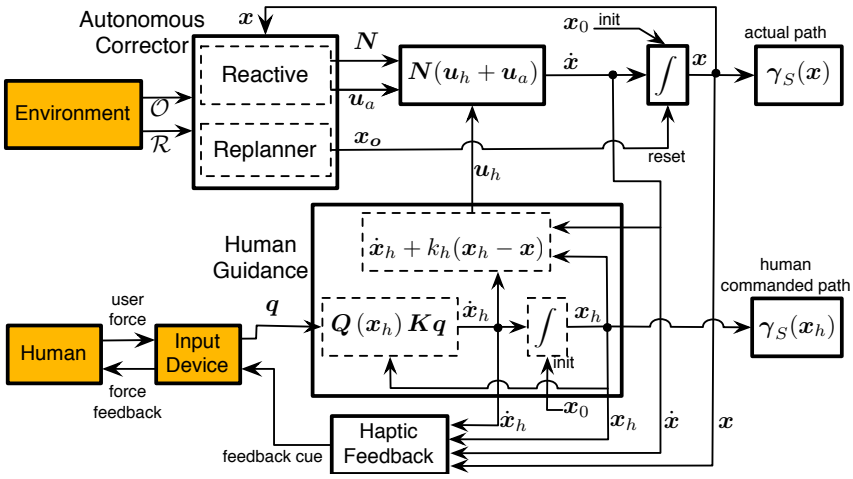


FIGURE V.8: Overview of the framework. The signals x_h and $\dot{x}_h$ indicate the desired corrections provided by the human operator.

V.2.2.4 Description of the architecture

Figure V.8 provides a detailed description of the various components of the shared planning architecture: as explained, the simple, but effective, idea behind the framework is to let a human operator to act in *real time* on the shape of the path with the assistance of an autonomous algorithm that corrects, if necessary, the operator's directives so that the path remains regular and satisfies the requirements regarding obstacles and PoIs. Path modifications are realized by introducing a time dependency in $\boldsymbol{x}$, so that $\boldsymbol{\gamma}(\boldsymbol{x}(t), s)$ becomes a time-varying point and $\boldsymbol{\gamma}_S(\boldsymbol{x}(t))$ a time-varying path. Note that, by introducing also a signal $s(t)$, then $\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))$ would provide the reference trajectory for the robot according to the well-known decoupled design in *path* and *timing law* shown in [Kant and Zucker, 1986], [Peng and Akella, 2005]. The design of a suitable timing law $s(t)$ has not been considered in the reported works, and all the simulations and experiments presented in Sect. V.2.6 have been run by considering a small enough $s(t)$ for keeping the traveling speed small[6].

In a nutshell, the overall architecture works as follows: the signal $\boldsymbol{x}(t)$ is generated online according to the following dynamical system

$$\dot{\boldsymbol{x}} = \boldsymbol{N}\,\big(\boldsymbol{u}_h + \boldsymbol{u}_a\big), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \qquad \boldsymbol{N} \in \mathbb{R}^{2n \times 2n} \tag{V.4}$$

where $\boldsymbol{u}_h \in \mathbb{R}^{2n}$ is the control term influenced by the human operator (described in Sect. V.2.3), and $\boldsymbol{u}_a \in \mathbb{R}^{2n}$ and $N \in \mathbb{R}^{2n \times 2n}$ are two control terms generated by the autonomous algorithm (described in Sect. V.2.4). The initial condition $\boldsymbol{x}_0$ is assumed to define a regular and collision free path and it can be specified by the human operator or by a preliminary planning algorithm tailored for the task at hand[7].

The framework of Fig. V.8 is then organized according to the following circular structure:

Human guidance: It provides the signal $\boldsymbol{u}_h$ in (V.4) that steers the *actual path* (sent to the robot) $\boldsymbol{\gamma}_S(\boldsymbol{x}(t))$ towards the *desired path* $\boldsymbol{\gamma}_S(\boldsymbol{x}_h(t))$ modified by the human via an actuated multi-DoF input device.

[6] Some algorithms meant to generate a suitable $s(t)$ can be found in, e.g., [Faulwasser et al., 2011], [Smith et al., 2012].
[7] For example, it can be an exploration algorithm planning the next move based on the current partial map, or a coverage method that selects one among predefined curve patterns.

Autonomous corrector: It consists of two parts: 1. a reactive algorithm that corrects, if necessary, the human commands such that the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x}(t))$ keeps *regular*, *collision free* and *attracted* by nearby points of interest; 2. a replanner that reinitializes the path in presence of obstacles whenever the local action of the previous step falls within local minimum.

Haptic feedback: It closes the interaction-loop between the human operator and the autonomous correction algorithm in order to increase his/her situational awareness. This is obtained by producing a haptic feedback informative of the changes brought by the autonomous correction to his/her suggested modifications to the current path (in an integral sense).

These three components of the framework are now briefly summarized in the following Sections V.2.3–V.2.4–V.2.5.

V.2.3 Human guidance

The human guidance is obtained by making use of an input device with $1 \leq m \leq 2n$ fully-actuated DoF[8]. An example of actuated input device is illustrated in Fig. V.9. The device is modeled as a generic (gravity pre-compensated) mechanical system

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} + \boldsymbol{\tau}_h \tag{V.5}$$

where $\boldsymbol{q} \in \mathbb{R}^m$ is the configuration vector of the device, $M(\boldsymbol{q}) \in \mathbb{R}^{m\times m}$ is the positive-definite and symmetric inertia matrix, $C(\boldsymbol{q}, \dot{q})\dot{q} \in \mathbb{R}^m$ are the Coriolis and centrifugal terms, and $\boldsymbol{\tau}$, $\boldsymbol{\tau}_h \in \mathbb{R}^m$ are the control and human forces, respectively. The computation of $\boldsymbol{\tau}$ is done automatically by the haptic feedback algorithm and it is described in Sect. V.2.5. Multiple input devices could also be used at once and in this case $\boldsymbol{q}$, $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_h$ are obtained by stacking in columns the corresponding vectors of each device, while M and C become block diagonal matrices.

The configuration vector $\boldsymbol{q}$ is used by the operator to generate $\boldsymbol{u}_h$ in (V.4) for modifying the reference path. The connection between $\boldsymbol{q}$ and $\boldsymbol{u}_h$ is provided by an auxiliary vector of control points $\boldsymbol{x}_h \in \mathbb{R}^{2n}$ that evolves according to the following dynamical system

$$\dot{\boldsymbol{x}}_h = Q\left(\boldsymbol{x}_h\right) K \boldsymbol{q}\,, \qquad \boldsymbol{x}_h(0) = \boldsymbol{x}_0, \tag{V.6}$$

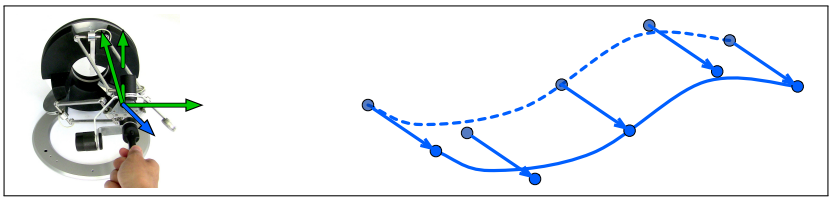
where $K \in \mathbb{R}^{m\times m}$ is a diagonal matrix of positive gains and $Q : \mathbb{R}^{2n} \to \mathbb{R}^{2n\times m}$ is a nonlinear mapping. The vector $\boldsymbol{x}_h$ defines a 'virtual' path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ controlled by the operator alone, without any autonomous correction, i.e., $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ is the *desired* path planned according to the human operator. The term $\boldsymbol{u}_h$ in (V.4) is then designed to steer the actual $\boldsymbol{x}$ towards the desired $\boldsymbol{x}_h$ by implementing a feedforward/proportional-like action, i.e.,

$$\boldsymbol{u}_h = \dot{\boldsymbol{x}}_h + k_h(\boldsymbol{x}_h - \boldsymbol{x}), \tag{V.7}$$

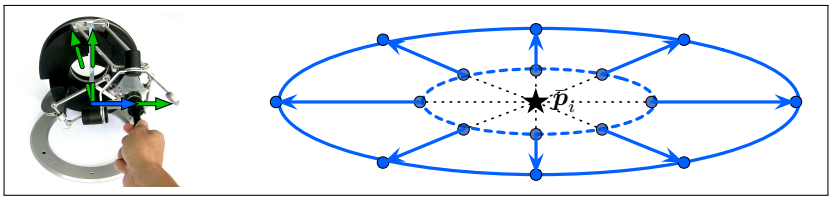
with $k_h > 0$.

The matrix $Q(\boldsymbol{x}_h)$ in (V.6) determines how the human operator is allowed to interact with the path. Clearly any mapping Q could be used, but it is clearly desirable to have a mapping (V.6), i.e., how $\boldsymbol{q}$

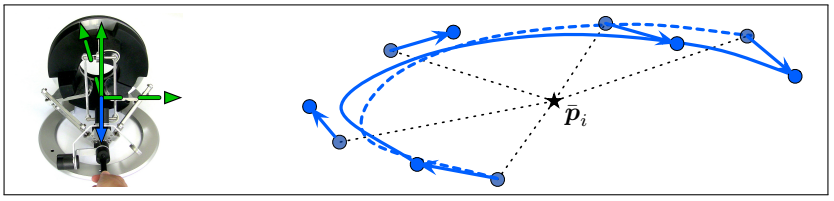
[8] In practice, input devices have at most seven fully-actuated DoF (m) while the number n of control points that specifies a path easily reaches the hundreds even in simple cases.



(A) Translation



(B) Scaling w.r.t. $\bar{\boldsymbol{p}}_i$



(C) Rotation around $\bar{\boldsymbol{p}}_i$

FIGURE V.9: Examples of three canonical path transformations applied to different paths and commanded with the same actuated input device. Green arrows represent the DoF. Continuous green arrows indicate the DoF used by the specific transformation: 2 DoF for translation, 1 DoF for scaling, 1 DoF for rotation. Blue arrows represent the commands and corresponding motion of the control points.

affects $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$, as intuitive as possible for a human operator. One can then map each dof (or group dofs) of the input device to a 'canonical' transformation of the path easily managed by the operator. Translations, scalings and rotations are very good candidates in this sense, as shown in the representative examples of Fig. V.9.

V.2.4 Autonomous corrector

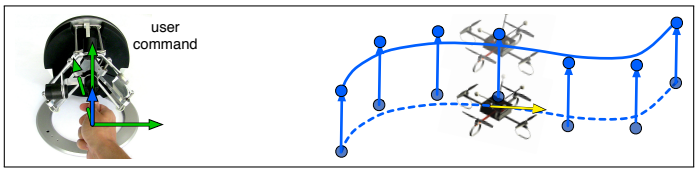
V.2.4.1 Reactive Path Deformation

The reactive part of the autonomous corrector is responsible for generating the control terms $\boldsymbol{N}$ and $\boldsymbol{u}_a$ in (V.4).

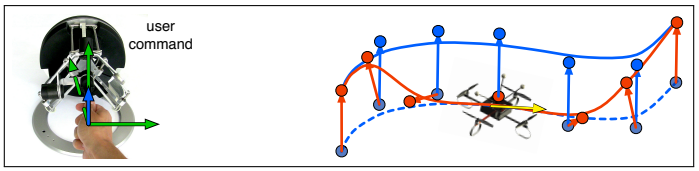
The design of $\boldsymbol{N}$ and $\boldsymbol{u}_a$ is chosen so as to meet several objectives:

Objective V.1. Suppose that an external algorithm provides a timing law $s(t) \in \mathcal{C}^k$ together with its first k derivatives, and denote with $\boldsymbol{p}(t) = \boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))$ the trajectory tracked by the robot. The trajectory time derivatives $\dot{\boldsymbol{p}}(t), \ddot{\boldsymbol{p}}(t), \ldots, \boldsymbol{p}^{(k)}(t)$ must not be affected by the time derivatives of the curve parameters $\boldsymbol{x}(t)$ at the current $s(t)$.

Objective V.2. The distance between any obstacle point $\boldsymbol{o} \in \mathcal{O}$ and $\boldsymbol{\gamma}_S(\boldsymbol{x})$ is always greater than $R_{\mathcal{O}}$.



(A) Without $\boldsymbol{N}$



(B) With $\boldsymbol{N}$

FIGURE V.10: User commanding a desired translation (blue arrow) to the path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ (blue line), while the robot is traveling it with nonzero speed (yellow arrows). In this example it is assumed that the control term $\boldsymbol{u}_a$ is null. a) Without the projection matrix $\boldsymbol{N}$, the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ follows exactly the command, but the resulting motion is unfeasible for the robot. b) When using (V.8), the local geometric properties of $\boldsymbol{\gamma}_S(\boldsymbol{x})$ are preserved and the path translation does not affect the instantaneous motion of the robot.

Objective V.3. The path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ is regular.

Objective V.4. The path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ is attracted by every PoI that is closer than $R_{\mathcal{R}}$ to the path itself.[9]

Realization of Objective V.1

Objective V.1 is important for preventing that path modifications caused by the exogenous human command $\boldsymbol{u}_h$ in (V.4) may result in an unfeasible reference trajectory for the robot *at its current location on the curve*. This could happen, for example, if the operator abruptly steers the path sideways with respect to the current velocity of the robot (see Fig. V.10).

Secondly, when exploiting the differential flatness of the system for the control design, the computation of the robot inputs requires knowledge of $\dot{\boldsymbol{p}}(t), \ddot{\boldsymbol{p}}(t), \ldots, \boldsymbol{p}^{(k)}(t)$ (see e.g. [Mellinger and Kumar, 2011] for the case of a quadrotor). However, since the derivatives of $\boldsymbol{u}_h$ are not assumed available, $\ddot{\boldsymbol{x}}(t), \ldots, \boldsymbol{x}^{(k)}(t)$ and, consequently, also $\ddot{\boldsymbol{p}}(t), \ldots, \boldsymbol{p}^{(k)}(t)$ can only be exactly computed when meeting Objective V.1.

Lastly, Objective V.1 can simplify the design of external algorithms providing the timing law $s(t)$, since the trajectory derivatives only depend on the derivatives of $s(t)$ and not on the derivatives of $\boldsymbol{x}(t)$. For instance, Objective V.1 allows to command the robot to remain still just by keeping $s(t)$ constant, regardless of any underlying path modification.

In order to achieve this goal, the control term $\boldsymbol{N}$ in (V.4) is designed as

$$\boldsymbol{N} = \boldsymbol{I}_{2n} - \boldsymbol{J}^{\dagger}\boldsymbol{J} \tag{V.8}$$

where $\boldsymbol{I}_{2n} \in \mathbb{R}^{2n\times 2n}$ is the identity matrix and $\boldsymbol{J}^{\dagger}$ indicates the Moore-Penrose pseudoinverse of $\boldsymbol{J} \in \mathbb{R}^{2k\times 2n}$, with $k < n$ and $\boldsymbol{J}$ is a Jacobian matrix relating variations of $\boldsymbol{x}$ to changes of local geometric properties of the path in $s(t)$, such as the position of the point $\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))$, the tangent vector

[9] Notice how this objective prescribes a qualitative behavior and therefore it intentionally represents a sort of soft constraint for the autonomous corrector.

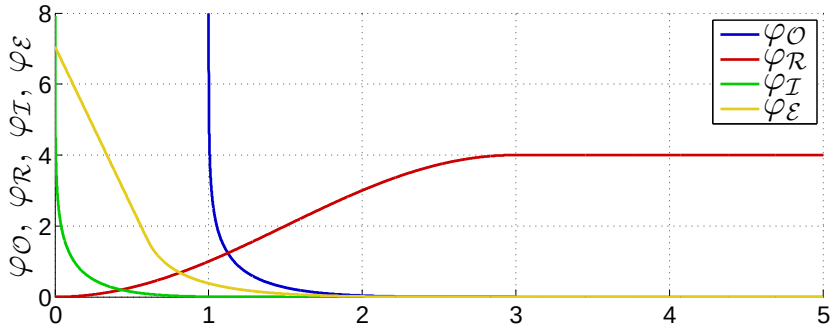


FIGURE V.11: Example of the artificial potentials $\varphi_{\mathcal{O}}$, $\varphi_{\mathcal{R}}$ and $\varphi_{\mathcal{I}}$ used to compute $\boldsymbol{u}_a$, and of the potential $\varphi_{\mathcal{E}}$ that is used in Chapter V.2.4.2.

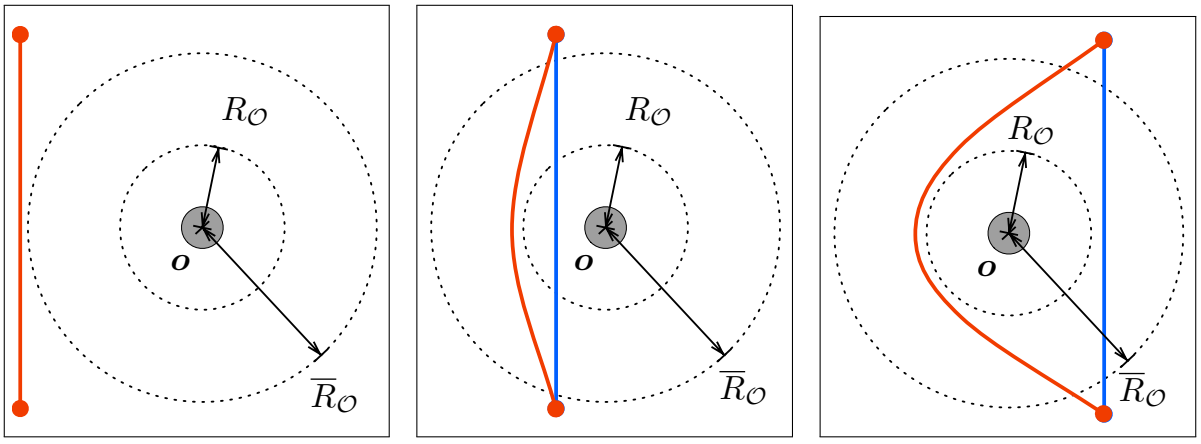


FIGURE V.12: a) to c): Sequence showing the deformation of a path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ (red curve) with respect to the desired path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ (blue line) that is moved through an obstacle (from left to right). In the end, the deformed path becomes a suboptimal w.r.t., e.g., a straight line.

$\frac{\partial}{\partial s}\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))$, the curvature vector $\frac{\partial^2}{\partial s^2}\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))$, and so on. Intuitively, this design imposes the invariance of the *local* geometric properties of the path at the current location of the robot regardless of the *global* changes brought by $\boldsymbol{u}_h$ and $\boldsymbol{u}_a$ in (V.4), as illustrated in the example of Fig. V.10b. This local geometric invariance of the path is beneficial to ease the tracking of the reference trajectory $\boldsymbol{p}(t)$.

Realization of Objectives V.2, V.3 and V.4

In order to satisfy Objectives V.2, V.3 and V.4, the control term $\boldsymbol{u}_a$ is designed as the sum of three terms

$$\boldsymbol{u}_a = \boldsymbol{u}_{a,\mathcal{O}}(\boldsymbol{x}, \mathcal{O}) + \boldsymbol{u}_{a,\mathcal{I}}(\boldsymbol{x}) + \boldsymbol{u}_{a,\mathcal{R}}(\boldsymbol{x}, \mathcal{R}), \tag{V.9}$$

where each term implements the gradient of a suitable potential function (attractive or repulsive) evaluated over the points of interest, obstacles and singular points. Figure V.11 shows some illustrative examples for the potential functions.

V.2.4.2 Generation of Non-homotopic Alternative Paths

The reactive part of the Autonomous Corrector described so far ensures that the path is collision free. However the reactive obstacle avoidance also prevents $\boldsymbol{\gamma}_S(\boldsymbol{x})$ to 'pass over an obstacle', thus leading to possible suboptimal paths (see Fig. V.12 for an example). This is a well known limitation of reactive

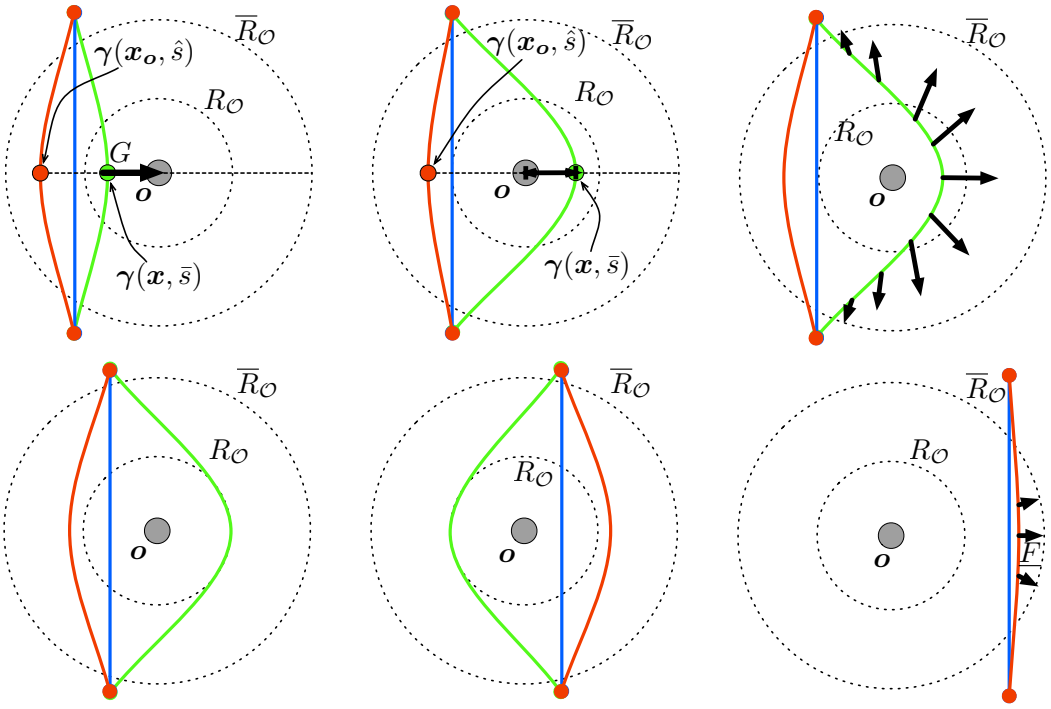


FIGURE V.13: Generation of an alternative path: $\boldsymbol{\gamma}_S(\boldsymbol{x_o})$ (green line), $\boldsymbol{\gamma}_S(\boldsymbol{x})$ (red line), $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ (blue line), obstacle $\boldsymbol{o}$ (gray disc). From a) to f), $\boldsymbol{\gamma}_S(\boldsymbol{x})$ is moving from left to right, passing over the obstacle.

planners and it can severely degrade the capability of the human operator to steer the path, especially in a cluttered environment. In order to overcome this problem one needs a strategy for generating new *alternative* paths in presence of obstacles. For example, in the elastic strip framework by [Brock and Khatib, 2002] this is done by allowing the separation of the elastic strip so that it can cross over the obstacle, however the strip cannot be reconnected if the obstacle remains in between.

The replanning method developed in this framework is still based on continuous deformations, but these deformations actively drive the path to the other side of an obstacle to create an alternative route. The underlying idea is that, given an obstacle $\boldsymbol{o}$ and a collision free path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ between two points[10], it is possible to find a new vector of control points $\boldsymbol{x_o} \in \mathbb{R}^{2n}$ such that $\boldsymbol{\gamma}_S(\boldsymbol{x_o})$ is collision-free, it has the same endpoints of $\boldsymbol{\gamma}_S(\boldsymbol{x_o})$, and it is *non-homotopic* ([LaValle, 2006]) to $\boldsymbol{\gamma}_S(\boldsymbol{x})$ (i.e., it cannot be continuously morphed into $\boldsymbol{\gamma}_S(\boldsymbol{x})$ without intersecting $\boldsymbol{o}$ (see Chapter V.12)). For each obstacle $\boldsymbol{o} \in \mathcal{O}$, the computation of $\boldsymbol{x_o}$ is done in three steps denoted as *Crossing*, *Expansion* and *Activation*. Full details can be found in [C41, C56], with Fig. V.13 giving a pictorial representation of the replanning step.

V.2.5 Haptic feedback

The haptic feedback algorithm computes the force $\boldsymbol{\tau}$ rendered by the input device (see (V.5)) to inform the operator about the discrepancies between the path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ generated by the Human Guidance and the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ modified by the Autonomous Corrector. The force $\boldsymbol{\tau}$ is designed as a function of two haptic cues, $\boldsymbol{e}_{\dot{\boldsymbol{x}}}$ and $\boldsymbol{e}_{\boldsymbol{x}}$. The first haptic cue $\boldsymbol{e}_{\dot{\boldsymbol{x}}}$ provides a feedback indicating how well the teleoperated system is following the *instantaneous* motion command given by the human. It is then designed as a suitable function of the mismatch between $\dot{\boldsymbol{x}}_h$ itself and the actual velocity $\dot{\boldsymbol{x}}$.

[10]It can also be a portion of the path.

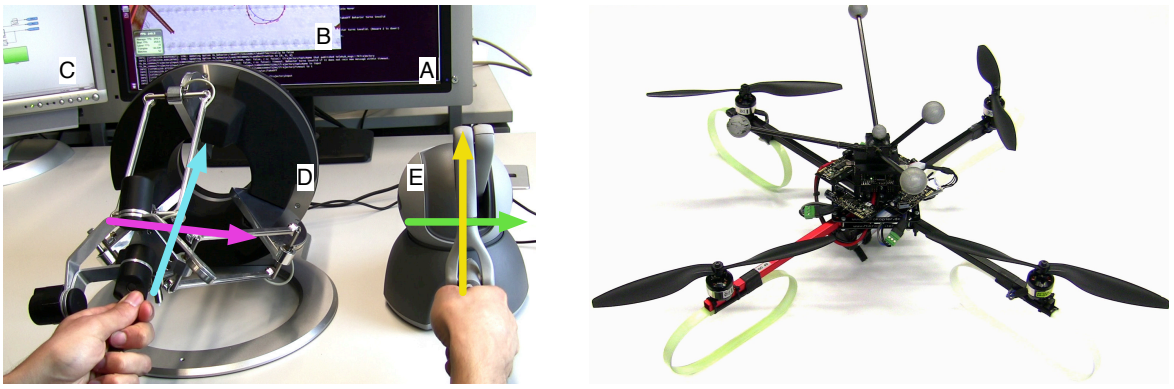


FIGURE V.14: Experimental setup. a) Software components of the framework and input devices: A) TeleKyb; B) SwarmSimX; C) Matlab; D) Omega.6, used to command changes of scale (magenta) and rotations (cyan); E) Phantom Omni, used to command translations along $\vec{X}_{\mathcal{W}}$ (yellow) and along $\vec{Y}_{\mathcal{W}}$ (green). b) The quadrotor used for the experiments.

The second haptic cue $\boldsymbol{e_x}$ is instead representative any mismatch between the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ and the planned path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$. The meaning of this cue is to provide a force feedback that 'guides' the operator so that his/her commands reduce the mismatch between $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ and $\boldsymbol{\gamma}_S(\boldsymbol{x})$. To achieve this result, $\boldsymbol{e_x}$ is taken as the velocity vector $k(\boldsymbol{x} - \boldsymbol{x}_h)$, with $k > 0$, which drives $\boldsymbol{x}_h$ towards $\boldsymbol{x}$.

Force feedback The force $\boldsymbol{\tau}$ corresponding to the two haptic cues $\boldsymbol{e}_{\dot{\boldsymbol{x}}}$ and $\boldsymbol{e_x}$ is

$$\boldsymbol{\tau} = -B\dot{\boldsymbol{q}} - K_M\boldsymbol{q} - K^*(\boldsymbol{e}_{\dot{\boldsymbol{x}}} - \boldsymbol{e_x}) \tag{V.10}$$

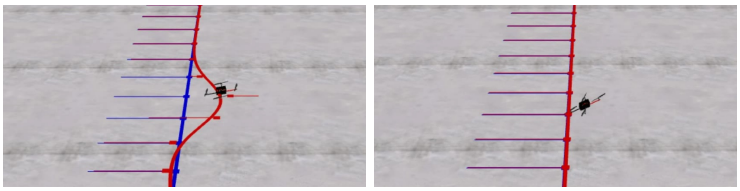
where B is a positive definite damping matrix used to stabilize the device, K_M is a diagonal non-negative matrix used to provide a perception of the distance from the zero-commanded velocity[11], and K^* a diagonal positive definite matrix of gains. As in all bilateral teleoperation applications, the presence of the force feedback $\boldsymbol{\tau}$ may cause unstable behaviors of the haptic interface because of non-modeled dynamics, communication delays and packet losses. In order to guarantee stability despite all these shortcomings, the PSPM approach by [Lee and Huang, 2010] has been adopted to guarantee stability (passivity) of the master side and of the closed-loop teleoperation system.

V.2.6 Simulations and Experiments

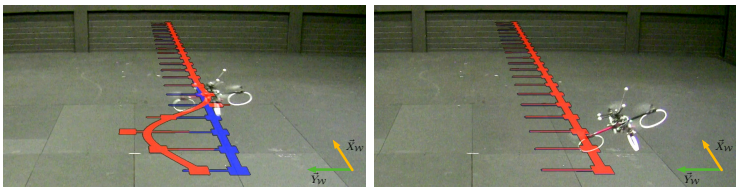
V.2.6.1 Experimental Testbed

The shared planning framework has been tested both in simulation and in experiments with a real robot. In both cases, the two haptic devices shown in Fig. V.14 have been used. The device on the left is an Omega.6[12] with 6 DoF (only 3 DoF are actuated), but only 2 DoF have been used to command changes of scale and rotations of the path with respect to the centre of mass of the control points. The device on the right is a Phantom Omni[13] with 6 DoF (only 3 DoF are actuated), and only 2 DoF have been used to command translations in the plane spanned by $(\vec{X}_{\mathcal{W}}, \vec{Y}_{\mathcal{W}})$.

[11] If this effect is not desired, one can alway disable it by taking $K_M = 0$.
[12] www.forcedimension.com
[13] www.geomagic.com



(A) Snapshot: $\boldsymbol{N}$ on. (B) Snapshot: $\boldsymbol{N}$ off.



(C) Snapshot: $\boldsymbol{N}$ on. (D) Snapshot: $\boldsymbol{N}$ off.

FIGURE V.15: Effects of the projection term $\boldsymbol{N}$. In the snapshots a) and b): the desired path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$, the control points $\boldsymbol{x}_h$ and the velocities $\dot{\boldsymbol{x}}_h$ are depicted as a thick blue line, blue squares and thin blue lines respectively. The actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$, the control points $\boldsymbol{x}$ and the velocities $\dot{\boldsymbol{x}}$ are depicted as a thick red line, red squares and thin red lines respectively. Thanks to the projection term $\boldsymbol{N}$, the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ is automatically modified so as to cope with the quadrotor dynamics (see Fig. (a))

A quadrotor has been used as mobile robot in the simulations and in experiments. Indeed, the use of a quadrotor allows to empirically validate the preliminary assumption made in Sect. V.2.2.1 because of the flatness of its central point $\boldsymbol{p}_r$ (part of the quadrotor flat output).

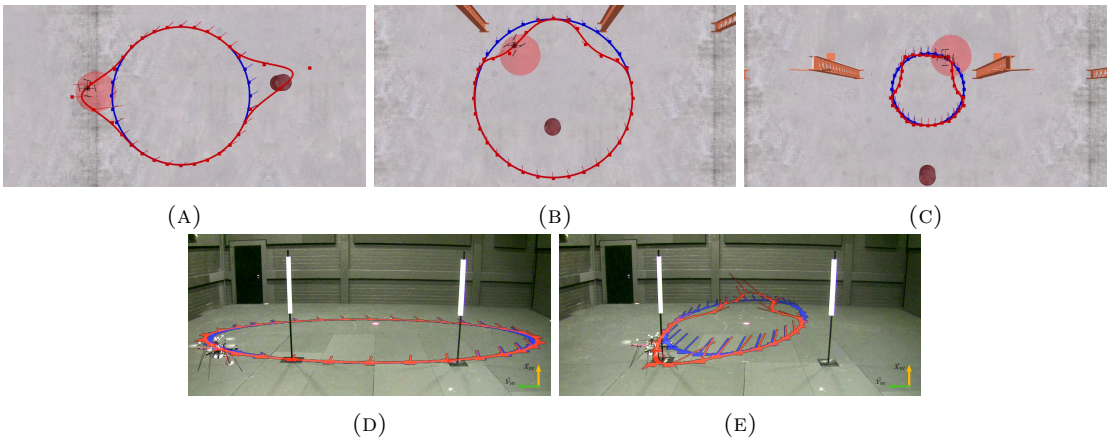
In the following we report some results and features of the approach. In particular, Fig. V.15 shows the effects of using or not using the projection term $\boldsymbol{N}$ for computing local modifications of the path due to the human commands in a simulated environment and by using a real quadrotor robot. As clear form the snapshots, by 'filtering' any trajectory modification via the projector $\boldsymbol{N}$, local feasibility of the modified planned trajectory is preserved.

Figure V.16 shows some examples of local modifications of the planned path due to attractive forces towards PoIs and repulsive forces due to nearby obstacles. These local modifications are automatically generated by the Autonomous Corrector while the human operator steers the quadrotor in the environment.

Finally, Fig. V.17 reports some illustrative details on the replanning algorithm for generating alternative paths in a cluttered environment.

V.3 Bilateral Teleoperation of Multiple Mobile Robots

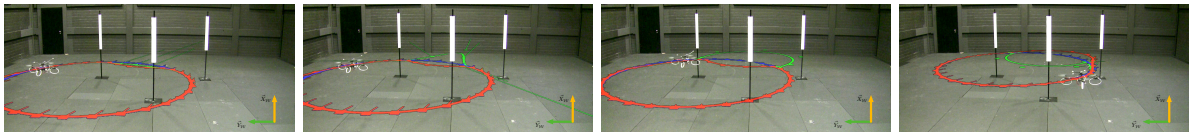
We now focus on the case of bilateral teleoperation of *multiple mobile robots* with, again, a special emphasis for quadrotor UAVs as experimental platforms. Compared with the previous example of Sect. V.2 which involved a *single mobile robot*, the exploitation of multiple mobile robots for accomplishing a mission has several advantages in terms of robustness to single point failures and flexibility (thanks to the internal redundancy). The use of multiple UAVs combines these benefits with the agility and pervasiveness of aerial platforms [Schwager et al., 2011a, Fink et al., 2010]. However, the degree of



(A) (B) (C)

(D) (E)

FIGURE V.16: Top: some snapshots of the effects of the reactive corrections in presence of obstacles and PoIs, while the human operator steers the desired path by commanding translations (2 DoFs) and scalings (1 DoF).



(A) (B) (C) (D)

FIGURE V.17: Illustration of the mechanism for the generation of alternative paths in a cluttered environment.

autonomy of the multi-UAV system should be tuned according to the specificities of the situation under considerations. For regular missions, fully-autonomous UAV systems are often appropriate, but in general the use of *semi-autonomous* groups of UAVs, *supervised* or *partially-controlled* by one or more human operators, is the only viable solution in order to deal with the complexity and unpredictability of real-world scenarios as in, e.g., the case of search and rescue missions or exploration of large/cluttered environments [Ding et al., 2009]. In addition to all this, the human presence is also often mandatory for taking the responsibility of critical decisions in highly risky situations [Murphy et al., 2009].

The goal of this Section is to then illustrate several possibilities for (*i*) allowing a group of UAVs to autonomously control its formation (i.e., internal motion) in a safe and stable manner, and for (*ii*) suitably incorporating some skilled human operators in the control loop. Indeed, contrarily to the previous case of shared control of *single* mobile robots, the multiple robot scenario necessarily requires a much higher level of autonomy for the robot group as the human operator can only focus on global/higher level tasks for the robot group such as steering a single leader robot or the position of the formation centroid. The robot group must then possess enough autonomy for being able to maintain a desired formation and cope with additional requirements (e.g., obstacle avoidance, maintenance of connectivity for the interaction graph) while following the human operator's commands. The examples reported in the following sections will illustrate some possible implementations of these general ideas: in all these examples, quadrotor UAVs will be exploited as robotic platform and different possibilities for the internal formation controller and for the coupling with the human operator will be considered. However, in all cases, the force feedback displayed to the human operator will always be a function of the mismatch between the operator (velocity) commands and their actual realization by the robot group. Indeed, this kind of 'differential' force feedback allows an easy understanding for the human operator of the feasibility of her/his motion commands and of any allowed/denied motion direction in her/his command space.

Figure V.18 illustrates in a general way the architecture for shared control underlying the following examples. Full details of this architecture can be found in [J12], while a brief description follows: a core

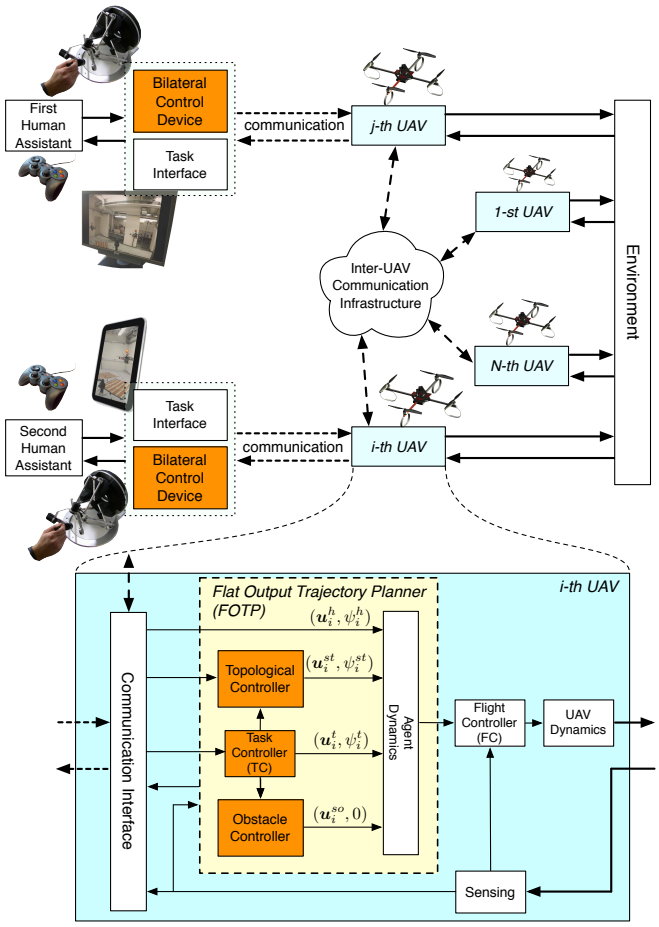


FIGURE V.18: Overall system architecture as seen from the point of view of the generic i-th UAV. The blocks in charge of the supportive features are: the bilateral control device, the topological controller, the obstacle controller, and the agent dynamics.

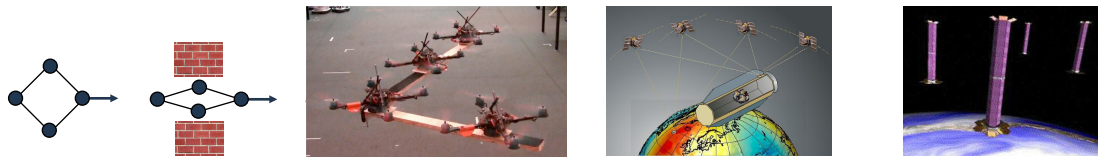
component is the *flat-output trajectory planner* (FOTP) providing the reference flat outputs $(\boldsymbol{p}_i, \psi_i)$, and their derivatives, to the *flight controller* (FC). The flight controller of each UAV acts on the UAV physical control inputs (e.g., the propeller speeds) in order to let the UAV outputs $(\boldsymbol{p}_{\mathcal{B}_i}, \psi_{\mathcal{B}_i})$ track the desired ones $(\boldsymbol{p}_i, \psi_i)$. The FOTP is designed so as to generate the quantities $(\boldsymbol{p}_i(t), \psi_i(t))$ as the time evolution of two virtual systems (henceforth unified under the name "*agent*"): one system for the desired yaw ψ_i (the *yaw-agent*), and one system for the desired position $\boldsymbol{p}_i$ (the *position-agent*).

We only consider kinematic yaw-agents, as this is in practice an acceptable assumption for many quadrotor-like UAVs. This then results in

$$\dot{\psi}_i = w_i, \tag{V.11}$$

where $w_i \in \mathbb{R}$ is the yaw-rate input. On the other hand, we consider to steer the position-agent either at the kinematic (first-order) level, i.e., by commanding a linear velocity $\boldsymbol{u}_i \in \mathbb{R}^3$:

$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i, \tag{V.12}$$



(A) (B) (C) (D)

FIGURE V.19: Examples of formations with *constant* topology where pair-wise geometrical constraints (e.g., distances) are kept among the robot pairs during the whole mission duration.

or at the dynamic (second-order) level, i.e., interpreting the command $\boldsymbol{u}_i \in \mathbb{R}^3$ as a force:

$$\dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \tag{V.13}$$

$$M_i \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i - B_i \boldsymbol{v}_i. \tag{V.14}$$

Here, $\boldsymbol{v}_i \in \mathbb{R}^3$ and $M_i \in \mathbb{R}^{3\times3}$ are the velocity and the (symmetric positive definite) inertia matrix of agent i, respectively, and $B_i \in \mathbb{R}^{3\times3}$ is a positive definite matrix representing an artificial damping added to asymptotically stabilize the behavior of the agent and also take into account typical physical phenomena such as wind/atmosphere drag.

Modeling UAVs as kinematic agents is a common assumption in the multi-robot literature (e.g., similar assumptions have been made in [Schwager et al., 2011a, Fink et al., 2010]). Due to their higher complexity, dynamic agents are less commonly adopted. Nevertheless dynamic agents provide a better approximation of the actual UAV dynamics, and therefore are more appropriate whenever the dynamical properties of the UAVs are more 'excited', as in, e.g., the case of intermittent interactions (see, e.g., the next variable topology cases). The interested reader can also find in [Schwager et al., 2011b] a comparison of stability and performance issues for a network of kinematic and dynamic agents. Finally, by considering dynamic agents one also gains the possibility to rely upon the *tank machinery* for passifying the slave side or the overall teleoperation channel as briefly explained in Sect. V.1.

We now proceed to illustrate several examples of shared control for multiple robots by considering the cases of *constant/fixed* topology and *unconstrained topology* for the robot formation. Subsequently, selected results of a user evaluation of some of proposed schemes will be presented.

In the following discussion we will purposely omit many technical details and inner machinery of the distributed multi-robot coordination algorithms exploited in the reported shared control scenarios. Indeed, these technical details will be more thoroughly addressed in Chapter VII to which the reader is referred.

V.3.1 Shared Control with Constant topology

Hereafter, *constant topology* refers to a UAV formation in which pair-wise geometrical constraints (e.g., relative positions, distances or bearing angles) are specified and maintained during the whole mission duration *without* any modification of the interacting pairs (i.e., of which geometrical constraints are actively monitored/controlled during the robot motion). Figure V.19(A) gives a pictorial view of a typical example of a constant topology formation: pair-wise geometrical constraints (distance constraints in this case) are maintaned over the whole mission duration. However, the formation shape may still have some degree of 'flexibility' due to the presence of external effects, such as repulsive obstacle avoidance forces as in the reported example. The human operator is allowed to steer some of the free DOFs of the formation,

i.e., those motion directions which do not affect maintenance of the inter-robot (distance) constraints (in the illustrative example of Fig. V.19(A), a planar translation and rotation could be steered by the human operator without affecting the formation shape).

The need for constant-topology (i.e., fixed) UAV-formations naturally arises from the specifications of many UAV applications, e.g., for interferometry, transportation [Fink et al., 2010], for guaranteeing inter-UAV visibility or environmental coverage [Schwager et al., 2011a, Ding et al., 2009], see Figs. V.19(B–D). The desired formation may be an output of the task controller, or directly specified by the human operators.

While, in principle several geometrical constraints can be considered for specifying a UAV formation, common choices are relative position, distance and/or bearing constraints motivated by the available sensing technology (e.g., laser scanners or onboard cameras). In any case, once the desired pair-wise geometrical constraints have been satisfied, the UAV formation will still possesses (in general) some free DOFs. For instance, in the case of (a suitable choice of) pair-wise distance constraints in 3D, a UAV formation can be shown to still possess 6 free DOFs: a collective translation of the centroid and a rotation about any axis in space [J16]. In case of pair-wise bearing constraints in 3D, the UAV formation still possesses 5 DOFs, namely, translation of the centroid, synchronized rotation about the vertical axis, and expansion [J11].
Presence of these free DOFs can be clearly exploited for accomplishing additional tasks of interest besides the formation control requirement. In our setting, one can naturally leverage these free DOFS for allowing an *easy interfacing* between a human operator and the underlying formation controller. In particular, the formation controller can represent the *autonomous* component of the shared control architecture by (*i*) steering the UAV configuration towards a *target manifold*, i.e., the set of all configurations satisfying the desired pair-wise geometrical constraints and (*ii*) meeting additional local requirements such as obstacle and inter-robot collision avoidance. At the same time, the human operator can easily act on the UAV formation by *sliding tangent* to the target manifold without affecting the desired constraints, i.e., while being *orthogonal* to the formation control action. For instance, as an illustration, the operator commands' could displace the position of the UAV centroid, or command a global rotation in space, without changing the desired inter-distances. Finally, a force feedback signal can be generated as a function of the instantaneous mismatch between human commands and their actual realization, thus providing the usual information of how well the human commands are being executed by the robot group.

We will now consider two specific cases of constant topology UAV formations defined in terms of distance constraints and bearing constraints in 3D. In all the following, the UAVs are modeled as kinematic agents as in (V.11–V.12). Generally speaking, the typical approaches for implementing formation control algorithms are based on artificial potentials that generate ($\boldsymbol{u}_i$, w_i) as the gradients of energy-like functions having a minimum at the desired inter-agent value (see [J16] for one of these cases). However, the drawback of artificial potential approaches is the presence of local minima. In this regard, the work in [J11] presents an interesting different strategy not based on artificial potentials and almost globally convergent.

V.3.1.1 Distance Constraints

As explained, by regulating a suitable set of inter-distance constraints makes the UAV group (when modeled as kinematic agents) to behave as a 'flexible' rigid body free to translate and rotate along any

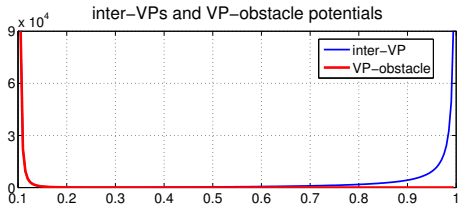


FIGURE V.20: Examples of φ^c_{ij} and φ^o_{ir}, to ensure inter-robot and obstacle collision distance > 0.1 and inter-robot separation distance < 1.

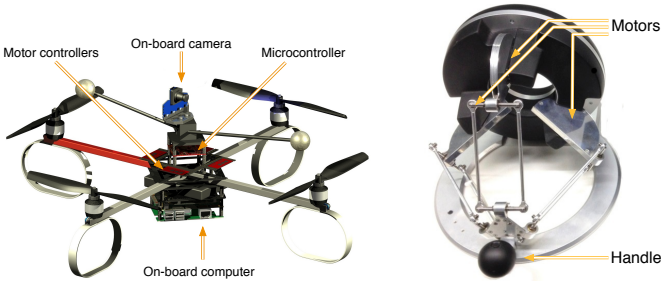


FIGURE V.21: On the left: our quadrotor UAV setup with its avionics parts. On the right: haptic interface used in the experimental testbed.

axis in space (the 6-DOF motions in the null-space of the formation constraint). The control approach taken for enforcing the distance formation constraint and allowing external 'forces' (e.g., environment and human operator) to act on the UAV group is to consider three control terms [J16]

$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}^t_i + \boldsymbol{u}^c_i + \boldsymbol{u}^o_i \tag{V.15}$$

where $\boldsymbol{u}^c_i$ embeds the inter-distance regulation action by following the anti-gradient of some suitable artificial potential function $\phi^c_{ij}(\|\boldsymbol{p}_i - \boldsymbol{p}_j\|)$, $\boldsymbol{u}^o_i$ represents an obstacle avoidance action generated, again, as the anti-gradient of some $\phi^o_{ir}(\|\boldsymbol{p}_i - \boldsymbol{p}^o_r\|)$ for all the obstacle points $\boldsymbol{p}^o_r$, and $\boldsymbol{u}^t_i$ represents the human operator command meant to affect the formation centroid. Figure V.20 shows an illustrative example of the shape of the potential functions.

By this choice (and under suitable assumptions about the boundedness of the operator commands and other technicalities), the UAV motion group then behaves as a coherent semi-rigid body which can partially deform under the action of external forces (the obstacle repulsive forces), and which can be easily steered as a whole by a human operator directly commanding the 3D velocity of its centroid. In this settings, the force feedback returned to the human operator is proportional to the mismatch between her/his commands $\boldsymbol{u}^t_i$ and the averaged UAV 3D velocity. Consequently, in the absence of any environmental force, the force feedback roughly informs about the inertia of the whole group (any delay in reaching and maintaining the commanded velocity), while, when environmental forces are present, the force feedback gives an indication of the 'forbidden' motion direction that would take the UAV group in collision with the obstacles. Stability of the overall bilateral teleoperation framework can be guaranteed by relying on the PSPM algorithm for a suitable passification action (see [J16] for all details).

The following plots and pictures illustrate some results of this approach. As in the previous case of single mobile robot teleoperation, quadrotor UAVs were used as actual robotic platform, and a 3-dof haptic device served as master side of the teleoperation system, see Fig. V.21.

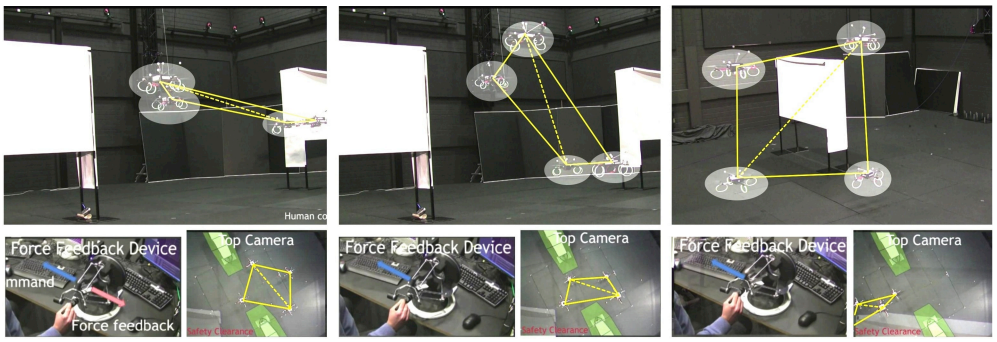


FIGURE V.22: Screenshot from the first experiment. Potentials are designed such that the formation is a square. The human operator is tasked to guide the UAV group into a narrow passage.

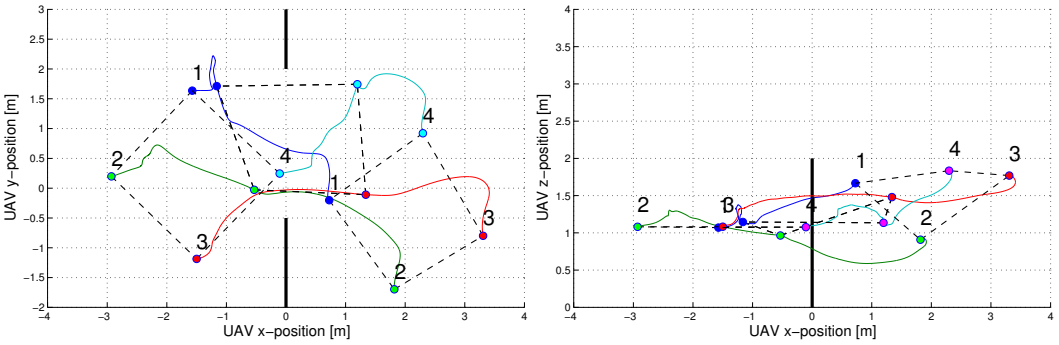


FIGURE V.23: Projections on the XY and XZ planes of the UAV trajectories in the time interval [0 s,10 s]. The black thick line represents the position of the narrow passage. The dashed lines and big dots highlight the formation at 0 s, 5 s, and 10 s.

In a first experiment, the quadrotor group was flown in the cluttered environment shown in Fig. V.22. The inter-distance constraints were chosen so as force the UAV group to lie on a planar square (arbitrarily orientated in space), with the human operator giving a linear velocity command in the direction of the two obstacles. As shown in the snapshots, the UAV group was able to maintain the desired square formation while automatically adjusting its orientation so as to pass through the 'narrow passage' created by the two obstacles. This automatic rotation was caused by the obstacle terms $\boldsymbol{u}_i^o$ in (V.15). At the same time, the discrepancy between the commanded linear velocity for the UAV group and their actual motion (perturbed by the obstacle forces) generated a force feedback signal for the human operator informing about the presence of the obstacles in the environment. This feedback then helped the operator in better guiding the UAV group through the narrow passage. Figure V.23 shows the evolution of the UAV trajectory where the rotation of the square formation can be better appreciated.

Finally, Fig. V.24 reports some screenshots of a second experiment where the UAVs are commanded to fly over an obstacle while keeping a tetrahedron formation.

V.3.1.2 Bearing Constraints

We now move to the second case of a UAV formation defined in terms of relative bearing vectors, that is unit vectors in 3D. The interest in this kind of formation, hereafter also termed 'bearing-formations', lies in the fact that relative bearings can be directly measured by onboard cameras segmenting the position of neighboring UAVs. Therefore, since cameras are a widespread sensing modality for small-scale UAVs (such as quadrotors), the issue of defining, controlling and exploiting UAV bearing-formations has

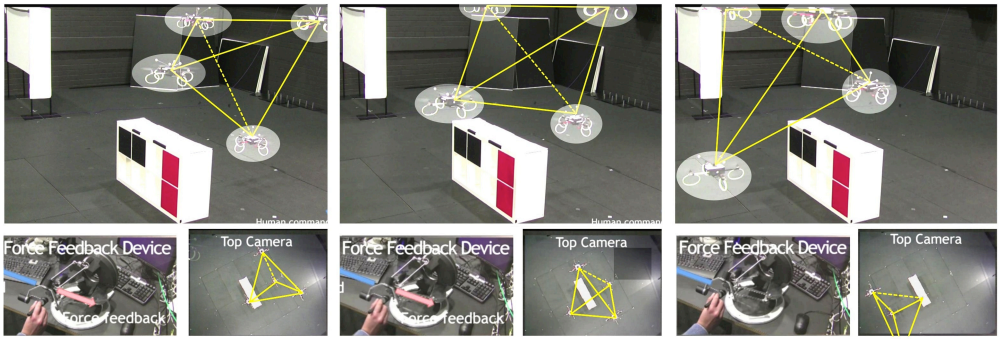


FIGURE V.24: Screenshot from the second experiment. Potentials are designed such that the formation is a tetrahedron. The human user is tasked to guide the UAV group into over a ground-based obstacle.

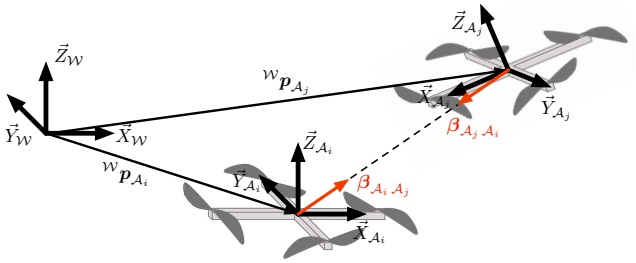


FIGURE V.25: Model of two UAVs (in this case two quadrotors), each of them with a body-fixed frame, and the corresponding relative bearings.

generated a high level of interest over the last years both in the control and robotics community (indeed, modeling and control of bearing formations presents some nice theoretical challenges and subtleties not present in the distance case).

We now proceed to briefly illustrate the definition and control of bearing formations, with all details available in [J11]. On a general note, the overall approach is conceptually similar to the previous case of *distance* constraints: a bearing controller will be in charge of maintaining a desired bearing formation, and the human operator will be allowed to steer the free DOFs of the robot group once all the pair-wise bearings are constrained to their desired values. As explained, these free DOFs, in the case of quadrotor UAVs, consist of a collective translation (3 DOFs), a collective expansion (1 DOF) and a synchronized rotation about a vertical axis (1 DOF). And, again, the force feedback displayed to the human operator will be proportional to the mismatch between the commanded velocities along the five free DOFs and their actual realization by the robot group. Since case of bearing formation control is more involved (and rich) than the previous distance constrains, we will keep the discussion at a more detailed level w.r.t. the previous case.

For our goals, a relative 3D bearing vector between two agents i and j is the unit vector pointing from the origin of agent i to the origin of agent j and expressed in the *local frame* of the measuring agent i. Figure V.25 gives a pictorial representation of these concepts.

A more formal definition can be given as follows: assume, as before, a kinematic model for the agents but expressed in the local body frame, that is

$$\begin{pmatrix} \dot{\boldsymbol{p}}_i \\ \dot{\psi}_i \end{pmatrix} = \begin{pmatrix} R_i & \mathbf{0}_3 \\ \mathbf{0}_3^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_i \\ w_i \end{pmatrix}, \tag{V.16}$$

for any agent i, where $\mathbf{0}_3 = (0\,0\,0)^T$, R_i is the rotation matrix between world frame and body frame of agent i, and the *body-frame* linear velocity $\boldsymbol{u}_i \in \mathbb{R}^3$ and yaw-rate $w_i \in \mathbb{R}$ are the velocity inputs. Vector $\boldsymbol{q}_i = (\boldsymbol{p}_i, \psi_i) \in \mathbb{R}^3 \times \mathbb{S}^1$ is defined as the i-th agent *configuration*, and vector $\boldsymbol{q} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N) \in (\mathbb{R}^3 \times \mathbb{S}^1)^N$ denotes the collection of configurations of all the agents. The direction between the i-th and j-th agent (seen from the body frame of the i-th agent) is defined as *agent relative bearing* and expressed as

$$\boldsymbol{\beta}_{ij}(\boldsymbol{q}) = \boldsymbol{\beta}_{ij}(\boldsymbol{q}_i, \boldsymbol{p}_j) = \frac{{}^i R\, \boldsymbol{p}_{ij}}{\delta_{ij}} \in \mathbb{S}^2, \tag{V.17}$$

where $\boldsymbol{p}_{ij} = \boldsymbol{p}_j - \boldsymbol{p}_i$, and

$$\delta_{ij}(\boldsymbol{q}) = \delta_{ij}(\boldsymbol{p}_i, \boldsymbol{p}_j) = \|\boldsymbol{p}_j - \boldsymbol{p}_i\| \tag{V.18}$$

is the *inter-distance* between agents i and j.

Let now $\mathcal{E}$ represent the collection of the all pairs $(i,\, j)$ for which agent i measures agent j (i.e., the edge set of the measuring/interaction graph, in the language of graph theory), and $\mathcal{T}$ the set of all possible pairs $(i,\, j)$, with, clearly, $\mathcal{E} \subseteq \mathcal{T}$. In a nutshell, the goal of the bearing controller is to steer all the relative bearings $\beta_{ij}(\boldsymbol{q})$, $(i,\, j) \in \mathcal{T}$ towards some desired values $\boldsymbol{\beta}^d_{ij}$, i.e.,

$$\boldsymbol{\beta}(\boldsymbol{q}) = (\ldots,\, \boldsymbol{\beta}_{ij}(\boldsymbol{q}), \ldots)_{(i,\, j)\in\mathcal{T}} \to \boldsymbol{\beta}^d = (\ldots,\, \boldsymbol{\beta}_{ij}(\boldsymbol{q}), \ldots)_{(i,\, j)\in\mathcal{T}}$$

by *only* exploiting the *measured* bearings $(\ldots,\, \boldsymbol{\beta}_{ij}(\boldsymbol{q}), \ldots)_{(i,\, j)\in\mathcal{E}}$. This goal can be achieved if the agent formation (i.e., the agent configuration in space *together with* the edge set $\mathcal{E}$) is, roughly speaking, *rigid*, see [J11]. Rigidity of a formation is a very general topic that has generated quite some interest in the control and robotics community and its definition, properties and implications go well beyond the scope of this report. A nice overview for a broad audience can be found in [Anderson et al., 2008]. For our goals, formation rigidity can be thought as follows: a formation is rigid if, by maintaining the geometric constraints over the edge set $\mathcal{E}$ (the available pair-wise measures), the only allowed collective motion are those that one would have obtained by maintaining the constraints over *all* the possible robot pairs in $\mathcal{T}$. Rigidity of a formation entails several useful properties for formation control and coopefrative localization. For instance, it allows to lower the complexity of any formation controller (both in terms of measured quantities, memory consumption and computation loads) reducing a $O(N^2)$ problem ($|\mathcal{T}| = N(N-1)$ for directed graphs) into a $O(N)$ problem ($|\mathcal{E}|$ can be shown to be $O(N)$ for all typical cases).

Because of these considerations, it is then desirable to use a *minimal* number of controlled and measured relative bearings for reducing at most the cardinality of $\mathcal{E}$ while retaining rigidity of the formation. One can then talk about the minimality for rigid bearing-formations, i.e., of the minimal cardinality of $|\mathcal{E}|$ needed to define a rigid bearing-formation. In [J11] it is shown that, for the case of 3D bearing formations for quadrotor UAVs, it is possible to construct a *minimally-linear rigid bearing formation* by choosing two leader/beacon agents, and by suitably selecting the edges present in $\mathcal{E}$ (i.e., which measurements need to be available among the UAV pairs) at *almost* all agent configurations (i.e., neglecting some special alignments among agents or other 'zero-measure' configurations). By following the procedure described in [J11], the cardinality of a minimally rigid bearing formation results $|\mathcal{E}| = 4N - 5$ (thus, linear in N as expected) and Fig. V.26 reports an illustrative example of a minimally rigid formation for five agents with agents 1 and 2 taken as leaders.

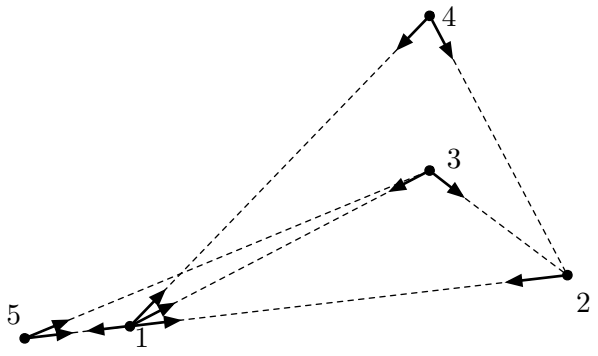


FIGURE V.26: Construction of the minimally-linear rigid bearing-formation described in [J11] for the case of 5 agents. In this case, one can construct $\mathcal{E}$ as $\mathcal{E} = \{(1,2),(1,3),(1,4),(1,5),(2,1),(3,1),(4,1),(5,1)\} \cup \{(3,2),(4,2)\} \cup \{(5,3)\}$. Note that $|\mathcal{E}| = 3 \cdot 5 - 4 = 11$ instead of $|\mathcal{T}| = (5-1) \cdot 5 = 20$ (the number of all possible relative bearings).

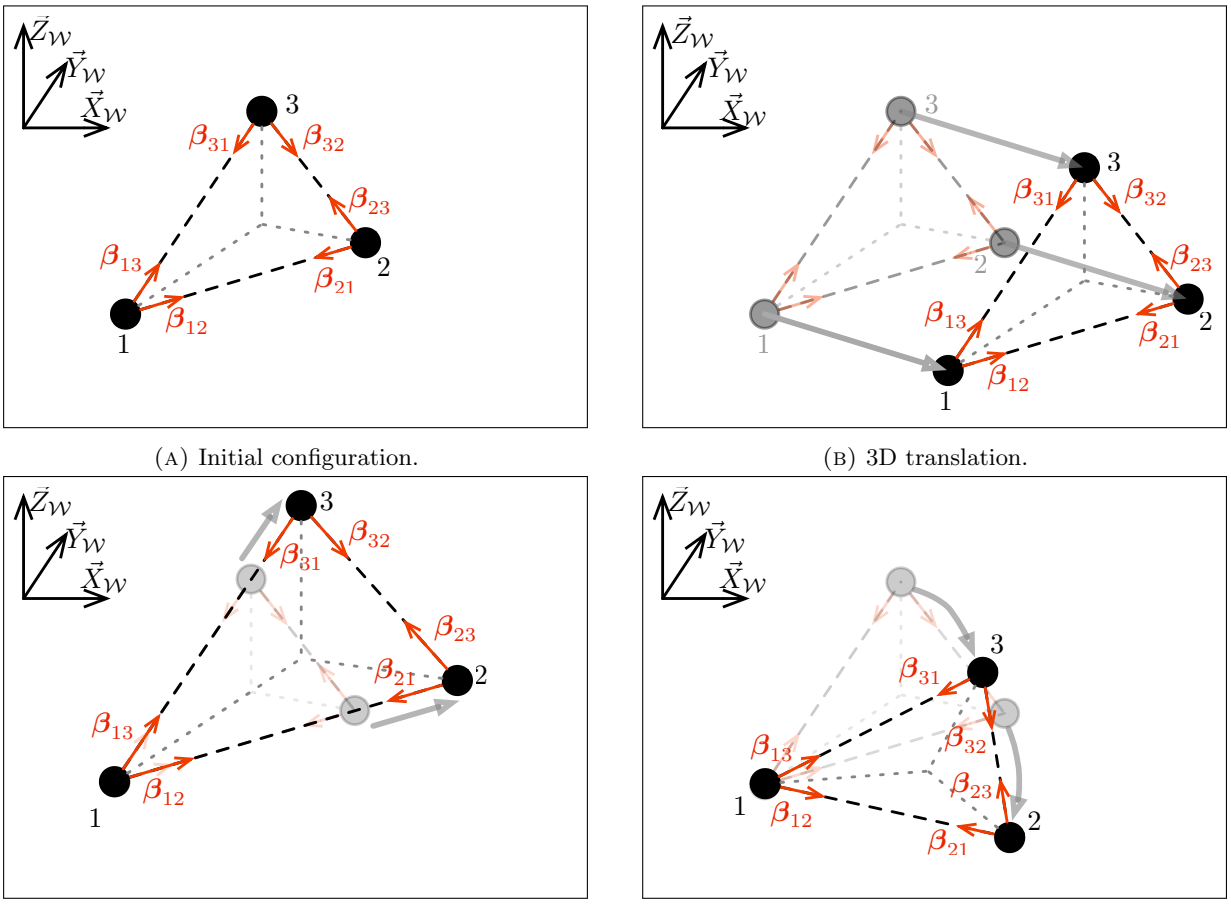
Assuming then that the current and desired bearing formations are (minimally) bearing-rigid, it is further possible to show that the formation control and human steering algorithms can be implemented by splitting the UAV velocity inputs into two 'orthogonal' control terms

$$(\boldsymbol{u}_i, w_i) = (\boldsymbol{u}_i^h, w_i^h) + (\boldsymbol{u}_i^f, w_i^f), \tag{V.19}$$

where the term $(\boldsymbol{u}_i^h, w_i^h)$ represents the action of the human in charge of steering the collective motion of the UAV group, and the term $(\boldsymbol{u}_i^f, w_i^f)$ enforces convergence to the desired bearing formation $\boldsymbol{\beta}^d$. These two control terms are orthogonal in the sense that $(\boldsymbol{u}_i^h, w_i^h)$ is designed by construction to lie on the tangent space of the bearing formation constraint, so that any agent motion induced by $(\boldsymbol{u}_i^h, w_i^h)$ will not change the value of the inter-robot bearings (but it will still produce some 'net' motion for the whole group). The term $(\boldsymbol{u}_i^f, w_i^f)$, on the other hand, will take the current value of the bearing $\boldsymbol{\beta}(\boldsymbol{q})$ towards the desired value $\boldsymbol{\beta}^d$ and keep this value over time against possible disturbances and non-modeled effects (e.g., the quadrotor dynamics not captured by the simple mode (V.16)). The reader is referred to [J11] for further details.

The formation controller $(\boldsymbol{u}_i^f, w_i^f)$ detailed in [J11] possesses the following nice features: it can be implemented by only resorting to the available bearing measurements (i.e., no additional distance measurement is needed), it imposes an almost globally convergent behavior to the closed-loop dynamics, and the induced transient behavior can be exactly characterised. In particular, it can be shown that, while converging towards the desired bearing formation, agent 1 will remain still in space, agent 2 will move along the surface of a sphere towards its goal location, and all the remaining agents will move in a straight line towards their final destinations.

The human steering action $(\boldsymbol{u}_i^h, w_i^h)$ can also be implemented by only resorting to measured bearings apart from a single distance measurement (e.g., the distance among agents 1 and 2). Furthermore, one can decompose $(\boldsymbol{u}_i^h, w_i^h)$ into five terms that can be exploited to *individually* and *independently* actuate the five free DOFs of the bearing formation. Therefore, the operator is given the possibility to command, in an independent way, a collective linear velocity $\boldsymbol{\nu} \in \mathbb{R}^3$, a collective expansion/retraction rate $s \in \mathbb{R}$, and a synchronized vertical angular velocity w for the whole group. Figure V.27 illustrates these motion for the case of three agents.



(A) Initial configuration.

(B) 3D translation.

(C) Expansion.

(D) Synchronized rotation.

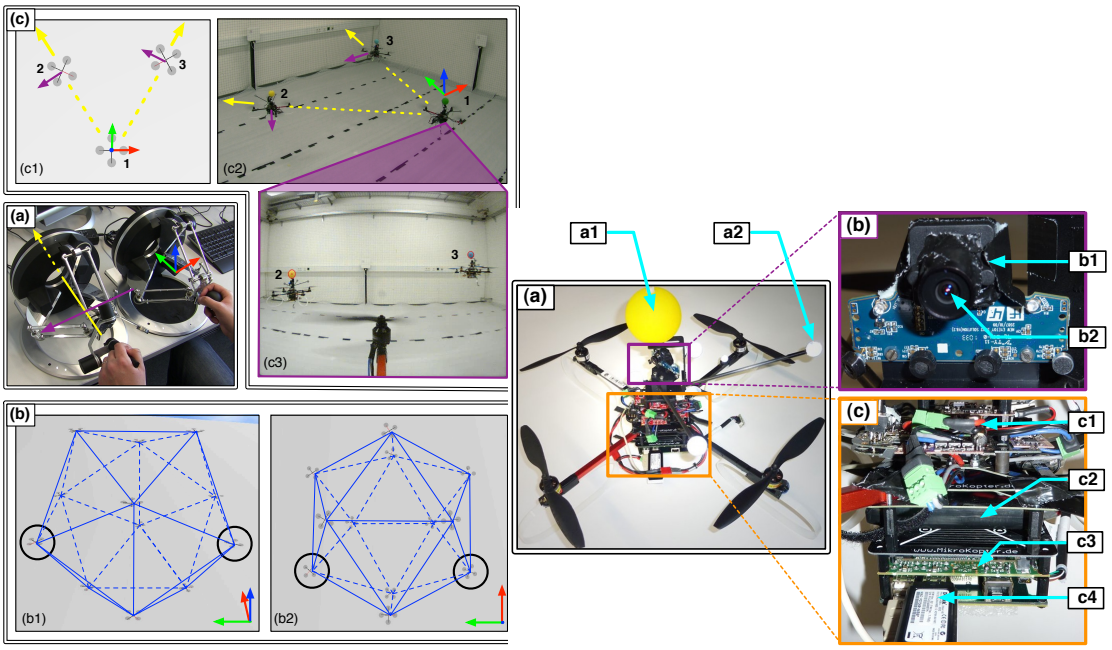
FIGURE V.27: Bearing invariant motions with a formation of three agents.

Finally, as explained one can design a force feedback signal displayed to the human operator proportional to the mismatch between the commanded steering motion (collective translation, expansion or rotation) and their (average) execution by the UAV group while mainanting the desired bearing formation. This feedback will then inform about any discrepancy in commanding the (individual) five free motion motions and their actual realization by the UAVs.

We now proceed to discuss some results of the proposed approach. Figure V.28 gives an illustration of the experimental setup employed for testing the bearing controller and human steering: three quadrotor UAVs equipped with onboard cameras were employed as robotic platforms, and two haptic devices were used by a human operator for actuating the five free DOFs of the bearing formation.

Figures V.29–V.30 report some screenshots taken during the experiment while the quadrotors were flown in the environment guided by the human operator, together with the onboard camera views where the position of the neighboring UAVs was segmented by means of a colored marker.

During part of the experiment, the human operator purposely commanded the individual 5 free motion of the UAV bearing formation (collective translation, expansion and rotation rate) *one at the time* in order to show the *orthogonality* and *decoupling* properties of the two control terms $(\boldsymbol{u}_i^h, w_i^h)$ and $(\boldsymbol{u}_i^f, w_i^f)$ in (V.19). Figure V.31 reports the behavior of several quantities of interest during the experiment where it is possible to appreciate how the quadrotor group did faithfully realize the commanded group motion



(A) (B)

FIGURE V.28: Left: Experimental setup: (a) 3 DoF haptic-feedback devices used to perform the bilateral high-level steering. (b) Simulation environment used to physically simulate the quadrotors: side (b1) and top (b2) views of 12 quadrotors in a icosahedron formation with agents 1 and 2 being highlighted. (c) Real UAV setup with 3 quadrotors: (c1) top-view of the formation in the 3D visualizer; (c2) triangular formation used during the experiments; (c3) onboard camera view of agent 1 with agent 2 and 3 detected by the image processing algorithm.
Right: Quadrotor setup. (a) Quadrotor in its flight configuration: a1) colored sphere used for the visual tracking, a2) reflective marker used for the ground-truth tracking system (b) Camera setup: b1) Consumer-market camera, b2) 140° lens. (c) Computational setup: c1) Microcontroller and IMU, c2) Battery, c3) GNU-Linux PC Board, c4) Wireless adapter.

while always maintaining the desired bearing formation as expected. Additional results and plots can be found in [J11].

V.3.2 Shared Control with Unconstrained topology

We now address the 'complementary' case of shared control of a multi-robot formation with *unconstrained* topology as opposed to the previous case of *constrained* ones. Intuitively, the main difference w.r.t. the illustrative case of Fig. V.19 is that the geometrical pair-wise constraints are now left free to be created/lost online as a function of the current group state. This modification makes it possible to consider sensing and/or communication limitations in the coordination algorithms: for instance, one can accommodate the possibility of gaining/losing relative measurements because of *maximum range* constraints or *occluded line-of-sight visibility*, thus rendering the robot group much more flexible when navigating in difficult environment. Indeed, typical applications in this context are (*i*) navigation in cluttered environments, where an unconstrained interaction topology is much more adaptable than a fixed one, and (*ii*) multi-robot exploration, where UAVs frequently need to divide or gather themselves into smaller or larger groups (as opposite to the previous constant topology case where the focus was on precise relative positioning for, e.g., optimal data collecting).

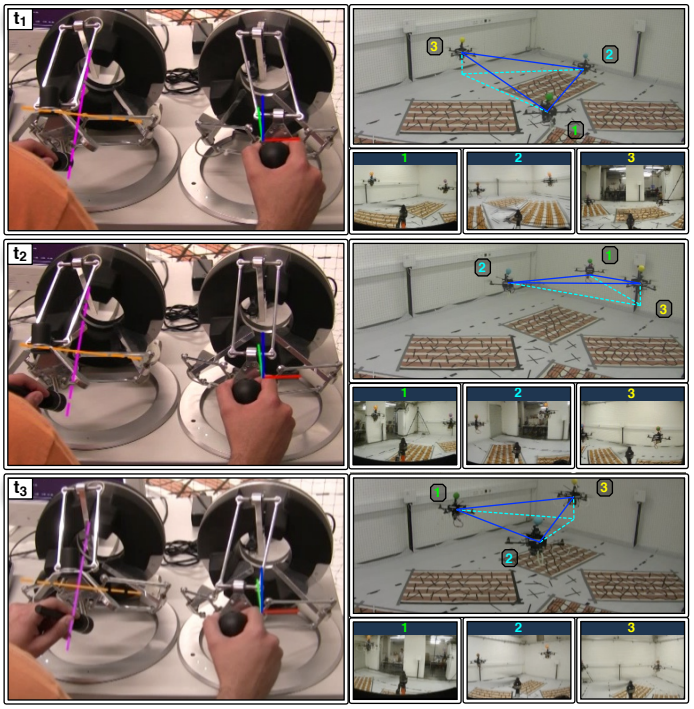
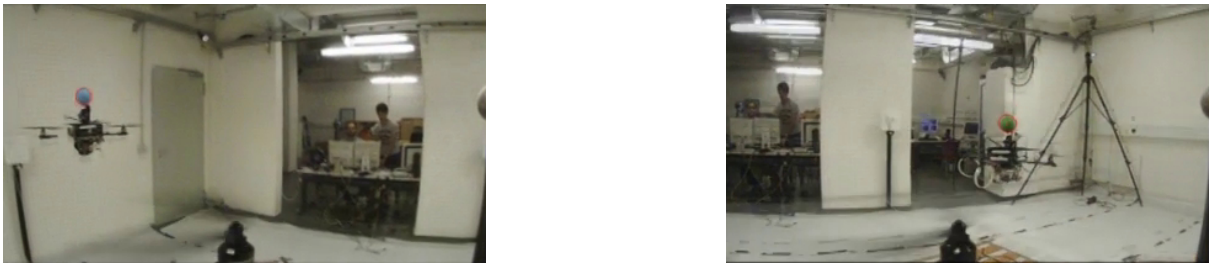


FIGURE V.29: Three snapshots of the experiment with 3 quadrotors. Pictures on the left show the haptic interfaces used to command the group motion and receive suitable haptic cues. Pictures on the right report the corresponding external views of the formation and, superimposed, the local views from the onboard cameras of each UAV.



(A) (B)

FIGURE V.30: Experiment with scanning: screenshots from onboard camera. a) Blue UAV acquired. b) Green UAV acquired.

Figure V.32 illustrates, in an informal way, the intended concept of shared control with an unconstrained topology for the robot group navigating a cluttered environment (to be compared against Fig. V.19). Assuming maximum range and non-occluded visibility constraints for the possibility of measuring neighboring agents, the picture shows the intended concept: the UAV group moves to the right because of an external motion command (e.g., originated from a human operator) applied to a *leader* agent, while the other agents in the group follow the leader agent by means of some 'loose' spring-like coupling that can be broken or created at runtime due to the sensing constraints. This way, the group is able to pass through a narrow passage by automatically rearranging its internal formation (a feature not possible with the approaches detailed in the previous section).

We will now focus on the description of two possible scenarios for implementing a shared control architecture in case of an *unconstrained* topology: specifically, we will first analyze a quite general framework for allowing a switching topology for the interaction graph with spring-like couplings among neighboring

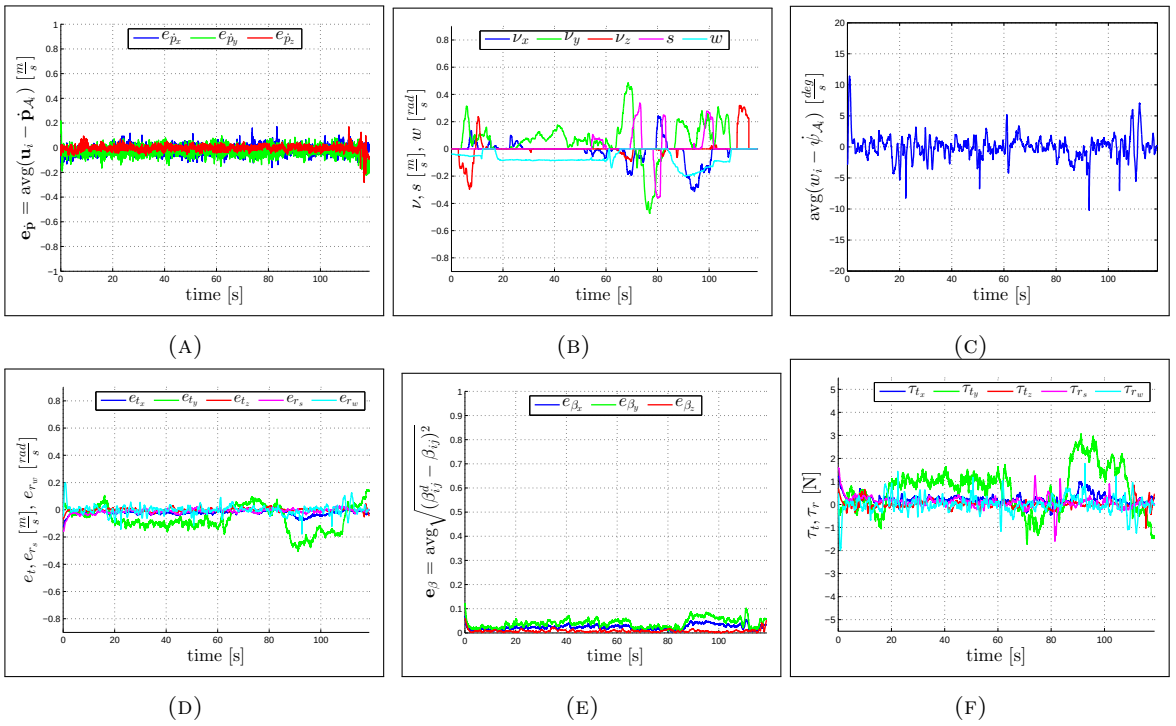


FIGURE V.31: Results with a group of 3 real quadrotors. a),c): Average velocity and yaw rate tracking errors; b): Operator commands: translation velocity, expansion rate and rotation speed; d): Mismatch between commands and executions; f): Force feedback; e): Mean square error w.r.t. desired bearing-formation.

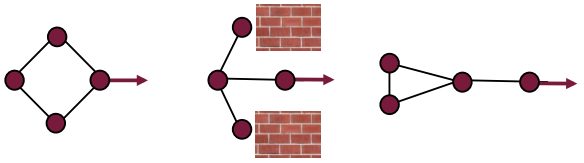


FIGURE V.32: A group of four robots navigates in a cluttered environment under the action of a human operator in a shared control framework. Sensing constraints (maximum range and visibility) determine the possible interactions among the agents with possible gain/loss of links over time. This possibility makes it easier for the group to navigate among obstacles as the group shape is not overconstrained and can adapt at runtime to the environment

pairs (meant to implement a 'loose' aggregation of the robots during motion). As we will see, this general modeling approach allows to capture many possible scenarios and constraints with (*i*) a guaranteed stable behavior for the robot group (despite the possible gain/loss of neighbors during motion) and (*ii*) the possibility for a human operator to command a desired linear velocity for a leader agent that can be used to steer the whole group in the environment. As usual, the force feedback cues for the human operator are then generated as a function of the mismatch between the commanded velocity and the actual leader velocity (which can be, e.g., 'slowed down' by the presence of the other agents in the group with their spring-like couplings, or by surrounding obstacles).

Next, we will move to consider a variation of this general *unconstrained* case in which the interaction topology is still left free to vary over time because of sensing limitation, but under the constraint of maintaining *connectivity* of the graph $\mathcal{G}$ despite the creation or disconnection of individual links. In fact, while flexibility of the formation topology is a desirable feature, connectedness of the underlying graph is often a prerequisite for implementing distributed control/sensing algorithms. This second approach

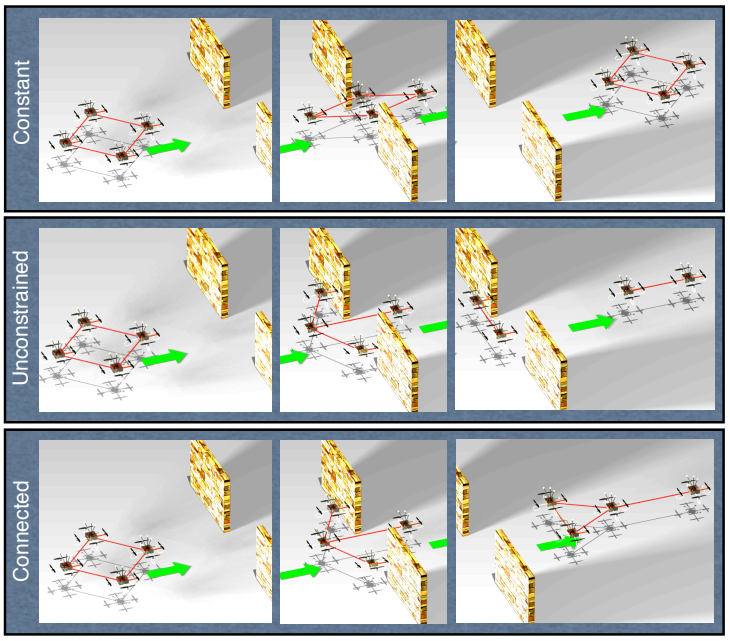


FIGURE V.33: The three topological behaviors considered in the previous Sect. V.3.1 and in this Section: in the first sequence, the interaction graph remains always *constant* regardless of the interaction with the environment; in the second sequence, the graph is *unconstrained*, thereby changing due to the environmental interaction and eventually becoming disconnected; in the third and last sequence, the graph is still allowed to change but under the constraint of remaining *connected*.

then allows to leverage the flexibility of having a 'loose' formation controller able to steer the robot formation in a cluttered environment with high maneuverability, but with the *guarantee* of maintaining a connected interaction graph at all times despite possible loss or gain of edges during motion. For the reader's convenience, Fig. V.33 gives an illustrative view of the three cases of *constant*, *unconstrained* and *connected* topologies during a shared control task.

A final important features of all the presented approaches is that the multi-robot coordination (in both the uncostrained and connected topology cases) is fully *decentralized*, required only local and 1-hop information from local communication. Full details can be found in [J10, J15] and references therein.

V.3.2.1 The general unconstrained case

In all the following derivations, agents will be modeled as the second-order systems (V.13–V.14) rather than as the kinematic agents considered so far. Indeed, as explained, this allows to better capture the dynamical properties of the agents in more 'stressed' conditions as in the case of intermittent interactions. Furthermore, from a control design perspective, considering dynamic agents allows for an easier *energetic* interpretation of the inter-agent and environment interactions ($\sim$ forces) that, in turn, simplifies, to some extent, the stability analysis when considering the whole group motion and its coupling with a human operator in a bilateral teleoperation setting.

Figure V.34 gives a schematic view of the intended bilateral teleoperation architecture for the (general) unconstrained case: the human operator sends a desired velocity command r_M to a leader agent l in the group that tries to execute the motion command while interacting with the other other robots in the group (also called followers). The interactions are intermittent in the sense explained above: a pair-wise

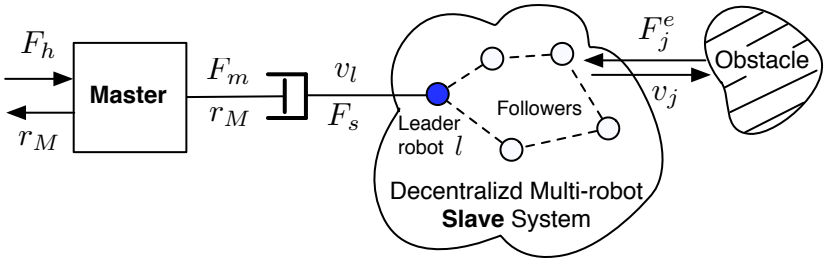


FIGURE V.34: The overall teleoperation system. From left to right: F_h is the human operator force applied to the master device; r_M is a velocity-like quantity which is almost proportional to the position of the haptic device; F_m is the control force applied to the master in order to provide haptic feedback; v_l is the velocity of the leader and F_s is a force applied to the leader to make it follow the desired velocity; finally v_j is the velocity of a generic j-th robot and F_j^{env} is its interaction force with the external environment (obstacles).

link among robots is allowed to be created or lost at runtime as a function of the robot states (e.g., relative distance, or occluded line-of-sight). Since the agents are modeled as second-order systems, the pair-wise interactions can be interpreted as mutual forces exchanged by the agents in order to regulate their relative state (e.g., relative distance, position or bearing). Obstacles in the environment can also apply repulsive forces to the agents in order to guarantee collision avoidance. Finally, the difference among the commanded velocity r_M and the actual leader agent v_l is transformed into a force cue F_m for the human operator.

In order to model *time-varying* interactions among the agents in a general way, in this framework we assume presence of a switching signal for every pair of UAVs $\sigma_{ij}(t) \in \{0, 1\}$ meant to represent the status of the interaction among agents i and j (with $\sigma_{ij} = 1$ indicating presence of the interaction, and $\sigma_{ij} = 0$ otherwise). The time behavior of $\sigma_{ij}(t)$ can model the effect of limited sensing capabilities of the UAVs (e.g., maximum sensing/communication range, occlusions of the line-of-sight), but can also be triggered by the task controller to account for any additional task (e.g., in order to split or join different subgroups). Letting d_{ij} be the interdistance among two robots, and D a maximum range, the only constraint assumed on $\sigma_{ij}(t)$ is that

1. $\sigma_{ij}(t) = 0$, if $d_{ij} > D \in \mathbb{R}^+$;
2. $\sigma_{ij}(t) = \sigma_{ji}(t)$.

This neighboring definition is purposely stated in a very general form in order to account for any additional task requirement independent of the main navigation command. In this sense, item 1) is meant to model a generic limited range capability of onboard sensors and/or communication complexity of the robot network: whatever the task at hand, two agents are never allowed to interact if their interdistance overcomes a certain threshold D. However, it is also possible for $\sigma_{ij}(t) = 0$ even though $d_{ij} \leq D$. This captures the intention of admitting the presence of additional subtasks or constraints the agents may be subject to during their motion. Finally, item 2) represents the fact that we aim for a *symmetric* neighboring condition: two agents always agree on their interaction state.

When two agents are interacting (i.e., $\sigma_{ij}(t) = 1$), we assume that their formation control goal is to maintain a *desired interdistance* $d_{ij}(t) = d_0$ in order to obtain a cohesive behavior during the group motion. This control action can be implemented by mimicking the nonlinear spring behavior depicted

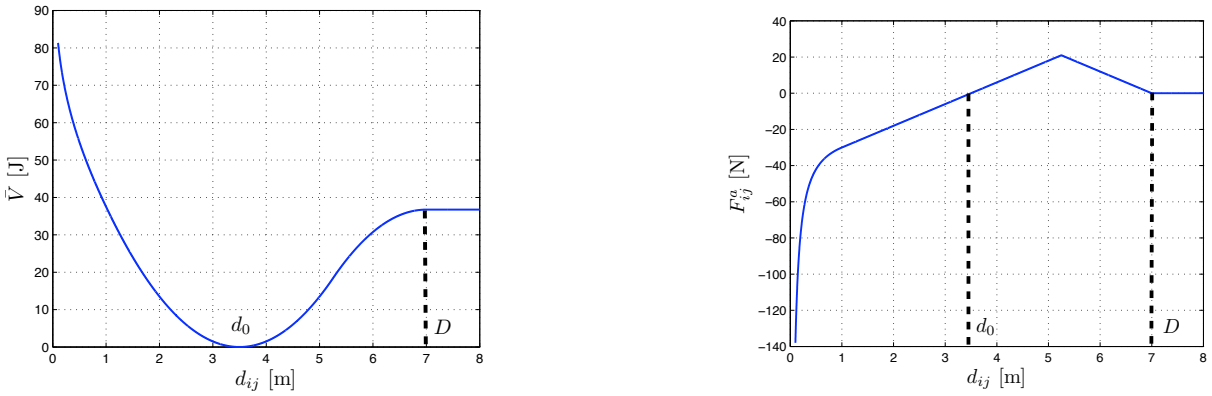


FIGURE V.35: The shape of the interagent potential as a function of the distance (left), and the corresponding coupling force (right)

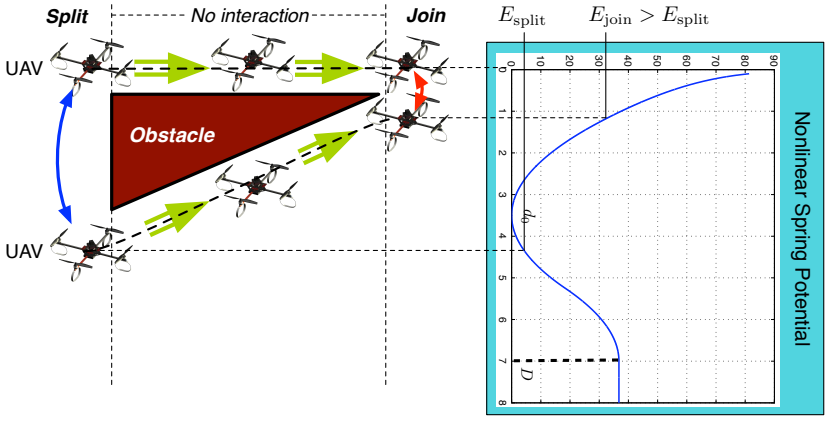


FIGURE V.36: Right: plot of the nonlinear spring potential modeling the agent interaction. Left: when the UAVs split, the energy E_{split} is stored in the spring, while when they join the energy $E_{\text{join}} > E_{\text{split}}$ is needed to implement the new desired coupling. In this case, without proper strategies, an amount $E_{\text{join}} - E_{\text{split}} > 0$ of energy would be introduced into the system, thus violating passivity.

in the potential function of Fig. V.35, i.e., a repulsive action F^a_{ij} if $d_{ij} < d_0$, an attractive action if $d_0 < d_{ij} \leq D$, and a null action if $d_{ij} > D$. By working out the detailed expression of F^a_{ij}, and by compacting several terms, it is then possible to show that such a network of agents coupled among themselves by sping-like forces can be modeled altogether as a dynamical system in port-Hamiltonian form [Secchi et al., 2007, Duindam et al., 2009]

$$\begin{cases} \begin{pmatrix} \dot{p} \\ \dot{x} \end{pmatrix} = \left[\begin{pmatrix} 0 & \mathcal{I}(t) \\ -\mathcal{I}^T(t) & 0 \end{pmatrix} - \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial x} \end{pmatrix} + GF^e \\ v = G^T \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial x} \end{pmatrix} \end{cases} \tag{V.20}$$

where

$$H = \sum_{i=1}^{N} \mathcal{K}_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V(x_{ij}) \tag{V.21}$$

is the total energy of the system, with $\mathcal{K}_i$ being the i-th kinetic energy and $V(x_{ij})$ the potential energy associated to the (i, j) spring coupling, and $\mathcal{I}(t)$ the Incidence matrix associated to the (time-varying) interaction graph defined by the interaction variables $\sigma_{ij}(t)$.

Because of its port-Hamiltonian form, *passivity* (and, thus, *stability*) of the robot group (the slave-side in the teleoperation architecture) can be easily proven for the case of constant topology of the graph $\mathcal{I}(t) = const$ as well as for *splits* among agents (i.e., events at which $\sigma_{ij} = 1 \rightarrow \sigma_{ij} = 0$). However, some special care is required for the join case ($\sigma_{ij} = 0 \rightarrow \sigma_{ij} = 1$) since a join decision can lead to a violation of the slave side passivity: allowing two agents to join means instantaneously switching from a state characterized by no interaction to the presence of an inter-agent interaction that can cause some extra energy to be produced during the join procedure. In fact, in the general case, the relative distance of two agents at the join decision can be different from their relative distance at the split decision, and this can result in a non passive behavior as shown in the illustrative example of Fig. V.36 where some extra energy is produced as the agents join. As explained in detail in [J10], a possible way to mitigate this possible loss of passivity during a join maneuver is to resort to the machinery of *energy Tanks* [Duindam and Stramigioli, 2004] for monitoring the energy flow of the agents and enforce the passivity (energetic) constraint. Specifically, an energy Tank is an additional energy storing element with state $t_i \in \mathbb{R}$ and energy function $T(t_i) = \frac{1}{2}t_i^2$ attached to each agent in the group and implemented so as to store any energy *dissipated* during the agent motion into its local reservoir. This stored energy can then be used later on for any desired goal *without violating the passivity of the system*. For instance, in the case under consideration, the tank reservoirs can be exploited for covering for any extra energy demand generated at a join event[14]. Possible *decentralized* redistribution strategies of the individual Tank energies across the whole group of agents are also possible, as detailed in [J10], in order to exploit at its most the total energy reservoir stored by the group over time.

Thanks to the Tank machinery, and a suitable join procedure, the slave side (the robot group) can then be proven to be a *passive* element w.r.t. external forces generated by the environment (obstacle repulsion) or by the master side (the human operator) also in the case of a *time-varying* topology for the interaction graph. For achieving the desired teleoperation behavior, the master and slave sides can then be joined by using the following interconnection:

$$\begin{cases} F_s = b_T(r_M - v_l) \\ F_m = -b_T(r_M - v_l) \end{cases}, \quad b_T > 0 \tag{V.22}$$

where, again, v_l represents the velocity of a leader agent, r_M is the human velocity command (rouhgly proportional to the master position, and b_T a suitable (damping) gain, see [Chopra et al., 2008]). This is equivalent to joining the master and the leader using a damper which generates a force proportional to the difference of the two velocity-like variables of the master and the leader. The overall teleoperation system is represented in Fig. V.34 then consists of the interconnection of a passive master side, a damper-like interconnection and a passive slave side, so that passivity (and stability) of the overall bilateral interconnection can be established.

With this setting, it is also possible to provide a formal analysis of the steady-state behavior of the teleoperation system so as to, e.g., characterize the agent motion in case of a constant human velocity command, as well as what force cue would be displayed to a human operator while navigating at steady-state. Indeed, by following the analysis of [J10], one can show that at steady-state the human operator needs to apply a force $F_h = F_{ss}$ proportional to the commanded velocity Kr_M by a factor which depends on the number of agents N in the group, and on the magnitude of their damping terms in B_i. For a given N, force F_h will mimic a spring centered on a zero velocity command. Thus, if the number of agents in

[14] We note that the Tank machinery has a very general scope that goes well beyond the particular application considered in this contexts. In general, the use of energy Tanks allows for exploiting at full any *passivity margin* present in a system (e.g., due to any internal dissipation).

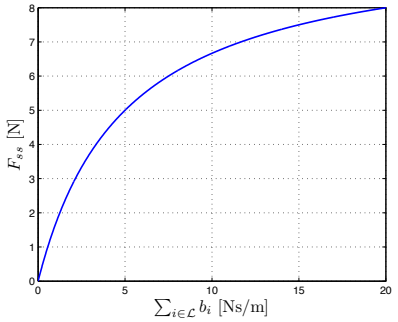


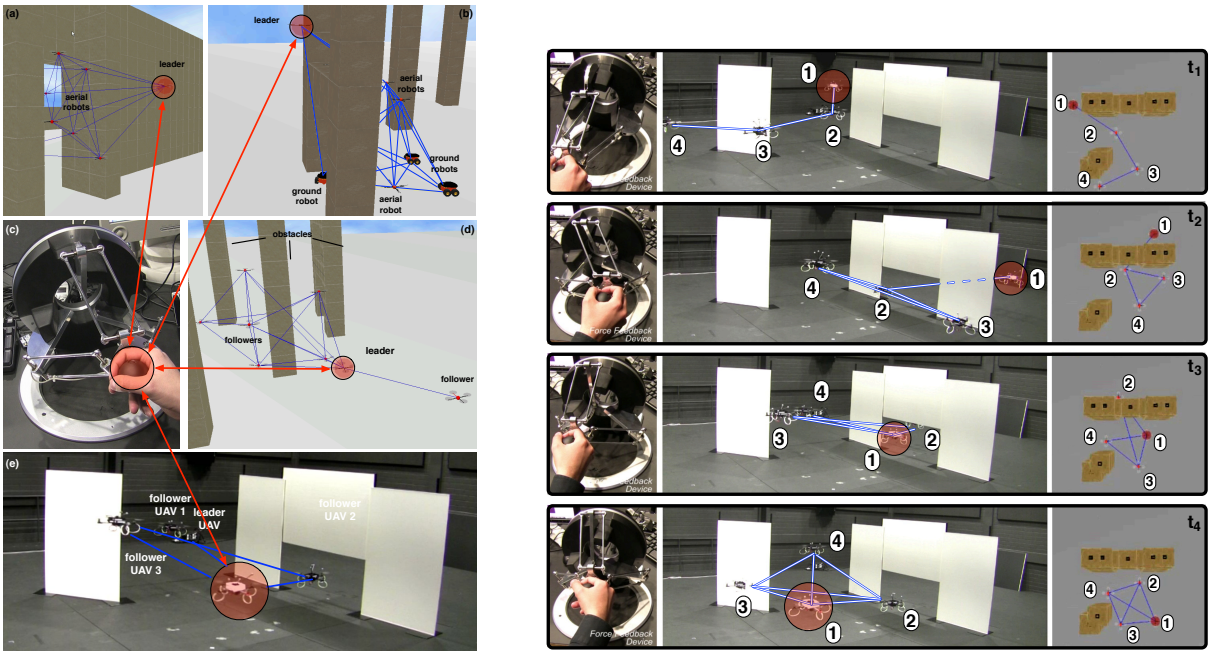
FIGURE V.37: The steady-state force F_{ss} during free-motion as a function of the total damping of the agents in the group. Note how F_{ss} increases with the number of agents N belonging to the group.

N is constant, this force cue will increase/decrease proportionally to the steady-state absolute speed of the whole group: this can provide an haptic cue informative of the overall group velocity. Figure V.37 shows an illustrative behavior of F_h. One can then check that the force F_h needed to keep a constant velocity Kr_M increases with the size of N. This behavior can constitute a beneficial feature of the proposed teleoperation design. In fact, the force F_h resulting from such master/slave velocity mismatch can provide the user with an additional information about the status of the group: if the operator is moving the whole fleet with a constant cruise speed, whenever a robot disconnects from the group, the human operator would feel a decrease in the force needed to keep the master device at r_M. This negative slope in F_h can be informative of the fact that the number of robots in the connected component of leader has decreased. Similarly, when a robot connects to the group, the user would feel a positive slope in F_h, thus informing him/her about the increased number of robots.

As number of simulations and experiments has been conducted to validate this theoretical framework. Figure V.41(A) depicts the testbed: the simulations involved a slave side consisting of a heterogeneous group of robots including both quadrotor UAVs and differentially-driven ground robots (UGVs). The experiments were performed by using four quadrotor UAVs as experimental platform. In all simulations and experiments, presence of a visibility sensor for retrieving the position of neighboring agents was assumed. Therefore, the boolean variable $\sigma_{ij} = 0$ whenever the line of sight between agents i and j was occluded, or the interdistance d_{ij} exceeded the maximum range D. The agents were then forced to split either because of too large interdistances ($d_{ij} > D$), or because of occlusions on their line of sights, an event which could also occur when $d_{ij} < D$. Of course, different choices for deciding splits are possible and this represents only a possible one meant to illustrate the approach. Figure V.41(B) shows some snapshots of the quadrotor navigation in a cluttered environment under the steering of the human operator. It is worth noting how the group topology changes over time because of the sensing (visibility) constraints triggered during motion.

V.3.2.2 The connected case

We now move to consider a variation of the previous case in which a time-varying topology is still possible for the robot group, but under the constraint of maintaining connectivity of the interaction graph $\mathcal{G}$ despite the creation or disconnection of individual links. In fact, while flexibility of the formation topology is a desirable feature, connectedness of the underlying graph is often a prerequisite for implementing any distributed control/sensing algorithms. The idea, illustrated in [J15], stems from the well-known fact that a measure of the connectivity of a graph can be obtained by examining the spectral properties of the



(A) (B)

FIGURE V.38: Left: the Human/Hardware-in-the-Loop simulative and experimental setup. Fig. V.38(c): The Omega-3 force-feedback master device handled by the human operator. Figs. V.38(a,b,d): 3 screenshots of a physically-based simulation in a cluttered environment involving a fleet of either 8 quadrotors (a,d), or 5 quadrotors and 3 differentially-driven ground robots (b). The leader is highlighted by a transparent red ball; inter-agent visibility and distance are considered as neighboring criteria; neighbor agents are linked by blue lines. Fig. V.38(e): a screenshot of an experiment with 4 real quadrotors in a cluttered environment where the leader is highlighted by a transparent red ball.
Right: each row denotes a different moment of a representative experiment. The left column shows the human operator commanding the leader robot with a 3DOF haptic device. The central column represents 4 UAVs in an environment with obstacles (the white walls with a narrow passage). The right column shows the corresponding top-view in a 3D visualizer. The time-varying interaction graph is enlighten by the presence of blue links representing its edges.

so-called *Laplacian* matrix L associated to the graph itself. In particular, a graph is connected if and only if the second smallest eigenvalue λ_2 of L is non-negative [Mesbahi and Egerstedt, 2010]. The eigenvalue λ_2 is then often referred to as the 'connectivity eigenvalue', and a number of works has addressed the problem of keeping $\lambda_2 > 0$ during the group motion in order to preserve connectedness of the graph, see, e.g., [Kim and Mesbahi, 2006, Zavlanos and Pappas, 2007, Stump et al., 2011].

The idea pursued in [J15] is to embed in the elements A_{ij} of the Adjacency matrix associated to the graph the *agent sensing/communication models* representing the actual (physical) ability to exchange mutual information because of the agent relative state. In particular, the (i, j)-th weight A_{ij} is designed as a measure of the quality of the interaction link among agents i and j so that $A_{ij} \to 0$ whenever a sensing/communication constraint is close to be invalid (for instance, as the interdistance d_{ij} among robot pairs starts approaching the maximum threshold D). This definition of *state-dependent* weights renders the adjacency matrix A and, as a consequence, the Laplacian matrix L and its eigenvalues a function of the current state of the robot group. It becomes then possible to devise a gradient-like controller that aims at maximizing λ_2 by acting on the robot velocities.

This idea has been explored in [J15] by embedding in the weights A_{ij} the aforementioned sensing/-communication constraints (maximum range and occluded visibility) *and* some additional terms able to

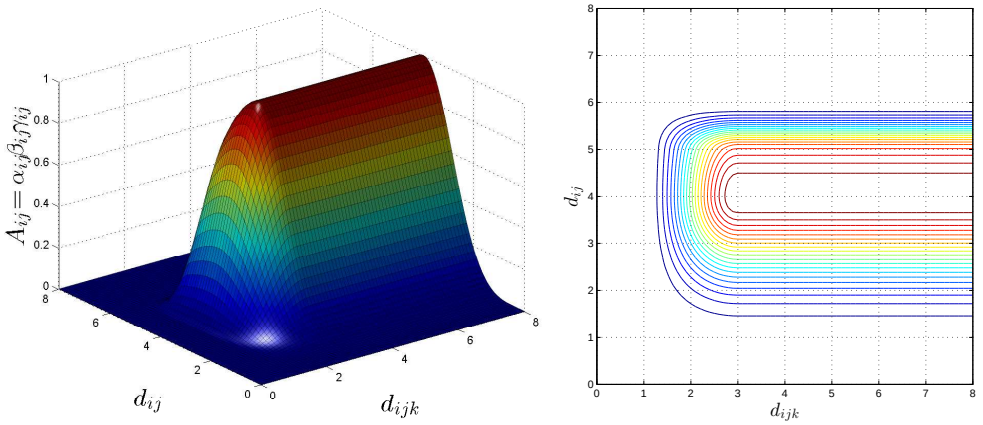


FIGURE V.39: Right: visualization as 3D surface of the total weight A_{ij} as a function of d_{ij} (the inter-robot distance) and d_{ijk} (the distance between the line-of-sight and the closest obstacle point). Right: corresponding contour plot.

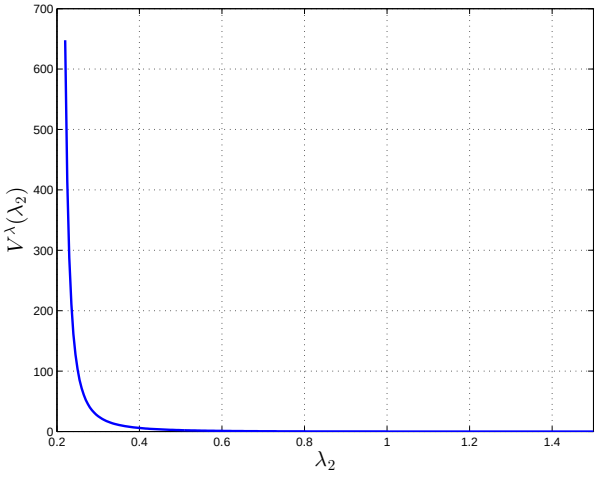


FIGURE V.40: An illustrative shape for $V^{\lambda}(\lambda_2) \geq 0$ with $\lambda_2^{\min} = 0.2$ and $\lambda_2^{\max} = 1$. The shape of $V^{\lambda}(\lambda_2)$ is chosen such that $V^{\lambda}(\lambda_2) \to \infty$ as $\lambda_2 \to \lambda_2^{\min}$, $V^{\lambda}(\lambda_2) \to 0$ (with vanishing slope) as $\lambda_2 \to \lambda_2^{\max}$, and $V^{\lambda}(\lambda_2) \equiv 0$ for $\lambda_2 \geq \lambda_2^{\max}$.

enforce disconnection of the whole graph as a collision with obstacles or other robots in the group is approached. This way, a *unified* control action based on the gradient of λ_2 could be devised for guaranteeing connectivity maintenance despite sensing/communication constraints and, at the same time, a guaranteed collision avoidance with the environment and neighboring robots. Figure V.39 shows a 3D visualization of the weight A_{ij} as a function of d_{ij} (the inter-robot distance) and d_{ijk} (the distance between the line-of-sight and the closest obstacle point o_k).

For the sake of illustration, the overall control strategy is similar to the one detailed in the previous Section with agents modeled as *second-order* systems with external forces playing the role of control inputs. However, a single/unified potential function $V^{\lambda}(\lambda_2)$ is now taken as 'potential' energy function for the group (instead of the several elastic elements V_{ij}). The scalar function $V^{\lambda}(\lambda_2)$ lives in the domain $(\lambda_2^{\min}, \infty)$ and is such that $V^{\lambda}(\lambda_2) \to \infty$ as $\lambda_2 \to \lambda_2^{\min} > 0$, and $V^{\lambda}(\lambda_2) \equiv 0$ if $\lambda_2 \geq \lambda_2^{\max} > \lambda_2^{\min}$, with $\lambda_2^{\max} > \lambda_2^{\min} > 0$ representing desired maximum and minimum values for λ_2. Figure V.40 shows a possible shape for $V^{\lambda}(\lambda_2)$.

By minimizing $V^{\lambda}(\lambda_2)$, the agents are not bound to keep a given fixed topology for the interaction graph $\mathcal{G}$. Creation or deletion of single or multiple links (also concurrently) is fully permitted as long as the current value of the connectivity does not fall below a minimum threshold, i.e., while ensuring that $\lambda_2 > \lambda_2^{\min}$. Minimization of $V^{\lambda}(\lambda_2)$ is formally achieved by letting every agent to follow the anti-gradient of $V^{\lambda}(\lambda_2)$ w.r.t. its own position, that is, by implementing the *connectivity* force

$$F_i^{\lambda} = -\frac{\partial V^{\lambda}(\lambda_2)}{\partial x_i}. \tag{V.23}$$

In [J15] it is shown how the force command F_i^{λ} can be actually computed in a *fully decentralized way* by agent i by resorting to only relative position measurements and to an online estimation $\hat{\lambda}_2$ of the true (but not measurable) λ_2 by leveraging the scheme introduced in [Yang et al., 2010]. As for the stability of the slave side, compared to the previous case no positive jumps of the elastic potential function V^{λ} are possible in this new settings thanks to the definition of the weights A_{ij} which, in summary, *smoothly* vanish as any sensing constraint is violated or met and, thus, prevent any jump in V^{λ}. However, possible non-passive behaviors can still arise because of the estimation scheme used to recover $\hat{\lambda}_2$ in place of the true λ_2. Nevertheless, the tank machinery can, again, be exploited for covering for any non-passive behavior induced by possible estimation errors in recovering $\hat{\lambda}_2$.

Summaring, the proposed *unconstrained* but *connected* shared control architecture possesses the following interesting features:

1. although V^{λ} is a global potential, reflecting global properties (connectivity) of the group, $\hat{F}_i^{\lambda}$ (an estimation of its gradient w.r.t. the i-th agent position) can be computed in a fully decentralized way. The only discrepancies among the true F_i^{λ} and $\hat{F}_i^{\lambda}$ are due to the use of the estimate $\hat{\lambda}_2$ in place of its real value, otherwise $\hat{F}_i^{\lambda}$ is evaluated upon actual information;
2. V^{λ} will grow unbounded as $\lambda_2 \to \lambda_2^{\min} > 0$, thus enforcing connectivity of the group. Note that, during the motion, the agents are fully allowed to break or create links (also concurrently) as long as $\lambda_2 > \lambda_2^{\min}$. Furthermore, the group motion will become *completely unconstrained* whenever $\lambda_2 \geq \lambda_2^{\max}$, since, in this case, the 'connectivity force' will vanish as the potential V^{λ} becomes flat. These features provide large amounts of flexibility to the group topology and geometry, as the agent motion is not forced to maintain a particular (given) graph topology, but is instead allowed to execute additional tasks in parallel to the connectivity maintenance action;
3. because of the various shapes chosen for the weights A_{ij}, minimization of V^{λ} will also enforce inter-agent and obstacle collisions avoidance in a unified way, i.e., the agents are only bound to follow the single control action (V.23) able to realize multiple goals at the same time.

The proposed approach has been experimentally tested by using four quadrotor UAVs as robotic platform. Figure V.41 shows the setup: in this case, two human operators were commanding the velocities of two leader robots via two interconnections in the form of (V.22). The other two robots were autonomously navigating under the influence of the connectivity maintenance action in the cluttered environment.

During motion, and despite the human commands, connectivity was always preserved against the sensing constraints, and obstacle and inter-robot avoidance was guaranteed, as clear from Fig. V.42(A) where the behavior of $V^{\lambda}(t)$ is depicted. Several edges could be created and destroyed (Fig. V.42(B)) without threatening the stability of the slave-side. The force cues displayed to the two human operators represented the mismatch between the commanded velocities and the actual leader velocities. Therefore, the

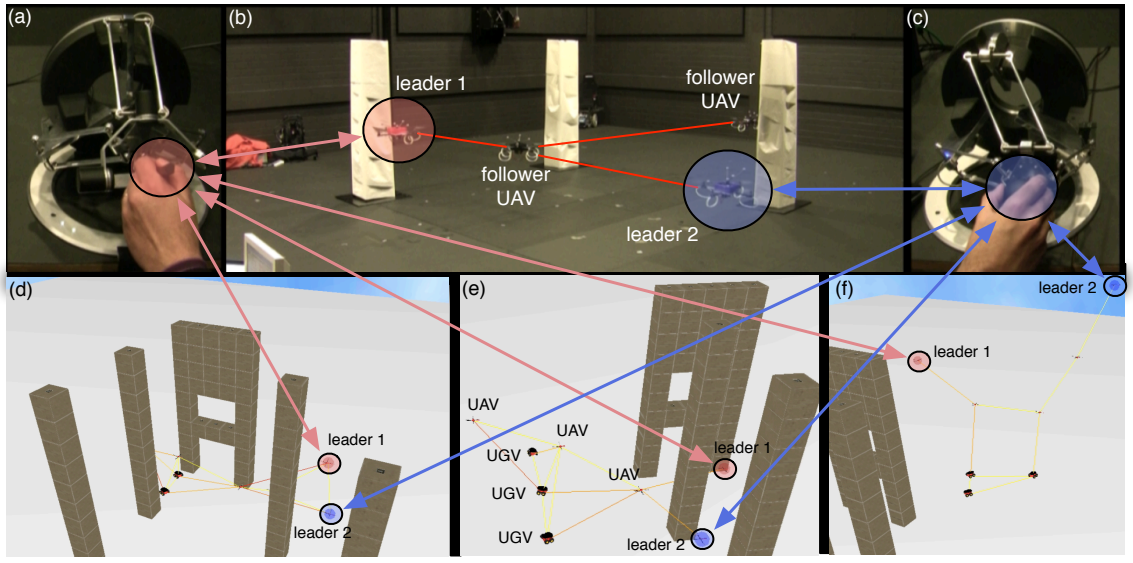
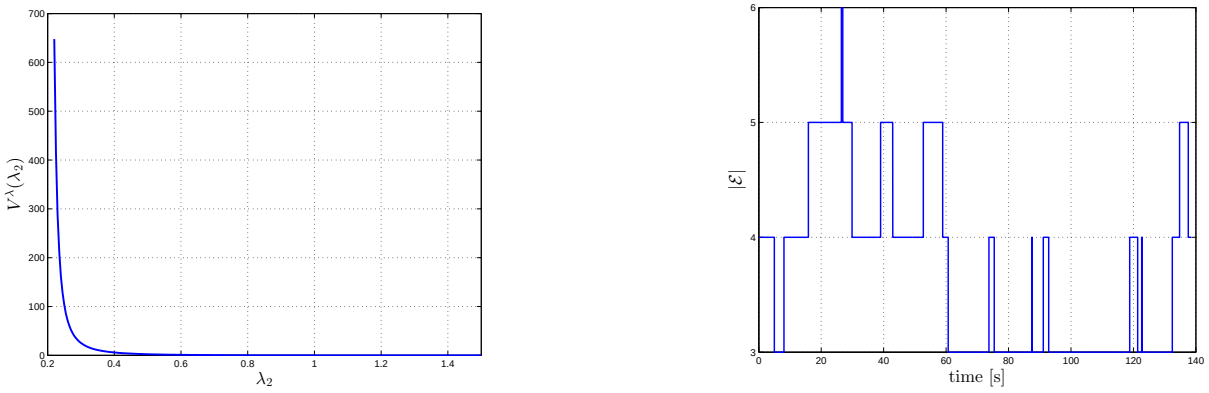


FIGURE V.41: The simulative and experimental setup employed in this work; (a),(c): the Omega.6 and Omega.3 force-feedback devices used by the two human operators; (b): a screenshot of our experimental setup with 4 quadrotor UAVs in a cluttered environment. The two leaders are highlighted with a semi-transparent red and blue circle; (d–f): three screenshots of our simulative setup: 5 quadrotor UAVs and 3 UGVs are maneuvered in a cluttered environment. The two leaders (two quadrotors) are again highlighted with a semi-transparent red and blue circle.



(A) (B)

FIGURE V.42: Left: results of the experiment: behavior of the Generalized Connectivity Potential $V^{\lambda}(\lambda_2(t))$ evaluated on the actual 'ground truth' value $\lambda_2(t)$. Boundedness of $V^{\lambda}(\lambda_2(t))$ confirms that $\lambda_2(t) > \lambda_2^{\min} > 0$ during the robot motion and, equivalently, that the requirements R1–R3 were always satisfied.
Results of the experiment: number of edges $|\mathcal{E}(t)|$ of the interaction graph $\mathcal{G}(t)$ over time. Note again how (i) the graph topology varies over time, and how (ii) the graph reaches in several phases the 'sparsest' possible topology to still ensure connectivity (3 edges for 4 robots).

operators could obtain a feeling on how well the leaders (and, as a consequence, the group) were following their commands: for instance, whenever a velocity command started conflicting with the connectivity maintenance action, a 'drag force' was generated and displayed to the human operator, with a magnitude proportional to the amount of conflict. On the other hand, whenever the operator commanded could be executed without threatening connectivity of the graph, almost no force was be displayed to the operators.

V.4 User Evaluation

In this final section we report selected results of some user studies conducted to investigate the effect of haptic cueing on the human operator's performance when bilaterally controlling the motion of a group of UAVs. These studies were run by considering the *constant* topology case with distance constraints detailed in Sect. V.3.1.1, and full details can be found in [J14]. In particular, two aspects of the human performance were considered in the study: *maneuverability* of group of mobile robots and *perceptual sensitivity* of the remote environment. In our contexts, maneuverability is defined in terms of the ease of the operator in maneuvering the slave robot for achieving an accurate tracking performance. Perceptual sensitivity, on the other hand, is aimed at measuring how well the operator is able to perceive the current state of the UAVs and their surrounding obstacles. Indeed, unlike conventional teleoperation [Lawrence, 1993] (e.g., telesurgery), hard contact between flying UAVs and their physical environment is especially undesirable because this could cause severe damages to the mechanical structure of the robots. Therefore, there are no actual physical forces to be transmitted to the multi-UAV teleoperator and the conventional objectives of force tracking are not applicable. For this reason, pure force tracking may not be a measurable concept that can be directly generalized to the teleoperation of mobile robots. In this sense, transparency [Hashtrudi-Zaad and Salcudean, 2001] is also not a suitable measure for the perceptual ability in the teleoperation of remote UAVs.

In order to obtain a reasonable measure for the maneuverability performance, the position tracking error of each robot in the slave side w.r.t. its commanded path $x_{ref}(t)$ was evaluated. In particular, by defining $\bar{x}(t) = \frac{1}{N}x_i(t)$ as the average of the robot positions (the formation centroid), the following cross-correlation index could be devised

$$CC_{position} = \frac{\int_0^T \bar{x}(t) \cdot x_{ref}(t)dt}{\sqrt{\int_0^T \bar{x}^2(t)dt}\sqrt{\int_0^T {x_{ref}}^2(t)dt}}. \tag{V.24}$$

This index allows to objectively compare the similarity of the actual path ($\bar{x}$) and the desired path (x_{ref}) of the slave robots. At this point, in order to assess the maneuverability performance, we considered the frequency response profile of the human operator, that is, the dynamical relationship between the operator's intended-force ($f_h \in \mathbb{R}^3$) as the input, and the position tracking accuracy ($CC_{position}$) as the output

$$\Phi_{maneuv}(s) := \frac{CC_{position}(s)}{f_h(s)}. \tag{V.25}$$

In this regard, maneuverability increases when f_h is reduced and when position tracking $CC_{position}$ is increased (i.e., the position tracking error is small). In particular a large bandwidth and H_2 norm of Φ_{maneuv} indicate higher level of maneuverability performance.

As for the perceptual sensitivity measure, one could consider a force tracking index for assessing how accurately the master can track the forces transmitted from the slave robot(s). This is, generally, evaluated by estimating the discrepancy between the force of the master and the slave. Unfortunately, force tracking accuracy is not a sensible measure in the context of teleoperation of mobile robots (or vehicles, in particular UAVs) since, as mentioned earlier, UAVs should ideally avoid any direct contact with their environment. Since haptic feedback cues should also serve to increase the human operator's sensitivity to the remote environment, which is inhabited by the controlled UAVs, we instead judged as a reasonable option to consider sensitivity in terms of their ability to perceptually discriminate between physical differences in the remote environment. For this, we considered the smallest change in the

magnitude of a physical variable (e.g., position, force, and impedance) that can be effectively perceived. Such a measure is commonly referred to as the *just noticeable difference* (JND) [Gescheider, 1997]. In particular, the JND was estimated from a psychometric function, which represents the probabilistic distribution of a human observer's perceptual response across a chosen physical variable (i.e., distance to an obstacle). For this, the JND was defined as

$$JND := \frac{\Delta f_{ref}}{f_{ref}}, \tag{V.26}$$

which represents the minimum difference in the reference force that can be effectively perceived by the human operator via force feedback.

As for the force feedback algorithm we compared three possible cueing schemes: in the general setting of rendering a force on the master device with expression $\tau = -B\dot{\boldsymbol{q}} - Ky$, we considered three choices for the signal $y(t)$

1. Force-cue feedback (*Force*):

$$y = y_{\mathrm{F}} := \frac{1}{\lambda N} \sum_{i=1}^{N} u_i^o. \tag{V.27}$$

   This haptic cue plays the role of a repulsive force from the environment w.r.t.the current position of the reference trajectory for the UAV (*environmental force*). It is only related to the difference between the positions where the UAVs are supposed to go and the location of obstacles. Therefore it is not related to the actual positions of the UAVs or to any real contact of the UAVs with their environment;

2. Velocity-cue feedback (*Velocity*)

$$y = y_{\mathrm{V}} := q - \frac{1}{\lambda N} \sum_{i=1}^{N} \dot{x}_i. \tag{V.28}$$

   This haptic cue represents the mismatch between the commanded velocity and the average velocity of real UAVs (*tracking-error*). The mismatch can be caused by the following reasons: *i*) inaccurate model of the UAV, imprecise parameter calibration (e.g., mass of the UAV), or poor quality of the trajectory tracker, *ii*) external disturbances on the UAV motion (e.g., wind, friction, actual contact with the environment), *iii*) presence of obstacles close the group with their repulsive actions. Therefore, this haptic cue provides the operator more information compared to the previous case. Note that the velocity-cue also includes the *Force* cue, albeit 'filtered' by the dynamical behavior of the real UAV positions/velocities rather than directly taken from the desired trajectory;

3. Velocity plus force-cue feedback (*Velocity+Force*)

$$y = y_{\mathrm{V}} + y_{\mathrm{F}}. \tag{V.29}$$

Thirty-two participants (25 males; age range: 20–33 years) took part in this study, which consisted of four experiments on maneuverability and perceptual sensitivity for low and large control gains. Five participants took part in all four experiments while the rest took part in at least one experiment. There were eighteen participants for each experiment and all participants possessed normal or corrected-to-normal eyesight and no physical disability. Figure V.43 depicts the apparatus used in the experiments,

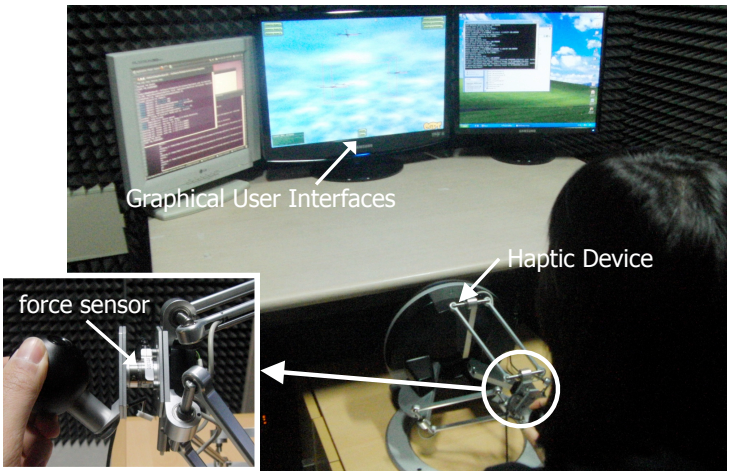
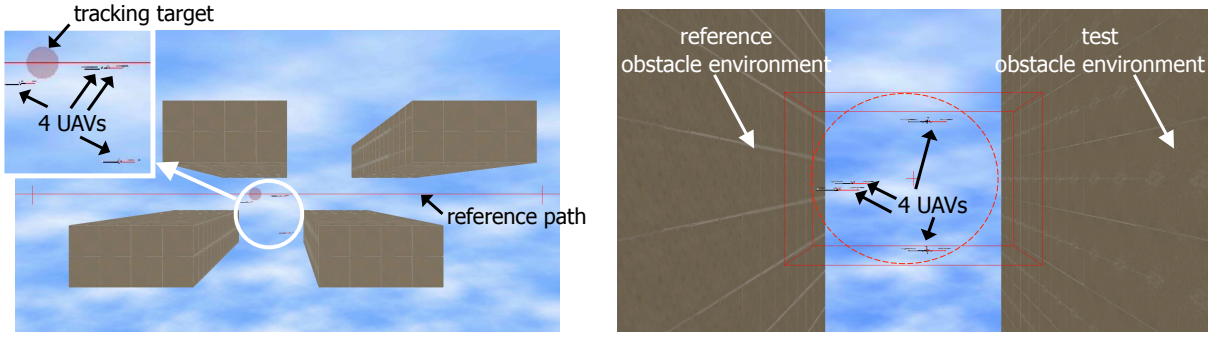


FIGURE V.43: Experimental setup. Subject with haptic device (Omega 3) and Graphical User Interfaces (GUIs).



(A) (B)

FIGURE V.44: Left: screenshot of the maneuverability experiment with visible reference path and obstacles, which were rendered invisible during the actual experiment. Right: Screenshot of test of perceptual sensitivity evaluation with visible obstacles, which were rendered invisible during the actual experiment.

that mainly consisted of a central display and a haptic device. The former presented a virtual environment for a swarm of four UAVs whose flight path could be controlled by the latter.

In the experiment for evaluating the maneuverability, participants were required to maneuver a swarm of UAVs, using the haptic control device. The objective was always to follow a moving target. This moving target preceded the UAV swarm and moved along a pre- assigned path, which was either straight or curved (see Fig. V.44(A)). On the other hand, in the experiment for evaluating the perceptual sensitivity, participants were presented with a swarm of four UAVs on each trial, which were located between two invisible obstacles (see Fig. V.44(B)). Their task was to move the multi-UAV swarm towards the direction of the obstacles and to determine which of the two obstacles returned a stiffer response on the haptic control device. Participants indicated the stiffer of the two obstacles by using a mouse to click on one of two possible buttons that were located at the left or right position of the screen, which corresponded to the respective positions of the obstacles.

The reader is referred to [J14] for a complete analysis and discussion of the results obtained during these studies. Nevertheless, the main findings can be summarized as follows: for what concerns the maneuverability measure, the force cue resulted in a multi-UAV control with the least amount of effort. This can be explained since the Force cue feedback is absent during free motion. In contrast, the Velocity

cue is absent only when UAVs move with a constant velocity vector, which rarely happens. Therefore, the latter case constantly requires the operator to counter-balance the UAVs felt inertia in order to obtain the desired motion. As for the Perceptual Sensitivity measure, it was found that the presence of proximal obstacles was rendered more detectable exploiting a force cue based on the UAVs velocity information w.r.t. force information. This is important to note, especially for instances where visual feedback is degraded or simply unavailable. Finally, an assessment on how to best choose the control parameters (i.e., gain K) was also performed. In this respect, the results of the experiments were unequivocal. Large gain values are necessary in order for haptic feedback cues to have a noticeable effect on performance. The positive contribution of the Force cue to maneuvering performance particularly benefited from larger gain values even if it would be wrong to assume that larger feedback forces will automatically deliver improved human performance, since larger gain values could decrease the effectiveness of the system in tracking the operator's control input.

V.5 Discussion

V.5.1 Summary

This second Chapter has addressed the shared control paradigm for *single* and *multiple* mobile robots, in particular quadrotor UAVs, with *force* and *visual* feedback for the human operator. As opposed to the case addressed in Chapter IV (visual-vestibular feedback), the visual-force combination for cueing a human operator in a shared control scenario is a more standard choice, since force cues can be easily provided by means of, e.g., low-cost force feedback devices. Nevertheless, visual-force shared control of single/multiple *mobile* robots (in particular quadrotor UAVs) is a less explored field (w.r.t. conventional bilateral teleoperation of manipulator arms), and a number of contributions could be proposed in the reported works. Indeed, as explained, several specificities of the mobile robot case makes the shared control problem interesting and challenging. These include, for instance, the kinematic dissimilarity of the master and slave workspaces, the need to fabricate environmental forces (as hard contact with the environment is often avoided), and finally the need to resolve the large redundancy of the slave side (in the multi-robot case) versus the small number of DOFs of the master side.

In this respect, in this Chapter we first addressed the case of shared control of a *single* mobile robot: here, the main focus has been on the trajectory generation algorithm, where a planning framework has been proposed to allow a human operator to control the *future* trajectory of the UAV over a given time window while coping with several constraints of interest. To this end, by leveraging the B-spline machinery, an *Autonomous Corrector* algorithm has been developed for generating a *path* free of collisions with obstacles, free of singularities, and with local geometric invariant properties meant to ease the tracking performance of the human motion commands. The human operator, on the other hand, was provided with the possibility of commanding online some global properties of the planned path, such as an overall translation, rotation or expansion rate. The force feedback signal for the human operator was then generated out of the *integral* mismatch between commanded path and executed path (that could be deformed w.r.t. the commanded one because of the several constraints imposed by the Autonomous Corrector algorithm). This force feedback signal can then inform the human operator about the feasibility of her/his intended motion commands over a *future* time window, making it thus easier for the operator to control the robot motion in highly cluttered or difficult environments.

Subsequently, the Chapter has focused on the case of shared control of *multiple* mobile robots. Here, the main contributions have been in the modeling of the inter-robot interactions in order to (*i*) obtain a desired behavior for the overall robot formation while, at the same time, (*ii*) guaranteeing the fulfilment of additional constraints/requirements such as collision avoidance and (*iii*) ensuring passivity of the closed-loop behavior of the slave side (the multi-robot group) for implementing a stable bilateral teloperation loop. Because of the large redundancy of the slave side (N robots for a single human operator), the shared control architecture was designed as follows for the various cases: a formation controller (in different settings) is in charge of controlling most of the internal DOFs of the robot group (e.g., by enforcing maintenance of a desired shape), and the human operator is given the possibility to act on those (few) DOFs left free by the formation controller action. As usual, the force feedback signal was generated as a function of the mismatch between the commanded velocities and their realization by the robot group.

Two conceptually distinct possibilities were considered for the formation control component of the slave side: a *constant* topology case, and an *unconstrained* topology case. The constant topology case attempts to force the agents to attain a desired shape defined in terms of pair-wise geometrical constraints such as relative distances or bearing vectors. This way, a well-defined shape is imposed to the N agents for applications involving, e.g., precise measurement, transportation, or environmental coverage. The human operator can then act on the free DOFs of the (constrained) robot formation, for instance a global translation and rotation for distance constraints, and a global translation, rotation about a vertical axis, and expansion for bearing constraints. The pair-wise couplings among the agents are assumed specified at the beginning of the task and constant during motion, from which the term *constant* topology for this class of algorithms.
The unconstrained topology case, on the other hand, addresses the case of a much looser coupling among agents where the pair-wise geometrical constraints can be created or lost at runtime as a function of the robot states. This makes it possible to take into account sensing/communication limitations that can prevent a robot in the group to obtain the needed state measurements of its neighbors for implementing a formation control action. This flexibility does not allow, in general, to impose a desired shape to the robot group. However, one typically aims at obtaining a coherent collective motion of the N robots without a special interest in the specific relative positions of each robot in the group. In this setting, the human operator is then in charge of controlling the motion of a small subset of the N robots (e.g., typically a single leader) for then steering the rest of the group in the environment. And, again, the force feedback signal is generated from the difference between commanded and actual leader(s) velocities. In the reported works, two possibilities have been considered for this unconstrained case: first, a general framework for ensuring passivity of a slave-side with *time-varying* topology where interaction links can be created/lost at runtime with no special requirements, and where spring-like coupling are established among interacting pairs (for ensuring a cohesive behavior to the whole group). Then, this framework has been extended to take into account the (global) constraint of preserving connectivity of the agent group despite the possible changes in the interaction topology. This way, the robots are still allowed to lose/gain neighbors over time, but with the guarantee of never threatening the group connectivity (which is a basic requirement for the convergence of any distributed control algorithm).

All the reported schemes for the *constant* and *unconstrained* cases can be implemented in a fully dencetralized way by only resorting to relative measurements (e.g., relative distances, positions or bearing angles) and 1-hop communication. Furthermore, an experimental validation of the various approaches have also been successfully performed by exploiting quadrotor UAVs as robotic platform. Besides the

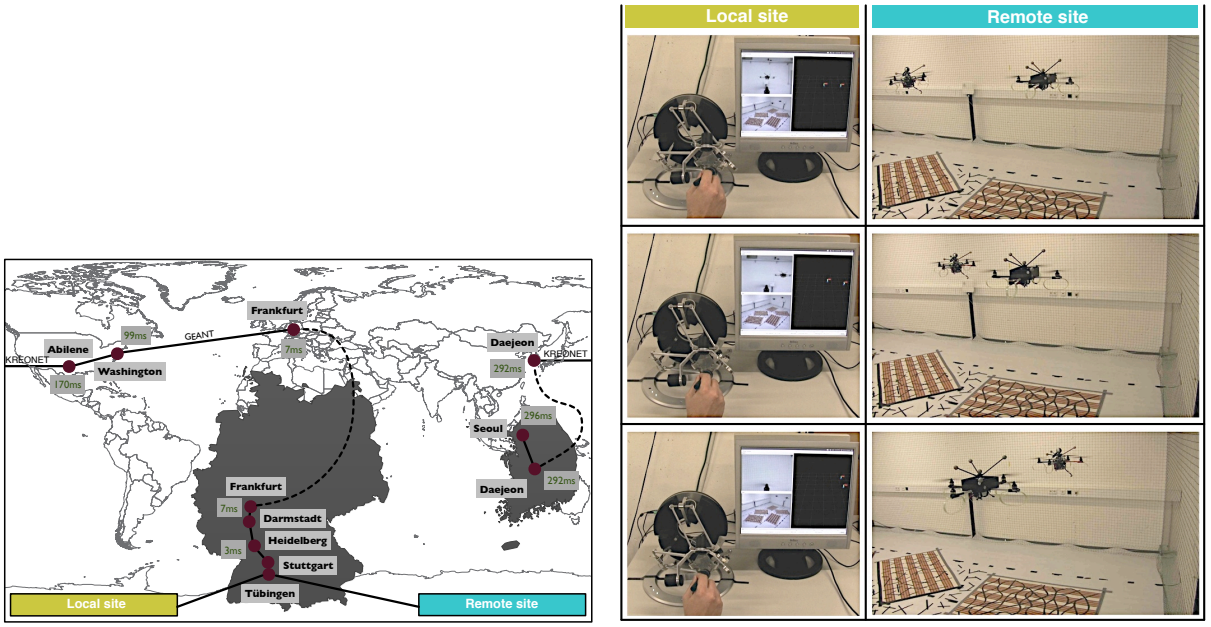


FIGURE V.45: Left: representation of the default packet routing and average delays between the MPI of Biological Cybernetics, Tübingen, and Korea University, Seoul. Some local hops were omitted to increase clarity. Right: a representative experiment of intercontinental bilateral teleoperation with 2 quadrotor UAVs. Left column: the local (human operator) site. Right column: the remote (UAV) environment. Each row represents a different time instant. The human operator is provided with a haptic interface in order to control the overall motion of the UAVs and 3 video streams: an onboard view, a global view, and a 3D representation of the UAV states. The two sites are connected through an intercontinental channel implementing a full Germany-South Korea roundtrip.

reported results, an intercontinental teleoperation experiment involving Germany and South Korea has also been performed [C38], see Fig. V.45. In this experiment, a human operator located at the Max Planck Institute for Biological Cybernetics, Germany, operated a group a quadrotor UAVs located at the same site, but by means of a real intercontinental Internet channel passing through a machine located at the Korea University, in Seoul (KU). Tracing packets between both endpoints revealed a route over the North American continent containing 25 hops resulting in a raw IMCP travel time of about 295 ms. The goal of this demo was to show the feasibility of the shared control framework by stressing in real conditions the robustness of the various theoretical claims (e.g., stability of the teleoperation loop). Finally, a user study has also been designed and conducted for assessing what combination of force cues led to an improved maneuverability and perceptual sensitivity for a human operator in the constant topology case. The findings of this study can be used to optimize several parameters of the force feedback algorithm in order to maximize the operator's awareness in 'tele-navigating' the robot group.

V.5.2 Perspectives

Given the broad range of the topics addressed in this chapter (multi-robot coordination strategies, trajectory planning algorithms, shared control architectures, user evaluations), there clearly exists a large number of open points and future directions w.r.t. what summarized in the previous sections. First of all, the reported works have mainly focused on the *algorithmic* challenges inherent to a multi-robot shared control scenario, namely, decentralized multi-robot coordination from relative measurements, passivity (stability) of the multi-robot slave side, autonomous generation of trajectories for mobile robots, and

so on. A smaller attention has been instead devoted to the need of running proper user evaluations in order to assess the effectiveness of the various force feedback schemes (also in conjunction with the multi-robot coordination algorithms). In this respect, apart from the studies reported in Sect. V.4 (that addressed the specific case of shared control for the *constant* topology case), the design of the force cues has often followed a heuristic approach in which the user has been provided with cues proportional to the difference between her/his commands and their actual realization. This design is clearly reasonable (and indeed exploited in many different bilateral teleoperation contexts), but obviously a much more rigorous user evaluation of all the various possibilities proposed in the reported works should be performed for obtaining a fair evaluation of the several approaches.

Coming to the algorithmic contributions in the multi-robot shared control scenarios, here, again, a quite large number of possible improvements can be conceived for addressing the many open points. Some of these have been covered by additional works not mentioned in the previous Sections. For instance, a (known) drawback of the coordination algorithms for the *constant* topology case with distance constraints of Sect. V.3.1.1 is the need of measuring *relative* positions (in a common shared frame of reference) in order to control the relative distances (which is the goal of the coordination algorithm). Therefore, the controller of Sect. V.3.1.1 cannot be implemented by only resorting to measured distances, but an additional relative bearing/position sensor needs to be assumed onboard the robots, together with the (assumed) possibility for the robots to agree on a *common reference frame* where to express all quantities. This issue has been resolved in the subsequent works [C37, J18], where *rigidity* maintenance for formations defined by distance constraints has been addressed. In short, if a formation is rigid w.r.t. the distance constraints, all the positions of the agents are 'fixed' w.r.t. a frame attached to the formation itself. It then becomes possible to estimate the relative positions of each agent w.r.t. a common (although arbitrary) frame of reference by only exploiting the measured distances. This idea has indeed been exploited in [C37, J18] for implementing formation controllers based on distance constraints by *only* resorting to distance measurements (via a concurrent estimation algorithm for retrieving the needed relative position information).

Along similar lines, the nice properties of the bearing formation controller discussed in Sect. V.3.1.2 (all the control actions can be computed from the sole measured bearings) are due to the very special structure assumed for the interaction graph among the agents, see Fig. V.26. Indeed, this graph assumes presence of two special agents (beacons) of which, e.g., one (the leader) needs to measure and to be measured by all the other agents in the group. The special structure of this interaction graph makes it possible for the agents to reconstruct all the needed information (e.g., relative orientations between their body frame) from the available bearings. However, this special structure also restricts the possible applications of the proposed algorithm to general situations since, as explained, one agent needs to 'see' and to 'be seen' by all the other ones. In this respect, the works reported in [C44, C57, C63] have proposed different control schemes able to stabilize a formation defined in terms of relative bearings without special requirements on the interaction graph apart from being 'bearing rigid' (a necessary requirement for solving the formation control problem). By exploiting these results, one could then devise an extension of what reported in Sect. V.3.1.2 for, e.g., allowing for a *time-varying* topology for the measurement graph in order to take into account sensory limitations such as the limited camera fov (which is now cursorily addressed in [J11]).

Furthermore, some extensions have also been proposed w.r.t. what reported in Sect. V.3.2 for the *unconstrained* topology cases. In particular, the work in [C36] has improved the scheme of Sect. V.3.2.1 by proposing a passification method (still based on the Tank machinery) for dealing with presence of

communication delays among the agents and with the master side. The *connected* topology case of Sect. V.3.2.2 has also been extended in [C45] in order to take into account the possibility of modifying *online* the minimum degree of connectivity $\lambda_2^{\min}$ for the robot group so as to render the group 'stiffer' or 'looser' during motion according to the particular situation. Finally, the works [C32, J24] have addressed the issue of selecting *online* and in a decentralied way the 'best' leader among the N robots for maximizing the tracking performance of the human commands.

A final open point only partially addressed in the reported works is the actual implementation of all the reported coordination schemes *without* the aid of external centralized facilities such as external motion capture systems (e.g., Vicon). Indeed, while the design stage of all the proposed schemes is *decentralized* and dependent on only *relative information*, the actual experimental realizations strongly relied, implicitly or explicitly, on the presence of a (centralized) Vicon system which, by tracking all the robots in the scene, allows mimicking the presence of onboard sensing out of a global/centralized overview of the group. Obviously, the use of Vicon is appropriate for ease of development, debugging and testing, or for applications designed for well-defined spaces/environments (e.g., indoor factory floors or external outdoor environments covered by GPS). However, it obviously lacks the flexibility for being deployed in real-world unstructured environments (e.g., indoor collapsed buildings, in highly crowded spaces, underwater or underground). In order to obtain a real experimental validation of the various approaches, one should then aim towards the design and implementation of a fully autonomous team of cooperating robots able to accomplish a mission by only resorting to their "own skills", i.e., local sensing (e.g., cameras) and communication capabilities. This is, indeed, the goal of the ANR Young Researcher Project "SenseFly – Sensor-Based Flying Multi-Robot Systems" funded by the French National Research Agency in 2014 and currently under progress.

V.5.3 Main references

The work presented in this Chapter is based on the following material:

- Ph.D. Thesis of Carlo Masone "Planning and Control for Robotic Tasks with a Human-in-the-Loop", July 2014, University of Stuttgart, Germany
- C. Masone, P. Robuffo Giordano, and A. Franchi. A Framework for Shared Planning and Control with Integral Haptic Feedback. *submitted to the The International Journal of Robotics Research*, 2015 (ref [J25])
- D. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano. Semi-Autonomous Haptic Teleoperation Control Architecture of Multiple Unmanned Aerial Vehicles. *IEEE/ASME Transactions on Mechatronics,* **Focused Section on Aerospace Mechatronics**, 18(4):1334–1345, 2013 (ref [J16])
- P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. A Passivity-Based Decentralized Strategy for Generalized Connectivity Maintenance. *The International Journal of Robotics Research*, 32(3):299–323, 2013 (ref [J15])
- H. I. Son, A. Franchi, L. L. Chuang, J. Kim, H. H. Bülthoff, and P. Robuffo Giordano. Human-Centered Design and Evaluation of Haptic Cueing for Teleoperation of Multiple Mobile Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 43(2):597–609, 2013 (ref [J14])

- A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Shared Control: Balancing Autonomy and Human Assistance with a Group of Quadrotor UAVs. *IEEE Robotics and Automation Magazine,* **Special Issue on Aerial Robotics and the Quadrotor Platform**, 19(3):57–68, 2012 (ref [J12])

- A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Modeling and Control of UAV Bearing-Formations with Bilateral High-Level Steering. *The International Journal of Robotics Research,* **Special Issue on 3D Exploration, Mapping and Surveillance**, 31(12):1504–1525, 2012 (ref [J11])

- A. Franchi, C. Secchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of Groups of Mobile Robots with Time-Varying Topology. *IEEE Transactions on Robotics*, 28(5):1019–1033, 2012 (ref [J10])

- C. Masone, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi. Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014 (ref [C56])

- C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Control of the Degree of Connectivity in Multiple Mobile-robot Teleoperation. In *Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*, 2013 (ref [C45])

- C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Interactive Planning of Persistent Trajectories for Human-Assisted Navigation of Mobile Robots. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, pages 2641–2648, Vilamoura, Portugal, Oct. 2012 (ref [C41])

- M. Riedel, A. Franchi, H. H. Bülthoff, P. Robuffo Giordano, and H. I. Son. Experiments on Intercontinental Haptic Control of Multiple UAVs. In *Proc. of the 12th International Conference on Intelligent Autonomous System (IAS 2012)*, pages 227–238, Jeju Island, Korea, Jun. 2012 (ref [C38])

- C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of a Group of UAVs with Communication Delays and Switching Topology. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 4307–4314, St. Paul, MN, May 2012 (ref [C36])

- A. Franchi, C. Masone, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of Multiple UAVs with Decentralized Bearing-only Formation Control. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* **Invited paper** *(IROS 2011)*, pages 2215–2222, San Francisco, CA, Sep. 2011 (ref [C31])

- P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. Experiments of Passivity-Based Bilateral Aerial Teleoperation of a Group of UAVs with Decentralized Velocity Synchronization. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* **Invited paper** *(IROS 2011)*, pages 163–170, San Francisco, CA, Sep. 2011 (ref [C30])

- H. I. Son, L. L. Chuang, A. Franchi, J. Kim, D. J. Lee, S. W. Lee, H. H. Bülthoff, and P. Robuffo Giordano. Measuring an Operator's Maneuverability Performance in the Haptic Teleoperation of Multiple Robots. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2011)*, pages 3039–3046, San Francisco, CA, Sep. 2011 (ref [C29])

- P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. Bilateral Teleoperation of Groups of UAVs with Decentralized Connectivity Maintenance. In *Proc. of the 2011 Robotics: Science and Systems Conference (RSS 2011)*, Los Angeles, CA, Jun. 2011 (ref [C28])
- H. I. Son and J. Kim and L. Chuang and A. Franchi and P. Robuffo Giordano and D. Lee and H. H. Bülthoff. An evaluation of haptic cues on the tele-operator's perceptual awareness of multiple UAVs' environments. In *Proc. of the 2011 IEEE World Haptics Conference (WHC 2011)*, pages 149–154, Istanbul, Turkey, Jun. 2011 (ref [C27])
- A. Franchi, P. Robuffo Giordano, C. Secchi, H. I. Son, and H. H. Bülthoff. A Passivity-Based Decentralized Approach for the Bilateral Teleoperation of a Group of UAVs with Switching Topology. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 898–905, Shanghai, China, May 2011 (ref [C25])
- D. Lee, A. Franchi, P. Robuffo Giordano, H. I. Son, and H. H. Bülthoff. Haptic Teleoperation of Multiple Unmanned Aerial Vehicles over the Internet. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 1341–1347, Shanghai, China, May 2011 (ref [C24])

V.5.4 Videos

- Shared control and planning of a single quadrotor UAV (Sect. V.2): simulations `https://youtu.be/ZY8pFydb0oY` and experiments `https://youtu.be/n9x7_pbDn2I`
- Shared control of multiple robots with constant topology and distance constraints (Sect. V.3.1.1): experiments `https://youtu.be/lYKvhTIddVY`
- Shared control of multiple robots with constant topology and bearing constraints (Sect. V.3.1.2): simulations `https://youtu.be/LkfB0l841zY` and experiments `https://youtu.be/9RiGd7DTk34`
- Shared control of multiple robots with time-varying *unconstrained* topology (Sect. V.3.2.1): simulations `https://youtu.be/MAlsH0Lbtqw` and experiments `https://youtu.be/OiI83mrVuYU`
- Shared control of multiple robots with time-varying *connected* topology (Sect. V.3.2.2): simulations `https://youtu.be/9r4yBh4R9Hw` and experiments `https://youtu.be/cokHQY46Rfw`,
- Intercontinental shared control of multiple robots (Fig. V.45): experiments `https://youtu.be/3MfhZ01x0Hw`

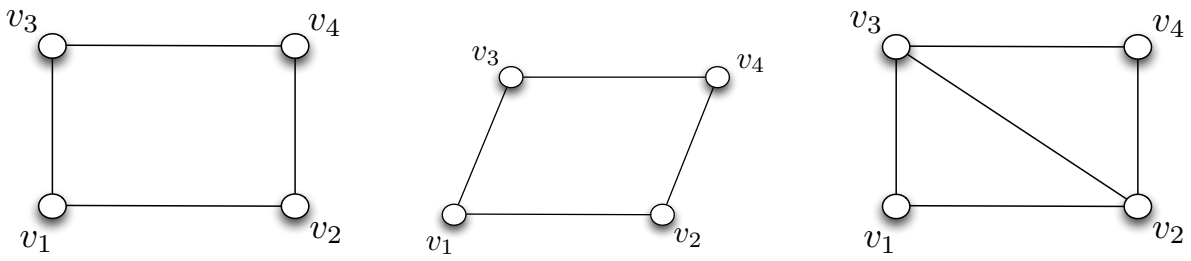
Chapter VI

Rigidity-based Decentralized Formation Control and Localization

These last Chapters are meant to provide a very high-level illustration of some additional activities related to the core material presented so far. For the reader's convenience, each chapter is paired with a corresponding Appendix where a publication representative of the reported material is attached.

As explained at the end of the previous Chapter, the notion of *rigidity* of a multi-robot spatial arrangement plays a central role for the development of cooperative (and decentralized) formation control and localization algorithms. An accessible overview of many fundamental concepts related to rigidity in the multi-robot domain can be found in [Anderson et al., 2008]. In essence, rigidity of a formation tries to characterize the global/local *uniqueness* of the robot spatial arrangement (relative poses) given the number and values of the pair-wise geometrical constraints that exist among interacting/neighboring pairs. For instance, considering the examples of the previous Chapter, when distance constraints exist (or are enforced) among pairs of robots one can analyze whether these constraints are enough to define a unique (desired) shape for the robot formation *up to a global roto-translation*. In case they do, the formation is termed 'rigid' in the sense that the relative positions of the agents are 'fixed' (in a frame attached to the formation itself) by the pair-wise distance constraints. Figure VI.1 shows a (typical) example of non-rigid and rigid formations in the plane in the case of distance constraints. The formation in Fig. VI.1(A) is non-rigid since it can be 'deformed' without changing the values of the constraints (distances). Indeed, the the agents in Fig. VI.1(B) have the same inter-distances over the edges of the graph but, clearly, their relative positions are different from those in Fig. VI.1(A). The formationin Fig. VI.1(C) is, instead, rigid since the agent positions are fixed (modulo a roto-translation on the plane) by maintaining the inter-distances specified by the interaction graph. Conceptually similar considerations can obviously be drawn for other kinds of pair-wise geometrical constraints such as the bearing vectors discussed in the previous Chapter.

In our context, the interest in rigidity of a multi-robot formation stems from the fact that, if a formation is rigid, then (i) the formation control problem can be univocally solved by only regulating the pair-wise geometrical constraints (e.g., the inter-robot distances), and (ii) the relative localization problem can also be univocally solved (in a common frame attached to the formation) by only processing the values of the pair-wise geometrical constraints (e.g., the inter-robot distances or bearing vectors). A multi-robot



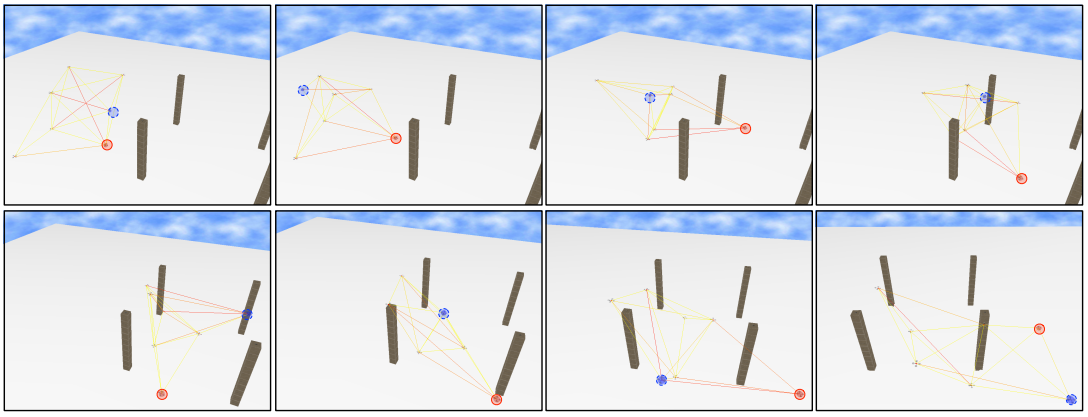
(A) (B) (C)

FIGURE VI.1: Examples of rigid and non-rigid formations on the plane in the case of distance constraints. The formation in (A) is non-rigid since it can be deformed into (B) without affecting the existing pair-wise constraints (the robot inter-distances). The formation in (C) is instead rigid since it cannot be deformed by preserving the values of the inter-robot distances specified by the interaction graph.

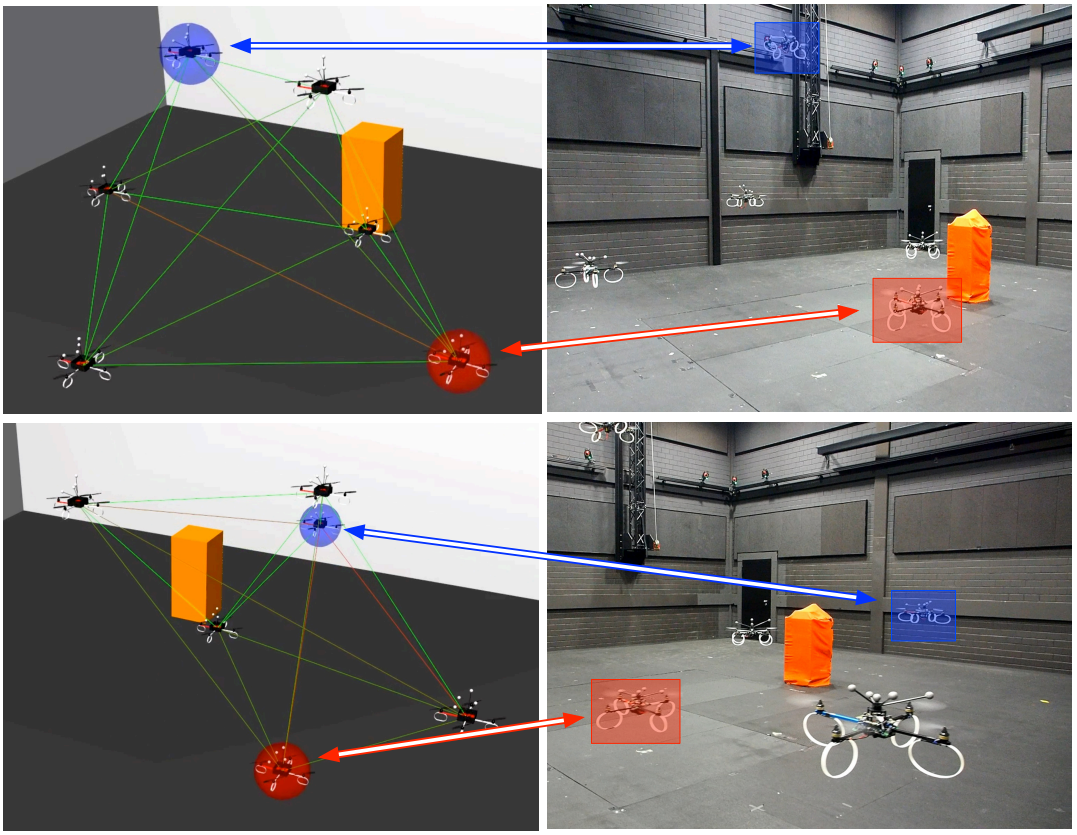
group can then, for instance, reach a desired geometrical shape (e.g., a regular polyhedron) and obtain a consistent estimation of all the inter-robot relative poses by only *measuring* and *processing* the values of the inter-robot distances and/or bearing vectors, which are quantities typically available to direct measurement.

Formation rigidity for the purpose of formation control and cooperative localization has been addressed in [C37, J18, C44, C57, C63]. In particular, the works [C37, J18] have considered the problem of *rigidity maintenance* for a group of mobile robots in 3D with *distance* measurements, that is, how to preserve rigidity of a formation by allowing online gain/loss of interacting pairs because of sensing constraints (thus, extending the *connectivity maintenance* algorithm of Sect. V.3.2.2 to the rigidity case). Indeed, it is possible to give an algebraic characterization to the notion of (infinitesimal) rigidity by looking at the spectral properties of the so-called *Rigidity matrix*, which plays a similar role to the Laplacian matrix for the connectivity case. By keeping the Rigidity matrix full-rank over time, one can then ensure maintenance of formation rigidity despite a possible time-varying topology for the interaction graph. In this respect, Figure VI.2 shows some snapshots of a simulation and experiment involving several quadrotor UAVs flying in a rigid formation but with the possibility of losing/gaining neighbors over time because of sensing constraints (maximum range and occluded visibility).

The works [C44, C57, C63] have instead addressed the problem of formation control and cooperative localization for formations defined in terms of relative bearing vectors for agents in $\mathbb{R}^2 \times SO(2)$, that is, point-wise planar agents equipped with a *local body frame* where all measurements and control actions are expressed. This scenario is particularly relevant when adopting *cameras* as onboard sensors for obtaining a relative measurement of neighboring robots. Cameras, indeed, allow obtaining bearing measurements expressed in the local camera/body frame of each agent in the group. The lack of a global/shared frame where all measurements and control actions can be expressed in poses some additional challenges to the localization and formation control problem, since the agents need also to agree on how to 'interpret' any measurement taken by neighboring robots and transmitted via local communication. While most of the literature simply assumes availability of a global frame to all agents in the group (e.g., by assuming presence of a compass measuring the magnetic north), the works in [C44, C57, C63] have instead resorted the theory of rigidity for bearing formations in $\mathbb{R}^2 \times SO(2)$ for solving the formation control and localization problem without explicitly requiring external availability of a global frame for the agents.



(A)



(B)

FIGURE VI.2: Top: snapshots of a simulation with 7 quadrotor UAVs. The two UAVs partially controlled by two human operators are marked with two semi-transparent spheres. Bottom: Two snapshots of the reported experiment. Left: simulated 3D views showing, in particular, the inter-agent links (red – almost disconnected link, green – optimally connected link). Right: corresponding pictures of the experimental setup. The two highlighted quadrotor UAVs are partially controlled by two human operators

VI.1 Main references

The work presented in this Chapter is based on the following material:

- D. Zelazo, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Decentralized Rigidity Maintenance Control with Range Measurements for Multi-Robot Systems. *The International Journal of Robotics Research*, 34(1):105–128, 2015 (ref [J18])
- D. Zelazo, A. Franchi, F. Allgöwer, H. H. Bülthoff, and P. Robuffo Giordano. Rigidity Maintenance Control for Multi-Robot Systems. In *Proc. of the 2012 Robotics: Science and Systems Conference (RSS 2012)*, Sydney, Australia, Jul. 2012 (ref [C37])
- A. Franchi and P. Robuffo Giordano. Decentralized Control of Parallel Rigid Formations with Direction Constraints and Bearing Measurements. In *Proc. of the 51st IEEE Conference on Decision and Control (CDC 2012)*, pages 5310–5317, Maui, HI, Dec. 2012 (ref [C44])
- D. Zelazo, A. Franchi, and P. Robuffo Giordano. Rigidity Theory in $SE(2)$ for Unscaled Relative Position Estimation using only Bearing Measurements. In *Proc. of the 2014 European Control Conference (ECC 2014)*, 2014 (ref [C57])
- D. Zelazo, P. Robuffo Giordano, and A. Franchi. Bearing-Only Formation Control Using an $SE(2)$ Rigidity Theory. In *2015 IEEE Conf. on Decision and Control (CDC 2015)*, 2015 (ref [C63])

Furthermore, an Invited Session with title "Rigidity Theory for Problems in Multi-Agent Coordination" has been organized by the applicant together with Prof. D. Zelazo (Technion, Israel) and Dr. A.Franchi (LAAS-CNRS, France) at the 54-th IEEE Conf. on Decision and Control (CDC2015).

VI.2 Videos

- Rigidity maintenance in 3D with distance measurements ([J18, C37]): simulations `https://youtu.be/Ni6rIrcA5Hw` and experiments `https://youtu.be/im-4ZfCoLec`

Chapter VII

Novel Mechanical Designs for Quadrotor UAVs

This Chapter summarizes the works [C35, C46, J19] in which a novel *overactuated* concept for a quadrotor UAV has been proposed. Indeed, standard quadrotor UAVs (such as those exploited in the experiments of the previous Chapters) possess a limited mobility because of their *underactuation*, i.e., availability of 4 independent control inputs (the 4 propeller spinning velocities) vs. the 6 DOFs parameterizing the quadrotor position/orientation in space. As a consequence, the quadrotor pose cannot track arbitrary trajectories in space (e.g., it can hover on the spot *only* when horizontal). Such a underactuation not only affects the quadrotor flying ability in free or cluttered space, but it also degrades the possibility of interacting with the environment by exerting desired forces in arbitrary directions, thus limiting the exploitation of quadrotors as as autonomous flying *service robots*. On the other hand, quadrotor UAVs are more and more employed as *service robots* for interaction with the environment, as proven by the recently funded EU projects "AIRobots" [AIRobots, 2013] and "ARCAS" [ARCAS, 2011], and by the works of several groups which addressed the design and implementation of strategies for interacting with the environment, either in the form of direct contact [Gentili et al., 2008, Naldi and Marconi, 2010, Marconi and Naldi, 2012] or by considering aerial grasping/manipulation tasks [Pound et al., 2011, Lindsey et al., 2011, C39, Sreenath and Kumar, 2013, Manubens et al., 2013].

In this respect, the works [C35, C46, J19] have instead explored different actuation strategies that can overcome the underactuation problem and allow for full motion/force control in all directions in space. In particular, we focused on a novel actuation concept for a quadrotor UAV in which all the (usually fixed) propellers are allowed to *tilt* (in a controlled way) about the axes connecting them to the main body frame. Indeed, by means of these additional 4 actuated tilting dofs, it is possible to gain *full controllability* over the quadrotor position and orientation, thus transforming it, as a matter of fact, in a *fully actuated flying rigid body*.

Figure VII.1 shows a picture of the first prototype of 'quadrotor with tilting propellers', denoted as *holocopter* in the following, while Figs. VII.2 report an illustrative view of the main quantities of interest (in particular, the four tilting angles α_i that can now be actuated), and a detailed view of the tilting mechanism with the main forces/torques acting on it during flight.

FIGURE VII.1: Picture of the holocopter prototype. The four propeller groups are slightly tilted. The red bar indicates the positive direction of the holocopter body axis

(A) (B)

FIGURE VII.2: Left: schematic view of the quadrotor considered in this paper. The overall center of mass is assumed to be in the body frame center. The symbol L represents the length of all propeller arms, ω_i, $i = 1 \dots 4$, the propeller rotation speed and α_i, $i = 1 \dots 4$, the orientation of the propeller group. Right: i-th tilting arm visualizing the body frame $\mathcal{F}_{P_i}$, the associated propeller thrust T_i, torque τ_{ext_i} and the propeller tilt angle α_i

By exploiting standard techniques (e.g., Newton-Euler procedure), one can derive a complete description of the quadrotor dynamic model by considering the forces/moments generated by the propeller motion, as well as any cross-coupling due to gyroscopic and inertial effects arising from the relative motion of the 5 bodies composing the quadrotor (main body and four tilting propellers). As shown in [C35, C46, J19], this complete dynamic model can be further simplified into the following simpler system

$$\begin{cases} \ddot{\boldsymbol{p}} &= \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} {}^{W}\boldsymbol{R}_B \boldsymbol{F}(\boldsymbol{\alpha}) \boldsymbol{w} \\ \dot{\boldsymbol{\omega}}_B &= \boldsymbol{I}_B^{-1} \boldsymbol{\tau}(\boldsymbol{\alpha}) \boldsymbol{w} \\ \dot{\boldsymbol{\alpha}} &= \boldsymbol{w}_\alpha \\ {}^{W}\dot{\boldsymbol{R}}_B &= {}^{W}\boldsymbol{R}_B [\boldsymbol{\omega}_B]_\wedge \end{cases}, \tag{VII.1}$$

still able to capture the main effects acting on the holocopter but more suited for *control design*. Here,

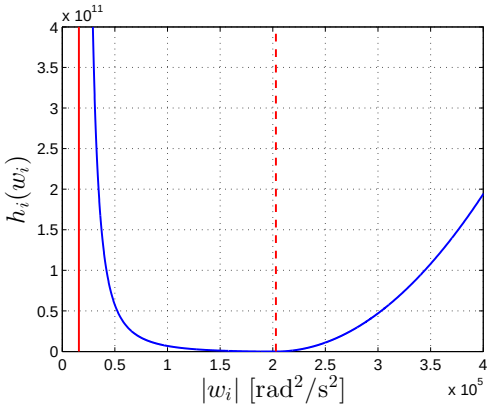


FIGURE VII.3: Example of a cost function $h_i(w_i)$ with $\bar{w}_{min} = 126$ [rad/s] (solid red vertical line), $\bar{w}_{rest} = 450$ [rad/s] (dashed red vertical line). Note that $h_i(w_i) \to \infty$ as $|w_i| \to w_{min}$ or $|w_i| \to \infty$, and that $h_i(w_i)$ has a unique minimum at w_{rest} with continuous derivative

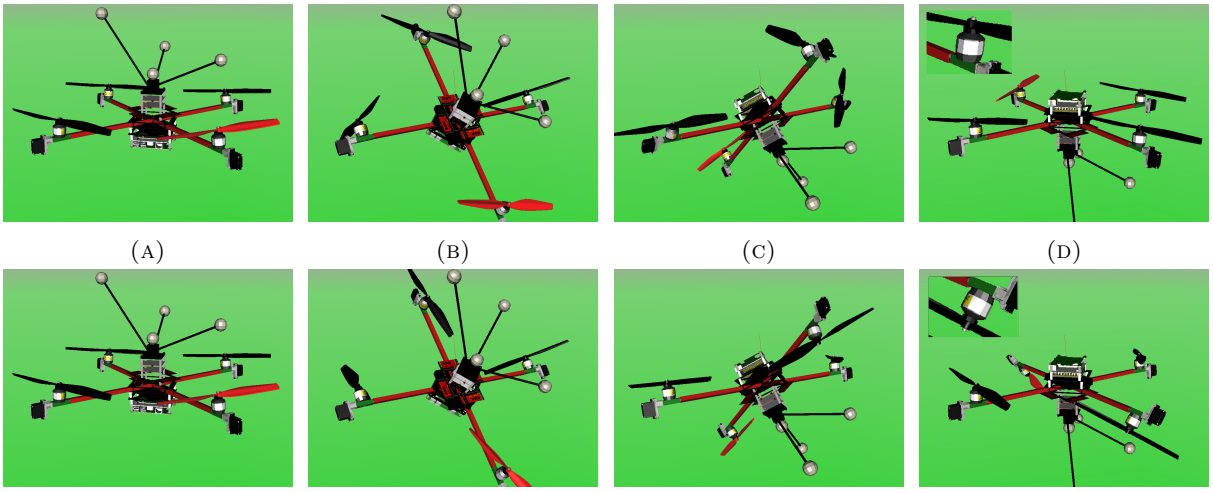


FIGURE VII.4: Simulation results. Left column: holocopter motion while minimizing the cost functions $h_i(w_i)$ (case (i)). Right column: holocopter motion without minimizing the cost functions $h_i(w_i)$ (case (ii)). Note the large reorientation of the propeller groups in case (i) which, thanks to the action of the optimization term, end up in a full 'upright' position (minimum energy consumption) w.r.t. case (ii)

$\boldsymbol{w} \in \mathbb{R}^4$ represents the set of four spinning velocities, i.e., the control inputs available to standard quadrotors. On the other hand, $\boldsymbol{w}_\alpha \in \mathbb{R}^4$ represents the *new* four tilting velocity commands available for the holocopter (for a total of then $4+4=8$ velocity commands). System (VII.1) can be shown to be dynamically feedback linearizable by taking the main body position $\boldsymbol{p}$ and orientation ${}^W\boldsymbol{R}_B$ as output functions and by inverting the input/output mapping at the *third* differential order. The resulting 6×8 decoupling matrix (to be pseudo-inverted by the control law) results to be a function of the spinning velocities $\boldsymbol{w}$ and tilting angles $\boldsymbol{\alpha}$, and can be proven to be full rank if and only if no propeller comes at rest during flight, i.e., $\boldsymbol{w}_i \neq 0$, $i = 1 \dots 4$. Since the actuation redundancy (eight control inputs for six controlled quantities) can be exploited for optimizing any criterium of interest, it is then possible to penalize either too small propeller velocities (for preventing singularities of the control scheme) or too large propeller velocities (for minimizing the total energy consumption because of air drag). This can be accomplished by minimizing, in the null-space of the main control action, a suitable cost function $h_i(w_i)$ for each propeller such as that shown in Fig. VII.3.

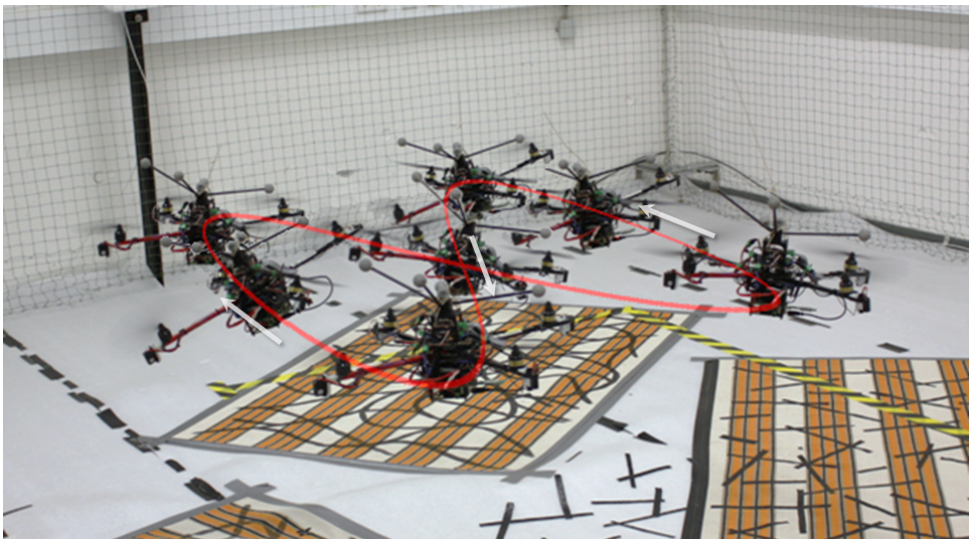
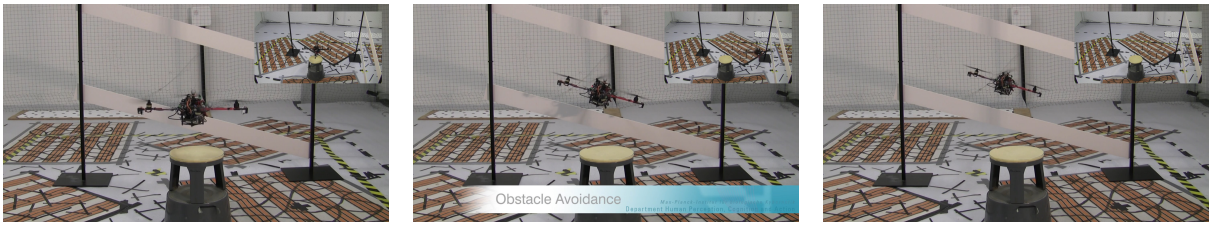


FIGURE VII.5: Experimental results. Overlay of several snapshots of the holocopter while performing eight-shape trajectory



(A) (B) (C)

FIGURE VII.6: Experimental results. The holocopter passes through a narrow passage along a straight horizontal path while keeping a tilted orientation (a maneuver not possible for a standard quadrotor UAV)

The proposed modeling approach and control design has been successfully validated in a number of simulation and experiments as reported in [C35, C46, J19]. For instance, Fig. VII.4 reports the results of a simulation meant to show the importance of minimizing the cost functions $h_i(w_i)$ in the null-space of the main control action. Indeed, in Fig. VII.4 the quadrotor is commanded to perform a 180 deg rotation about a horizontal axis while staying fixed in space (a maneuver clearly not possible with a standard quadrotor UAV). In one case (top row), the cost functions $h_i(w_i)$ are minimized, and in the other case (bottom row) they are not minimized. As a consequence, in the first case the quadrotor can perform the commanded maneuver by rearranging the propeller groups so as to have them vertical again at the end of the motion (with, thus, the minimum energy consumption needed for hovering in place). In the second case, on the other hand, the quadrotor ends up with two propellers pointing upwards and two downwards, with an internal cancellation of the four propeller thrusts (and, thus, a very inefficient configuration for hovering).

Figure VII.5 reports some snapshots taken during a flight experiment obtained with the first prototype. In this experiment, the holocopter was commanded to track a horizontal eight-shape trajectory with superimposed a sinusoidal rotation about one of its body axis which is, again, a maneuver not possible for a standard quadrotor UAV. In the same spirit, Figs. VII.6 report some snapshots of the holocopter passing through a narrow passage along a straight horizontal path while keeping a titled orientation (needed to avoid collision with the obstacles).

FIGURE VII.7: CAD image of the second-generation prototype with improved mechanics, actuation system, electronics and an overall reduced mass and inertia

Finally, Fig. VII.7 shows a CAD model of a second-generation prototype (currently under construction) meant to overcome the main technological limitations of the first prototype, namely, a poorly performant tilting actuation system, an excessive weight w.r.t. the generated thrust, and the presence of mechanical end-stops for the tilting propellers (which, thus, could not rotate endlessly about their tilting angles).

VII.1 Main references

The work presented in this Chapter is based on the following material:

- Ph.D. Thesis of Markus Ryll "A Novel Overactuated Quadrotor UAV", February 2015, University of Stuttgart, Germany
- M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. A Novel Overactuated Quadrotor UAV: Modeling, Control and Experimental Validation. *IEEE Transactions on Control Systems Technology*, 23(2):540–556, 2015 (ref [J19])
- M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. First Flight Tests for a Quadrotor UAV with Tilting Propellers. In *Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*, 2013 (ref [C46])
- M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Modeling and Control of a Quadrotor UAV with Tilting Propellers. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 4606–4613, St. Paul, MN, May 2012 (ref [C35])

VII.2 Videos

- Simulation and experimental results of holocopter performing different maneuvers `https://youtu.be/hA-uNHW8MLE`

Chapter VIII

Active Structure from Motion

This last Chapter will summarize the works [J23, J20, J17, C62, C61, C60, C55, C54, C53, C51, C49, C40, C34] that addressed, in different contexts and scenarios, the general problem of optimizing online the convergence rate of Structure from Motion (SfM) estimation schemes by acting on the motion of a monocular camera observing a (partially) unknown scene. Structure from Motion (SfM) is a very classical and well-studied topic in computer and robot vision and, broadly speaking, it refers to the problem of recovering the 'missing structure' of the observed scene by processing the images taken by a moving camera. Within the scope of the reported works, by 'missing structure' we will mostly refer to some *metric/3D* information associated to the tracked image features that cannot be directly measured on the image plane because of the perspective projection performed by monocular cameras. Common examples include the depth of point features, the distance of a planar scene, or the size of a tracked 3D object.

When processing consecutive images over time, a popular possibility is to treat SfM as a recursive/filtering task where images and (known) camera motion are elaborated *online* for obtaining an incremental estimation of the scene structure. A large number of works has addressed SfM as a filtering problem: for instance, Extended Kalman Filter (EKF)-based solutions have been proposed in [Matthies et al., 1989, Soatto et al., 1996, Civera et al., 2008, Civera et al., 2010, Omari and Ducard, 2013], while other approaches exploiting tools from (deterministic) nonlinear observation can instead be found in [Dixon et al., 2003, Metni and Hamel, 2007, De Luca et al., 2008, Robuffo Giordano et al., 2008, Morbidi et al., 2010, Sassano et al., 2010, Durand Petiteville et al., 2010, Martinelli, 2012] and references therein.

Structure from Motion is, however, a *nonlinear* estimation problem because of the nonlinear sensor mapping from the observed environment to the sensor space (i.e., the perspective projection performed by cameras). A consequence of this nonlinearity is that the estimation convergence rate and accuracy depend on the particular trajectory followed by the robot/camera which, loosely speaking, must guarantee a sufficient level of *excitation* during motion[1]. For example, a poor choice of the system inputs (e.g., the camera linear velocity) can make the 3D scene structure non-observable *whatever* the employed estimation strategy, resulting, in practice, in inaccurate state estimation for trajectories with low information content. This, in turn, can degrade the performance of any planner/controller that needs to generate actions as a function of the reconstructed states, possibly leading to failures/instabilities in case of too large estimation errors [De Luca et al., 2008, Malis et al., 2010].

[1] The observability conditions for SfM are not (in general) time-invariant but may both depend on the current state *and* on the current applied inputs (i.e., the camera motion). Indeed, SfM falls into the class of *nonuniformly observable systems* where there exist 'singular inputs' that prevent state observation [Besançon and Hammouri, 1996].

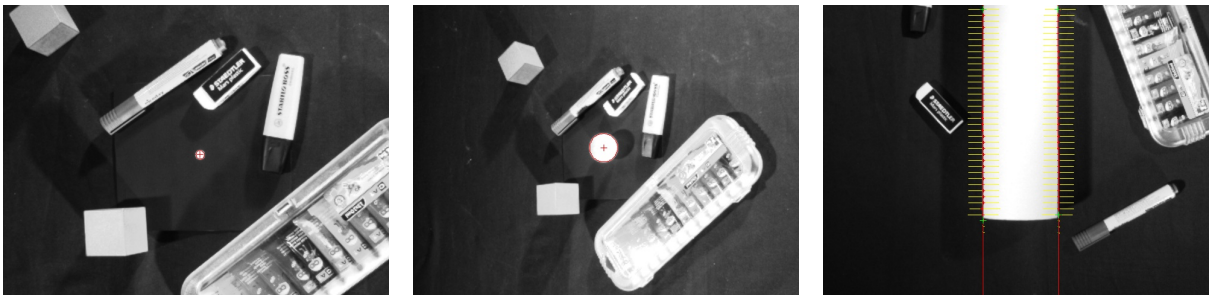
Because of the role played by the camera trajectory for the convergence of SfM algorithms, it is then meaningful to study how to optimize the camera motion in order to maximize (some measure of) the SfM performance. An *online* solution to this problem has been proposed in [C51] by detailing a general strategy able to *actively* impose to the SfM estimation error a desired convergence behavior equivalent to that of a reference linear second-order system with *desired/assigned* poles. The main sketch of the idea is as follows: let $\boldsymbol{s} \in \mathbb{R}^m$ the set of measured visual features (e.g., the x and y coordinates of a point feature), $\boldsymbol{\chi} \in \mathbb{R}^p$ the vector of unmeasurable 3D quantities associated to $\boldsymbol{s}$ (e.g., the depth Z for a point feature), and $(\boldsymbol{v}, \boldsymbol{\omega})$ the camera linear/angular velocities in the camera frame. It is possible to show that the dynamics of $\boldsymbol{s}$ and $\boldsymbol{\chi}$ obeys the following general form

$$\begin{cases} \dot{\boldsymbol{s}} &= \boldsymbol{f}_m(\boldsymbol{s}, \boldsymbol{\omega}) + \boldsymbol{\Omega}^T(\boldsymbol{s}, \boldsymbol{v})\boldsymbol{\chi} \\ \dot{\boldsymbol{\chi}} &= \boldsymbol{f}_u(\boldsymbol{s}, \boldsymbol{\chi}, \boldsymbol{u}) \end{cases} \tag{VIII.1}$$

that exhibits, in particular, *linearity* of the unknown $\boldsymbol{\chi}$ (the 3D quantities to be estimated) w.r.t. the dynamics of $\boldsymbol{s}$. As a consequence, matrix $\boldsymbol{\Omega}^T(\boldsymbol{s}, \boldsymbol{v})$ (which represents the 'sensitivity' of $\dot{\boldsymbol{s}}$ w.r.t. $\boldsymbol{\chi}$) results a function of only *known* quantities (the measured $\boldsymbol{s}$ and the camera linear velocity $\boldsymbol{v}$), and it becomes possible to act on $\boldsymbol{v}$ in order to increase the conditioning of the 'sensitivity' $\boldsymbol{\Omega}^T(\boldsymbol{s}, \boldsymbol{v})$ during the camera motion.

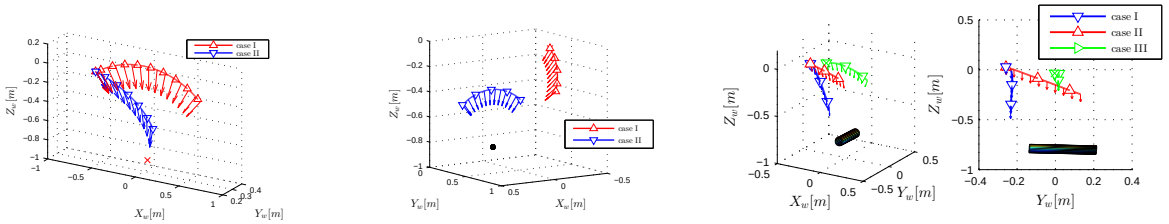
This insight has been exploited in [C51] for proposing an *active* SfM scheme built upon the dynamics (VIII.1) and yielding an estimation error with assignable convergence rate (by suitably acting on the camera linear velocity $\boldsymbol{v}$ and estimation gains). By letting $\boldsymbol{z}(t) = \boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t)$ represent the SfM estimation error, the convergence rate of $\boldsymbol{z}(t)$ results dictated by the norm of the square matrix $\alpha\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ (in particular by its smallest eigenvalue $\alpha\sigma_1^2$) which then plays the role of an observability measure for system (VIII.1). For a given choice of gain α (a free parameter of the SfM scheme), the larger σ_1^2 the faster the error convergence, with in particular $\sigma_1^2 = 0$ if $\boldsymbol{v} = \boldsymbol{0}$ (as well-known, only a translating camera can estimate the scene structure). Furthermore, the eigenvalue σ_1^2 and its Jacobian w.r.t. the camera linear velocity $\boldsymbol{v}$, i.e., $\boldsymbol{J}_v = \partial\sigma_1^2/\partial\boldsymbol{v}$, have a closed form expression function of $(\boldsymbol{s}, \boldsymbol{v})$ (known quantities). It is then possible to optimize *online* the camera linear velocity $\boldsymbol{v}$ in order to, e.g., maximize the value of σ_1^2 by following its positive gradient $\boldsymbol{J}_v$ and, as a consequence, increase the convergence rate of the estimation error $\boldsymbol{z}(t)$. This online optimization of $\boldsymbol{v}$ represents the *active* component of the SfM algorithm and it brings several added values compared to more classical inactive estimation strategies: for instance, it allows obtaining the 'best' estimation error convergence when subject to real-world constraints such as limited camera velocity or upper bounds on the estimation gains due to noise, discretization, or other typical non-idealities. Furthermore, from a more theoretical perspective, one can also get insights into the optimal camera trajectories needed to estimate the scene structure for particular classes of SfM problems (e.g., when dealing with point features or specific 3D geometrical primitives).

In this respect, the general active SfM machinery introduced in [C51] has then been applied to several case studies and application scenarios. For instance, in [J17, C54] a number of successful experimental results have been obtained by considering the case of a eye-in-hand camera tracking a point feature (with an unknown depth Z to be estimated), and a sphere and a cylinder (with an unknown radius R to be estimated). Figures VIII.1 report some snapshots of the experiments, while Figs. VIII.2 show the optimized camera trajectories vs. the non-optimized ones. All cases showed a very good match between the theoretical analysis and obtained experimental results (in terms of predicted/assigned convergence rate to the estimation error $\boldsymbol{z}(t)$).



(A) (B) (C)

FIGURE VIII.1: Camera snapshots for the point feature, the sphere and the cylinder

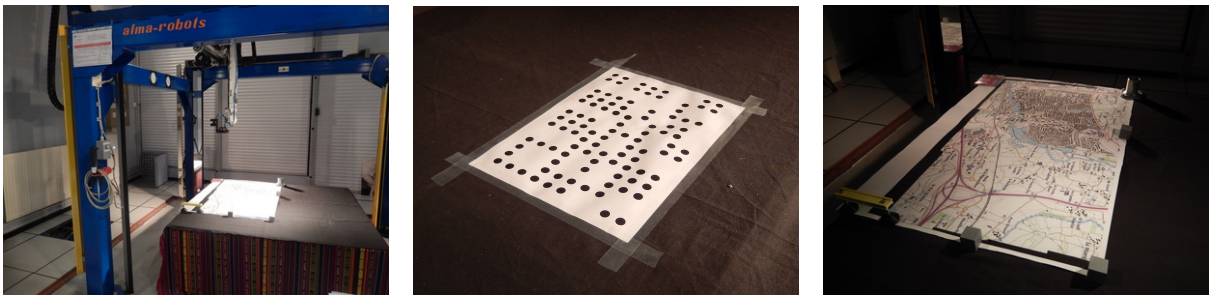


(A) (B) (C)

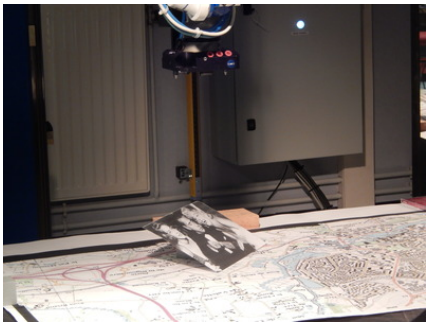
FIGURE VIII.2: A – point feature case: optimized camera trajectory (red line) vs. non-optimized one (blue line). B – sphere case: two different trajectories yielding the same convergence rate for $\boldsymbol{z}(t)$. C – cylinder case: optimized camera trajectories (red and green lines) vs. non-optimized one (red line)

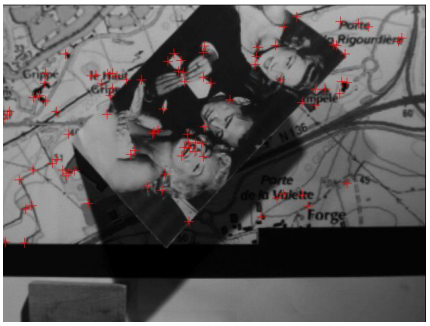
The works in [C53, C60] considered, instead, the case of a planar scene made of N 3D point $\boldsymbol{P}_k = (X_k, Y_k, Z_k)$ belonging to a common plane with equation $\boldsymbol{n}^T\boldsymbol{E} + d = 0$, with $\boldsymbol{n} \in \mathbb{S}^2$ being the unit normal vector and $d \in \mathbb{R}$ the distance to the plane. In this case, the estimation problem consisted in retrieving the 3D plane parameters $(\boldsymbol{n}, d)$ from the tracked N point features $\boldsymbol{p}_k$ (projection of the N 3D points $\boldsymbol{P}_k$) gathered during motion. Detection and estimation of 3D planes from the raw visual data is, in fact, a typical problem faced by, e.g., ground mobile robots or UAVs autonomously navigating in unknown indoor environments. Indeed, in many situations, and especially in artificial environments, the surrounding scene can be reasonably approximated as a collection of planes. This simple modelization is often sufficient for solving tasks such as navigating in a corridor or positioning inside a room. In [C53, C60] three different possibilities were considered and compared for solving the active SfM problem: a first one (the baseline method) based on the classical decomposition of the homography constraint [Ma et al., 2003], a second one based on a least-square fitting of the 3D estimated positions $\hat{\boldsymbol{P}}_k$ of the 3D points on the plane, and a third one based on the machinery of (discrete) image moments made of the tracked point features [Tahri and Chaumette, 2005]. The advantage of this latter method is that it directly yields the estimated plane parameters $(\boldsymbol{n}, d)$ (3 independent quantities) as output, instead of first retrieving the N estimated points $\hat{\boldsymbol{P}}_k$ for then solving algebraically a least-square fitting problem. Thus, the complexity of the SfM scheme results reduced w.r.t. the previous two methods as the number of estimated states is independent of the number of tracked points and, since $(\boldsymbol{n}, d)$ are directly estimated via a filtering process, one can also expect an increased level of robustness w.r.t. non perfectly planar scenes as no algebraic step is involved (contrarily to the first two methods).

The three methods were experimentally tested with the setup shown in Figs. VIII.3, i.e., a dotted pattern and a topographic map used for feature extraction and tracking. Presence of outliers was also considered, as shown in Figs. VIII.3(D–E) for assessing the robustness of the approach. Overall, good



(A) (B) (C)





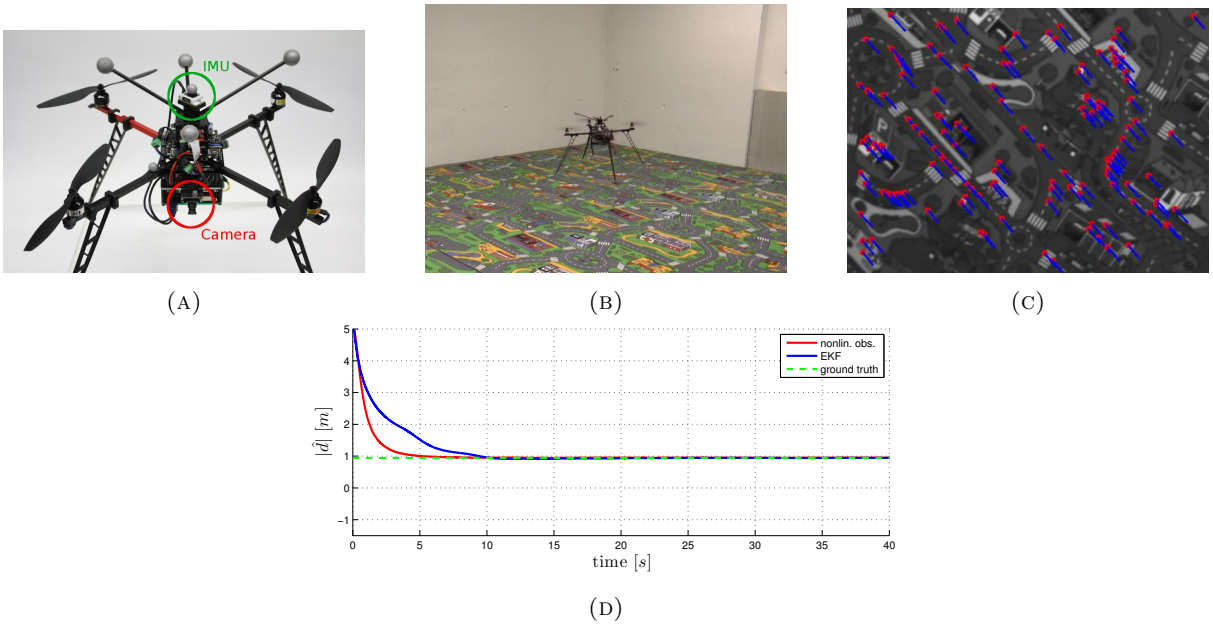
(D) (E)

FIGURE VIII.3: A–C: experimental setup with the dotted pattern and the topographic map used for feature extraction and tracking. D–E: (intentional) presence of an outlier object w.r.t. the main dominant plane

results were obtained in estimating $(\boldsymbol{n}, d)$ in real conditions, also in the (intentional) presence of some outliers w.r.t. the main dominant plane. In particular, the third (moment-based) method resulted the most robust against outliers because of its better filtering capabilities.

In [J20, C49, C40, C34], we instead considered the case of a quadrotor UAV equipped with an onboard camera and an Inertial Measurement unit (IMU) flying over a planar scene. The goal, in this case, was to recover the quadrotor linear velocity $\boldsymbol{v}$ and distance to the plane d by fusing together the *scaled* velocity $\boldsymbol{v}/d$ (obtained from an algebraic decomposition of the perceived optical flow) with the IMU measurements, that is, angular velocity $\boldsymbol{\omega}$ and specific acceleration $\boldsymbol{f}$ (which provides the needed 'metric' information for disambiguating $\boldsymbol{v}$ and d from the measured $\boldsymbol{v}/d$). The active SfM framework introduced in [C51] was then suitably extended to cover this case as well, by spelling out the various implementation details and observability conditions. In particular, it was possible to show that the pair $(\boldsymbol{v}, d)$ can be disambiguated from the measured $\boldsymbol{v}/d$ if and only if the camera undergoes a physical acceleration w.r.t. the observed plane, and, consequently, moving at constant velocity cannot allow the estimation to converge.

A detailed comparison against a more standard EKF (with same process and measurement models) was also provided by means of a number of experiments where the two filters were tested in parallel on the same trajectories and measurements. Overall, the active SfM algorithm performed consistently better than the EKF in terms of convergence rate of the estimation error. Figures VIII.4(A–C) show the experimental setup, a snapshot of the quadrotor flying over the planar scene, and an onboard view of the extracted optical flow. Figure VIII.4(D), on the other hand, reports the results of a comparison in estimating the plane distance d between the EKF (blue line) and the active SfM algorithm (red line) during one experimental run.



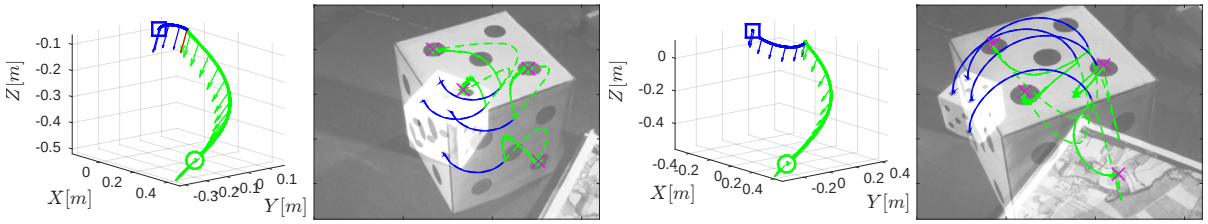
(A) (B) (C)

(D)

FIGURE VIII.4: A–C: picture of the employed quadrotor UAV with onboard camera and IMU, and snapshots of the quadrotor flying over the planar scene and of the onboard camera view (with the segmented optical flow vectors). D: comparison among a EKF (blue line) and the active SFM scheme (red line) in estimating the plane distance d

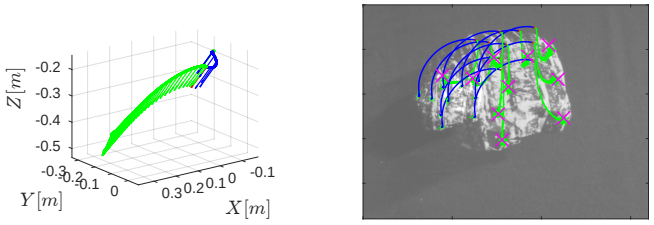
Finally, the works in [J23, C55] addressed the issue of coupling the active SfM algorithm with the concurrent execution of a Image-Based Visual Servoing (IBVS) task [Chaumette and Hutchinson, 2006] in an attempt to bridge the field of visual control (the IBVS task) with that of active perception (the active SfM scheme). Indeed, there clearly exist a tight coupling between visual perception and control: the convergence/behavior of a visual control task also depends on the estimation accuracy of some unmeasurable 3D metric information associated to the tracked features (e.g., again, the depths of point features). However, the performance in estimating the needed 3D quantities is also dictated by the particular camera trajectory generated by the visual controller. Therefore, perception should be (actively) optimized for the sake of improving the visual control performance, and the chosen visual controller should generate a camera trajectory that maximizes the information gathered during motion for facilitating the estimation task.

In [J23, C55] this perception/action coupling has been addressed by considering an eye-in-hand camera carried by a manipulator arm, and by projecting the action generated by the active SfM scheme in the null-space of the main visual servoing task (regulation of a set of point features). In order to obtain the largest degree of redundancy w.r.t. the visual task, the norm controller introduced in [Marey and Chaumette, 2010] has been suitably extended to the case at hand. Indeed, the control strategy proposed in [Marey and Chaumette, 2010] allows (under suitable conditions) to replace the regulation of an error vector $\boldsymbol{e} \in \mathbb{R}^m$ (the classical visual error) by its norm $\|\boldsymbol{e}\|$ which, being a one-dimensional task, grants the largest possible degree of redundancy w.r.t. the camera motion. The nom controller suffers however from some structural singularities when the error norm $\|\boldsymbol{e}\|$ becomes too small, so that a switch to the classical regulation of the error vector $\boldsymbol{e}$ is needed. A suitable strategy for implementing this switch in a smooth way, and with the guarantee of a monotonic decrease of the error norm $\|\boldsymbol{e}\|$ has been developed in [J23, C55].



(A) (B) (C) (D)

FIGURE VIII.5: Camera trajectories and initial/final camera views during the regulation of 4 point features extracted from a cubic target. Blue lines represent the active phase (regulation of the error norm), while green lines represent the non-active phase (regulation of the full error vector)



(A) (B)

FIGURE VIII.6: Same pattern as in Fig. VIII.5, but by considering a non-structured object where a number of points was arbitrarily segmented and tracked

Figures VIII.5–VIII.6 report some results obtained using a cubic object with some black dots on its surface, and a non-textured object where a number of points was arbitrarily segmented and tracked. Figures. VIII.5(A)–(C) and Fig. VIII.6(B) depict the camera trajectory during the IBVS task, with the blue line representing the *active* phase (during the regulation of the error norm), and the green line the non-active phase (during the regulation of the full error vector). It is worth noticing how the active SfM induces a 'spiralling' motion to the camera in order to maximize the convergence rate of the 3D estimation error. Similar results can also be found in Figs. VIII.7(A–B) where multiple spiralling motions are present: indeed, in this case the target object was purposely displaced during the servoing for introducing an "external disturbance" able to increase both the servoing and the estimation errors above their minimum thresholds with a corresponding (re-)activation of the optimization of the camera motion. In all the tested cases, the proposed coupling between active perception and visual control showed some clear benefits in terms of: (*i*) obtaining a faster convergence of the structure estimation error during the servoing transient w.r.t. non-active cases, and (*ii*) imposing an improved closed-loop IBVS behavior by significantly mitigating the negative effects of an inaccurate knowledge of the scene structure.

VIII.1 Main references

The work presented in this Chapter is based on the following material:

- Ph.D. Thesis of Volker Grabe "Towards Robust Visual-Controlled Flight of Single and Multiple UAVs in GPS-Denied Indoor Environments", March 2014, University of Tübingen, Germany
- Ph.D. Thesis of Riccardo Spica "Contributions to Active Visual Estimation and Control of Robotic Systems" (tentative), December 2015 (expected date), University of Rennes 1, France

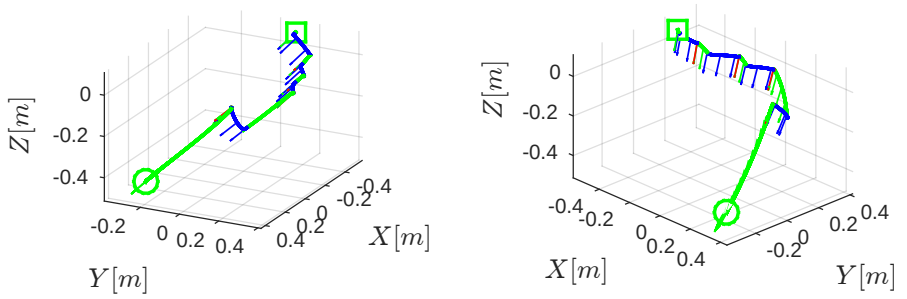


FIGURE VIII.7: Same pattern as in Fig. VIII.5, but, in this case, the target object was intentionally displaced for repeatedly triggering the activation of the optimization of the camera motion

- V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. Robuffo Giordano. Nonlinear Ego-Motion Estimation from Optical Flow for Online Control of a Quadrotor UAV. *The International Journal of Robotics Research*, 34(8):1114–1135, 2015 (ref. [J20])

- V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 491–497, St. Paul, MN, May 2012 (ref. [C34])

- V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. Robust Optical-Flow Based Self-Motion Estimation for a Quadrotor UAV. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Portugal, Oct. 2012 (ref. [C40])

- V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. A Comparison of Scale Estimation Schemes for a Quadrotor UAV based on Optical Flow and IMU Measurements. In *Proc. of the 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2013)*, 2013 (ref. [C49])

- R. Spica, P. Robuffo Giordano, and F. Chaumette. Active Structure from Motion: Application to Point, Sphere and Cylinder. *IEEE Transactions on Robotics*, 30(6):1499–1513, 2014 (ref. [J17])

- R. Spica, P. Robuffo Giordano, and F. Chaumette. Bridging Visual Control and Active Perception via a Large Projector Operator. *submitted to the IEEE Transactions on Robotics*, 2015 (ref. [J23])

- R. Spica and P. Robuffo Giordano. A Framework for Active Estimation: Application to Structure from Motion. In *Proc. of the 2013 IEEE Conf. on Decision and Control (CDC 2013)*, 2013 (ref. [C51])

- P. Robuffo Giordano, R. Spica, and F. Chaumette. An Active Strategy for Plane Detection and Estimation for a Monocular Camera. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014 (ref. [C53])

- R. Spica, P. Robuffo Giordano, and F. Chaumette. Active Structure from Motion for Spherical and Cylindrical Targets. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014 (ref. [C54])

- R. Spica, P. Robuffo Giordano, and F. Chaumette. Coupling Image-Based Visual Servoing with Active Structure from Motion. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014 (ref. [C55])

- R. Spica, P. Robuffo Giordano, and F. Chaumette. Plane Estimation by Active Vision from Point Features and Image Moments. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015 (ref. [C60])
- P. Robuffo Giordano, R. Spica, and F. Chaumette. Learning the Shape of Image Moments for Optimal 3D Structure Estimation. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015 (ref. [C61])
- Omar Tahri, Paolo Robuffo Giordano, and Youcef Mezouar. Rotation Free Active Vision. In *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2015)*, 2015 (ref. [C62])

VIII.2 Videos

- Active SfM for estimating the structure of point features, sphere and cylinder ([J17, C54]): experiments `https://youtu.be/u_UcCKtL214`
- Active SfM for estimating the structure of planar scenes ([C53, C60]): experiments `https://youtu.be/ydXfom-SDpA`
- Active SfM for estimating the ground velocity and distance to the plane of quadrotor UAV ([J20, C49, C40, C34]): experiments `http://youtu.be/ALo5xfH0LGo`
- Coupling visual control with active perception ([J23, C55]): experiments `http://youtu.be/kgoWUu-9fhs`

Chapter IX

Conclusions and Perspectives

This final Chapter is meant to summarize the main contributions of the works reported in the Thesis and to provide a general discussion on the several research directions discussed so far[1].

As often stated throughout the Thesis, a unified underlying theme has been the design (and implementation) of *shared control* architectures for interfacing a human operator with single or multiple mobile robots, and in particular quadrotor UAVs. As in any shared control scenario, this interfacing was meant to realize a *win-win* situation: the mobile robots contribute with their autonomy, precision, motion capability (but also 'expendability'); the human operator contributes with her/his superior cognitive skills and general situational awareness. This synergy ought to outperform the possibilities of (current) robotic systems which still possess quite limited capabilities when facing the complexity of real-world scenarios.

In this respect, we believe that the main contributions of the activities reported in the Thesis can be summarized as follows:

- two different combinations of sensory cues for a human operator in shared control of single/multiple mobile robots have been considered: a (novel) *visual/vestibular* feedback and a (more classical) *visual/force* feedback. The vestibular (i.e., self-motion) feedback is a less explored modality for providing cues to a human operator by stimulating his sense of self-motion via the use of a motion platform/motion simulator. This feedback has proven to be useful in, e.g., piloting a remote flying vehicle for increasing the operator's situational awareness during a shared task execution via a mechanism also known as *shared fate* [Hing and Oh, 2009]. The force feedback is, instead, a more classical solution exploiting the operator's ability to perceive forces when giving commands via an actuated input device. The force channel is known to have a limited 'bandwidth' in terms of amount of transmitted information, but it also represents a fast/reactive channel that can be exploited to quickly convey simple information such as allowed/forbidden motion directions, presence of obstacles or other impediments;
- a number of strategies for delegating some *autonomy* to the remote robot(s) has been proposed according to the particular tasks and scenarios. For instance, in the single robot case, Chapters IV–V have discussed two general approaches for providing cues to the human operator: a (classical) feedback based on the *instantaneous mismatch* between commanded and actual motion commands,

[1] The reader is referred to the end of Chapters IV–VIII for a more in-depth summary and discussion of the individual topics treated in the Thesis.

and a *shared planning framework* (Sect. V.2) able to generate a future trajectory (over some finite time window) accounting for both the robot mobility constraints and the operator's commands. The resulting (force) feedback becomes then representative of the *future* consequences of the operator's actions, as opposed to the standard case of *instantaneous* mismatch between commanded and actual motions, thus contributing to easing the operator's mental load while steering the mobile robot in cluttered/difficult environments.
In the multi-robot case, Chapters V–VI have instead detailed a number of possible *coordination schemes* for the robot group with the operator playing the role of an 'external planner' in charge of providing desired velocity commands to single/multiple nodes (or leaders). The main contributions here have been in the different approaches taken for resolving the (large) discrepancy between the few DOFs actuated by the operator (e.g., three translational velocity commands) vs. the many DOFs of the whole robot group. Two main paradigms have been explored: a *fixed topolgy* case in which the robot formation is supposed specified at the beginning of the task and the operator is allowed to act on the motions in the 'null-space' of the formation geometrical constraints (e.g., a collective translation/rotation for distance constraints), and a *time-varying topology* case in which the formation is left free to change/adapt online to cope with, e.g., inter-robot sensing constraints, but with the guarantee of still preserving some fundamental global properties (such as *connectivity* and *rigidity*) needed to allow for an effective coordination among the robots;

- a novel mechanical design for an overactuated quadrotor UAV has been discussed in Chapter VII, with the aim of providing full 6D motion capabilities to the widespread (but underactuated) quadrotor platform. Indeed, the underactuation of standard quadrotors (four control inputs for six DOFs) can pose some difficulties when navigating in harsh environments, or when needing to interact (i.e., exchange forces) with the external world. The overactuated design of Chapter VII can overcome these difficulties by guaranteeing full 6D motion capabilities during flight, and it thus represents, in our opinion, a good trade-off between increased mechatronic complexity (need to actuate the four propeller tilting axes) and improved flight performance;

- finally, a quite general *active perception* framework has been presented in Chapter VIII for optimizing *online* the performance (i.e., convergence rate and final accuracy) of 'visual-based' state estimation schemes by acting on the camera/robot motion. This has been applied to both the cases of a fixed manipulator carrying an eye-in-hand camera, and a quadrotor UAV observing a planar scene with a downlooking camera. Furthermore, the possibility of executing a *visual servoing task* (i.e., positioning the camera/robot w.r.t. a given target) concurrently to the active optimization of the visual state estimation task has also been explored from a theoretical and implementation perspective. In all cases, very good results have been obtained in terms of performance improvement w.r.t. non-active (classical) cases. As cameras are a widespread sensor found onboard mobile robots (and, in particular, on quadrotor UAVs), the proposed active perception techniques have then the potential (in our opinion) to improve the execution of many visual-based navigation tasks for single or multiple mobile robots (such as, e.g., the bearing formation control scenario described in Sect. V.3.1).

Overall, we believe that the research activities documented in the Thesis have succeeded in combining different ideas in original ways: for instance, while bilateral teleoperation of fixed robot manipulators, or even of single mobile robots, is a well-explored topic in the robotics community, the proposed shared control frameworks for single/multiple robots were, to the best of our knowledge, new contributions to the field, bridging together (at least) three different sub-areas: bilateral teleoperation, online trajectory

(re)planning, and multi-robot distributed coordination and estimation schemes. Similarly, the use of vestibular feedback for providing cues to a human operator in control of a remote mobile robot was a novel (and almost unexplored) possibility. The same goes for the overactuated quadrotor UAV concept reported in Chapter VII: while flying vehicles with tilting propellers are not a new idea, the particular application to the quadrotor UAV case was, to the best of our knowledge, an original contribution, as also proven by the interest generated in some specialized press (IEEE Spectrum Magazine).
Finally, another important achievement of most of the reported works has been, in our opinion, the *strong* emphasis placed on a thorough experimental validation of the various approaches. This experimental validation has indeed made it possible to (*i*) prove the *feasibility* of our contributions on real hardware and non-perfect environments (although still in partially controlled lab conditions), but also to (*ii*) assess the limits of the various modeling assumptions, simplifications and idealizations that are always (explicitly or implicitly) made at the design/theoretical stage. Besides the specific results reported here, we believe that these findings can also be of practical interest for any researcher working the field of multi-robot(-UAV) coordination and bilateral shared control.

Having listed what, in our opinion, are the core Thesis achievements, we shall now discuss the main limitations and possible improvements of the reported works which are, obviously, far from being "flawless". First of all, many of the experimental validations involved, to some extent, partially controlled lab conditions that eased the implementation side (such as the extensive use of external motion tracking facilities for tracking the robot poses). The reported validation of the various approaches can then be only considered as a 'partial' result, or as a solid 'proof-of-concept', preliminary to all the additional testing/developments that would be needed for facing more realistic scenarios (e.g., for flying outdoor in harsh conditions, or indoor but by only relying or local onboard sensing without any *centralized* aid). Indeed, the deployment truly autonomous (i.e., only exploiting their 'own skills' such as local sensing/-communication) groups of mobile robots, and in particular to flying robots — quadrotor UAVs, still presents many unsolved challenges, mainly on the perception/decision-making side. The reasons for these difficulties are manifold and span a considerable range of robotics/computer vision/engineering areas: within the scope of this Thesis, the use of onboard local sensing and communication introduces, for instance, noise, delays, low update rates, but also state-dependent constraints on the possibility to acquire the needed information. For example, non-obstructed visibility and limited field of view hinder the use of onboard cameras in real scenarios where the assumption of a continuous perfect visibility is clearly unrealistic. Furthermore, lack of a global centralized measurement of the whole group state requires the agents to run (decentralized) local estimations of any global quantity of interest (e.g., the relative position w.r.t. the formation barycenter) with the inherent loss of performance and optimality w.r.t. centralized solutions. Finally, use of only local (i.e., expressed in the agent body-frame) relative measurements creates the need of agreeing on a common reference frame where to, e.g., express relative positions. This, in turn, requires the (decentralized) maintenance of global group properties, such as the mentioned connectivity and formation rigidity, that allow for convergence of the employed estimation schemes — an additional (and often complex) constraint to meet during the mission. It is then our opinion that, despite the positive results reported in this Thesis (and in many other works in the current state-of-the-art), a big gap still exists between the design and the actual realization of truly autonomous sensor-based applications for single/multiple mobile robots able to perform tasks in highly unstructured real-world conditions. And this gap is due to existing methodological, algorithmic and implementative limitations that still need to be overcome in the coming years.

A second limitation of the reported works, as also stressed at the end of Chapter V, is the discrepancy

between the (large) efforts spent on the *algorithmic* difficulties for implementing a shared control architecture for single/multiple mobile robots vs. the attention placed on the validation of the proposed approaches by means of principled user's studies. Indeed, apart from the analyses of Sect. IV.4.2 and Sect. V.4, the remaining shared control architectures have always exploited the *differential paradigm* of providing a force/vestibular cue proportional to the (instantaneous or integral) mismatch between commanded and actual robot motion. This choice is quite standard in many bilateral teleoperation settings (e.g., when operating remote robots in impedance mode), and it thus represents the first natural attempt when designing shared control architectures. However, the pertinence of such a choice might be questioned as not being always supported by a solid user's study assessing, e.g., the various pros/cons of different cueing possibilities. This is especially true in view of the specificities of the particular robotic systems considered in the Thesis (single and, even more, multiple mobile robots) that would certainly require a dedicated study for assessing which cue combination is 'best' according to the particular scenario/application. Therefore, in our opinion any future development or improvement of the proposed shared control approaches should definitely focus on the 'perceptual optimization' of the various cueing possibilities.

Finally, even by restricting the attention to the sole algorithmic contributions, the Thesis has still left some open points that should be addressed in future extensions. In our opinion, it is for instance worth mentioning the (unsolved) issue affecting the *connectivity maintenance* and *rigidity maintenance* coordination schemes of Chapters V–VI. Indeed, the implementation of these two methods is based on the possibility of estimating, in a decentralized way, a particular eigenvalue/eigenvector pair of the corresponding Laplacian and Rigidity matrixes. Nevertheless, the employed estimation methods (taken from the state-of-the-art) are known to become ill-conditioned (e.g., to lose accuracy or even diverge) in presence of *repeated* eigenvalues. This known shortcoming is often overlooked (as in the reported works), and it is just assumed that the relevant eigenvalues remain 'simple' during motion. This assumption is, however, not supported by any theoretical guarantee, possibly causing poor performance of the eigenvector/eigenvalue estimation (and of the overall coordination scheme as well). A proper treatment of the problem would instead require the development of novel estimation methods *robust* against the occurrence of repeated eigenvalues in order to cope with this issue in a rigorous way.
As an additional limitation, all the proposed schemes did not consider, in an explicit way, the possible presence of measurement noise, delays, or packet losses and link failures arising when relying on onboard sensing and local communication among robots. These are clear shortcomings which would affect any real (multi-)robot application. A typical workaround, also exploited in the reported works, is to rely on gain tuning and/or on the possibility to move "slow enough" so that the effect of these non-idealities becomes negligible w.r.t. the main group motion. However, more advanced techniques should be exploited for dealing with noise and delays. For instance, for what concerns communication reliability, a very interesting possibility could be to consider *stochastic* extensions of the notion of connectivity (and of rigidity) that would take into account the possibility of losing/gaining communication links with some given probability over time [Kar and Moura, 2010]. This probabilistic modeling could be inspired by physical considerations of the employed inter-robot communication medium as nicely discussed in [Fink et al., 2012]. One could then try to maximize the expected value of the "probabilistic connectivity eigenvalue" as, e.g., defined in [Kar and Moura, 2010], so as to ensure an average level of connectivity in the group over time. The same technique could obviously also be extended to the rigidity maintenance case, or similar scenarios. Finally, in order to cope with limited refresh rate and delays of the employed sensors, advanced control techniques, such as Model-Predictive Control (MPC), could be useful [Bemporad and Rocchi, 2011] thanks to their ability to predict (to some extent) the state evolution over a future time

(A) Copyright of Amazon

(B) Copyright of DHL

FIGURE IX.1: Amazon and DHL have recently started to consider the use of quadrotor UAVs as platforms for shipping goods, thus indicating the very strong interest of industry (and society) in the use of mobile (flying) robots for solving everyday needs

window. A quadrotor UAV could, for instance, accept a very slow sampling rate of its camera by propagating over time the (simulated) positions of its surrounding UAVs until a new image becomes available. Use of MPC techniques would also prove very effective for coping with unavoidable sensing limitations such as, e.g., the typical limited field of view of onboard cameras; one could, for instance, enable a quadrotor to temporarily fly 'blindly' over short phases with a guarantee of subsequently regaining full visual control. Finally, we believe that It would also be very interesting to investigate the applicability of MPC techniques (or similar ideas) to the active perception machinery discussed in Chapter VIII, in order to allow for consecutive re-planning of a future trajectory with maximum information content to be executed by the robot(s). This would indeed make it easier to take into account a number of constraints, such as limited robot mobility, environmental obstructions (e.g., obstacles), or, again, sensing limitations (e.g., presence of temporary blind phases).

To conclude: we believe that the field of shared control of single and multiple mobile robots (and, in particular, of quadrotor UAVs) is a very promising and exciting one, with a bright future in terms of perspectives, application scenarios with practical utility for society and industry, and, last but not least, also a number of technical challenges that still need to be solved, but whose resolution would significantly advance the general robotics state-of-the-art. Focusing on the specific case of quadrotor UAVs, it is for instance very indicative that big companies such as DHL and Amazon have concretely started to consider the use of quadorotor UAVs for delivery tasks, see Fig. IX.1. Small-size flying robots have been used to investigate disaster sites after earthquakes in Christchurch (New Zealand) and Emilia Romagna (Italy), and most prominently at the Fukushima Daiichi power plant in Japan. In [Michael et al., 2012] impressive results in multi-floor exploration and mapping were obtained by exploiting the cooperation of a ground and an aerial robot in an earthquake-damaged building in Sendai, Japan. Three large-scale (and very ambitious) European research projects (SHERPA, ARCAS and AEROARMS [ARCAS, 2011, SHERPA, 2012, AEROARMS, 2015] have been recently funded on the use of multiple (ground/flying) robots for addressing societal needs such as cooperative transportation/assembly and search and rescue in difficult environments. The total amount of EU funding for these two projects is in the order of 20M€, thus clearly confirming even further the strong strategic interest in the multi-robot technology also at the European level. A number of researchers have started addressing the complex problem of controlling a flying robot (or multiple flying robots) equipped with an onboard manipulator arm, see [Yang and Lee, 2014, Gioioso et al., 2014, Yang and Lee, 2015, Yüksel et al., 2015] and in particular [Kondak et al.,

2014] for an impressive technological demonstrator involving a full-scale helicopter equipped with a 7-DOF lightweight KUKA arm.
The list could still be long: it is our hope that the research activities reported in this Thesis will be able, over the next years, to bring an added value to the community and to foster additional research in the field.

Appendix A

Annex to Chapter VI

Article

The International Journal of
Robotics Research
2015, Vol. 34(1) 105–128

DOI: 10.1177/0278364914546173
ijr.sagepub.com
SAGE

Decentralized rigidity maintenance control with range measurements for multi-robot systems

Daniel Zelazo[1], Antonio Franchi[2], Heinrich H. Bülthoff[3,4] and Paolo Robuffo Giordano[5]

Abstract

This work proposes a fully decentralized strategy for maintaining the formation rigidity of a multi-robot system using only range measurements, while still allowing the graph topology to change freely over time. In this direction, a first contribution of this work is an extension of rigidity theory to weighted frameworks and the rigidity eigenvalue, which when positive ensures the infinitesimal rigidity of the framework. We then propose a distributed algorithm for estimating a common relative position reference frame amongst a team of robots with only range measurements in addition to one agent endowed with the capability of measuring the bearing to two other agents. This first estimation step is embedded into a subsequent distributed algorithm for estimating the rigidity eigenvalue associated with the weighted framework. The estimate of the rigidity eigenvalue is finally used to generate a local control action for each agent that both maintains the rigidity property and enforces additional constraints such as collision avoidance and sensing/communication range limits and occlusions. As an additional feature of our approach, the communication and sensing links among the robots are also left free to change over time while preserving rigidity of the whole framework. The proposed scheme is then experimentally validated with a robotic testbed consisting of six quadrotor unmanned aerial vehicles operating in a cluttered environment.

Keywords

Graph rigidity, decentralized control, multi-robot, distributed algorithms, distributed estimation

1. Introduction

The coordinated and decentralized control of multi-robot systems is an enabling technology for a variety of applications. Multi-robot systems benefit from an increased robustness against system failures due to their ability to adapt to dynamic and uncertain environments. There are also numerous economic benefits by considering the price of small and cost-effective autonomous systems as opposed to their more expensive monolithic counterparts. Currently, there is a great interest in implementing these systems from deep-space interferometry missions and distributed sensing and data collection, to civilian search and rescue operations, among others (Bristow et al., 2000; Akyildiz et al., 2002; Murray, 2006; Anderson et al., 2008a; Michael et al., 2009; Mesbahi and Egerstedt, 2010; Lindsey et al., 2011).

The challenges associated with the design and implementation of multi-agent systems range from hardware and software considerations to the development of a solid theoretical foundation for their operation. In particular, the *sensing* and *communication* capabilities of each agent will dictate the distributed protocols used to achieve team objectives. For example, if each agent in a multi-robot system is equipped with a GPS-like sensor, then tasks such as formation keeping or localization can be trivially accomplished by communication between robots of their state information in a common world-frame. However, in

[1]Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa, Israel
[2]Centre National de la Recherche Scientifique (CNRS), Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), Toulouse, France
[3]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[4]Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea
[5]CNRS at Irisa and Inria Rennes Bretagne Atlantique, Rennes, France

Corresponding author:
Paolo Robuffo Giordano, CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France.
Email: prg@irisa.fr

applications operating in harsher environments, i.e. indoors, underwater, or in deep space, GPS is not a viable sensing option (Scaramuzza et al., 2014). Indeed, in these situations, agents must rely on sensing without knowledge of a common inertial reference frame (Franchi et al., 2012a). In these scenarios, *relative sensing* can provide accurate measurements of, for example, range or bearing, but without any common reference frame.

A further challenge related to the sensing capabilities of multi-robot systems is the availability of these measurements. Sensing constraints such as line-of-sight requirements, range, and power limitations introduce an important system-level requirement, and also lead to an inherently time-varying description of the sensing network. Successful decentralized coordination protocols, therefore, must also be able to manage these constraints.

These issues lead to important architectural requirements for the sensing and communication topology in order to achieve the desired higher-level tasks (i.e. formation keeping or localization). The *connectivity* of the sensing and communication topology is one such property that has received considerable attention in the multi-robot communities (Ji and Egerstedt, 2007; Robuffo Giordano et al., 2011, 2013). However, connectivity alone is not sufficient to perform certain tasks when only relative sensing is used. For these systems, the concept of *rigidity* provides the correct framework for defining an appropriate sensing and communication topology architecture. Rigidity is a combinatorial theory for characterizing the "stiffness" or "flexibility" of structures formed by rigid bodies connected by flexible linkages or hinges.

The study of rigidity has a rich history with contributions from mathematics and engineering disciplines (Laman, 1970; Tay and Whiteley, 1985; Jacobs, 1997; Eren et al., 2004; Connelly and Whiteley, 2009; Krick et al., 2009; Shames et al., 2009). Recently, rigidity theory has taken an outstanding role in the motion control of mobile robots. The rigidity framework allows for applications, such as formation control, to employ control algorithms relying on only *relative distance measurements*, as opposed to *relative position measurements* from a global or relative internal frame (Olfati-Saber and Murray, 2002; Baillieul and McCoy, 2007; Smith et al., 2007; Anderson et al., 2008a,b; Krick et al., 2009). For example, Krick et al. (2009) showed that formation stabilization using only distance measurements can be achieved only if rigidity of the formation is maintained. Moreover, rigidity represents also a necessary condition for estimating relative positions using only relative distance measurements (Aspnes et al., 2006; Calafiore et al., 2010b).

In a broader context, rigidity turns out to be an important architectural property of many multi-agent systems when a common inertial reference frame is unavailable. Applications that rely on sensor fusion for localization, exploration, mapping and cooperative tracking of a target, can all benefit from notions in rigidity theory (Aspnes et al., 2006; Shames et al., 2009; Calafiore et al., 2010a; Wu et al., 2010; Williams et al., 2014). The concept of rigidity, therefore, provides the theoretical foundation for approaching *decentralized* solutions to the aforementioned problems using *distance measurement sensors*, and thus establishing an appropriate framework for relating system-level architectural requirements to the sensing and communication capabilities of the system.

1.1. Main contributions

In general, rigidity as a property of a given formation (i.e. of the robot spatial arrangement) has been studied from either a purely combinatorial perspective (Laman, 1970), or by providing an algebraic characterization via the state-dependent *rigidity matrix* (Tay and Whiteley, 1985). In our previous work (Zelazo et al., 2012), we introduced a related matrix termed the *symmetric rigidity matrix*. A main result of Zelazo et al. (2012) was to provide a necessary and sufficient condition for rigidity in the plane in terms of the positivity of a particular eigenvalue of the symmetric rigidity matrix; this eigenvalue we term the *rigidity eigenvalue*. This result is in the same spirit as the celebrated Fiedler eigenvalue[1] and its relation to the connectivity of a graph (Godsil and Royle, 2001). A first contribution of this work is the extension of the results on the rigidity eigenvalue provided by Zelazo et al. (2012) to three-dimensional frameworks, as well as the introduction of the concept of *weighted rigidity* and the corresponding *weighted rigidity matrix*. This notion allows for the concept of rigidity to include state-dependent weight functions on the edges of the graph, weights which can then be exploited to take into account inter-agent sensing and communication constraints and/or requirements.

A gradient-based *rigidity maintenance action* aimed at 'maximizing' the rigidity eigenvalue was also proposed by Zelazo et al. (2012). However, while this gradient control law was decentralized in structure, there was still a dependence on the availability of several global quantities, namely, of the robot relative positions in some *common reference frame*, of the value of the rigidity eigenvalue, and of the *rigidity eigenvector* associated with the rigidity eigenvalue. A main contribution of this work is then the development of the machinery needed to distributedly estimate all of these global quantities by resorting to only *relative distance measurements* among neighbors, so as to ultimately allow for a *fully distributed and range-based* implementation of the rigidity maintenance controller. To this end, we first show that if the formation is infinitesimally rigid, it is possible to *distributedly* estimate the relative positions of neighboring robots in a common reference frame from only range-based measurements. Our approach relies explicitly on the form of the symmetric rigidity matrix developed here, in contrast to other approaches focusing on distributed implementations of centralized estimation schemes, such as a Gauss–Newton approach used by Calafiore et al. (2010a). This first step is then instrumental for the subsequent development of the distributed

estimation of the rigidity eigenvalue and eigenvector needed by the rigidity gradient controller. This is obtained by exploiting an appropriate modification of the *power iteration method* for eigenvalue estimation following from the works Yang et al. (2010) and Robuffo Giordano et al. (2011) for the distributed estimation of the connectivity eigenvalue of the graph Laplacian and now applied to rigidity. Finally, we show how to exploit the weights on the graph edges to embed constraints and requirements such as inter-robot and obstacle avoidance, limited communication and sensing ranges, and line-of-sight occlusions, into a unified gradient-based *rigidity maintenance control law.*

Our approach, therefore, can be considered as a contribution to the general problem of distributed strategies for maintaining certain architectural features of a multi-robot system (i.e. connectivity or rigidity) with minimal sensing requirements (only relative distance measurements). In addition, we also provide a thorough experimental validation of the entire framework by employing a group of six quadrotor unmanned aerial vehicles (UAVs) as robotic platforms to demonstrate the feasibility of our approach in real-world conditions.

The organization of this paper is as follows. Section 1.2 provides a brief overview of some notation and fundamental theoretical properties of graphs. In Section 2, the theory of rigidity is introduced, and our extension of the rigidity eigenvalue to three-dimensional weighted frameworks is given. We then proceed to present a general strategy for a distributed rigidity maintenance controller in Section 3. This section will provide details on certain operational constraints of the multi-robot team and how these constraints can be embedded in the control law. This section also highlights the need to develop distributed algorithms for estimating a common reference frame for the team, outlined in Section 4, and estimation of the rigidity eigenvalue and eigenvector, detailed in Section 5. The results of the previous sections are then summarized in Section 6 where the full distributed rigidity maintenance controller is given. The applicability of these results are then experimentally demonstrated on a robotic testbed consisting of six quadrotor UAVs operating in a obstacle populated environment. Details of the experimental setup and results are given in Section 7. Finally, some concluding remarks are offered in Section 8.

1.2. Preliminaries and notation

The notation employed is standard. Matrices are denoted by capital letters (e.g. A), and vectors by lowercase letters (e.g. x). The ij-th entry of a matrix A is denoted $[A]_{ij}$. The rank of a matrix A is denoted $\mathbf{rk}[A]$. Diagonal matrices will be written as $D = \mathbf{diag}\{d_1,\ldots,d_n\}$; this notation will also be employed for block-diagonal matrices. A matrix and/or a vector that consists of all zero entries will be denoted by $\mathbf{0}$; whereas, '0' will simply denote the scalar zero. Similarly, the vector $\mathbb{1}_n$ denotes the $n \times 1$ vector of all ones. The $n \times n$ identity matrix is denoted as I_n. The set of real numbers will be denoted as $\mathbb{R}$, and $\|\cdot\|$ denotes the standard Euclidean 2-norm for vectors. The Kronecker product of two matrices A and B is written as $A \otimes B$ (Horn and Johnson, 1991).

Graphs and the matrices associated with them will be widely used in this work (see, e.g. Godsil and Royle, 2001). An undirected (simple) weighted graph $\mathcal{G}$ is specified by a vertex set $\mathcal{V}$, an edge set $\mathcal{E}$ whose elements characterize the incidence relation between distinct pairs of $\mathcal{V}$, and diagonal $|\mathcal{E}| \times |\mathcal{E}|$ weight matrix W, with $[W]_{kk} \geq 0$ the weight on edge $e_k \in \mathcal{E}$. In this work we consider only finite graphs and denote the cardinality of the node and edge sets as $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$. Two vertices i and j are called *adjacent* (or neighbors) when $\{i,j\} \in \mathcal{E}$. The *neighborhood* of the vertex i is the set $\mathcal{N}_i = \{j \in \mathcal{V} | \{i,j\} \in \mathcal{E}\}$. An *orientation* of an undirected graph $\mathcal{G}$ is the assignment of directions to its edges, i.e. an edge e_k is an ordered pair (i, j) such that i and j are, respectively, the initial and the terminal nodes of e_k.

The incidence matrix $E(\mathcal{G}) \in \mathbb{R}^{n \times m}$ is a $\{0, \pm 1\}$-matrix with rows and columns indexed by the vertices and edges of $\mathcal{G}$ such that $[E(\mathcal{G})]_{ik}$ has the value '$+1$' if node i is the initial node of edge e_k, '-1' if it is the terminal node, and '0' otherwise. The degree of vertex i, d_i, is the cardinality of the set of vertices adjacent to it. The degree matrix, $\Delta(\mathcal{G})$, and the adjacency matrix, $A(\mathcal{G})$, are defined in the usual way (Godsil and Royle, 2001). The (graph) Laplacian of $\mathcal{G}$, $L(\mathcal{G}) = E(\mathcal{G})E(\mathcal{G})^{\mathrm{T}} = \Delta(\mathcal{G}) - A(\mathcal{G})$, is a positive-semidefinite matrix. One of the most important results from algebraic graph theory in the context of collective motion control states that a graph is connected if and only if the second smallest eigenvalue of the Laplacian is positive (Godsil and Royle, 2001).

Table 1 provides a summary of the notations used throughout the document.

2. Rigidity and the rigidity eigenvalue

In this section we review the fundamental concepts of graph rigidity (Graver et al., 1993; Jackson, 2007). A contribution of this work is an extension of our previous results on the concepts of the *symmetric rigidity matrix* and *rigidity eigenvalue* for three-dimensional ambient spaces (Zelazo et al., 2012), and the notion of *weighted frameworks*.

2.1. Graph rigidity and the rigidity matrix

We consider graph rigidity from what is known as a *d-dimensional bar-and-joint framework*. A framework is the pair $(\mathcal{G}, p)$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph, and $p : \mathcal{V} \rightarrow \mathbb{R}^d$ maps each vertex to a point in $\mathbb{R}^d$. In this work we consider frameworks in a three-dimensional ambient space, i.e. $d = 3$. Therefore, for node $u \in \mathcal{V}$, $p(u) = [\,p_u^x \quad p_u^y \quad p_u^z\,]^{\mathrm{T}}$ is the position vector in $\mathbb{R}^3$ for the mapped node. We refer to the matrix $p(\mathcal{V}) = [\,p(v_1) \quad \cdots \quad p(v_n)\,]^{\mathrm{T}} \in \mathbb{R}^{n \times 3}$ as the *position matrix*. We now provide some basic definitions.

Definition 2.1. *Frameworks* $(\mathcal{G}, p_0)$ *and* $(\mathcal{G}, p_1)$ *are* equivalent *if* $\|p_0(u)-p_0(v)\| = \|p_1(u) - p_1(v)\|$ *for all* $\{u, v\} \in \mathcal{E}$, *and are* congruent *if* $\|p_0(u)-p_0(v)\| = \|p_1(u)-p_1(v)\|$ *for all* $\{u, v\} \in \mathcal{V}$.

Definition 2.2. *A framework* $(\mathcal{G}, p_0)$ *is* globally rigid *if every framework which is equivalent to* $(\mathcal{G}, p_0)$ *is congruent to* $(\mathcal{G}, p_0)$.

Definition 2.3. *A framework* $(\mathcal{G}, p_0)$ *is* rigid *if there exists an* $\epsilon > 0$ *such that every framework* $(\mathcal{G}, p_1)$ *which is equivalent to* $(\mathcal{G}, p_0)$ *and satisfies* $\|p_0(v) - p_1(v)\| < \epsilon$ *for all* $v \in \mathcal{V}$, *is congruent to* $(\mathcal{G}, p_0)$.

Table 1. Notation.

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	A graph defined by its vertex and edge sets
$\mathcal{N}_i(t)$	*Time-varying* neighborhood of node $v_i \in \mathcal{V}$
$p(i)$	Position vector in $\mathbb{R}^3$ of the mapped node $v_i \in \mathcal{V}$
p_i^s	$s \in \{x, y, z\}$ coordinate of position vector for node i
$p(\mathcal{V})$	Stacked position matrix of all nodes ($\mathbb{R}^{n\times 3}$)
$\xi(i)$	Velocity vector in $\mathbb{R}^3$ of the node $v_i \in \mathcal{V}$
$(\mathcal{G}, p, \mathcal{W})$	A weighted framework
$R(p, \mathcal{W})$	Rigidity matrix of a weighted framework
$\mathcal{R}$	Symmetric rigidity matrix of a weighted framework
λ_7, $\mathbf{v}_7$ $(\mathbf{v})$	Rigidity eigenvalue and eigenvector
ℓ_{ij}	Distance between nodes $v_i, v_j \in \mathcal{V}$, i.e., $\|p(v_i) - p(v_j)\|$
$\hat{\lambda}_7^i$	Agent i's estimate of the rigidity eigenvalue
$\hat{\mathbf{v}}_i^s$	s coordinate of the agent i estimation of the rigidity eigenvector
$\hat{p}_{i,c}$	Agent i estimate of relative position vector $p_i - p_c$
$\hat{p}$	Stacked vector of the relative position vector estimate $p_i - p_c$, $i = 1 \ldots n$
avg (x)	The average of a vector $x \in \mathbb{R}^n$, $\mathbf{avg}(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$
$\overline{\mathbf{v}}_i^x$	Agent i estimate of $\mathbf{avg}(\hat{\mathbf{v}}^x)$
$\overline{\mathbf{v}}_i^{2x}$	Agent i estimate of $\mathbf{avg}(\hat{\mathbf{v}}^x \circ \hat{\mathbf{v}}^x)$
z_i^{xy}	Agent i estimate of $\mathbf{avg}(\hat{p}^{y,c} \circ \hat{\mathbf{v}}^x - \hat{p}^{x,c} \circ \hat{\mathbf{v}}^y)$
z_i^{xz}	Agent i estimate of $\mathbf{avg}(\hat{p}^{z,c} \circ \hat{\mathbf{v}}^x - \hat{p}^{x,c} \circ \hat{\mathbf{v}}^z)$
z_i^{yz}	Agent i estimate of $\mathbf{avg}(\hat{p}^{y,c} \circ \hat{\mathbf{v}}^z - \hat{p}^{z,c} \circ \hat{\mathbf{v}}^y)$

Definition 2.4. *A* minimally rigid graph *is a rigid graph such that the removal of any edge results in a non-rigid graph*.

Figure 1 shows three frameworks illustrating the above definitions. The frameworks in Figure 1(a) are both minimally rigid and are equivalent to each other, but are not congruent, and therefore not globally rigid. By adding an additional edge, as in Figure 1(b) (the edge $\{v_4, v_5\}$), the framework becomes globally rigid. The key feature of global rigidity, therefore, is that the distances between *all* node pairs are maintained for different framework realizations, and not just those defined by the edge set.

By parameterizing the position map by a positive scalar representing time, we can also consider *trajectories* of a framework. That is, the position map now becomes $p : \mathcal{V} \times \mathbb{R} \to \mathbb{R}^3$ and is assumed to be continuously differentiable with respect to time. We then explicitly write $(\mathcal{G}, p, t)$ so as to represent a *time-varying framework*. In this direction, we can define a set of trajectories that are *edge-length preserving*, in the sense that for each time $t \geq t_0$, the framework $(\mathcal{G}, p, t)$ is equivalent to the framework $(\mathcal{G}, p, t_0)$. More formally, an edge-length preserving framework must satisfy the constraint

$$\|p(v,t) - p(u,t)\| = \|p(v,t_0) - p(u,t_0)\| = \ell_{vu}, \quad \text{for all } t \geq t_0 \tag{1}$$

and for all $\{v, u\} \in \mathcal{E}$.

One can similarly assign velocity vectors $\xi(u, t) \in \mathbb{R}^3$ to each vertex $u \in \mathcal{V}$ for each point in the configuration space such that

$$(\xi(u,t) - \xi(v,t))^{\mathrm{T}}(p(u,t) - p(v,t)) = 0, \text{ for all } \{u, v\} \in \mathcal{E} \tag{2}$$

Note that this relation can be obtained by time-differentiation of the length constraint described in (1). These motions are referred to as *infinitesimal motions* of the mapped vertices $p(u, t)$, and one has

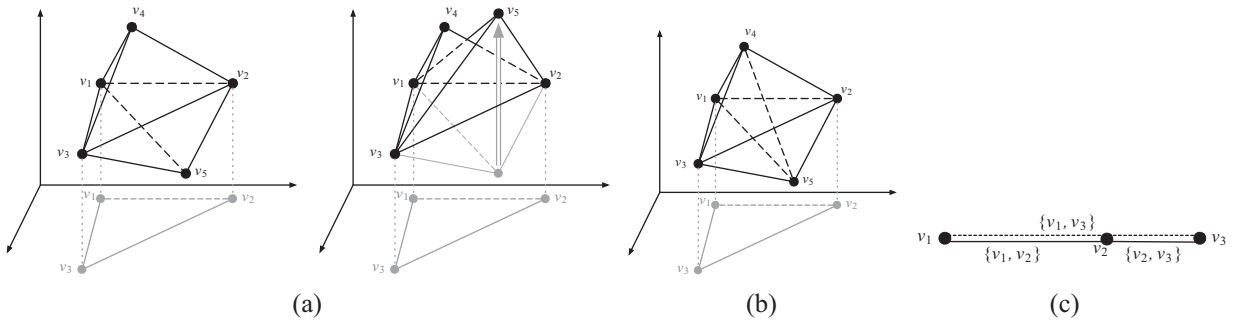


Fig. 1. Examples of rigid and infinitesimally rigid frameworks in $\mathbb{R}^3$. (a) Two equivalent minimally rigid frameworks in $\mathbb{R}^3$. The framework on the right-hand side is obtained by the reflection of the position of v_5 with respect to the plane characterized by the positions of v_1, v_2, and v_3 (as illustrated in gray). (b) An infinitesimally and globally rigid framework in $\mathbb{R}^3$. (c) A non-infinitesimally rigid framework (note that vertexes v_1 and v_3 are connected). Note that in (a) and (b) the 3D points associated with each vertex do not lie on the same plane, while in (c) the 3D points are aligned.

$$\dot{p}(u,t) = \xi(u,t) \tag{3}$$

For the remainder of this paper, we drop the explicit inclusion of time for frameworks and simply write $(\mathcal{G},p)$ and $p(u)$ and $\xi(u)$ for the time-varying positions and velocities. The velocity vector $\xi(u)$ will be treated as the *agent velocity input* throughout the rest of the paper (see Section 3).

Infinitesimal motions of a framework can be used to define a stronger notion of rigidity.

Definition 2.5. *A framework is called* infinitesimally rigid *if every possible motion that satisfies (2) is trivial (i.e. consists of only global rotations and translations of the whole set of points in the framework).*

An example of an infinitesimally rigid graph in $\mathbb{R}^3$ is shown in Figure 1(b). Furthermore, note that infinitesimal rigidity implies rigidity, but the converse is not true (Tay and Whiteley, 1985), see Figure 1(c) for a rigid graph in $\mathbb{R}^3$ that is not infinitesimally rigid.

The infinitesimal motions in (2) define a system of m linear equations in the vector of unknown velocities $\xi = [\xi^{\mathrm{T}}(v_1)\ldots\xi^{\mathrm{T}}(v_n)]^{\mathrm{T}} \in \mathbb{R}^{3n}$. This system can be equivalently written as the linear matrix equation

$$R(p)\xi = \mathbf{0}$$

where $R(p) \in \mathbb{R}^{m\times 3n}$ is called *rigidity matrix* (Tay and Whiteley, 1985). Each row of $R(p)$ corresponds to an edge $e = \{u, v\}$ and the quantity $(p(u) - p(v))$ represents the non-zero coefficients for that row. For example, the row corresponding to edge e has the form

$$\Big[-\mathbf{0}- \underbrace{(p(u)-p(v))^{\mathrm{T}}}_{\text{vertex } u} -\mathbf{0}- \underbrace{(p(v)-p(u))^{\mathrm{T}}}_{\text{vertex } v} -\mathbf{0}- \Big]$$

The definition of infinitesimal rigidity can then be restated in the following form:

Lemma 2.6 (Tay and Whiteley, 1985). *A framework* $(\mathcal{G},p)$ *in* $\mathbb{R}^3$ *is infinitesimally rigid if and only if* $\mathbf{rk}[R(p)] = 3n - 6$.

Note that, as expected from Definition 2.5, the six-dimensional kernel of $R(p)$ for an infinitesimally rigid graph only allows for six independent *feasible* framework motions, that is, the above-mentioned collective roto-translations in $\mathbb{R}^3$ space. Note also that, despite its name, the rigidity matrix is actually characterizing *infinitesimal rigidity* rather than *rigidity* of a framework.

2.2. Rigidity of weighted frameworks

We now introduce an important generalization to the concept of rigidity and the rigidity matrix by introducing weights to the framework. Indeed, as discussed in the introduction, our aim is to propose a control law able to not only maintain *infinitesimal rigidity* of the formation as per Definition 2.5, but to also concurrently manage additional constraints typical of multi-robot applications such as collision avoidance and limited sensing and communication. This latter objective will be accomplished via the introduction of suitable state-dependent weights, thus requiring an extension of the traditional results on rigidity to a weighted case.

Definition 2.7. *A d-dimensional* weighted *framework is the triple* $(\mathcal{G},p,\mathcal{W})$, *where* $\mathcal{G} = (\mathcal{V},\mathcal{E})$ *is a graph,* $p : \mathcal{V} \to \mathbb{R}^d$ *is a function mapping each vertex to a point in* $\mathbb{R}^d$, and $\mathcal{W} : (\mathcal{G},p) \to \mathbb{R}^m$ *is a function of the framework that assigns a scalar value to each edge in the graph.*

Using this definition, we can also define the corresponding *weighted rigidity matrix*, $R(p,\mathcal{W})$, as

$$R(p,\mathcal{W}) = W(\mathcal{G},p)R(p) \tag{4}$$

where $W(\mathcal{G},p) \in \mathbb{R}^{m\times m}$ is a diagonal matrix containing the elements of the vector $\mathcal{W}(\mathcal{G},p)$ on the diagonal. Often we will simply refer to the weight matrix $W(\mathcal{G},p)$ as W when the underlying graph and map p is understood.

Remark 2.8. *Note that the rigidity matrix R(p) can also be considered as a weighted rigidity matrix with* $W(\mathcal{G},p) = I$. *Another useful observation is that the unweighted framework* $(\mathcal{G},p)$ *can also be cast as a weighted framework* $(K_n,p,\mathcal{W})$, *where* K_n *is the complete graph on n nodes and* $[W(\mathcal{G},p)]_{ii}$ *is 1 whenever* $e_i \in \mathcal{E}(K_n)$ *is also an edge in* $\mathcal{G}$, *and 0 otherwise.*

Weighted rigidity can lead to a slightly different interpretation of infinitesimal rigidity, where the introduced weights might cause the rigidity matrix to lose rank. That is, an unweighted framework might be infinitesimally rigid, whereas a weighted version might not. This observation is trivially observed by considering a minimally infinitesimally rigid framework $(\mathcal{G},p)$ and introducing a weight with a 0 entry on any edge. We formalize this with the following definitions.

Definition 2.9. *The* unweighted counterpart *of a weighted framework* $(\mathcal{G},p,\mathcal{W})$ *is the framework* $(\widehat{\mathcal{G}},p)$ *where the graph* $\widehat{\mathcal{G}} = (\mathcal{V},\widehat{\mathcal{E}})$ *is such that* $\widehat{\mathcal{E}} \subset \mathcal{E}$ *and the edge* $e_i \in \mathcal{E}$ *is also an edge in* $\widehat{\mathcal{G}}$ *if and only if the corresponding weight is non-zero (i.e.* $[W(\mathcal{G},p)]_{ii} \neq 0$).

Definition 2.10. *A weighted framework is called infinitesimally rigid if its unweighted counterpart is infinitesimally rigid.*

We now present a corollary to Lemma 2.6 for weighted frameworks.

Corollary 2.11. *A weighted framework* $(\mathcal{G},p,\mathcal{W})$ *in* $\mathbb{R}^3$ *is infinitesimally rigid if and only if* $\mathbf{rk}[R(p,\mathcal{W})] = 3n - 6$.

Proof. The statement follows from the fact that $\mathbf{rk}[R(p,\mathcal{W})] = \mathbf{rk}[\hat{R}(p)]$, where $\hat{R}(p)$ is the rigidity matrix for the unweighted counterpart of $(\mathcal{G},p,\mathcal{W})$.

2.3. The rigidity eigenvalue

In our previous work (Zelazo et al., 2012), we introduced an alternative representation of the rigidity matrix that transparently separates the underlying graph from the

positions of each vertex. Here we recall the presentation and extend it to the case of three-dimensional frameworks.

Definition 2.12 (Zelazo et al. (2012)). *Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and its associated incidence matrix with arbitrary orientation $E(\mathcal{G})$. The* directed local graph *at node v_j is the sub-graph $\mathcal{G}_j = (\mathcal{V}, \mathcal{E}_j)$ induced by node v_j such that*

$$\mathcal{E}_j = \{(v_j, v_i) | e_k = \{v_i, v_j\} \in \mathcal{E}\}$$

The local incidence matrix *at node v_j is the matrix*

$$E_l(\mathcal{G}_j) = E(\mathcal{G})\mathbf{diag}\{s_1, \ldots, s_m\} \in \mathbb{R}^{n \times m}$$

where $s_k = 1$ if $e_k \in \mathcal{E}_j$ and $s_k = 0$ otherwise.

Note, therefore, that the local incidence matrix will contain columns of all zeros in correspondence to those edges not adjacent to v_j. This also implicitly assumes a predetermined labeling of the edges.

Proposition 2.13 (Zelazo et al., 2012). *Let $p(\mathcal{V}) \in \mathbb{R}^{n \times 3}$ be the position matrix for the framework $(\mathcal{G}, p)$. The rigidity matrix $R(p)$ can be defined as*

$$R(p) = \begin{bmatrix} E_l(\mathcal{G}_1)^{\mathrm{T}} & \cdots & E_l(\mathcal{G}_n)^{\mathrm{T}} \end{bmatrix} (I_n \otimes p(\mathcal{V})), \qquad (5)$$

where $E_l(\mathcal{G}_i)$ is the local incidence matrix for node v_i.

A more detailed discussion and examples of these definitions are provided in Appendix B.

Lemma 2.6 and Corollary 2.11 relate the property of infinitesimal rigidity for a given (weighted) framework to the rank of a corresponding matrix. A contribution of this work is the translation of the rank condition to that of a condition on the spectrum of a corresponding matrix that we term the *symmetric rigidity matrix*. For the remainder of this work, we will only consider weighted frameworks, since from the discussion in Remark 2.8, any framework can be considered as a weighted framework with appropriately defined weights.

The symmetric rigidity matrix for a weighted framework $(\mathcal{G}, p, \mathcal{W})$ is a symmetric and positive-semidefinite matrix defined as

$$\mathcal{R} := R(p, \mathcal{W})^{\mathrm{T}} R(p, \mathcal{W}) \in \mathbb{R}^{3n \times 3n} \qquad (6)$$

An immediate consequence of the construction of the symmetric rigidity matrix is that $\mathbf{rk}[\mathcal{R}] = \mathbf{rk}[R(p, \mathcal{W})]$ (Horn and Johnson, 1985), leading to the following corollary.

Corollary 2.14. *A weighted framework $(\mathcal{G}, p, \mathcal{W})$ is infinitesimally rigid if and only if $\mathbf{rk}[\mathcal{R}] = 3n - 6$.*

The rank condition of Corollary 2.14 can be equivalently stated in terms of the eigenvalues of $\mathcal{R}$. Denoting the eigenvalues of $\mathcal{R}$ as $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{3n}$, note that infinitesimal rigidity is equivalent to $\lambda_i = 0$ for $i = 1, \ldots, 6$ and $\lambda_7 > 0$. Consequently, we term λ_7 the *rigidity eigenvalue*. We will now show that, in fact, for any connected graph,[2] the first six eigenvalues are always 0.

The first result in this direction shows that the symmetric rigidity matrix is similar to a weighted Laplacian matrix.

Proposition 2.15. *The symmetric rigidity matrix is similar to the weighted Laplacian matrix via a permutation of the rows and columns as*

$$P\mathcal{R}P^{\mathrm{T}} = (I_3 \otimes E(\mathcal{G})W)Q(p(\mathcal{V}))\left(I_3 \otimes WE(\mathcal{G})^{\mathrm{T}}\right) \qquad (7)$$

with

$$Q(p(\mathcal{V})) = \begin{bmatrix} Q_x^2 & Q_xQ_y & Q_xQ_z \\ Q_yQ_x & Q_y^2 & Q_yQ_z \\ Q_zQ_x & Q_zQ_y & Q_z^2 \end{bmatrix} \in \mathbb{R}^{3m \times 3m} \qquad (8)$$

where Q_x, Q_y, and Q_z are $m \times m$ diagonal weighting matrices for each edge in $\mathcal{G}$ such that for the edge $e_k = (v_i, v_j)$,

$$[Q_s]_{kk} = (p_i^s - p_j^s), \quad s \in \{x, y, z\}$$

and p_i^x (p_i^y,p_i^z) represents the x coordinate (y coordinate, z coordinate) of the position of agent i.

Proof. The proof is by direct construction using Proposition 2.13 and (6). Consider the permutation matrix P as

$$P = \begin{bmatrix} I_n \otimes \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ I_n \otimes \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \\ I_n \otimes \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \end{bmatrix} \qquad (9)$$

and let $\hat{E} = \begin{bmatrix} E_l(\mathcal{G}_1)^{\mathrm{T}} & \cdots & E_l(\mathcal{G}_n)^{\mathrm{T}} \end{bmatrix}$. It is straightforward to verify that

$$(I_n \otimes (p^x)^{\mathrm{T}})\hat{E}^{\mathrm{T}}W = E(\mathcal{G})W \underbrace{\begin{bmatrix} \ddots & & \\ & (p_i^x - p_j^x) & \\ & & \ddots \end{bmatrix}}_{\text{diagonal matrix of size } m \times m}$$

where p^x represents the first column of the position vector. The structure of the matrix in (7) then follows directly.[3]

The representation of the symmetric rigidity matrix as a weighted Laplacian allows for a more transparent understanding of certain eigenvalues related to this matrix. The next result shows that the first six eigenvalues of $\mathcal{R}$ must equal zero for any connected graph $\mathcal{G}$.

Theorem 2.16. *Assume that a weighted framework $(\mathcal{G}, p, \mathcal{W})$ has weights such that the weight matrix $W(\mathcal{G}, p)$ is invertible and the underlying graph $\mathcal{G}$ is connected. Then the symmetric rigidity matrix has at least six eigenvalues at the origin; that is, $\lambda_i = 0$ for $i \in \{1, \ldots, 6\}$. Furthermore, a possible set of linearly independent eigenvectors associated with each 0 eigenvalue is*

$$\left\{ P^{\mathrm{T}} \begin{bmatrix} \mathbb{1}_n \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, P^{\mathrm{T}} \begin{bmatrix} \mathbf{0} \\ \mathbb{1}_n \\ \mathbf{0} \end{bmatrix}, P^{\mathrm{T}} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbb{1}_n \end{bmatrix}, P^{\mathrm{T}} \begin{bmatrix} (p^y) \\ -(p^x) \\ \mathbf{0} \end{bmatrix}, P^{\mathrm{T}} \begin{bmatrix} (p^z) \\ \mathbf{0} \\ -(p^x) \end{bmatrix}, P^{\mathrm{T}} \begin{bmatrix} \mathbf{0} \\ (p^z) \\ -(p^y) \end{bmatrix} \right\},$$

where P is defined in (9).

Proof. Recall that for any connected graph, one has $E(\mathcal{G})^{\mathrm{T}}\mathbb{1}_n = 0$ (Godsil and Royle, 2001). Therefore, $P\mathcal{R}P^{\mathrm{T}}$ must have three eigenvalues at the origin, with eigenvectors $u_1 = \begin{bmatrix} \mathbb{1}_n^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, $u_2 = \begin{bmatrix} \mathbf{0}^{\mathrm{T}} & \mathbb{1}_n^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, and $u_3 = \begin{bmatrix} \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & \mathbb{1}_n^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. We now demonstrate that the remaining three eigenvectors proposed in the theorem are indeed in the null-space of the symmetric rigidity matrix.

Let $u_4 = \begin{bmatrix} (p^y)^{\mathrm{T}} & -(p^x)^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Observe that $(I_3 \otimes WE(\mathcal{G})^{\mathrm{T}})u_4 = \begin{bmatrix} b_1^{\mathrm{T}} & b_2^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ is such that b_1 is $\pm[W]_{kk}(p_i^y - p_j^y)$ only for edges $e_k = \{v_i, v_j\} \in \mathcal{E}$, and 0 otherwise. Similarly, b_2 is $\pm[W]_{kk}(p_j^x - p_i^x)$ only for edges $e_k = \{v_i, v_j\} \in \mathcal{E}$. The invertibility assumption of the weight matrix also guarantees that $[W]_{kk} \neq 0$. It can now be verified that from this construction one has

$$\begin{bmatrix} Q_x^2 & Q_xQ_y & Q_xQ_z \\ Q_yQ_x & Q_y^2 & Q_yQ_z \\ Q_zQ_x & Q_zQ_y & Q_z^2 \end{bmatrix} (I_3 \otimes WE(\mathcal{G})^{\mathrm{T}})u_4 = 0.$$

The remaining two eigenvectors follow the same argument as above. It is also straightforward to verify that u_4, u_5, and u_6 are linearly independent of the first three eigenvectors.

Theorem 2.16 provides a precise characterization of the eigenvectors associated with the null-space of the symmetric rigidity matrix for an infinitesimally rigid framework.

Remark 2.17. *It is important to note that the chosen eigenvectors associated with the null-space of the symmetric rigidity matrix are expressed in terms of the* absolute positions *of the nodes in the framework. We note that these eigenvectors can also be expressed in terms of the* relative position *of each node to any arbitrary reference point* $p_c = [p_c^x \quad p_c^y \quad p_c^z]^{\mathrm{T}} \in \mathbb{R}^3$. *For example, vector* u_4 *could be replaced by*

$$u_4^{p_c} = P^{\mathrm{T}} \begin{bmatrix} p^y - p_c^y\mathbb{1}_n \\ p_c^x\mathbb{1}_n - p^x \\ \mathbf{0} \end{bmatrix}$$

that is a linear combination of the null-space eigenvectors u_1,u_2 *and* u_4. *The use of eigenvectors defined on relative positions, in fact, will be necessary for the implementation of a distributed estimator for the rigidity eigenvector and eigenvalue based on only* relative measurements *available from onboard sensing.*

Theorem 2.16 can be used to arrive at the main result relating infinitesimal rigidity to the rigidity eigenvalue.

Theorem 2.18. *A weighted framework* $(\mathcal{G}, p, \mathcal{W})$ *is infinitesimally rigid if and only if the rigidity eigenvalue is strictly positive, i.e.* $\lambda_7 > 0$.

Proof. The proof is a direct consequence of Corollary 2.14 and Theorem 2.16.

Another useful observation relates infinitesimal rigidity of a framework to connectedness of the underlying graph.

Corollary 2.19. *Rigidity of the weighted framework* $(\mathcal{G}, p, \mathcal{W})$ *implies connectedness of the graph* $\mathcal{G}$.

The connection between infinitesimal rigidity of a framework and the spectral properties of the symmetric rigidity matrix inherits many similarities between the well studied relationship between graph connectivity and the graph Laplacian matrix (Mesbahi and Egerstedt, 2010).

In the next section, we exploit this similarity and propose a *rigidity maintenance* control law that aims to ensure the rigidity eigenvalue is always positive. Such a control action will be shown to depend on the rigidity eigenvalue, on its eigenvector, and on relative positions among neighboring pairs expressed in a common frame. The issue of how every agent in the group can distributedly estimate these quantities will be addressed in Sections 4 and 5.

3. A decentralized control strategy for rigidity maintenance

The results of Section 2 highlight the role of the rigidity eigenvalue λ_7 as a measure of the "degree of infinitesimal rigidity" of a weighted framework $(\mathcal{G}, p, \mathcal{W})$. It provides a linear algebraic condition to test the infinitesimal rigidity of a framework and, especially in the case of weighted frameworks, provides a means of quantifying "how rigid" a weighted framework is. Moreover, the symmetric rigidity matrix was shown to have a structure reminiscent of a weighted graph Laplacian matrix, and thus can be considered as a naturally *distributed* operator.

The basic approach we consider for the maintenance of rigidity is to define a scalar potential function of the rigidity eigenvalue, $V_\lambda(\lambda_7) > 0$, with the properties of growing unbounded as $\lambda_7 \to \lambda_7^{\min} > 0$ and vanishing (with vanishing derivative) as $\lambda_7 \to \infty$ (see Figure 2 for one possible shape or V_λ with $\lambda_7^{\min} = 5$). Here, $\lambda_7^{\min}$ represents some predetermined minimum allowable value for the rigidity eigenvalue determined by the needs of the application. In addition to maintaining rigidity, the potential function should also capture additional constraints in the system, such as collision avoidance or formation maintenance. Each agent should then follow the anti-gradient of this potential function, that is

$$\xi(u) = \dot{p}_u(t) = -\frac{\partial V_\lambda}{\partial p_u(t)} = -\frac{\partial V_\lambda}{\partial \lambda_7}\frac{\partial \lambda_7}{\partial p_u(t)} \qquad (10)$$

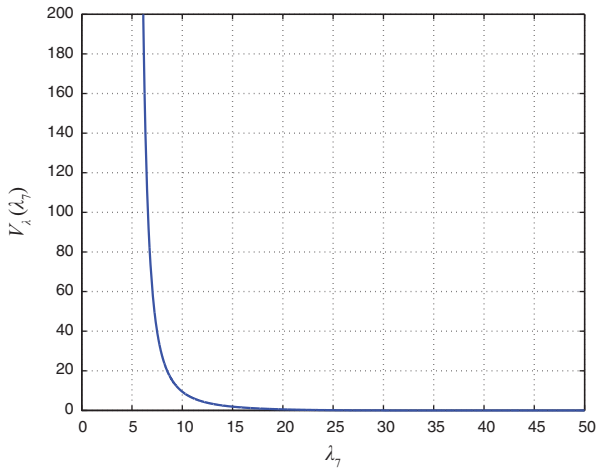


Fig. 2. A possible shape for the rigidity potential function $V_\lambda(\lambda_7)$ with $\lambda_7^{\min} = 5$.

where $\xi(u)$ is the velocity input of agent u, as defined in (3), and $p_u = [p_u^x \quad p_u^y \quad p_u^z]^{\mathrm{T}}$ is the position vector of the u-th agent. This strategy will ensure that the formation maintains a "minimum" level of rigidity (i.e. $\lambda_7^{\min}$) at all times. Of course, this strategy is an inherently *centralized* one, as the computation of the rigidity eigenvalue and of its gradient require full knowledge of the symmetric rigidity matrix. Nevertheless, we will proceed with this strategy and demonstrate that it can be implemented in a fully decentralized manner.

In the following, we examine in more detail the structure of the control scheme (10). First, we show how the formalization of weighted frameworks allows to embed additional weights within the rigidity property that enforce explicit inter-agent sensing and communication constraints and group requirements such as collision avoidance and formation control. For instance, the weighting machinery will be exploited so as to induce the agents to keep a desired inter-agent distance ℓ_0 and to ensure a minimum safety distance $\ell_{\min}$ from neighboring agents and obstacles. With these constraints, the controller will simultaneously maintain a minimum level of rigidity while also respecting the additional inter-agent constraints. We then provide an explicit characterization of the gradient of the rigidity eigenvalue with respect to the agent positions, and highlight its distributed structure. Finally, we present the general control architecture for implementing (10) in a fully decentralized way.

3.1. Embedding constraints in a weighted framework

In real-world applications a team of mobile robots may not be able to maintain the same interaction graph throughout the duration of a mission because of various sensing and communication constraints preventing mutual information exchange and relative sensing. Furthermore, additional requirements such as collision avoidance with obstacles and among robots, as well as some degree of formation control, must be typically satisfied during the mission execution. Building on the design guidelines proposed in Robuffo Giordano et al. (2013) for dealing with *connectivity* maintenance, we briefly discuss here a possible design of weights $\mathcal{W}$ aimed at taking into account the above-mentioned sensing and communication constraints and group requirements within the rigidity maintenance action.

To this end, we start with the following definition of *neighboring agents*:

Definition 3.1. *Two agents u and v are considered* neighbors *if and only if* (*i*) *their relative distance* $\ell_{uv} = \|p(u) - p(v)\|$ *is smaller than* $D \in \mathbb{R}^+$ (the sensing range), (*ii*) *the distance* ℓ_{uvo} *between the segment joining u and v and the closest obstacle point o is larger than* $\ell_{\min}$ (*the minimum* line-of-sight visibility), *and (iii) neither u nor v are closer than* $\ell_{\min}$ *to any other agent or obstacle*.

Conditions (i) and (ii) are meant to take into account two typical sensing constraints in multi-robot applications: maximum communication and sensing ranges and line-of-sight occlusions. The purpose of condition (iii), which will be better detailed later on, is to force disconnection from the group if an agent is colliding with any other agent or obstacle in the environment. In the following we will denote with $\mathcal{S}_u$ the set of neighbors of agent u induced by Definition 3.1.

This neighboring definition can be conveniently taken into account by designing the inter-agent weights $\mathcal{W}_{uv}$ as state-dependent functions smoothly vanishing as any of the above constraints and requirements are not met by the pair (u, v) with the desired accuracy. Indeed, the use of state-dependent weights allows us to consider the ensemble of robots in the context of weighted frameworks, as introduced in Definition 2.7. In particular, we take the underlying graph to be the complete graph K_n and the map p corresponds to the physical position state of each agent in a common global frame. The weights are the maps $\mathcal{W}_{uv}$, and the weighted framework is the triple $(K_n, p, \mathcal{W})$ with, therefore, $\mathcal{N}_u = \{v \in \mathcal{V} | \mathcal{W}_{uv} \neq 0\}$.

Following what was proposed in Robuffo Giordano et al. (2013), and recalling that ℓ_{uvo} represents the distance between the segment joining agents u and v and the closest obstacle point o, we then take

$$\mathcal{W}_{uv} = \alpha_{uv}\beta_{uv}\gamma_{uv}^a\gamma_{uv}^b \tag{11}$$

with $\alpha_{uv} = \alpha_{uv}(\ell_{uk}|_{k\in\mathcal{S}_u}, \ell_{vk}|_{k\in\mathcal{S}_v})$, $\beta_{uv} = \beta_{uv}(\ell_{uv})$, $\gamma_{uv}^a = \gamma_{uv}^a(\ell_{uv})$, $\gamma_{uv}^b = \gamma_{uv}^b(\ell_{uvo})$ and such that:

- we have
  - $\lim_{\ell_{uk}\to\ell_{\min}} \alpha_{uv} = 0$, for all $k \in \mathcal{S}_u$,
  - $\lim_{\ell_{uk}\to\ell_{\min}} \alpha_{uv} = 0$, for all $k \in \mathcal{S}_v$, and
  - $\alpha_{uv} \equiv 0$ if $\ell_{uk} \leq \ell_{\min}$ or $\ell_{vk} \leq \ell_{\min}$, for any $k \in \mathcal{S}_u$, $k \in \mathcal{S}_v$;
- $\lim_{|\ell_{uv}-\ell_0|\to\infty} \beta_{uv} = 0$ with $\beta(\ell_{uv}) < \beta(\ell_0)$ for all $\ell_{uv} \neq \ell_0$;

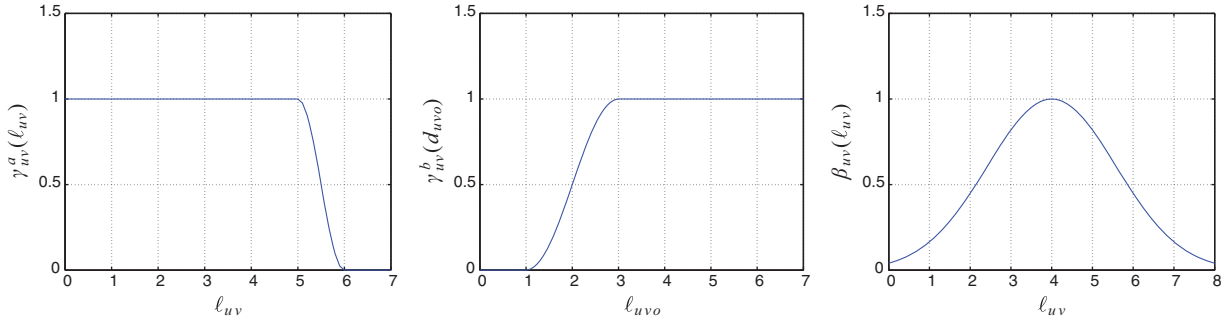


Fig. 3. The shape of $\gamma^a_{uv}(\ell_{uv})$ for $D = 6$ (a), $\gamma^b_{uv}(\ell_{uvo})$ for $\ell_{\min} = 1$ (b), and $\beta_{uv}(\ell_{uv})$ for $\ell_0 = 4$ (c).

- $\lim_{\ell_{uv}\to D} \gamma^a_{uv} = 0$ with $\gamma^a_{uv} \equiv 0$ for all $\ell_{uv} \geq D$;
- $\lim_{\ell_{uvo}\to \ell_{\min}} \gamma^b_{uv} = 0$ with $\gamma^b_{uv} \equiv 0$ for all $\ell_{uvo} \leq \ell_{\min}$.

As explained, $\ell_{\min}$ is a predetermined minimum safety distance for avoiding collisions and line-of-sight occlusions. Figures 3(a)–(c) show an illustrative shape of weights γ^a_{uv}, γ^b_{uv} and β_{uv}. The shape of the weights α_{uv} is conceptually equivalent to that of weights γ^b_{uv} in Figure 3(b).

This weight design results in the following properties: for a given pair of agents (u, v), the weight $\mathcal{W}_{uv}$ will vanish (because of the term $\gamma^a_{uv}\gamma^b_{uv}$) whenever the sensing and communication constraints of Definition 3.1 are violated (maximum range, obstacle occlusion), thus resulting in a decreased degree of connectivity of the graph $\mathcal{G}$ (edge $\{u, v\}$ is lost). The same will happen as the inter-distance ℓ_{uv} deviates too much from the desired ℓ_0 because of the term β_{uv}. Finally, the term α_{uv} will force *complete disconnection* of vertexes u and v from the other vertexes and therefore a complete loss of connectivity for the graph $\mathcal{G}$ whenever a collision with another agent is approached.[4]

We now recall from Corollary 2.19 that infinitesimal rigidity implies graph connectivity. Therefore, any decrease in the degree of graph connectivity due to the weights $\mathcal{W}_{uv}$ vanishing will also result in a decrease of rigidity of the weighted framework $(K_n, p, \mathcal{W})$ (in particular, rigidity is obviously lost for a disconnected graph). By maintaining $\lambda_7 > 0$ (in the context of weighted frameworks) over time, it is then possible to preserve formation rigidity while, at the same time, explicitly considering and managing the above-mentioned sensing and communication constraints and requirements.

Remark 3.2. *We note that the purpose of the weight β_{uv} in (11) is to embed a basic level of formation control into the rigidity maintenance action: indeed, every neighboring pair will try to keep the desired distance ℓ_0 thanks to the shape of the weights β_{uv}. More complex formation control behaviors could be obtained by different choices of functions β_{uv} (e.g. for maintaining given relative positions). Furthermore, formation shapes can be uniquely specified owing to the infinitesimal rigidity property of the configuration.*

Remark 3.3. *We further highlight the following properties whose explicit proof can be found in Robuffo Giordano et al. (2013): the chosen weights $\mathcal{W}_{uv}$ are functions of only relative distances to other agents and obstacles, while their gradients with respect to the agent position p_u (respectively p_v) are functions of* relative positions *expressed in a common reference frame. Furthermore, $\mathcal{W}_{uv} = \mathcal{W}_{vu}$ and $\frac{\partial \mathcal{W}_{uv}}{\partial p_u} = 0$, for all $v \notin \mathcal{N}_u$. Finally, the evaluation of weights $\mathcal{W}_{uv}$ and of their gradients can be performed in a decentralized way by agent u (respectively v) by only resorting to local information and* 1 *-hop communication.*

As shown in the next developments, these properties will be instrumental for expressing the gradient of the rigidity eigenvalue as a function of purely relative quantities with respect to only 1-hop neighbors.

3.2. The gradient of the rigidity eigenvalue

We now present an explicit characterization of the gradient of the rigidity eigenvalue with respect to the agent positions, as used in the control (10). We first recall that the rigidity eigenvalue can be expressed as

$$\lambda_7 = \mathbf{v}_7^{\mathrm{T}} \mathcal{R} \mathbf{v}_7$$

where $\mathbf{v}_7$ is the *normalized* rigidity eigenvector associated with λ_7. For notational convenience, we consider the permuted rigidity eigenvector $P\mathbf{v}_7 = \left[(\mathbf{v}^x)^{\mathrm{T}} \quad (\mathbf{v}^y)^{\mathrm{T}} \quad (\mathbf{v}^z)^{\mathrm{T}} \right]^{\mathrm{T}}$, where P is defined in Theorem 2.16. For the remainder of the work, we drop the subscript and reserve the bold font $\mathbf{v}$ for the rigidity eigenvector. Note that in fact, the rigidity eigenvalue and eigenvector are state-dependent, and therefore also time-varying when the formation is induced by the spatial orientation of a mobile team of robots, or due to the action of state-dependent weights on the sensing and communication links.

We can now exploit the structure of the symmetric rigidity matrix for weighted frameworks. Using the form of the symmetric rigidity matrix given in (7), we define $\tilde{Q}(p(\mathcal{V})) = (I_3 \otimes W) Q(p(\mathcal{V}))(I_3 \otimes W)$ as a generalized weight matrix, and observe that

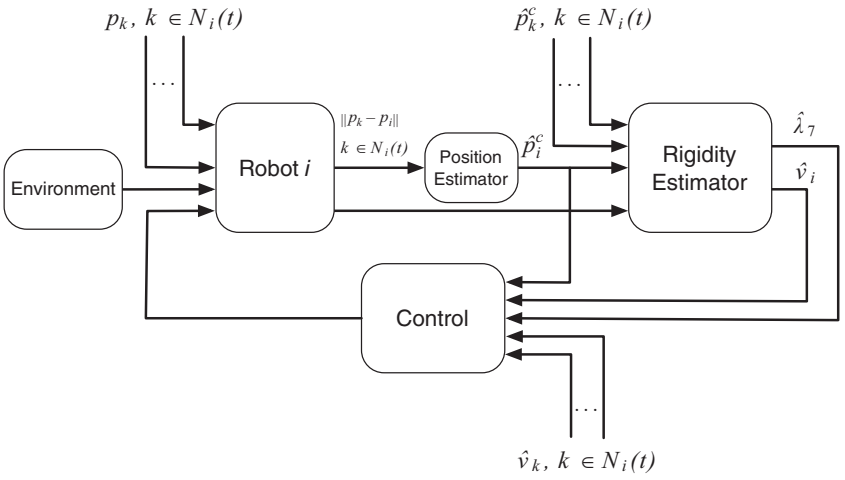


Fig. 4. Control architecture for distributed rigidity maintenance.

$$P\mathcal{R}P^{\mathrm{T}} = (I_3 \otimes E(\mathcal{G}))\tilde{Q}(p(\mathcal{V}))\left(I_3 \otimes E(\mathcal{G})^{\mathrm{T}}\right).$$

The elements of $\tilde{Q}(p(\mathcal{V}))$ are entirely in terms of the relative positions of each agent and the weighting functions defined on the edges as in (11).

The rigidity eigenvalue can now be expressed explicitly as

$$\begin{aligned}\lambda_7 = \sum_{(i,j)\in\mathcal{E}} \mathcal{W}_{ij}\Big((p_i^x - p_j^x)^2(\mathbf{v}_i^x - \mathbf{v}_j^x)^2 + (p_i^y - p_j^y)^2(\mathbf{v}_i^y - \mathbf{v}_j^y)^2 + \\ (p_i^z - p_j^z)^2(\mathbf{v}_i^z - \mathbf{v}_j^z)^2 + 2(p_i^x - p_j^x)(p_i^y - p_j^y)(\mathbf{v}_i^x - \mathbf{v}_j^x)(\mathbf{v}_i^y - \mathbf{v}_j^y) + \\ 2(p_i^x - p_j^x)(p_i^z - p_j^z)(\mathbf{v}_i^x - \mathbf{v}_j^x)(\mathbf{v}_i^z - \mathbf{v}_j^z) + \\ 2(p_i^y - p_j^y)(p_i^z - p_j^z)(\mathbf{v}_i^y - \mathbf{v}_j^y)(\mathbf{v}_i^z - \mathbf{v}_j^z)\Big) = \sum_{(i,j)\in\mathcal{E}} \mathcal{W}_{ij}S_{ij}.\end{aligned} \tag{12}$$

From (12), one can then derive a closed-form expression for $\frac{\partial\lambda_7}{\partial p_i^s}$, $s\in\{x, y, z\}$, i.e. the gradient of λ_7 with respect to each agent's position. In particular, by exploiting the structure of the terms S_{ij} and the properties of the employed weights $\mathcal{W}_{ij}$ (see, in particular, the previous Remark 3.3), it is possible to reduce $\frac{\partial\lambda_7}{\partial p_i^x}$ to the following *sum over the neighbors*,

$$\begin{aligned}\frac{\partial\lambda_7}{\partial p_i^x} = \sum_{j\in\mathcal{N}_i} \mathcal{W}_{ij}\Big(2(p_i^y - p_j^y)(\mathbf{v}_i^x - \mathbf{v}_j^x)(\mathbf{v}_i^y - \mathbf{v}_j^y) + \\ 2(p_i^x - p_j^x)(\mathbf{v}_i^x - \mathbf{v}_j^x)^2 + 2(p_i^z - p_j^z)(\mathbf{v}_i^x - \mathbf{v}_j^x)(\mathbf{v}_i^z - \mathbf{v}_j^z)\Big) \\ + \frac{\partial\mathcal{W}_{ij}}{\partial p_i^x}S_{ij}\end{aligned} \tag{13}$$

and similarly for the y and z components.

The gradient (13) possesses the following key feature: it is a function of relative quantities, in particular of (i) relative components of the eigenvector $\mathbf{v}$, (ii) relative distances, and (iii) relative positions with respect to *neighboring agents* (see, again, Remark 3.3 for what concerns weights $\mathcal{W}_{ij}$), thus allowing for a distributed computation of its value once these quantities are locally available. Sections 4 and 5 will detail two estimation schemes able to recover all of these relative quantities by resorting to only *measured distances* with respect to 1-hop neighbors owing to the infinitesimal rigidity of the group formation.

3.3. The control architecture

The explicit description of the gradient of the rigidity eigenvalue in (13) motivates the general control architecture for the implementation of the rigidity maintenance action in (10). We observe that each agent requires knowledge of the rigidity eigenvalue, appropriate components of the rigidity eigenvector, and relative positions with respect to neighboring agents in a common reference frame. As already mentioned, all of these quantities are inherently global quantities, and thus a fully distributed implementation of (10) must include appropriate estimators for recovering these parameters in a distributed manner.

As a preview of the next sections in this work, Figure 4 depicts the general architecture needed by each agent to implement the rigidity maintenance control action (10).

1. Exploiting measured distances with respect to its 1-hop neighbors, and owing to the formation rigidity, each agent distributedly estimates relative positions in a common reference frame, labeled as the *position estimator* in the figure. This block is fully explained in Section 4.
2. The output of the position estimator is then used by each agent to perform a distributed estimation of the rigidity eigenvalue ($\hat{\lambda}_7$) and of the relative components of the eigenvector ($\hat{\mathbf{v}}$), labeled as the *rigidity estimator* in the figure. This procedure is explained in Section 5.
3. Thanks to these estimated quantities (relative positions, $\hat{\lambda}_7$ and $\hat{\mathbf{v}}$), each agent can finally implement the control action (10) in a distributed way for maintaining infinitesimal rigidity of the formation during the group motion

(while also coping with the various constraints and requirements embedded into weights $\mathcal{W}$). Maintaining infinitesimal rigidity guarantees in turn convergence of the position estimator from measured distances of step 1), and thus closes the 'estimation-control loop.'

We finally note that the proposed control architecture also implicitly assumes the *initial* spatial configuration of the agents (i.e. their positions $p(\mathcal{V})$ at time 0) to be infinitesimally rigid (with, in particular, a $\lambda_7 > \lambda_7^{\min}$). This assumption on the group initial condition is formally stated below.

Assumption 3.4. *The initial spatial configuration of the agents, $p(\mathcal{V})$ at time $t = 0$, is infinitesimally rigid with $\lambda_7 > \lambda_7^{\min}$.*

The purpose of requiring a minimum level of rigidity ($\lambda_7^{\min}$) is discussed in greater detail in Section 7.

4. Decentralized estimation of positions in a common frame

As explained, evaluation of the gradient control (13) requires that each agent has access to the *relative positions* of its neighboring agents. A main focus of this work, however, is to achieve rigidity maintenance using only *relative distance* measurements. In this section, we leverage the infinitesimal rigidity of the formation to estimate the relative position with respect to a *common reference point*, p_c, shared by all agents. In particular, each agent i, with $i = 1 \ldots n$, will be able to compute an estimate $\hat{p}_{i,c}$ of its relative position $p_{i,c} = p_i - p_c$ to this common point. By exchanging their estimates over 1-hop communication channels, two neighboring agents i and j can then build an estimate $\hat{p}_{j,c} - \hat{p}_{i,c}$ of their actual relative position $p_j - p_j$ in a common reference frame. Note that both the graph (i.e. neighbor sets, edges, etc.) and the robot positions are time-varying quantities. However, in this section we omit dependency on time for the sake of conciseness.

We also note that this common reference point does not need to be stationary, i.e. it can move over time. In the following, we choose the point p_c to be attached to a *special agent* in the group, determined *a priori*. This agent will be denoted with the index i_c and, in the remainder of this section, we set $p_c = p_{i_c}$. We now proceed to describe a distributed scheme able to recover an estimation of the relative position $p_{i,c} = p_i - p_{i_c}$ for any agent in the group by exploiting the measured relative distances and the rigidity property of the formation.

To achieve this estimation, we first introduce additional assumptions on the capabilities of the special agent i_c. While all agents other than i_c are able to measure only the *relative distance* to their neighbors, the special agent i_c is required to be endowed with an additional sensor able to also measure, at any time t, the *relative position* (i.e. distance *and* bearing angles) of at least two non-collinear neighbors;[5] these two sensed neighbors will be denoted with the indexes $(\iota(t), \kappa(t)) \in \mathcal{N}_{i_c}(t)$.

Remark 4.1. *We stress that the agent indexes $\iota(t)$ and $\kappa(t)$ are* time-varying; *indeed, contrarily to the special agent i_c, $\iota(t)$ and $\kappa(t)$ are not preassigned to any particular agent in the multi-robot team. Therefore the special agent i_c only needs to measure its relative positions $p_{\iota(t)} - p_{i_c}$ and $p_{\kappa(t)} - p_{i_c}$ with respect to* any *two agents within its neighborhood (ι and κ are effectively arbitrary), with the points p_{i_c}, $p_{\iota(t)}$ and $p_{\kappa(t)}$ being non-collinear for all $t \geq t_0$. We believe this assumption is not too restrictive in practice, as it only require the presence of at least one robot equipped with a range plus bearing sensor while all the remaining ones can be equipped with simple range-only sensors.*

In the following we omit for brevity the dependency upon the time t of the quantities ι and κ.

In order to perform the distributed estimation of $p_{i,c} = p_i - p_c$, for all $i \in \{1, \ldots, n\}$, we follow the approach presented in Calafiore et al. (2010b), with some slight modifications dictated by the nature of our problem. Consistently with our notation, we define $\hat{p} = \left[\hat{p}_{1,c}^{\mathrm{T}} \ \ldots \ \hat{p}_{n,c}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{3n}$. For compactness, we also denote by ℓ_{ij} the measured distance $\|p_j - p_i\|$, as introduced in Definition 3.1. We then consider the following least-squares estimation error:

$$e(\hat{p}) = \frac{1}{4} \sum_{\{i,j\} \in \mathcal{E}} \left(||\hat{p}_{j,c} - \hat{p}_{i,c}||^2 - \ell_{ij}^2 \right)^2 + \frac{1}{2} ||\hat{p}_{i_c,c}||^2 + \frac{1}{2} ||\hat{p}_{\iota,c} - (p_\iota - p_{i_c})||^2 + \frac{1}{2} ||\hat{p}_{\kappa,c} - (p_\kappa - p_{i_c})||^2 \tag{14}$$

Note that the quantities $\ell_{ij}, p_\iota - p_{i_c}$, and $p_\kappa - p_{i_c}$ are measured while all of the other quantities represent local estimates of the robots.

The non-negative error function $e(\hat{p})$ is zero if and only if:

- $||\hat{p}_{j,c} - \hat{p}_{i,c}||$ is equal to the measured distance ℓ_{ij} for all the pairs $\{i, j\} \in \mathcal{E}$;
- $||\hat{p}_{i_c,c}|| = 0$;
- $\hat{p}_{\iota,c}$ and $\hat{p}_{\kappa,c}$ are equal to the measured relative positions $p_\iota - p_{i_c}$ and $p_\kappa - p_{i_c}$, respectively.

Note that the estimates $\hat{p}_{i_c,c}$, $\hat{p}_{\iota,c}$ and $\hat{p}_{\kappa,c}$ could be directly set to 0, $(p_\iota - p_{i_c})$, and $(p_\iota - p_{i_c})$, respectively, since the first quantity is known and the last two are measured. Nevertheless, we prefer to let the estimator obtaining these values via a 'filtering action' for the following reasons: first, the estimator provides a relatively simple way to filter out noise that might affect the relative position measurements; second, implementation of the rigidity maintenance controller only requires that $(\hat{p}_{j,c} - \hat{p}_{i,c}) \to (p_j - p_i)$, which is achieved if $\hat{p}_{j,c} \to p_j - \hat{p}_{i_c,c}$ and $\hat{p}_{i,c} \to p_i - \hat{p}_{i_c,c}$ for *any* common value of $\hat{p}_{i_c,c}$. Therefore, any additional hard constraint on $\hat{p}_{i_c,c}$ (e.g. $\hat{p}_{i_c,c} \equiv 0$) might unnecessarily over-constrain the estimator.

Applying a first-order gradient descent method to $e(\hat{p})$, we finally obtain the following *decentralized* update rule for the i-th agent ($i \neq i_c$):

$$\dot{\hat{p}}_{i,c} = -\frac{\partial e}{\partial \hat{p}_{i,c}} = \sum_{j \in \mathcal{N}_i} (||\hat{p}_{j,c} - \hat{p}_{i,c}||^2 - \ell_{ij}^2)(\hat{p}_{j,c} - \hat{p}_{i,c}) - \delta_{ii_c}\hat{p}_{i,c} - \delta_{i\iota}(\hat{p}_{\iota,c} - (p_\iota - p_{i_c})) - \delta_{i\kappa}(\hat{p}_{\kappa,c} - (p_\kappa - p_{i_c})), \quad (15)$$

where δ_{ij} is the well-known Kronecker's delta.[6] The estimator (15) is clearly decentralized since:

- ℓ_{ij} is locally measured by agent i;
- $\hat{p}_{i,c}$ is locally available to agent i;
- $\hat{p}_{j,c}$ can be transmitted using one-hop communication from agent j to agent i, for every $j \in \mathcal{N}_i$;
- $(p_\iota - p_{i_c})$ and $(p_\kappa - p_{i_c})$ are measured by agent i_c and can be transmitted using one-hop communication to agents ι and κ, respectively.

In order to show the relation between the proposed decentralized position estimator scheme and the infinitesimal rigidity property, one can restate (15) in matrix form as

$$\dot{\hat{p}} = -\mathcal{R}(\hat{p})\hat{p} + R(\hat{p})\ell + \Delta^c \quad (16)$$

where $\mathcal{R}(\hat{p})$ and $R(\hat{p})$ are the symmetric rigidity matrix and the rigidity matrix computed with the estimated positions, $\ell \in \mathbb{R}^{|\mathcal{E}|}$ is a vector whose entries are ℓ_{ij}^2, for all $\{i,j\} \in \mathcal{E}$, and $\Delta^c \in \mathbb{R}^{|\mathcal{E}|}$ contains the remaining terms of the right-hand side of (15).

Proposition 4.2. *If the framework is (infinitesimally) rigid, then the vector of true values* $p - (\mathbb{1}_n \otimes p_c) = \left[(p_1 - p_c)^{\mathrm{T}} \ \cdots \ (p_n - p_c)^{\mathrm{T}}\right]^{\mathrm{T}}$ *is an isolated local minimizer of* $e(\hat{p})$. *Therefore, there exists an* $\epsilon > 0$ *such that, for all initial conditions satisfying* $||\hat{p}(0) - p - (\mathbb{1}_n \otimes p_c)|| < \epsilon$, *the estimation* $\hat{p}$ *converges to* $p - (\mathbb{1}_n \otimes p_c)$.

We point out that the estimator in the form (16) is identical to the formation controller proposed in Krick et al. (2009). Consequently, we refer the reader to this work for a discussion on the stability and convergence properties of this model. A similar estimation scheme is also proposed in Calafiore et al. (2010b). We briefly emphasize that the property of having the true value of relative positions $p - (\mathbb{1}_n \otimes p_c)$ as an isolated local minimizer of (14) is a consequence of the definition of infinitesimal rigidity and of the non-collinearity assumption of the agents i_c, ι, and κ.

We finally note that, in general, the rate of convergence of a gradient descent method is known to be slower than other estimation methods. However, we opted for this method since is its directly amenable to a distributed implementation and requires only first-order derivative information.

5. Distributed estimation of the rigidity eigenvalue and eigenvector

As seen in Section 4, when the multi-robot team possesses the infinitesimal rigidity property, it is possible to distributedly estimate the relative positions in a common reference frame for each agent. However, the proposed distributed rigidity maintenance control action (10) requires knowledge of some additional global quantities that are explicitly expressed in the expressions (13) and (10). In particular, each agent must know also the current value of the rigidity eigenvalue and certain components of the rigidity eigenvector. In this section we propose a distributed estimation scheme inspired by the distributed connectivity maintenance solution proposed in Yang et al. (2010) for obtaining the rigidity eigenvalue and eigenvector.

For the reader's convenience, we first provide a brief summary of the *power iteration method* for estimating the eigenvalues and eigenvectors of a matrix. We then proceed to show how this estimation process can be distributed by employing *PI consensus filters* and by suitably exploiting the structure of the symmetric rigidity matrix.

5.1. Power iteration method

The power iteration method is one of a suite of iterative algorithms for estimating the dominant eigenvalue and eigenvector of a matrix. Following the same procedure as in Yang et al. (2010), we employ a continuous-time variation of the algorithm that will compute the smallest non-zero eigenvalue and eigenvector of the symmetric rigidity matrix.

The discrete-time power iteration algorithm is based on the following iteration,

$$x^{(k+1)} = \frac{Ax^{(k)}}{||Ax^{(k)}||} = \frac{A^k x^{(0)}}{||A^k x^{(0)}||}$$

Under certain assumptions for the matrix A (i.e. no repeated eigenvalues), the iteration converges to the eigenvector associated to the largest eigenvalue of the matrix.

To adapt the power iteration to compute the rigidity eigenvector and eigenvalue, we leverage the results of Theorem 2.16 and consider the iteration on a *deflated* version of the symmetric rigidity matrix, i.e. $\widetilde{\mathcal{R}} = I - TT^{\mathrm{T}} - \alpha\mathcal{R}$ for some small enough $\alpha > 0$. The power iteration method estimates the largest eigenvalue of a matrix. As all of the eigenvalues of the symmetric rigidity matrix are non-negative, the largest eigenvalue of the deflated version $\widetilde{\mathcal{R}}$ will correspond to $1 - \alpha\lambda_7$, and thus can be used to estimate λ_7. The constant α ensures the matrix $\widetilde{\mathcal{R}}$ is positive semi-definite. The columns of the matrix $T \in \mathbb{R}^{3n \times 6}$ contain the eigenvectors corresponding to the zero eigenvalues of $\mathcal{R}$, for example, as characterized in Theorem 2.16. Note that the power iteration applied to the matrix $\widetilde{\mathcal{R}}$ will compute the eigenvector associated with the rigidity eigenvalue.[7]

The continuous-time counterpart of the power iteration algorithm now takes the form (Yang et al., 2010)

$$\dot{\widehat{\mathbf{v}}}(t) = -\left(k_1 TT^{\mathrm{T}} + k_2\mathcal{R} + k_3\left(\frac{\widehat{\mathbf{v}}(t)^{\mathrm{T}}\widehat{\mathbf{v}}(t)}{3n} - 1\right)I\right)\widehat{\mathbf{v}}(t) \tag{17}$$

where $\widehat{\mathbf{v}}$ is the *estimate* of the rigidity eigenvector, and the constants $k_1, k_2, k_3 > 0$ are chosen to ensure the trajectories converge to the rigidity eigenvector.[8] We present here the main result and refer the reader to Yang et al. (2010) for details of the proof, noting that the proof methodologies are the same for the system (17) as that proposed in Yang et al. (2010).

Theorem 5.1. *Assume that the weighted framework* $(\mathcal{G}, p, \mathcal{W})$ *with symmetric rigidity matrix* $\mathcal{R}$ *is infinitesimally rigid and has distinct non-zero eigenvalues, and let* $\mathbf{v}$ *denote the rigidity eigenvector. Then for any initial condition* $\widehat{\mathbf{v}}(t_0) \in \mathbb{R}^{3n}$ *such that* $\mathbf{v}^{\mathrm{T}}\widehat{\mathbf{v}}(t_0) \neq 0$, *the trajectories of (17) converge to the subspace spanned by the rigidity eigenvector, i.e.* $\lim_{t\to\infty}\widehat{\mathbf{v}}(t) = \gamma\mathbf{v}$ *for* $\gamma \in \mathbb{R}$, *if and only if the gains* k_1,k_2 *and* k_3 *satisfy the following conditions:*

1. $k_1, k_2, k_3 > 0$;
2. $k_1 > k_2\lambda_7$;
3. $k_3 > k_2\lambda_7$.

Furthermore, for any choice of constants $k_1, k_2, k_3 > 0$, *the trajectories of (17) remain bounded and satisfy*

$$||\widehat{\mathbf{v}}(t)|| \leq \max\left\{||\widehat{\mathbf{v}}(t_0)||, \sqrt{3n}\right\}, \quad \text{for all } t \geq t_0$$

In particular, the trajectory converges to the rigidity eigenvector with

$$\lim_{t\to\infty}||\widehat{\mathbf{v}}(t)|| = \sqrt{3n\left(1 - \frac{k_2}{k_3}\right)\lambda_7}$$

Remark 5.2. *The power iteration proposed in (17) assumes that the symmetric rigidity matrix is static. However, in a dynamic setting the parameters of the rigidity matrix are a function of the state of the robots in a multi-robot system, and both the symmetric rigidity matrix and the expression of its null space are inherently* time-varying. *While the proof provided in Yang et al. (2010) does not explicitly address the time-varying case, our experience suggests that the dynamics of (17) is able to track even a time-varying rigidity eigenvector, so long as the dynamics of the robots are slower than the estimator. The speed of convergence of (17), of course, is also tunable by the constants* k_i.

Remark 5.3. *Another important subtlety of the dynamics (17) is the requirement that the rigidity eigenvalue is unique. When the rigidity eigenvalue is not unique, the associated eigenvector can belong to (at least) a two-dimensional subspace* $\mathcal{L}$, *so that (17) cannot be expected to converge to a unique eigenvector but rather to an equilibrium point in* $\mathcal{L}$ *(see, e.g., Yang et al., 2010). This can pose difficulties in real-world conditions since non-idealities such as noise in measuring the agent states (used in evaluating the symmetric rigidity matrix* $\mathcal{R}$ *), and discretization when numerically integrating (17), can make the equilibrium point for (17) in* $\mathcal{L}$ *to abruptly vary over time, thus preventing a successful convergence of the estimation of* $\mathbf{v}$.

5.2. A distributed implementation

The results of Section 5.1 provide a continuous-time estimator for estimating the rigidity eigenvalue and eigenvector of the symmetric rigidity matrix. The estimator given in (17), however, is a *centralized* implementation. Moreover, certain parameters used in (17) are expressed using a common reference frame (i.e. the quantity TT^{T}, see Theorem 2.16 and Remark 2.17) or require each robot to know the entire estimator state (i.e. the quantity $\widehat{\mathbf{v}}(t)^{\mathrm{T}}\widehat{\mathbf{v}}(t)$ in (17)). We propose in this section a distributed implementation for the rigidity estimator that overcomes these difficulties, in particular by leveraging the results of Section 4. In the same spirit as the solution proposed in Yang et al. (2010), we make use of the *PI average consensus filter* (Freeman et al., 2006) to distributedly compute the necessary quantities of interest, and strongly exploit the particular structure of the symmetric rigidity matrix.

Our approach to the distribution of (17) is to exploit both the built-in distributed structure (i.e. the symmetric rigidity matrix $\mathcal{R}$) and the reduction of the other parameters to values that all agents can obtain via a distributed algorithm. In this direction, we now proceed to analyze each term in (17) and discuss the appropriate strategies for implementing the estimator in a distributed fashion.

Concerning the first term $TT^{\mathrm{T}}\widehat{\mathbf{v}}$, Theorem 2.16 provides an analytic characterization of the eigenvectors associated with the zero eigenvalues of the symmetric rigidity matrix (assuming the graph is infinitesimally rigid). To begin the analysis, we explicitly write out the matrix T and examine the elements of the matrix TT^{T}. Following the comments of Remark 2.17, we express the null-space vectors in terms of *relative positions* to an arbitrary point $p_c = [p_c^x \;\; p_c^y \;\; p_c^z] \in \mathbb{R}^3$; in particular, the point p_c will be the special agent i_c described in Section 4:

$$T = \begin{bmatrix} \mathbb{1}_n & \mathbf{0} & \mathbf{0} & p^y - p_c^y\mathbb{1}_n & p^z - p_c^z\mathbb{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbb{1}_n & \mathbf{0} & p_c^x\mathbb{1}_n - p^x & \mathbf{0} & p^z - p_c^z\mathbb{1}_n \\ \mathbf{0} & \mathbf{0} & \mathbb{1}_n & \mathbf{0} & p_c^x\mathbb{1}_n - p^x & p_c^y\mathbb{1}_n - p^y \end{bmatrix}$$

For the remainder of this discussion, we assume that all agents have access to their state in an estimated coordinate frame relative to the point p_{i_c}, the details of which were described in Section 4:

$$TT^{\mathrm{T}} = \begin{bmatrix} \mathbb{1}_n\mathbb{1}_n^{\mathrm{T}} + p^{y,c}(p^{y,c})^{\mathrm{T}} + p^{z,c}(p^{z,c})^{\mathrm{T}} & -p^{y,c}(p^{x,c})^{\mathrm{T}} & -p^{z,c}(p^{x,c})^{\mathrm{T}} \\ -p^{x,c}(p^{y,c})^{\mathrm{T}} & \mathbb{1}_n\mathbb{1}_n^{\mathrm{T}} + p^{x,c}(p^{x,c})^{\mathrm{T}} + p^{z,c}(p^{z,c})^{\mathrm{T}} & -p^{z,c}(p^{y,c})^{\mathrm{T}} \\ -p^{x,c}(p^{z,c})^{\mathrm{T}} & -p^{y,c}(p^{z,c})^{\mathrm{T}} & \mathbb{1}_n\mathbb{1}_n^{\mathrm{T}} + p^{x,c}(p^{x,c})^{\mathrm{T}} + p^{y,c}(p^{y,c})^{\mathrm{T}} \end{bmatrix} \quad (18)$$

To simplify notation, we write as in Section 4, for example, $p^{y,c} = p^y - p_c^y\mathbb{1}_n$ and $p_{i,c} = p_i - p_c$. Following our earlier notation, we also partition the vector $\widehat{\mathbf{v}}$ into each coordinate, $\widehat{\mathbf{v}}^x$, $\widehat{\mathbf{v}}^y$, and $\widehat{\mathbf{v}}^z$. Let $\mathbf{avg}(r)$ denote the average value of the elements in the vector $r \in \mathbb{R}^n$, i.e. $\mathbf{avg}(r) = \frac{1}{n}\mathbb{1}_n^{\mathrm{T}} r$. Then it is straightforward to verify that

$$\mathbb{1}_n\mathbb{1}_n^{\mathrm{T}}\widehat{\mathbf{v}}^k(t) = n\,\mathbf{avg}(\widehat{\mathbf{v}}^k(t))\mathbb{1}_n, \quad k \in \{x,y,z\} \quad (19)$$

$$p_{i,c}(p_{j,c})^{\mathrm{T}}\widehat{\mathbf{v}}^k(t) = n\,\mathbf{avg}(p_{j,c} \circ \widehat{\mathbf{v}}^k)p_{i,c}, \quad i,j,k \in \{x,y,z\} \quad (20)$$

where '$\circ$' denotes the element-wise multiplication of two vectors.

This characterization highlights that, in order to evaluate the term $TT^{\mathrm{T}}\widehat{\mathbf{v}}$, each agent must compute the average amongst all agents of a certain value that is a function of the current state of the estimator and of the positions in some common reference frame whose origin is the point p_c. It is well known that the *consensus protocol* can be used to distributedly compute the average of a set of numbers (Mesbahi and Egerstedt, 2010). The speed at which the consensus protocol can compute this value is a function of the connectivity of the underlying graph and the weights used in the protocol. In this framework, however, a direct application of the consensus protocol will not be sufficient. Indeed, it is expected that each agent will be physically moving, leading to a time-varying description of the matrix TT^{T} (see Remark 5.2). In addition, the underlying network is also dynamic as sensing links between agents are inherently state dependent.

The use of a *dynamic* consensus protocol introduces additional tuning parameters that can be used to ensure that the distributed average calculation converges faster than the underlying dynamics of each agent in the system, as well as the ability to track the average of a time-varying signal. We employ the following *PI average consensus filter* proposed in Freeman et al. (2006),

$$\begin{bmatrix} \dot{z}(t) \\ \dot{w}(t) \end{bmatrix} = \begin{bmatrix} -\gamma I_n - K_P L(\mathcal{G}(t)) & K_I L(\mathcal{G}(t)) \\ -K_I L(\mathcal{G}(t)) & 0 \end{bmatrix} \begin{bmatrix} z(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} \gamma I_n \\ 0 \end{bmatrix} u(t) \quad (21)$$

$$y(t) = \begin{bmatrix} I_n & 0 \end{bmatrix} \begin{bmatrix} z(t) \\ w(t) \end{bmatrix} \quad (22)$$

The parameters $K_P, K_I \in \mathbb{R}$ and $\gamma \in \mathbb{R}$ are used to ensure stability and tune the speed of the filter. An analysis of the stability and performance of this scheme with time-varying graphs is given in Freeman et al. (2006). Figure 5 provides a block diagram representation of how the PI consensus filters are embedded into the calculation of $TT^{\mathrm{T}}\widehat{\mathbf{v}}(t)$ (in only the x coordinate).

As for the second term in (17), as shown in Section 2.3 the symmetric rigidity matrix is by construction a distributed operator. The term $\mathcal{R}\widehat{\mathbf{v}}(t)$ can be computed using only information exchanged between neighboring agents, as determined by the sensing graph.

The final term in (17) is a normalization used to drive the eigenvector estimate to the surface of a sphere of radius $\sqrt{3n}$. Using the same analysis as above, it can be verified that

$$\left(\frac{\widehat{\mathbf{v}}(t)^{\mathrm{T}}\widehat{\mathbf{v}}(t)}{3n} - 1\right)\widehat{\mathbf{v}}(t) = (\mathbf{avg}(\widehat{\mathbf{v}}(t) \circ \widehat{\mathbf{v}}(t)) - 1)\widehat{\mathbf{v}}(t) \quad (23)$$

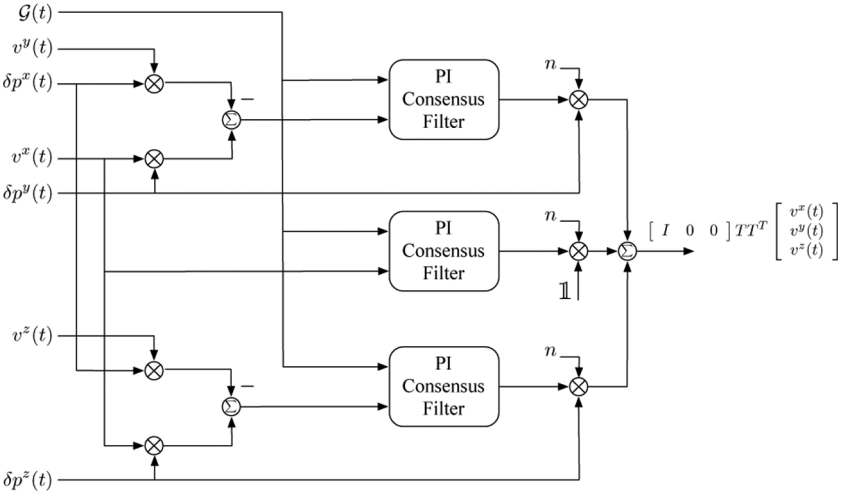


Fig. 5. Block diagram showing PI consensus filters in calculation of $TT^{\mathrm{T}}\widehat{\mathbf{v}}(t)$.

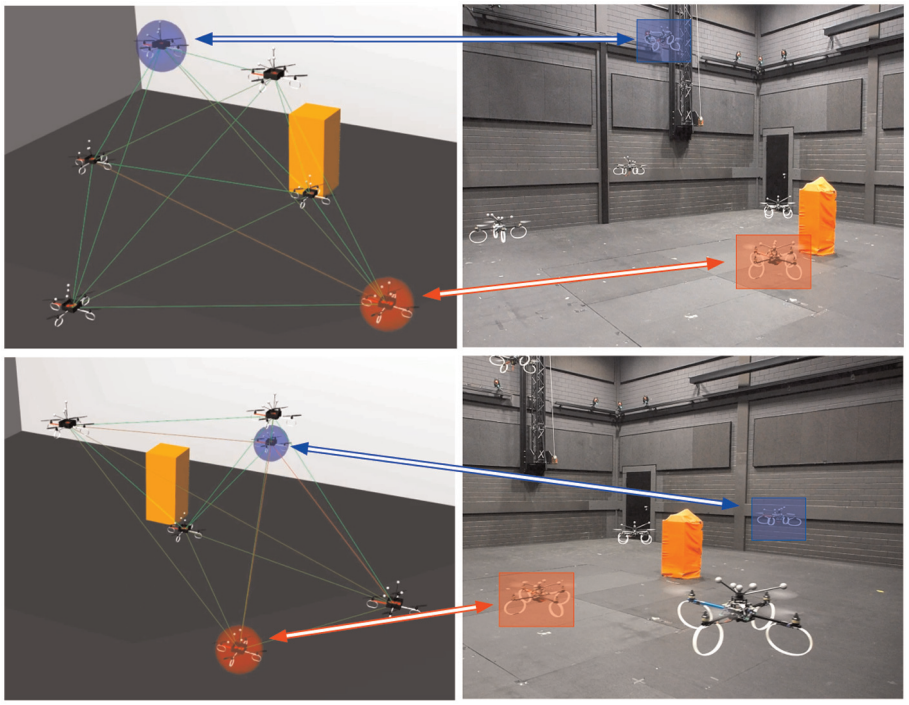


Fig. 6. Two snapshots of the reported experiment. Left: Simulated 3D views showing, in particular, the inter-agent links (red, almost disconnected link; green, optimally connected link). Right: Corresponding pictures of the experimental setup. The two highlighted quadrotor UAVs are partially controlled by two human operators.

This quantity can therefore be distributedly computed using an additional PI consensus filter.

Using the result of Theorem 5.1 and the PI consensus filters, each agent is also able to estimate the rigidity eigenvalue.

Corollary 5.4. *Let* $\overline{\mathbf{v}}_i^2(t)$ *denote the output of the PI consensus filter for estimating the quantity* $\mathbf{avg}(\widehat{\mathbf{v}}(t) \circ \widehat{\mathbf{v}}(t))$ *for agent* i*. Then agent* i*'s estimate of the rigidity eigenvalue,* $\hat{\lambda}_7^i$*, can be obtained as*

$$\hat{\lambda}_7^i = \frac{k_3}{k_2}\left(1 - \overline{\mathbf{v}}_i^2(t)\right).$$

In summary, each agent implements the following filters:

- estimation of a common reference frame using (15);
- estimation of the rigidity eigenvector using (17);
- a PI-consensus filter for tracking the average of the estimate of the rigidity eigenvector (19);
- a PI-consensus filter for tracking the quantity described in (20);
- a PI-consensus filter for tracking the average of the square of the rigidity eigenvector estimate (23).

For completeness, we now present the full set of filters that each robot executes:

$$\begin{aligned}\dot{\widehat{\mathbf{v}}}_i^x = &-k_1 n\left(\overline{\mathbf{v}}_i^x + z_i^{xy}(t)\hat{p}_{i,c}^y + z_i^{xz}\hat{p}_{i,c}^z(t)\right)\\ &- k_2 \sum_{j\in\mathcal{N}_i(t)} W_{ij}\left(\widehat{\mathbf{v}}_i^x(t) - \widehat{\mathbf{v}}_j^x\right) - k_3\left(\overline{\mathbf{v}}_i^x - 1\right)\widehat{\mathbf{v}}_i^x\end{aligned} \tag{24}$$

$$\begin{aligned}\dot{\hat{p}}_{i,c} = &\sum_{j\in\mathcal{N}_i(t)} (||\hat{p}_{j,c} - \hat{p}_{i,c}||^2 - \ell_{ij}^2)(\hat{p}_{j,c} - \hat{p}_{i,c})\\ &- \delta_{ii_c}\hat{p}_{i,c} - \delta_{i\iota}(\hat{p}_{\iota,c} - (p_\iota - p_{i_c}))\\ &- \delta_{i\kappa}(\hat{p}_{\kappa,c} - (p_\kappa - p_{i_c}))\end{aligned} \tag{25}$$

$$\dot{\widehat{\mathbf{v}}}_i^x = \gamma\left(\widehat{\mathbf{v}}_i^x - \overline{\mathbf{v}}_i^x\right) - K_P \sum_{j\in\mathcal{N}_i}\left(\overline{\mathbf{v}}_i^x - \overline{\mathbf{v}}_j^x(t)\right) + K_I \sum_{j\in\mathcal{N}_i(t)}\left(\bar{w}_i^x - \bar{w}_j^x\right) \tag{26}$$

$$\dot{\bar{w}}_i^x = -K_I \sum_{j\in\mathcal{N}_i(t)}\left(\overline{\mathbf{v}}_i^x - \overline{\mathbf{v}}_j^x\right) \tag{27}$$

$$\begin{aligned}\dot{\widehat{\mathbf{v}}}_i^{2x} = &\gamma\left((\widehat{\mathbf{v}}_i^x)^2 - \overline{\mathbf{v}}_i^{2x}\right) - K_P \sum_{j\in\mathcal{N}_i(t)}\left(\overline{\mathbf{v}}_i^{2x} - \overline{\mathbf{v}}_j^{2x}\right)\\ &+ K_I \sum_{j\in\mathcal{N}_i(t)}\left(\bar{w}_i^{2x} - \bar{w}_j^{2x}\right)\end{aligned} \tag{28}$$

$$\dot{\bar{w}}_i^{2x} = -K_I \sum_{j\in\mathcal{N}_i(t)}\left(\overline{\mathbf{v}}_i^{2x} - \overline{\mathbf{v}}_j^{2x}\right) \tag{29}$$

$$\begin{aligned}\dot{z}_i^{xy} = &\gamma\left((\hat{p}^y \circ \widehat{\mathbf{v}}^x - \hat{p}^x \circ \widehat{\mathbf{v}}^y) - z_i^{xy}\right) - K_P \sum_{j\in\mathcal{N}_i(t)}\left(z_i^{xy} - z_j^{xy}\right)\\ &+ K_I \sum_{j\in\mathcal{N}_i(t)}\left(w_i^{xy}(t) - w_j^{xy}\right)\end{aligned} \tag{30}$$

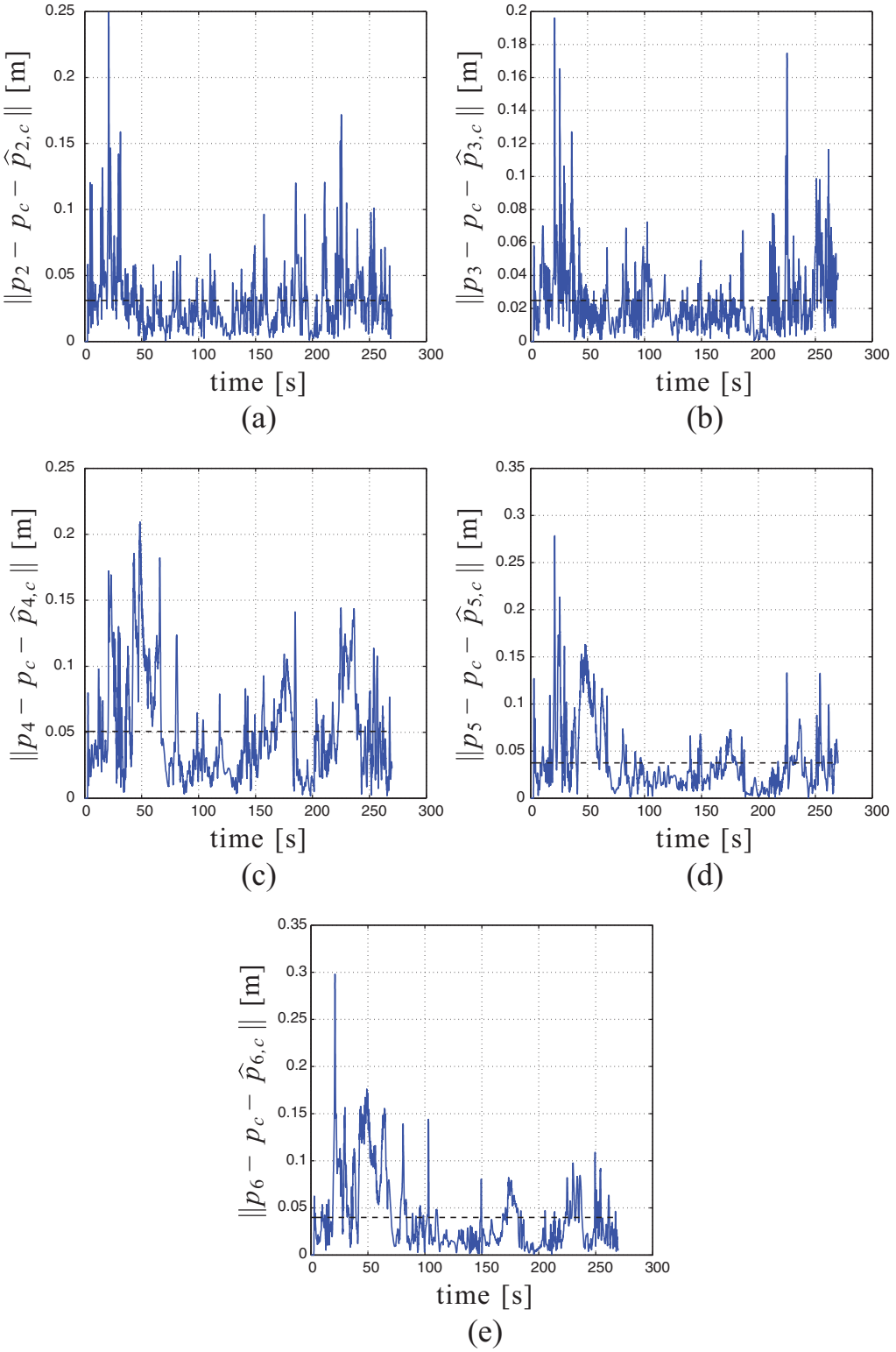


Fig. 7. Behavior of $||p_i - p_c - \hat{p}_{i,c}||$, $i = 2\ldots6$, the norm of the estimation error for the relative positions of agents 2…6 with respect to agent $i_c = 1$. The horizontal dashed black line represents the mean value of each error norm over time. Note how the estimation errors keep a low value during the group motion and thus indicate the ability of each robot to recover its relative position with respect to the robot $i_c = 1$ by only exploiting measured distances with respect to its neighbors and the infinitesimal rigidity of the formation.

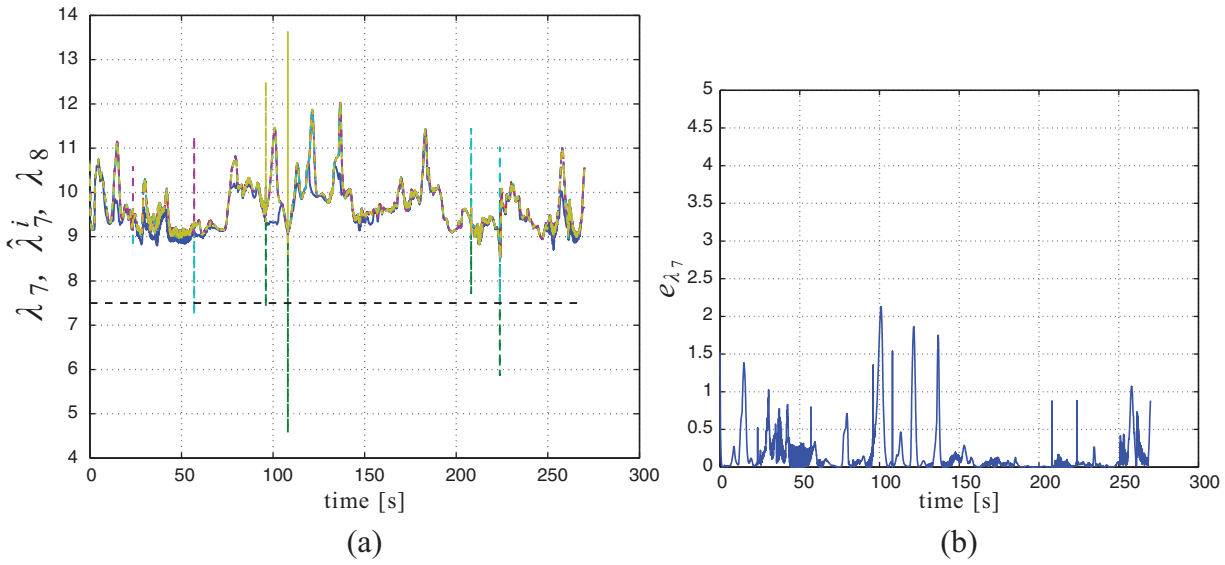


Fig. 8. (a) Behavior of $\lambda_7(t)$ (blue line) and the six estimations $\hat{\lambda}_7^i(t)$ (dashed colored lines) which result almost coincident. (b) Behavior of the overall rigidity eigenvalue estimation error $e_\lambda(t)$ as defined in (35).

$$\dot{w}_i^{xy} = -K_I \sum_{j\in\mathcal{N}_i(t)} \left(z_i^{xy} - z_j^{xy}\right) \tag{31}$$

$$\dot{z}_i^{xz} = \gamma\left((\hat{p}^z \circ \hat{\mathbf{v}}^x - \hat{p}^x \circ \hat{\mathbf{v}}^z) - z_i^{xz}\right) - K_P \sum_{j\in\mathcal{N}_i(t)} \left(z_i^{xy} - z_j^{xy}\right) + K_I \sum_{j\in\mathcal{N}_i(t)} \left(w_i^{xy} - w_j^{xy}\right) \tag{32}$$

$$\dot{w}_i^{xz} = -K_I \sum_{j\in\mathcal{N}_i(t)} \left(z_i^{xz} - z_j^{xz}\right). \tag{33}$$

These equations are written only for the x coordinate associated with all of the quantities. Observe, however, that the filters needed for the y and z coordinates do not require additional integrators, as similar filters can be vectorized (for example, the PI filters can be combined as in (21)). For the readers convenience, a summary of the notation and variable definitions used in (24)–(33) is provided in Table 1.

Remark 5.5. *Equations (24)–(33) show that each agent requires a 10th-order dynamic estimator for estimating the rigidity eigenvector and eigenvalue. This filter is composed of three PI-consensus filters, a relative position estimation filter, and the power iteration filter. An important point to emphasize is the order of the overall filter is independent of the number of agents in the ensemble, and thus is a scalable solution.*

6. The rigidity maintenance controller

The primary focus of this work until now was a detailed description of how the rigidity of a multi-robot formation can be maintained in a distributed fashion. The basic idea was to follow the gradient of an appropriately defined potential function of the rigidity eigenvalue; this control strategy was presented in (13). The fundamental challenge for the implementation of this control strategy was twofold: on the one hand, rigidity of a formation is an inherently *global* property of the network, and on the other hand, the control law depended on relative position measurements in a *common* reference fame.

A truly distributed solution based on this control strategy requires each agent to estimate a common inertial reference frame and also estimate the rigidity eigenvalue and eigenvector of the formation. The solution to these estimation problems was presented in Sections 4 and 5, with the complete set of filter equations summarized in (24)–(33). Note that both estimation strategies implicitly require that the underlying formation is infinitesimally rigid (see also Assumption 3.4). The final step for implementation of the rigidity maintenance controller is then to replace all of the state variables given in (13) with the appropriate estimated states computed by the relative position estimators and rigidity eigenvalue estimators. The local controller for each agent is thus given as,[9]

$$\begin{aligned}\xi_i^x = -\frac{\partial V(\hat{\lambda}_7^i)}{\partial \lambda_7} \sum_{j\in\mathcal{N}_i} W_{ij}\Big(&2(\hat{p}_{i,c}^x - \hat{p}_{j,c}^x)(\hat{\mathbf{v}}_i^x - \hat{\mathbf{v}}_j^x)^2 + \\ &2(\hat{p}_{i,c}^y - \hat{p}_{j,c}^y)(\hat{\mathbf{v}}_i^x - \hat{\mathbf{v}}_j^x)(\hat{\mathbf{v}}_i^y - \hat{\mathbf{v}}_j^y) \\ &+ 2(\hat{p}_{i,c}^z - \hat{p}_{j,c}^z)(\hat{\mathbf{v}}_i^x - \hat{\mathbf{v}}_j^x)(\hat{\mathbf{v}}_i^z - \hat{\mathbf{v}}_j^z)\Big) \\ &+ \frac{\partial W_{ij}}{\partial p_i^x}\hat{S}_{ij}\end{aligned} \tag{34}$$

in conjunction with all of the estimation filters of (24)–(33).

Remark 6.1. *The interconnection of the relative position estimator, rigidity eigenvalue estimator, and gradient controller leads to a highly non-linear dynamics for which a*

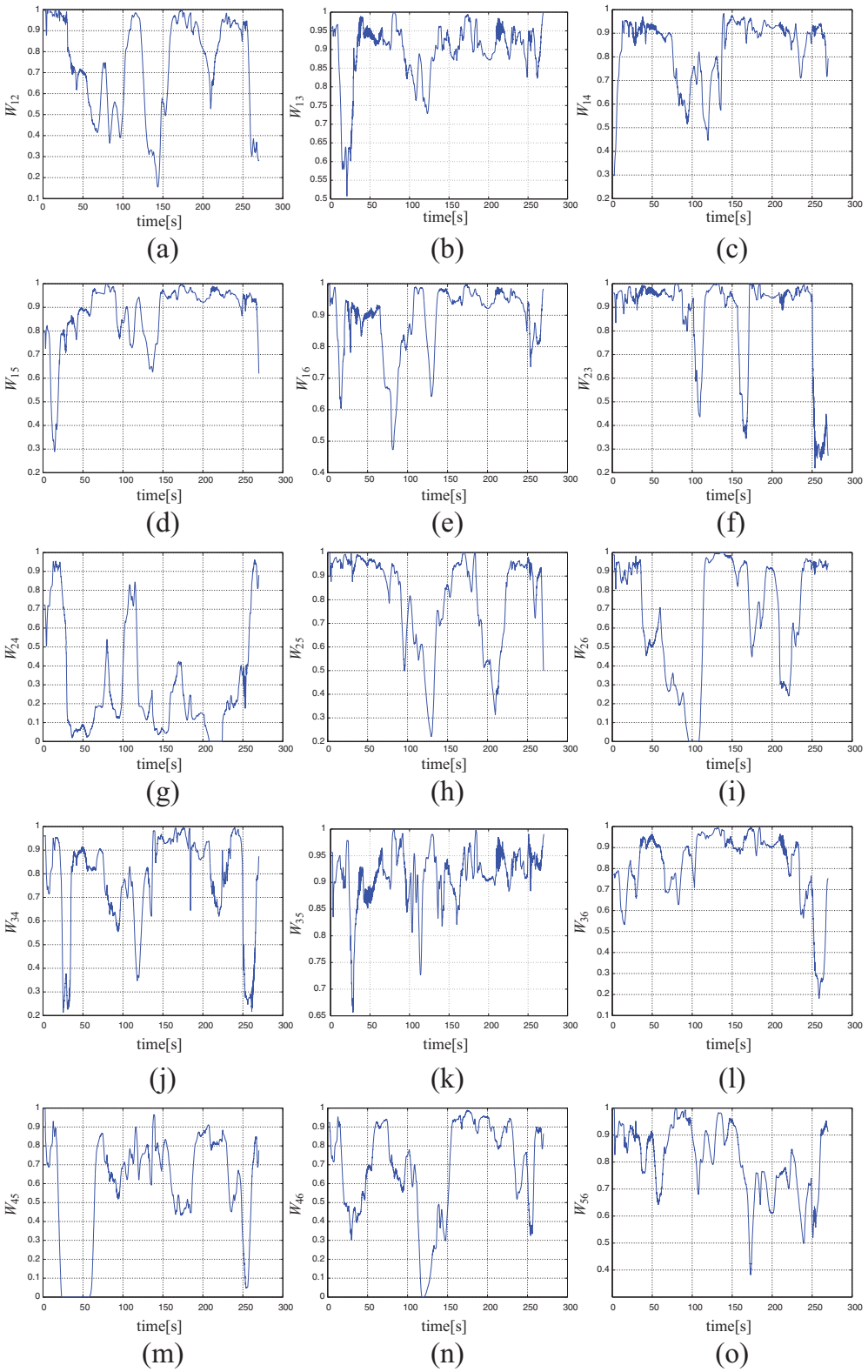


Fig. 9. Behavior of the 15 weights $W_{uv}(t)$ for all the possible edges of graph $\mathcal{G}$. Note how the values of weights $W_{uv}(t)$ vary over time because of the sensing/communication constraints and requirements embedded within their definition (see Section 3.1). Some weights (e.g. W_{24} and W_{45}) also temporarily vanish indicating loss of the corresponding edge (and, thus, the time-varying nature of graph $\mathcal{G}$).

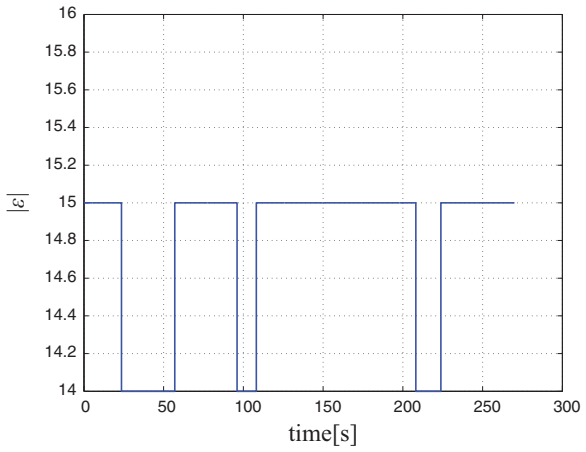


Fig. 10. Total number of edges in the graph $\mathcal{G}$ during the group motion.

formal proof analysis is not straightforward. While we are currently working towards a deeper analysis in this sense, the approach taken in this work is to exploit the typical (although informal) time-scale separation argument commonly found in many robotics applications relying on feedback control from an estimated state (as, e.g., when using an extended Kalman filter). Basically, the estimator dynamics is assumed "fast enough" such that its transient behavior can be considered as a second-order perturbation with respect to the robot motion (see also Yang et al., 2010) for an equivalent assumption in the context of decentralized connectivity maintenance control.

7. Experimental results

In this section we report some experimental results aimed at illustrating the machinery proposed so far for distributed rigidity maintenance. The experiments involved a total of $N = 6$ quadrotor UAVs (five real and one simulated) flying the environment shown in Figure 6. A video illustrating the various phases of the experiment (Extension 1) is attached to the paper.

All of the quadrotor UAVs were implementing the rigidity maintenance action (34) in addition to the estimation filters presented in (24)–(33). In addition, for two of the quadrotor UAVs (namely, quadrotors 1 and 2) an exogenous bounded velocity term $\xi_i^* \in \mathbb{R}^3$ was also added to (34); this allows for two human operators to independently control the motion of quadrotors 1 and 2 during the experiment, so as to steer the whole formation and trigger the various behaviors embedded in the weights $\mathcal{W}_{uv}$ (formation control, obstacle avoidance, sensing limitations).[10]

Our experimental quadrotor platform is a customized version of the MK-Quadro (see http://www.mikrokopter.de) implementing the TeleKyb ROS framework (see http://www.ros.org/wiki/telekyb) for flight control, experimental workflow management and human inputting. Attitude is stabilized with a fast inner loop that takes advantage of high-rate/onboard accelerometer and gyroscope measurements while the velocity stabilization is achieved by a slower control loop that measures the current velocity thanks to an external motion capture system. The motion capture system is also used to obtain relative distance measurements among the robots and the two bearing measurements needed by the special robot i_c. The reader is referred to Franchi et al. (2012b) for a detailed description of the quadrotor-based experimental setup.

We start illustrating the behavior of the relative position estimator described in Section 4 and upon which all of the subsequent steps are based (estimation of λ_7 and $\mathbf{v}$ and evaluation of the control action (10)). As explained in Section 4, owing to the formation infinitesimal rigidity, the scheme (15) allows each agent i to build an estimation $\hat{p}_{i,c}$ of its relative position $p_i - p_c$ with respect to the agent i_c, with $i_c = 1$ in this experiment. Figures 7(a)–(e) report the behavior of the norm of the estimation errors $||p_i - p_c - \hat{p}_{i,c}||$ for $i = 2 \ldots 6$ together with their mean values (dashed horizontal black line). It is then possible to verify how the relative position estimation errors keep low values over time, thus effectively allowing every agent to recover its correct relative position with respect to p_c from the measured relative distances.

As for the rigidity eigenvalue estimation of Section 5, Figure 8(a) reports the behavior of $\lambda_7(t)$ (solid blue line), of the six estimations $\hat{\lambda}_7^i(t)$ (solid colored lines almost superimposed to $\lambda_7(t)$), and of the minimum threshold $\lambda_7^{\min} = 7.5$ (horizontal dashed line). From the plot one can verify: (i) the accuracy in recovering the value of $\lambda_7(t)$ (note how the six estimations are almost superimposed on the real value) and (ii) that $\lambda_7(t) > \lambda_7^{\min}$ at all times apart from few isolated spikes, implying that *formation rigidity* was maintained during the task execution. As an additional indication of the eigenvalue estimation performance, Figure 8(b) shows the total estimation error for the rigidity eigenvalue

$$e_\lambda(t) = \frac{\sum_{i=1}^{N} \lambda_7(t) - \hat{\lambda}_7^i(t)|}{N} \tag{35}$$

which again confirms the accuracy of the estimation strategy.

Figures 9(a–o) report the behavior of the 15 weights W_{uv} defined in (11) and associated with all of the possible edges of graph $\mathcal{G}$ in order to show their *time-varying* nature because of the constraints and requirements listed in Section 3.1. Note how the value of some weight drops to zero over time (e.g. $W_{45}(t)$ at about $t = 25$ s or $W_{24}(t)$ at about $t = 210$ s), thus indicating loss of the corresponding edge. In the same spirit, Figure 10 shows the total number of edges $|\widehat{\mathcal{E}}|$ of the unweighted graph $\widehat{\mathcal{G}}$ (i.e. of non-zero weights W_{uv}, see Definition 2.9) during the group motion. These results highlight the time-varying nature of graph $\mathcal{G}$ which, as explained in the previous sections, is not constrained to keep a given fixed topology but is free to lose

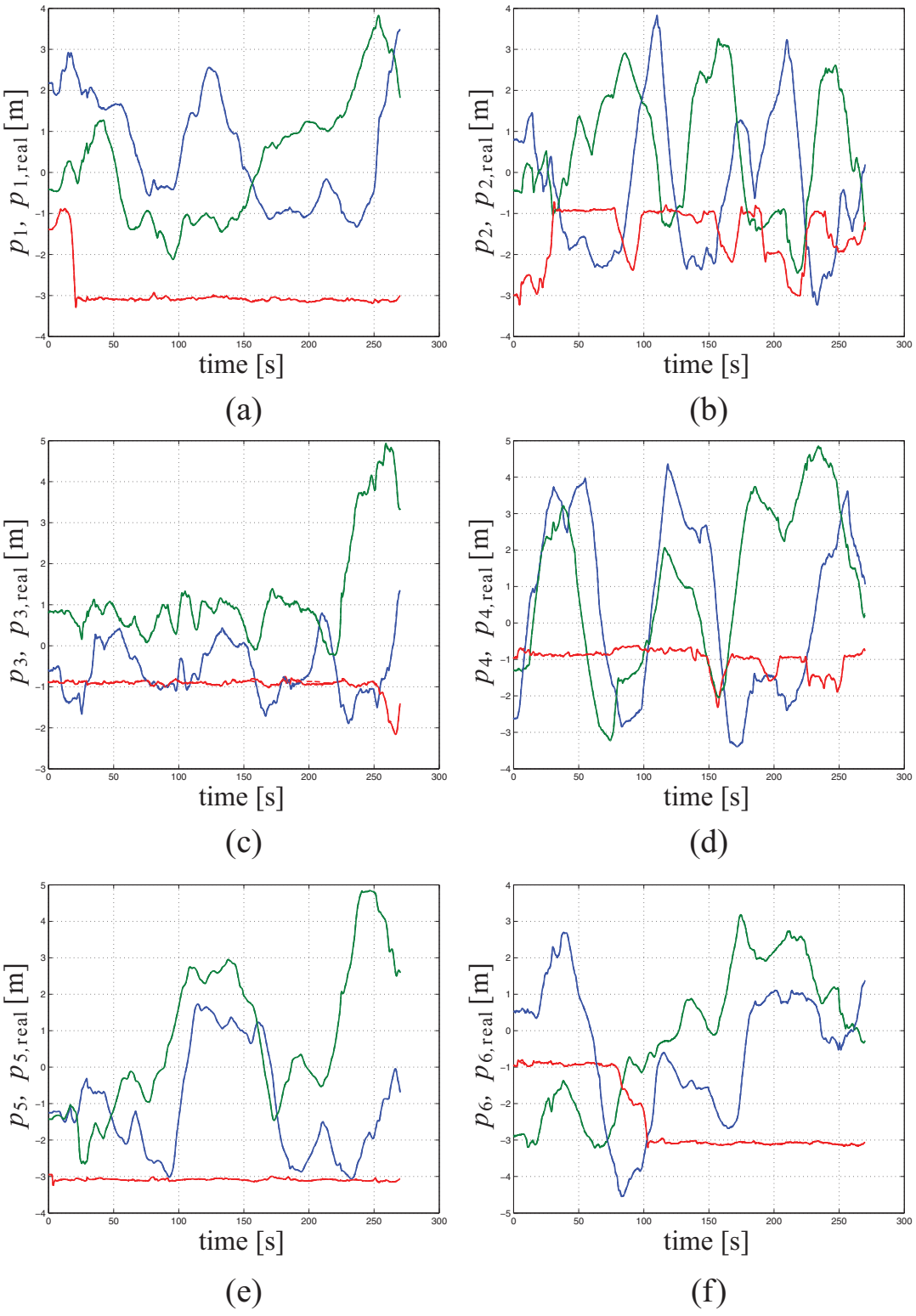


Fig. 11. Behavior of $p_i(t)$ (solid) and $p_{i,real}$ (dashed): these are basically superimposed, showing the accuracy of the quadrotors in tracking the reference trajectory $p_i(t)$. In the plots the following color code is used: blue/red/green solid/dashed lines correspond to the x / y / z components of $p_i(t)$ and $p_{i,real}$.

or gain edges as long as infinitesimal rigidity of the formation is preserved.

Finally, Figures 11(a)–(f) report the behavior over time of $p_i(t)$ (the ith agent position, solid lines) and of $p_{i,\text{real}}$ (t) (the i th quadrotor position, dashed lines) while tracking the motion of $p_i(t)$. The two position vectors result almost perfectly coincident, thus indicating a successful tracking performance of the quadrotors (and the soundness of our

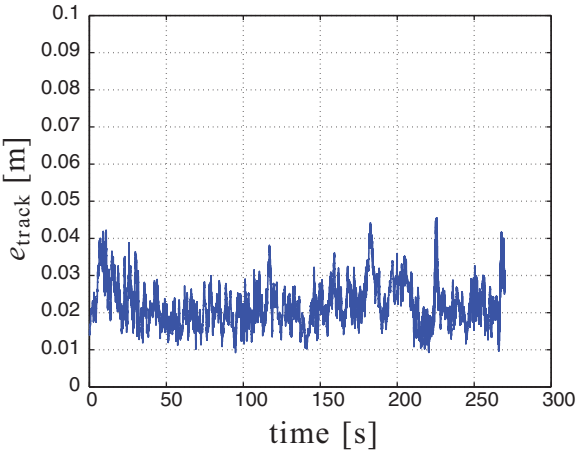


Fig. 12. Behavior of the tracking error e_{track} (t) defined in (36) showing again the good tracking performance of the six quadrotors.

modeling assumptions). As a further confirmation of this fact, the norm of the overall tracking error defined as

$$e_{\text{track}}(t) = \frac{\sum_{i=1}^{N} ||p_i(t) - p_{i,real}(t)||}{N} \tag{36}$$

is also reported in Figure 12.

8. Concluding remarks

This work presented a fully distributed solution for the rigidity maintenance control of a multi-robot system. As discussed in the introduction, rigidity is an important architectural feature for multi-robot systems that enables, for example, formation keeping and localization using only range-based measurements. The main theme of this work, therefore, was the distributed implementation of a number of algorithms for estimation and control in a multi-robot system related to rigidity maintenance. In particular, we demonstrated how the *rigidity eigenvalue* and eigenvector, used to decide whether a formation is infinitesimally rigid, can be distributedly estimated using a suite of estimators based on dynamic consensus filters and the power iteration method for eigenvalue estimation. The rigidity property also allowed for estimation of a common inertial reference frame using only *range-based measurements*, along with one single endowed agent that is able to sense both range and bearing. The estimation of these quantities were then embedded in a gradient-based distributed control action ensuring each agent moves in a way that guarantees rigidity of the formation is maintained. This control scheme also explicitly handles a variety of practical multi-robot constraints, including sensing and communication ranges, collision and obstacle avoidance, and line-of-sight requirements. The validity of the proposed algorithms was demonstrated by a team of six quadrotor UAVs flying in a cluttered environment.

This work also highlighted a number of directions for future research. In particular, the estimation of the rigidity eigenvalue assumed that there is a separation between the rigidity eigenvalue and the next largest eigenvalue, i.e. $|\lambda_7 - \lambda_8| > 0$. While the reported experimental results showed a large degree of robustness with respect to this effect, there remain both theoretical and practical questions related to this problem. For instance, it would be interesting to complement the rigidity maintenance controller with an additional term meant to maintain a minimum separation among λ_8 and λ_7. Another extension is to relax the requirement for having a special agent endowed with additional sensing capabilities (i.e. range and bearing). This would lead to a distributed solution involving only range measurements for all robots in the ensemble.

Despite these remaining challenges, this work has successfully demonstrated the power of distributed strategies for multi-robot systems. Indeed, it is remarkable to observe the behavior of the multi-robot team running many distributed filters to achieve a common global objective. The refinement of these strategies will no doubt become an important requirement as autonomous multi-robot systems are integrated more into a variety of application domains.

Funding

Part of Heinrich Bülthoff's research was supported by the Brain Korea 21 PLUS Program through the National Research Foundation of Korea funded by the Ministry of Education.

Notes

1. The second smallest eigenvalue of the graph Laplacian matrix.
2. If the graph is not connected, there will be additional eigenvalues at the origin corresponding to the number of connected components of the graph, see Godsil and Royle (2001).
3. A more detailed proof for the two-dimensional case is provided in Zelazo et al. (2012).
4. As for collision with obstacles, an equivalent behavior is automatically obtained from weights γ_{uv}^{b}, see again Robuffo Giordano et al. (2013) for a full explanation. Also note that, because of the definition of weights W_{uv}, one has $\mathcal{N}_u \subseteq \mathcal{S}_u$ but $\mathcal{S}_u \not\subset \mathcal{N}_u$.
5. Formation rigidity implies presence of at least two non-collinear neighbors for each agent (Laman, 1970).
6. $\delta_{ij} = 0$ if $I \neq j$ and $\delta_{ij} = 1$ otherwise.
7. Assuming the rigidity eigenvalue is unique and the framework is infinitesimally rigid (i.e. the rigidity eigenvalue is positive). We will discuss the implications of this assumption later.
8. Note that the constant α used to describe the deflated symmetric rigidity matrix is effectively replaced by k_2 in this formulation.
9. The control is shown in the x coordinate; a similar expression can be obtained for the y and z coordinates.
10. We note that, being ξ^*_i bounded, its effect does not threaten rigidity maintenance since the control action ξ_i in (10) always results dominant as $V_\lambda(\lambda_7) \to \infty$ if $\lambda_7(t) \to \lambda_7^{\min}$

11. Here, we assume that the directed edges (v_i,v_j) and (v_j,v_i) are equivalent to the undirected edge $\{v_i,v_j\}$.
12. This representation also assumes that all of the edges have been assigned a label, and this labeling is maintained even for the local graphs (local graphs do not relabel their edges; for example if edge 2 is not in local graph $\mathcal{G}_j$, then the second column of $E(\mathcal{G}_j)$ will be zero).

References

Akyildiz IF, Sankarasubramaniam Y and Cayirci E (2002) A survey on sensor networks. *IEEE Communications Magazine* 40(8): 102–114.

Anderson BDO, Fidan B, Yu C and van der Walle D (2008a) UAV formation control: Theory and application. In: Blondel VD, Boyd SP and Kimura H (eds.) *Recent Advances in Learning and Control* (Lecture Notes in Control and Information Sciences, vol. 371). New York: Springer, pp. 15–34.

Anderson BDO, Yu C, Fidan B and Hendrickx JM (2008b) Rigid graph control architectures for autonomous formations. *IEEE Control Systems Magazine* 28(6): 48–63.

Aspnes J, Eren T, Goldenberg DK, et al. (2006) A theory of network localization. *IEEE Transactions on Mobile Computing* 5(12): 1663–1678.

Baillieul J and McCoy L (2007) The combinatorial graph theory of structured formations. In: *2007 46th IEEE conference on decision and control*, pp. 3609–3615.

Bristow J, Folta D and Hartman K (2000) A formation flying technology vision. In: *AIAA space 2000 conference and exposition,* vol. 21, Long Beach, CA.

Calafiore GC, Carlone L and Wei M (2010a) A distributed Gauss–Newton approach for range-based localization of multi agent formations. In: *2010 IEEE international symposium on computer-aided control system design (CACSD)*, pp. 1152–1157.

Calafiore GC, Carlone L and Wei M (2010b) A distributed gradient method for localization of formations using relative range measurements. In: *2010 IEEE international symposium on computer-aided control system design*, Yokohama, Japan, pp. 1146–1151.

Connelly R and Whiteley WJ (2009) Global rigidity: the effect of coning. *Discrete Computational Geometry* 43(4): 717–735.

Eren T, Goldenberg OK, Whiteley W, et al. (2004) Rigidity, computation, and randomization in network localization. In: *IEEE INFOCOM* 2004, vol. 4. Piscataway, NJ: IEEE, pp. 2673–2684.

Franchi A, Masone C, Grabe V, Ryll M, Bülthoff HH and Robuffo Giordano P (2012a) Modeling and control of UAV bearing-formations with bilateral high-level steering. *The International Journal of Robotics Research* 31(12): 1504–1525.

Franchi A, Secchi C, Ryll M, Bülthoff HH and Robuffo Giordano P (2012b) Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs. *IEEE Robotics and Automation Magazine* 19(3): 57–68.

Freeman RA, Yang P and Lynch KM (2006) Stability and convergence properties of dynamic average consensus estimators. In: *45th IEEE conference on decision and control*, San Diego, CA, pp. 338–343.

Godsil CD and Royle G (2001) *Algebraic Graph Theory*. New York: Springer.

Graver J, Servatius B and Servatius H (1993) *Combinatorial Rigidity* (Graduate Studies in Mathematics). Providence, RI: American Mathematical Society.

Horn R and Johnson C (1985) *Matrix Analysis*. New York: Cambridge University Press.

Horn RA and Johnson CR (1991) *Topics in Matrix Analysis*. New York: Cambridge University Press.

Jackson B (2007) Notes on the rigidity of graphs. In: *Levico Conference Notes*.

Jacobs D (1997) An algorithm for two-dimensional rigidity percolation: the pebble game. *Journal of Computational Physics* 137(2): 346–365.

Ji M and Egerstedt M (2007) Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics* 23(4): 693–703.

Krick L, Broucke ME and Francis BA (2009) Stabilisation of infinitesimally rigid formations of multi-robot networks. *International Journal of Control* 82(3): 423–439.

Laman G (1970) On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics* 4(4): 331–340.

Lindsey Q, Mellinger D and Kumar V (2011) Construction of cubic structures with quadrotor teams. In: *2011 robotics: science and systems*, Los Angeles, CA.

Mesbahi M and Egerstedt M (2010) *Graph Theoretic Methods in Multiagent Networks*. (Princeton Series in Applied Mathematics), 1st edn. Princeton, NJ: Princeton University Press.

Michael N, Fink J and Kumar V (2009) Cooperative manipulation and transportation with aerial robots. In: *2009 robotics: science and systems*, Seattle, WA.

Murray RM (2006) Recent research in cooperative control of multi-vehicle systems. *ASME Journal on Dynamic Systems, Measurement, and Control* 129(5): 571–583.

Olfati-Saber R and Murray RM (2002) The combinatorial graph theory of structured formations. In: *41th IEEE conference on decision and control*, Las Vegas, NV, pp. 3609–3615.

Robuffo Giordano P, Franchi A, Secchi C and Bülthoff HH (2011) Bilateral teleoperation of groups of UAVs with decentralized connectivity maintenance. In: *2011 Robotics: Science and Systems*, Los Angeles, CA.

Robuffo Giordano P, Franchi A, Secchi C and Bülthoff HH (2013) A passivity-based decentralized strategy for generalized connectivity maintenance. *The International Journal of Robotics Research* 32(3): 299–323.

Scaramuzza D, Achtelik MC, Doitsidis L, et al. (2014) Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics and Automation Magazine* (in press).

Shames I, Fidan B and Anderson BDO (2009) Minimization of the effect of noisy measurements on localization of multi-agent autonomous formations. *Automatica* 45(4): 1058–1065.

Smith B, Egerstedt M and Howard A (2007) Automatic generation of persistent formations for multi-agent networks under range constraints. In: *1st international conference on robot communication and coordination*, pp. 1–8.

Tay T and Whiteley W (1985) Generating isostatic frameworks. *Structural Topology* 11(1): 21–69.

Williams RK, Gasparri A, Priolo A and Sukhatme GS (2014) Evaluating network rigidity in realistic systems: decentralization, asynchronicity, and parallelization. *IEEE Transactions on Robotics* (in press).

Wu C, Zhang Y, Sheng W and Kanchi S (2010) Rigidity guided localisation for mobile robotic sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing* 6(2): 114.

Yang P, Freeman RA, Gordon GJ, Lynch KM, Srinivasa SS and Sukthankar R (2010) Decentralized estimation and control of

graph connectivity for mobile sensor networks. *Automatica* 46(2): 390–396.

Zelazo D, Franchi A, Allgöwer F, Bülthoff HH and Robuffo Giordano P (2012) Rigidity maintenance control for multi-robot systems. In: *2012 Robotics: Science and Systems*, Sydney, Australia.

Appendix A: Index to Multimedia Extension

Archives of IJRR multimedia extensions published prior to 2014 can be found at http://www.ijrr.org, after 2014 all videos are available on the IJRR YouTube channel at http://www.youtube.com/user/ijrrmultimedia

Table of Multimedia Extension

Extension	Media type	Description
1	Video	Experiments of rigidity maintenance with a group of UAVs

Appendix B: Rigidity matrix example

The development of the alternative representation of the rigidity matrix given in Proposition 2.13 of the document is aided by a simple example. To begin, we make some qualitative observations of the rigidity matrix. For this example we consider a framework in $\mathbb{R}^2$ with the complete graph on three nodes (denoted K_3). The rigidity matrix can be written by inspection as

$$R(p) = \begin{bmatrix} p_1^x - p_2^x & p_1^y - p_2^y & p_2^x - p_1^x & p_2^y - p_1^y & 0 & 0 \\ p_1^x - p_3^x & p_1^y - p_3^y & 0 & 0 & p_3^x - p_1^x & p_3^y - p_1^y \\ 0 & 0 & p_2^x - p_3^x & p_2^y - p_3^y & p_3^x - p_2^x & p_3^y - p_2^y \end{bmatrix}$$

For the complete graph and an arbitrary orientation assigned to each edge, the incidence matrix $E(\mathcal{G})$ can be written as

$$E(\mathcal{G}) = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix}$$

The transpose of the incidence matrix functions as a "difference" operator. If the position of each agent is formed into a vector, we have

$$E(\mathcal{G})^{\mathrm{T}} \begin{bmatrix} p_1^x & p_1^y \\ p_2^x & p_2^y \\ p_3^x & p_3^y \end{bmatrix} = \begin{bmatrix} p_1^x - p_2^x & p_1^y - p_2^y \\ p_1^x - p_3^x & p_1^y - p_3^y \\ p_2^x - p_3^x & p_2^y - p_3^y \end{bmatrix}$$

The point to illustrate here is that this difference operation between positions is *redundantly embedded inside the rigidity matrix*. This fact can be made more precise by defining a *directed local graph at node* v_i from the graph $\mathcal{G}$ as in Definition 2.12 in the main text. Intuitively, the idea is that each node only has some local information about the connectivity of the entire graph; indeed, it only knows of the existence of other nodes that it can sense. In this way, we can define a sub-graph induced by each node in the graph as follows.

Let $\mathcal{G}_j = (\mathcal{V}, \mathcal{E}_j)$ be a sub-graph induced by node v_j such that

$$\mathcal{E}_j = \{(v_j, v_i) | \{v_i, v_j\} \in \mathcal{E}\}$$

Here we emphasize that the original graph $\mathcal{G}$ is undirected, while in the new induced graph $\mathcal{G}_i$ we assign a direction to the edge such that node v_j is always the tail. Furthermore, observe that $\cup_j \mathcal{G}_j = \mathcal{G}$.[11] This is illustrated in Figure 13.

To continue with the K_3 example, we can write the local incidence matrix for node v_1 as

$$E_l(\mathcal{G}_1) = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Note that this matrix is not truly an incidence matrix for the graph $\mathcal{G}_1$; "placeholders" for the other edges in the graph $\mathcal{G}$

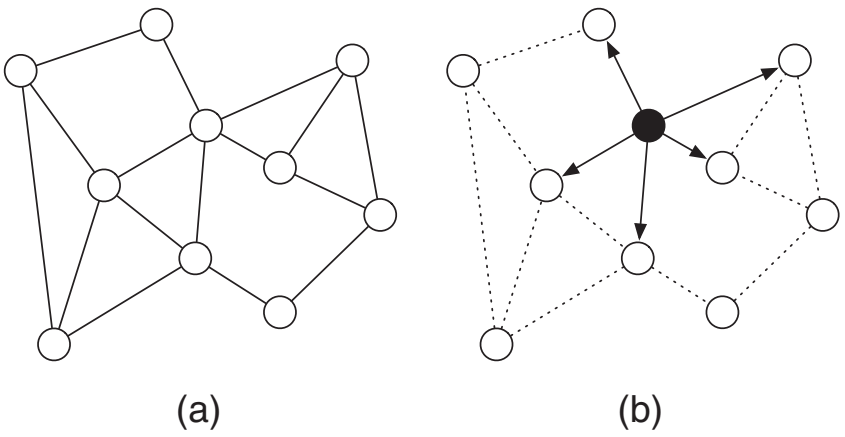


Fig. 13. Example of a directed local graph: (a) a graph; (b) local directed graph at a node.

are kept. As a result, the local incidence matrix is defined as $E_l(\mathcal{G}_j) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$ to have zero columns corresponding to the edges not in $\mathcal{E}_j$.[12]

Now, consider the local incidence matrix as the difference operator,

$$E_l(\mathcal{G}_1)^{\mathrm{T}} \begin{bmatrix} p_1^x & p_1^y \\ p_2^x & p_2^y \\ p_3^x & p_3^y \end{bmatrix} = \begin{bmatrix} p_1^x - p_2^x & p_1^y - p_2^y \\ p_1^x - p_3^x & p_1^y - p_3^y \\ 0 & 0 \end{bmatrix}$$

Note that this is identical to the first two columns of the rigidity matrix $R(p)$. In fact, this shows that the rigidity matrix can be written entirely in terms of local incidence matrices, as formally stated in Proposition 2.13.

Appendix B

Annex to Chapter VII

A Novel Overactuated Quadrotor Unmanned Aerial Vehicle: Modeling, Control, and Experimental Validation

Markus Ryll, Heinrich H. Bülthoff, *Member, IEEE*, and Paolo Robuffo Giordano, *Member, IEEE*

***Abstract*— Standard quadrotor unmanned aerial vehicles (UAVs) possess a limited mobility because of their inherent underactuation, that is, availability of four independent control inputs (the four propeller spinning velocities) versus the 6 degrees of freedom parameterizing the quadrotor position/orientation in space. Thus, the quadrotor pose cannot track arbitrary trajectories in space (e.g., it can hover on the spot only when horizontal). Because UAVs are more and more employed as service robots for interaction with the environment, this loss of mobility due to their underactuation can constitute a limiting factor. In this paper, we present a novel design for a quadrotor UAV with tilting propellers which is able to overcome these limitations. Indeed, the additional set of four control inputs actuating the propeller tilting angles is shown to yield full actuation to the quadrotor position/orientation in space, thus allowing it to behave as a fully actuated flying vehicle. We then develop a comprehensive modeling and control framework for the proposed quadrotor, and subsequently illustrate the hardware and software specifications of an experimental prototype. Finally, the results of several simulations and real experiments are reported to illustrate the capabilities of the proposed novel UAV design.**

***Index Terms*— Aerial robotics, aerodynamical modeling, dynamical modeling, flight control, overactuation, quadrotor unmanned aerial vehicles (UAVs), redundancy resolution.**

I. Introduction

COMMON unmanned aerial vehicles (UAVs) are underactuated mechanical systems, that is, systems possessing less control inputs than available degrees of freedom (DoF). This is, for instance, the case of helicopters and quadrotor UAVs for which only the Cartesian position and yaw angle with respect to an inertial frame can be independently controlled (4 DoF), whereas the behavior of the remaining roll and pitch angles (2 DoF) is completely determined by the trajectory chosen for the former 4 DoF (the so-called quadrotor flat outputs [1]–[3]).

Manuscript received August 19, 2013; revised March 5, 2014; accepted May 18, 2014. Date of publication July 30, 2014; date of current version February 11, 2015. Manuscript received in final form June 10, 2014. Recommended by Associate Editor L. Marconi.

M. Ryll is with Max Planck Institute for Biological Cybernetics, Tübingen 72076, Germany (e-mail: markus.ryll@tuebingen.mpg.de).

H. H. Bülthoff is with Max Planck Institute for Biological Cybernetics, Tübingen 72076, Germany, and also with the Department of Brain and Cognitive Engineering, Korea University, Seoul 136-713, Korea (e-mail: hhb@tuebingen.mpg.de).

P. Robuffo Giordano is with the CNRS, Irisa and Inria Rennes Bretagne Atlantique, Rennes Cedex 35042, France (e-mail: prg@irisa.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2014.2330999

Over the last decades, a variety of control techniques have been proposed to deal with the quadrotor underactuation to allow for an effective and robust flight performance (see [3]–[5] for an overview). The constant improvements during the miniaturization of microelectromechanical systems and sensors (MEMS) and during the computational power of microcontrollers have led to impressive achievements by employing quadrotor UAVs as robotics platforms: planning and control for aggressive flight maneuvers [6], collective control of multiple small- and micro-quadrotors [7], [8], and vision-based state estimation for autonomous flight [9] are just a few examples.

Nevertheless, the underactuated quadrotor design still limits its flying ability in free or cluttered space, and it also degrades the possibility of interacting with the environment by exerting desired forces in arbitrary directions. This is a particularly a limiting factor because quadrotor UAVs are being more and more envisaged and exploited as autonomous flying service robots, for example, as proven by the recently funded EU projects AIRobots [10] and ARCAS [11]. Indeed, several groups have been addressing the possibility to allow for an actual interaction with the environment, either in the form of direct contact [12]–[14] or by considering aerial grasping and manipulation tasks [15]–[19]. In this respect, as also recognized in [20] and [21], it is interesting to explore different actuation strategies that can overcome the aforementioned underactuation problem and allow for full motion and force control in all directions in space.

Motivated by these considerations, several solutions have been proposed in the past literature spanning different concepts as, for example, tilt-wing mechanisms [22], [23], UAVs with nonparallel (but fixed) thrust directions [24], or tilt-rotor actuations [25], [26]. In [27], the possibility of combining several modules of underactuated ducted-fan vehicles to achieve full 6-dof actuation for the assembled robot is theoretically explored, with a special focus on the optimal allocation of the available (redundant) control inputs. In contrast, Hua *et al.* [20] consider the possibility of a thrust-tilted quadrotor UAV in which the main thrust direction (2 DoF) can be actively regulated. A trajectory tracking control strategy is then proposed, which is able to explicitly consider a limited range of the thrust-tilting angles. Finally, in [21] a UAV made of two central coaxial counter-rotating propellers surrounded by three tilting thrusters has been presented along with some preliminary experimental results. The prototype is capable of two flight modalities: a fixed configuration in which it

Fig. 1. Picture of the holocopter prototype. The four propeller groups are slightly tilted. Red bar: positive direction of the $\boldsymbol{X}_B$ holocopter body axis.

essentially behaves as a standard underactuated UAV, and a variable angle configuration which guarantees some degree of full actuation as shown in the reported results.

A. Paper Contributions

Taking inspiration from these works, in this paper we focus on a novel actuation concept for a quadrotor UAV in which all the (usually fixed) propellers are allowed to tilt about the axes connecting them to the main body frame [28]–[30]. Indeed, as explained, one of the limitations of the classical quadrotor design depends on its inherent underactuation: presence of only four independent control inputs (the four propeller spinning velocities) does not allow to independently control the position and orientation of the quadrotor at the same time. For instance, in quasi-hover conditions, an horizontal translation necessarily implies a change in the attitude or, symmetrically, a quadrotor can hover in place only when being horizontal with respect to the ground plane. In contrast, in this paper we will show that, by means of the additional four actuated tilting DoF, it is possible to gain full controllability over the quadrotor position and orientation, thus transforming it, as a matter of fact, in a fully actuated flying rigid body.[1] Fig. 1 shows a picture of our current prototype of quadrotor with tilting propellers, which will be denoted as holocopter throughout the following developments.

The rest of this paper, as well as its main contributions, are then organized as follows.

1) A complete dynamical model of the holocopter is first derived in Section II by considering the dominant aerodynamic forces and torques (the propeller actuation), and by analyzing the effects of the main neglected terms.
2) A trajectory tracking controller is then presented in Section III aimed at exploiting the actuation capabilities of the holocopter for tracking arbitrary trajectories for its body position and orientation. As the holocopter is actually overactuated (eight control inputs for six controlled DoF), suitable strategies to exploit the actuation redundancy are also discussed; these are aimed at preserving full controllability of the holocopter pose and at minimizing the total energy consumption during flight.

[1]This, of course, without considering possible limitations of the actuation systems such as, for example, finite range for the tilting angles. Section IV-A will further discuss these points with respect to the experimental prototype considered in this paper.

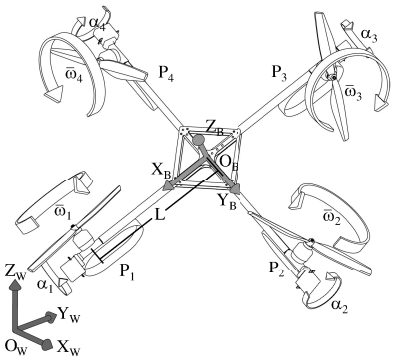


Fig. 2. Schematic view of the quadrotor considered in this paper. The overall center of mass is assumed to be in the body frame center. The symbol L represents the length of all propeller arms, ω_i, $i = 1, \ldots, 4$, the propeller rotation speed and α_i, $i = 1, \ldots, 4$, and the orientation of the propeller group.

3) A thorough description of the hardware/software architecture of the prototype shown in Fig. 1 is then given in Section IV, including the identification of its dynamical parameters and a discussion of the main nonidealities with respect to the dynamical model developed in Section II. In particular, a predictive scheme complementing the control action described in Section III is introduced to cope with the poor performance of the employed servo motors.
4) An extensive set of ideal/realistic simulations and experimental results on the holocopter prototype is then presented in Sections V–VII, showing the appropriateness of the various modeling assumptions and of the adopted control design. A video collecting several experimental flights is also attached to this paper.
5) Conclusions and some future discussions are then given in Section VII with a particular focus on the second-generation holocopter prototype currently under development.

II. Dynamical Modeling of the Holocopter

The quadrotor analyzed in this paper can be considered as a connection of 5 main rigid bodies in relative motion among themselves: the quadrotor body itself B and the 4 propeller groups P_i. These consist of the propeller arm hosting the motor responsible for the tilting actuation mechanism, and the propeller itself connected to the rotor of the motor responsible for the propeller spinning actuation[2] (see Figs. 1–3). The aim of this section is to derive the equations of motion of this multibody system.

A. Preliminary Definitions

Let $\mathcal{F}_W : \{\boldsymbol{O}_W;\ \boldsymbol{X}_W,\ \boldsymbol{Y}_W,\ \boldsymbol{Z}_W\}$ be a world inertial frame and $\mathcal{F}_B : \{\boldsymbol{O}_B;\ \boldsymbol{X}_B,\ \boldsymbol{Y}_B,\ \boldsymbol{Z}_B\}$ a moving frame attached to the quadrotor body at its center of mass (see Fig. 2). We also define $\mathcal{F}_{P_i} : \{\boldsymbol{O}_{P_i};\ \boldsymbol{X}_{P_i},\ \boldsymbol{Y}_{P_i},\ \boldsymbol{Z}_{P_i}\}$, $i = 1, \ldots, 4$, as the frames associated to the ith propeller group, with $\boldsymbol{X}_{P_i}$

[2]For simplicity, we are here considering each propeller groups P_i as a single-body approximation of both the propeller/rotor and its hoisting mechanism.

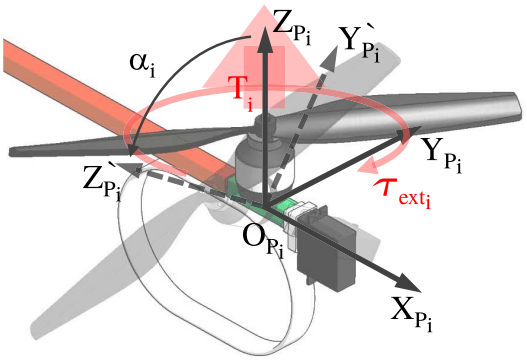


Fig. 3. ith tilting arm visualizing the body frame $\mathcal{F}_{P_i}$, the associated propeller thrust T_i, torque τ_{ext_i}, and the propeller tilt angle α_i.

representing the tilting actuation axis and $\boldsymbol{Z}_{P_i}$ the propeller-actuated spinning (thrust T_i) axis (see Fig. 3).

As usual, we let ${}^1\boldsymbol{R}_2 \in SO(3)$ be the rotation matrix representing the orientation of frame 2 with respect to frame 1; therefore, ${}^W\boldsymbol{R}_B$ will represent the orientation of the body frame with respect to the world frame, while ${}^B\boldsymbol{R}_{P_i}$ the orientation of the propeller group ith frame with respect to the body frame. By denoting with $\alpha_i \in \mathbb{R}$ the propeller tilt angle about axis $\boldsymbol{X}_{P_i}$, it follows from Fig. 2 that:[3]

$$ {}^B\boldsymbol{R}_{P_i} = \boldsymbol{R}_Z\left((i-1)\frac{\pi}{2}\right)\boldsymbol{R}_X(\alpha_i), \qquad i = 1,\ldots,4. $$

Similarly, we also let

$$ {}^B\boldsymbol{O}_{P_i} = \boldsymbol{R}_Z\left((i-1)\frac{\pi}{2}\right)\begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix}, \qquad i = 1,\ldots,4 $$

be the origin of the propeller frames $\mathcal{F}_{P_i}$ in the body frame with L being the distance of $\boldsymbol{O}_{P_i}$ from $\boldsymbol{O}_B$.

Summarizing, the quadrotor configuration is completely determined by the body position $\boldsymbol{p} = {}^W\boldsymbol{O}_B \in \mathbb{R}^3$ and orientation ${}^W\boldsymbol{R}_B$ in the world frame, and by the four tilt angles α_i specifying the propeller group orientations in the body frame (rotations about $\boldsymbol{X}_{P_i}$). We omit the propeller spinning angles about $\boldsymbol{Z}_{P_i}$ as configuration variables, although the propeller spinning velocities $\bar{\omega}_i$ about $\boldsymbol{Z}_{P_i}$ will be part of the system model (see the following sections).

B. Equations of Motion

By exploiting standard techniques (e.g., Newton–Euler procedure), it is possible to derive a complete description of the quadrotor dynamic model by considering the forces/moments generated by the propeller motion, as well as any cross coupling due to gyroscopic and inertial effects arising from the relative motion of the five bodies composing the quadrotor. As aerodynamic forces/torques, we will only consider those responsible for the quadrotor actuation and neglect any additional second-order effects/disturbances. Indeed, as discussed in the next Section II-C, for the typical slow flight regimes considered in this paper, the propeller actuation forces/torques result significantly dominant with respect to other aerodynamic effects. We now discuss in detail the main conceptual steps needed to derive the quadrotor dynamical model.

To this end, let $\boldsymbol{\omega}_B \in \mathbb{R}^3$ be the angular velocity of the quadrotor body B expressed in the body frame,[4] and consider the i-th propeller group P_i. The angular velocity of the i-th propeller (i.e., of $\mathcal{F}_{P_i}$) with respect to $\mathcal{F}_W$ and expressed in $\mathcal{F}_{P_i}$ is just

$$ \boldsymbol{\omega}_{P_i} = {}^B\boldsymbol{R}_{P_i}^T\boldsymbol{\omega}_B + [\dot{\alpha}_i \;\; 0 \;\; \bar{\omega}_i]^T $$

where $\dot{\alpha}_i$ is the tilting velocity about $\boldsymbol{X}_{P_i}$ and $\bar{\omega}_i \in \mathbb{R}$ is the spinning velocity about $\boldsymbol{Z}_{P_i}$, both with respect to $\mathcal{F}_B$ (see Section II-A). This results in an angular acceleration

$$ \dot{\boldsymbol{\omega}}_{P_i} = {}^B\boldsymbol{R}_{P_i}^T\dot{\boldsymbol{\omega}}_B + {}^B\dot{\boldsymbol{R}}_{P_i}^T\boldsymbol{\omega}_B + [\ddot{\alpha}_i \;\; 0 \;\; \dot{\bar{\omega}}_i]^T. $$

By applying the Euler equations of motion, it follows that:

$$ \boldsymbol{\tau}_{P_i} = \boldsymbol{I}_{P_i}\dot{\boldsymbol{\omega}}_{P_i} + \boldsymbol{\omega}_{P_i} \times \boldsymbol{I}_{P_i}\boldsymbol{\omega}_{P_i} - \boldsymbol{\tau}_{\text{ext}_i}. \tag{1} $$

Here, $\boldsymbol{I}_{P_i} \in \mathbb{R}^{3\times3}$ is the (constant) symmetric and positive definite inertia matrix of the ith propeller/rotor assembly approximated as an equivalent disc (the inertia of the tilting mechanism is supposed lumped into the main body B), and $\boldsymbol{\tau}_{\text{ext}_i}$ any external torque applied to the propeller. As usual, see, for example, [31], we assume presence of a counter-rotating torque about the $\boldsymbol{Z}_{P_i}$ axis caused by air drag and modeled as

$$ \boldsymbol{\tau}_{\text{ext}_i} = \left[0 \;\; 0 \;\; -k_m\omega_{P_{i_Z}}|\omega_{P_{i_Z}}|\right]^T, \qquad k_m > 0 \tag{2} $$

with $\omega_{P_{i_Z}}$ being the third component of ω_{P_i}.

Let now

$$ \boldsymbol{T}_{P_i} = [0 \;\; 0 \;\; k_f\bar{\omega}_i|\bar{\omega}_i|]^T, \qquad k_f > 0 \tag{3} $$

represent the ith propeller force (thrust) along the $\boldsymbol{Z}_{P_i}$ axis and acting at ${}^B\boldsymbol{O}_{P_i}$ in $\mathcal{F}_B$. By considering the quadrotor body B and the torques generated by the four propellers P_i, one then obtains

$$ \sum_{i=1}^{4}\left({}^B\boldsymbol{O}_{P_i} \times {}^B\boldsymbol{R}_{P_i}\boldsymbol{T}_{P_i} - {}^B\boldsymbol{R}_{P_i}\boldsymbol{\tau}_{P_i}\right) = \boldsymbol{I}_B\dot{\boldsymbol{\omega}}_B + \boldsymbol{\omega}_B \times \boldsymbol{I}_B\boldsymbol{\omega}_B \tag{4} $$

with $\boldsymbol{I}_B \in \mathbb{R}^{3\times3}$ being the (constant) symmetric and positive definite inertia matrix of B.

As for the translational dynamics, we assume for simplicity that the barycenter of each propeller group P_i coincides with $\boldsymbol{O}_{P_i}$. This allows us to neglect inertial effects on the propeller groups owing to the quadrotor body acceleration in space. Therefore, by recalling that $\boldsymbol{p} = {}^W\boldsymbol{O}_B$ is the quadrotor body position in world frame, one has

$$ m\ddot{\boldsymbol{p}} = m\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + {}^W\boldsymbol{R}_B\sum_{i=1}^{4}{}^B\boldsymbol{R}_{P_i}\boldsymbol{T}_{P_i} \tag{5} $$

where m is the total mass of the quadrotor and propeller bodies and g is the scalar gravitational acceleration of Earth.

[3] Throughout the following, $\boldsymbol{R}_X(\theta)$, $\boldsymbol{R}_Y(\theta)$, $\boldsymbol{R}_Z(\theta)$ will denote the canonical rotation matrixes about the $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ axes of angle θ, respectively.

[4] In the following, we will assume that every quantity is expressed in its own frame, for example, $\boldsymbol{\omega}_B = {}^B\boldsymbol{\omega}_B$.

TABLE I
MAIN QUANTITIES AND DEFINITIONS FOR THE HOLOCOPTER DYNAMIC MODEL

Symbols	Definitions
$\mathcal{F}_W$	inertial world frame
$\mathcal{F}_B$	quadrotor body frame B
$\mathcal{F}_{P_i}$	i-th propeller group frame P_i
$\boldsymbol{p}$	position of B in $\mathcal{F}_W$
${}^W\boldsymbol{R}_B$	rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$
${}^B\boldsymbol{R}_{P_i}$	rotation matrix from $\mathcal{F}_{P_i}$ to $\mathcal{F}_B$
α_i	i-th propeller tilt angle about $\boldsymbol{X}_{P_i}$
$\bar{\omega}_i$	i-th propeller spinning velocity about $\boldsymbol{Z}_{P_i}$
$\boldsymbol{\omega}_B$	angular velocity of B in $\mathcal{F}_B$
$\boldsymbol{\tau}_{ext_i}$	i-th propeller air drag torque about $\boldsymbol{Z}_{P_i}$
$\boldsymbol{T}_i$	i-th propeller thrust along $\boldsymbol{Z}_{P_i}$
τ_{P_i}	motor torque actuating $\boldsymbol{X}_{P_i}$
$\tau_{\bar{\omega}_i}$	motor torque actuating $\boldsymbol{Z}_{P_i}$
m	total mass
$\boldsymbol{I}_P$	inertia of the i-th propeller group P_i
$\boldsymbol{I}_B$	inertia of the quadrotor body B
k_f	propeller thrust coefficient
k_m	propeller drag coefficient
L	distance of $\mathcal{F}_{P_i}$ from $\mathcal{F}_B$
g	gravitational acceleration of Earth

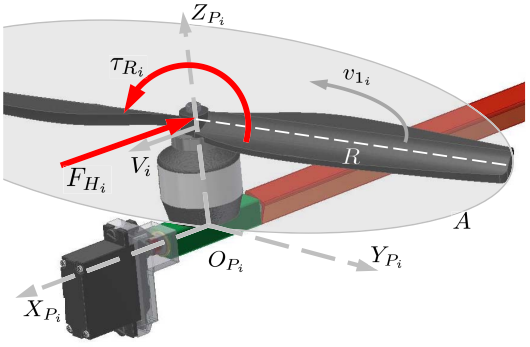


Fig. 4. Visualization of hub force $\boldsymbol{F}_{H_i}$ and rolling moment $\boldsymbol{\tau}_{R_i}$ acting on a single blade.

Summarizing, (1)–(5) describe the rotational/translational dynamics of the quadrotor body and propeller groups. Note that the inputs of this model are the motor torques actuating the propeller tilting axes $\boldsymbol{X}_{P_i}$ and spinning axes $\boldsymbol{Z}_{P_i}$. These are denoted as $\tau_{\alpha_i} = \boldsymbol{\tau}_{P_i}^T \boldsymbol{X}_{P_i} \in \mathbb{R}$ and $\tau_{\bar{\omega}_i} = \boldsymbol{\tau}_{P_i}^T \boldsymbol{Z}_{P_i} \in \mathbb{R}$, $i = 1, \ldots, 4$, respectively, for a total of $4 + 4 = 8$ independent control torques (inputs). The propeller spinning velocities $\bar{\omega}_i$ (actuated by $\tau_{\bar{\omega}_i}$) will then generate the forces and torques affecting the translational/rotational motion of the quadrotor body B as a function of its current configuration, in particular of the tilting angles α_i actuated by τ_{α_i}. For the reader's convenience, Table I lists the main quantities introduced in this section.

C. Additional Aerodynamic Effects

The derivation of the most significant aerodynamic effects besides the already considered propeller torques/forces (2–3) can be obtained by considering momentum and blade element theory. In this section, we will focus on the influence of the hub force $\boldsymbol{F}_{H_i}$ (a force perpendicular to the rotor shaft acting on the single blade elements) and of the rolling moment $\boldsymbol{\tau}_{R_i}$ (a torque around $\boldsymbol{X}_{P_i}$ caused by different thrust on the retreating and the advancing blade of the propeller). Fig. 4 shows an illustration of these quantities, and Table II summarizes the main symbols introduced hereafter.

TABLE II
SYMBOLS AND DEFINITIONS DESCRIBING THE MAIN AERODYNAMIC EFFECTS ON A SINGLE PROPELLER

Symbols	Definitions
$\rho\ [kg/m^3]$	Air density
$A\ [m^2]$	Propeller disk area
$v_1\ [m/s]$	Propeller inflow velocity
$\boldsymbol{F}_H\ [N]$	Hub force
$\boldsymbol{\tau}_R\ [Nm]$	Rolling moment
$R\ [m]$	Propeller radius
a	Lift slope
$\sigma\ [rad^{-1}]$	Solidity ratio
C_d	Average drag coefficient
λ_i	Inflow ratio
μ_i	Rotor advance ratio
$\theta_0\ [rad]$	Pitch of incidence
$\theta_{tw}\ [rad]$	Twist pitch
$\dot{x}_{P_i}, \dot{y}_{P_i}, \dot{z}_{P_i}\ [m/s]$	Velocity of the propeller w.r.t. ground

Let $(\dot{x}_{P_i}, \dot{y}_{P_i}, \dot{z}_{P_i}) = {}^W\boldsymbol{R}_{P_i}^T\, {}^W\dot{\boldsymbol{O}}_{P_i}$ be the velocity of the ith propeller with respect to the world frame and expressed in the propeller frame $\mathcal{F}_{P_i}$. We define

$$V_i = \sqrt{\dot{x}_{P_i}^2 + \dot{y}_{P_i}^2}$$

as the sideways velocity of the ith propeller in propeller group frame. Momentum theory models the generated thrust of a single propeller as

$$\|\boldsymbol{T}_{P_i}\| = 2\rho A v_{1_i} \sqrt{V_i^2 + v_{1_i}^2},$$

with v_{1_i} being the so-called ith propeller inflow velocity [32]. Solving for v_{1_i} we get

$$v_{1_i} = \sqrt{\frac{V_i^2}{2} + \sqrt{\left(\frac{V_i^2}{2}\right)^2 + \left(\frac{\|\boldsymbol{T}_{P_i}\|}{2\rho A}\right)^2}} \tag{6}$$

where ρ is the air density and A the rotor area.

Following the assumptions and simplifications of [33], the magnitude of the hub force $\boldsymbol{F}_{H_i}$ can then be modeled as

$$\begin{cases} \|\boldsymbol{F}_{H_i}\| = C_{H_i} \rho A (\bar{\omega}_i R)^2 \\ \dfrac{C_{H_i}}{\sigma a} = \dfrac{1}{4a}\mu_i \bar{C}_d + \dfrac{1}{4}\lambda_i \mu_i \left(\theta_0 - \dfrac{\theta_{tw}}{2}\right) \end{cases} \tag{7}$$

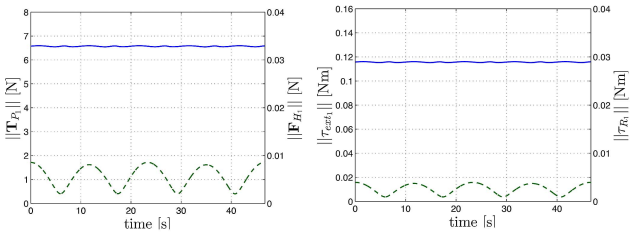
where

$$\lambda_i = \frac{v_{1_i} - \dot{z}_{P_i}}{\bar{\omega}_i R}, \qquad \mu_i = \frac{V_i}{\bar{\omega}_i R} \tag{8}$$

and R is the propeller radius, σ is the solidity ratio, a is the lift slop, $\bar{C}_d$ is the average drag coefficient, θ_0 is the pitch of incidence, and θ_{tw} is the twist pitch. One can similarly model the magnitude of the rolling moment $\boldsymbol{\tau}_{R_i}$ acting on P_i and owing to the different lift of the retrieving and the advancing blade of the propeller as

$$\begin{cases} \|\boldsymbol{\tau}_{R_i}\| = C_{R_m i} \rho A (\bar{\omega}_i R)^2 R \\ \dfrac{C_{R_m i}}{\sigma a} = -\mu_i \left(\frac{1}{6}\theta_0 - \frac{1}{8}\theta_{tw} - \frac{1}{8}\lambda_i\right). \end{cases} \tag{9}$$

To assess the influence of these aerodynamic effects, we compared them with the thrust $\boldsymbol{T}_{P_i}$ and torque $\boldsymbol{\tau}_{\text{ext}_i}$ of



(a) (b)

Fig. 5. (a) Solid line: thrust $\|\boldsymbol{T}_{P_1}\|$ in $\mathcal{F}_{P_1}$ (left y-axis). Dashed line: hub force $\|\boldsymbol{F}_{H_1}\|$ in $\mathcal{F}_{P_1}$ (right y-axis). (b) Solid line: drag $\|\boldsymbol{\tau}_{\text{ext}_1}\|$ in $\mathcal{F}_{P_1}$ (left y-axis). Dashed line: rolling moment $\|\boldsymbol{\tau}_{R_1}\|$ in $\mathcal{F}_{P_1}$ (right y-axis). Note the different scales of the y-axes in both plots.

a single propeller during the simulated trajectory described in Section V-A 2 and relying, for the various parameters, on the physical properties of our prototype and on values taken from literature. The trajectory consists of a horizontal eight-shape planar curve with a superimposed sinusoidal rotation about the body $\boldsymbol{Y}_B$ axis, and has been chosen as being representative of the typical operational regimes of our prototype. The results are shown in Fig. 5: as clear from the plots, it is $\|\boldsymbol{T}_{P_i}\|/\|\boldsymbol{F}_{H_i}\| \simeq 600$ and $\|\boldsymbol{\tau}_{\text{ext}_i}\|/\|\boldsymbol{\tau}_{R_i}\| \simeq 30$ (note the two different scales of the y-axes in both plots). This then confirms the minor significance of these (neglected) second-order aerodynamics effects with respect to the propeller actuation forces/torques (2) and (3) which are instead taken into account in the holocopter dynamical model. As for the significance of blade flapping and induced drag (typically modeled as the first-order contributions, see [3]), a similar analysis showed an even smaller effect with respect to the aforementioned second-order forces/torques in the same flying regimes. Therefore, these first-order terms were also neglected in the holocopter model.

III. Motion Control of the Holocopter

We now proceed to illustrate the control approach adopted in this paper for exploiting the holocopter capabilities in tracking arbitrary desired trajectories for the position and orientation of its main body B.

A. Simplified Holocopter Model

We start with some preliminary considerations: the dynamic model illustrated in the previous section is useful for simulation purposes as it captures the main effects of the quadrotor motion in space (apart from unmodeled aerodynamics forces/torques). Some simplifications are, however, useful for transforming it into a reduced model more suited to control design. First, as in many practical situations, we assume that the motors actuating the tilting/spinning axes are implementing a fast high-gain local controller able to impose desired speeds $\omega_{\alpha_i} = \dot{\alpha}_i$ and $\bar{\omega}_i$ with negligible transients.[5] This allows to neglect the motor dynamics, and to consider ω_{α_i} and $\bar{\omega}_i$, $i = 1, \ldots, 4$, as (virtual) control inputs in place of the motor torques τ_{α_i} and $\tau_{\bar{\omega}_i}$. Second, in this simplified model we also neglect the internal gyroscopic/inertial effects by considering them as second-order disturbances to be rejected by the controller.[6] We note that the validity of these assumptions will be discussed in Section V-A where the proposed controller will be tested against the complete dynamic model described in Section II representing the actual dynamics of the quadrotor.

Let us then define $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_4]^T \in \mathbb{R}^4$, $\boldsymbol{w}_\alpha = [\omega_{\alpha_1}, \ldots, \omega_{\alpha_4}]^T \in \mathbb{R}^4$ and $\boldsymbol{w} = [\bar{\omega}_1|\bar{\omega}_1|, \ldots, \bar{\omega}_4|\bar{\omega}_4|]^T \in \mathbb{R}^4$. Note that the elements of vector $\boldsymbol{w}$ are the signed squares of the spinning velocities $\bar{\omega}_i$, as the torques and forces in (2) and (3) are a (approx. linear) function of these quantities. Therefore, in the following analysis, $\omega_i = \bar{\omega}_i|\bar{\omega}_i|$ will be considered as input spinning velocity of the ith propeller, with the understanding that one can always recover the actual speed $\bar{\omega}_i = \text{sign}(\omega_i)\sqrt{|\omega_i|}$. Under the stated assumptions, the quadrotor dynamic model can be simplified into

$$\begin{cases} \ddot{\boldsymbol{p}} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \dfrac{1}{m} {}^W\boldsymbol{R}_B \boldsymbol{F}(\boldsymbol{\alpha})\boldsymbol{w} \\ \dot{\boldsymbol{\omega}}_B = \boldsymbol{I}_B^{-1}\boldsymbol{\tau}(\boldsymbol{\alpha})\boldsymbol{w} \\ \dot{\boldsymbol{\alpha}} = \boldsymbol{w}_\alpha \\ {}^W\dot{\boldsymbol{R}}_B = {}^W\boldsymbol{R}_B[\boldsymbol{\omega}_B]_\wedge \end{cases} \tag{10}$$

with $[\cdot]_\wedge$ being the usual map taking a vector $\boldsymbol{a} \in \mathbb{R}^3$ into the associated skew-symmetric matrix $[\boldsymbol{a}]_\times \in so(3)$, and

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{\alpha}) &= \begin{bmatrix} 0 & -k_f s_2 & 0 & k_f s_4 \\ -k_f s_1 & 0 & k_f s_3 & 0 \\ k_f c_1 & -k_f c_2 & k_f c_3 & -k_f c_4 \end{bmatrix} \\ \boldsymbol{\tau}(\boldsymbol{\alpha}) &= \begin{bmatrix} 0 & -Lk_f c_2 - k_m s_2 & 0 & Lk_f c_4 + k_m s_4 \\ -Lk_f c_1 + k_m s_1 & 0 & Lk_f c_3 - k_m s_3 & 0 \\ -Lk_f s_1 - k_m c_1 & Lk_f s_2 - k_m c_2 & -Lk_f s_3 - k_m c_3 & Lk_f s_4 - k_m c_4 \end{bmatrix} \end{aligned} \tag{11}$$

the 3×4 input coupling matrixes ($s_i = \sin(\alpha_i)$ and $c_i = \cos(\alpha_i)$). Note that input $\boldsymbol{w}$ appears linearly in (10) as expected. The subsequent control design is then performed on the simplified model (10) and (11).

B. Control Design

The control problem considered in this paper is an output-tracking problem: how to track, with the available inputs, a desired (and arbitrary) trajectory $(\boldsymbol{p}_d(t), \boldsymbol{R}_d(t)) \in \mathbb{R}^3 \times SO(3)$ for the body position $\boldsymbol{p}$ and orientation ${}^W\boldsymbol{R}_B$ taken as output functions. We note again that this problem is clearly ill-posed for a standard quadrotor with fixed propellers because of its underactuation. However, the chosen overactuated design of the holocopter (eight independent control inputs) can guarantee full controllability over the position/orientation of its main body as it is shown in the following developments.

As in many output-tracking problems, a possible solution is to resort to output feedback linearization techniques

[5]For instance, in the standard quadrotor case, the spinning velocities $\bar{\omega}_i$ are usually taken as control inputs.

[6]Obviously, this assumption holds as long as the inertia of the propeller group is small with respect to the main holocopter body.

(either static or dynamic, see [34] for a detailed treatment). Therefore, we rewrite the first two rows of (10) as

$$\begin{aligned}\begin{bmatrix}\ddot{\boldsymbol{p}}\\ \dot{\boldsymbol{\omega}}_B\end{bmatrix} &= \begin{bmatrix}\begin{bmatrix}0\\0\\-g\end{bmatrix}\\ \boldsymbol{0}\end{bmatrix} + \begin{bmatrix}\frac{1}{m}{}^W\boldsymbol{R}_B & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_B^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{F}(\boldsymbol{\alpha}) & \boldsymbol{0}\\ \boldsymbol{\tau}(\boldsymbol{\alpha}) & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{w}_\alpha\end{bmatrix}\\ &= \boldsymbol{f} + \boldsymbol{J}_R\begin{bmatrix}\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha}) & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{w}_\alpha\end{bmatrix}\\ &= \boldsymbol{f} + \boldsymbol{J}_R\boldsymbol{J}_\alpha(\boldsymbol{\alpha})\begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{w}_\alpha\end{bmatrix}\\ &= \boldsymbol{f} + \boldsymbol{J}(\boldsymbol{\alpha})\begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{w}_\alpha\end{bmatrix} \qquad (12)\end{aligned}$$

where $\boldsymbol{f} \in \mathbb{R}^6$ is a constant drift vector, $\bar{\boldsymbol{J}}(\boldsymbol{\alpha}) \in \mathbb{R}^{6\times 4}$, $\boldsymbol{J}_R \in \mathbb{R}^{6\times 6}$, and the 6×8 matrix $\boldsymbol{J}(\boldsymbol{\alpha})$ will be referred to as the output Jacobian. When $\rho_J = \text{rank}(\boldsymbol{J}(\boldsymbol{\alpha})) = 6$, it is always possible to statically feedback linearize (12) by means of the law

$$\begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{w}_\alpha\end{bmatrix} = \boldsymbol{K}(\boldsymbol{\alpha})\left(-\boldsymbol{f} + \begin{bmatrix}\ddot{\boldsymbol{p}}_r\\ \dot{\boldsymbol{\omega}}_r\end{bmatrix}\right) \qquad (13)$$

where $\boldsymbol{K}(\boldsymbol{\alpha})$ is a generalized inverse of $\boldsymbol{J}(\boldsymbol{\alpha})$, for example, the pseudoinverse $\boldsymbol{J}^\dagger(\boldsymbol{\alpha})$, and $[\ddot{\boldsymbol{p}}_r^T \;\; \dot{\boldsymbol{\omega}}_r^T]^T \in \mathbb{R}^6$ an arbitrary reference linear/angular acceleration vector to be imposed to the output dynamics in (10).

This solution is, however, not viable in the case under consideration. Indeed, $\rho_J = \text{rank}(\boldsymbol{J}) = \text{rank}(\boldsymbol{J}_R\boldsymbol{J}_\alpha) = \text{rank}(\boldsymbol{J}_\alpha)$ since $\boldsymbol{J}_R$ is a nonsingular square matrix. Furthermore, $\rho_J = \text{rank}(\boldsymbol{J}_\alpha) = \text{rank}(\bar{\boldsymbol{J}}_\alpha) \leq 4 < 6$ because of the structural null matrix $\boldsymbol{0} \in \mathbb{R}^{6\times 4}$ in matrix $\boldsymbol{J}_\alpha(\boldsymbol{\alpha})$ weighting the inputs $\boldsymbol{w}_\alpha$. Presence of this null matrix is due to the fact that inputs $\boldsymbol{w}_\alpha$ affect the output dynamics at a higher differential level compared with inputs $\boldsymbol{w}$. Therefore, a direct inversion at the acceleration level is bound to exploit only inputs $\boldsymbol{w}$ resulting in a loss of controllability for the system. Intuitively, the instantaneous linear/angular acceleration of the quadrotor body is directly affected by the propeller speeds $\boldsymbol{w}$ and tilting configuration $\boldsymbol{\alpha}$ (thanks to the dependence in $\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})$), but not by $\dot{\boldsymbol{\alpha}} = \boldsymbol{w}_\alpha$, that is, the tilting velocities.[7]

A possible way to circumvent these difficulties is to resort to a dynamic output linearization scheme and seek to invert the input–output map at a higher differential level where inputs $\boldsymbol{w}_\alpha$ will explicitly appear. This can be achieved by expanding the term $\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})\boldsymbol{w}$ in (12) as follows:

$$\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})\boldsymbol{w} = \sum_{i=1}^{4}\bar{\boldsymbol{j}}_i(\boldsymbol{\alpha})w_i$$

and noting that

$$\frac{\mathrm{d}\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})\boldsymbol{w}}{\mathrm{d}t} = \bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})\dot{\boldsymbol{w}} + \sum_{i=1}^{4}\frac{\partial\bar{\boldsymbol{j}}_i(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}}\boldsymbol{w}_\alpha w_i$$

differentiation of (12) with respect to time yields

$$\begin{aligned}\begin{bmatrix}\dddot{\boldsymbol{p}}\\ \ddot{\boldsymbol{\omega}}_B\end{bmatrix} &= \boldsymbol{J}_R\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})\dot{\boldsymbol{w}} + \boldsymbol{J}_R\sum_{i=1}^{4}\frac{\partial\bar{\boldsymbol{j}}_i(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}}\boldsymbol{w}_\alpha w_i + \dot{\boldsymbol{J}}_R\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})\boldsymbol{w}\\ &= \boldsymbol{J}_R\begin{bmatrix}\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha}) & \sum_{i=1}^{4}\frac{\partial\bar{\boldsymbol{j}}_i(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}}w_i\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{w}}\\ \boldsymbol{w}_\alpha\end{bmatrix} + \begin{bmatrix}\frac{{}^W\dot{\boldsymbol{R}}_B}{m}\boldsymbol{F}(\boldsymbol{\alpha})\boldsymbol{w}\\ \boldsymbol{0}\end{bmatrix} \qquad (14)\\ &= \boldsymbol{J}_R\boldsymbol{J}'_\alpha(\boldsymbol{\alpha},\,\boldsymbol{w})\begin{bmatrix}\dot{\boldsymbol{w}}\\ \boldsymbol{w}_\alpha\end{bmatrix} + \boldsymbol{b}(\boldsymbol{\alpha},\,\boldsymbol{w},\,\boldsymbol{\omega}_B)\\ &= \boldsymbol{A}(\boldsymbol{\alpha},\boldsymbol{w})\begin{bmatrix}\dot{\boldsymbol{w}}\\ \boldsymbol{w}_\alpha\end{bmatrix} + \boldsymbol{b}(\boldsymbol{\alpha},\,\boldsymbol{w},\,\boldsymbol{\omega}_B) \qquad (15)\end{aligned}$$

where the new input $\dot{\boldsymbol{w}}$ is the dynamic extension of the former (and actual) input $\boldsymbol{w}$ obtained by adding four integrators on its channel.[8]

We note that the new 6×8 input–output decoupling matrix $\boldsymbol{A}(\boldsymbol{\alpha},\,\boldsymbol{w})$ consists of two column blocks: although the first block $\boldsymbol{J}_R\bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})$ is exactly the first block of the former output Jacobian $\boldsymbol{J}(\boldsymbol{\alpha})$, the second block is not a null matrix as in the previous case. Rather, a new set of 4 columns, weighting inputs $\boldsymbol{w}_\alpha$, are now present and contributing to the rank of matrix $\boldsymbol{A}$. Furthermore, it is $\boldsymbol{A}(\boldsymbol{\alpha},\,\boldsymbol{0}) = \boldsymbol{J}(\boldsymbol{\alpha})$ and, as proven in [28], $\rho_A = \text{rank}(\boldsymbol{A}) = 6$ as long as $w_i \neq 0$, $i = 1\ldots 4$; in other words, full rankness of matrix $\boldsymbol{A}$ can always be ensured by preventing the propellers from stopping their spinning motion. Section III-C proposes a strategy able to meet this requirement.

With $\rho_A = \text{rank}(\boldsymbol{A}) = 6$ system (14) can be inverted by the law

$$\begin{bmatrix}\dot{\boldsymbol{w}}\\ \boldsymbol{w}_\alpha\end{bmatrix} = \boldsymbol{A}^\dagger\left(\begin{bmatrix}\dddot{\boldsymbol{p}}_r\\ \ddot{\boldsymbol{\omega}}_r\end{bmatrix} - \boldsymbol{b}\right) + (\mathbf{I}_8 - \boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{z} \qquad (16)$$

with $\mathbf{I}_N$ being the identity matrix of dimension N and $\boldsymbol{A}^\dagger \in \mathbb{R}^{8\times 6}$ denoting the Moore–Penrose pseudoinverse of matrix $\boldsymbol{A}$, to achieve full input–output linearization

$$\begin{bmatrix}\dddot{\boldsymbol{p}}\\ \ddot{\boldsymbol{\omega}}_B\end{bmatrix} = \begin{bmatrix}\dddot{\boldsymbol{p}}_r\\ \ddot{\boldsymbol{\omega}}_r\end{bmatrix}. \qquad (17)$$

Vector $\boldsymbol{z} \in \mathbb{R}^8$ in (16) is an additional free quantity projected onto the 2-D null-space of $\boldsymbol{A}$ whose use will be detailed in Section III-C. We note that presence of a 2-D null space for matrix $\boldsymbol{A}$ is a direct (and expected) consequence of the actuation redundancy of degree 2 of the considered holocopter (eight control inputs for 6 controlled DoF).

Assuming now $\boldsymbol{p}_d(t) \in \bar{\mathcal{C}}^3$, it is then sufficient to set in (17)

$$\dddot{\boldsymbol{p}}_r = \dddot{\boldsymbol{p}}_d + \boldsymbol{K}_{p_1}(\ddot{\boldsymbol{p}}_d - \ddot{\boldsymbol{p}}) + \boldsymbol{K}_{p_2}(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}) + \boldsymbol{K}_{p_3}(\boldsymbol{p}_d - \boldsymbol{p}) \qquad (18)$$

for obtaining exponential and decoupled convergence of the position error to $\boldsymbol{0}$ as long as the (diagonal) positive definite gain matrixes $\boldsymbol{K}_{p_1}$, $\boldsymbol{K}_{p_2}$, $\boldsymbol{K}_{p_3}$ define Hurwitz polynomials. For the stabilization of the orientation-tracking error, several choices are possible depending on the particular parameterization chosen for the rotation matrix $\boldsymbol{R}$. Besides the usual

[7]It is interesting to note that this inhomogeneity in the differential levels at which inputs are affecting the output dynamics is not a specificity of the system at hand. As an example, the same structural property is also present in other robotic structures such as mobile manipulators with steering wheels [35] where the role of $\boldsymbol{w}_\alpha$ is played by the wheel steering velocities.

[8]By means of this dynamic extensions, vector $\boldsymbol{w}$ becomes an internal state of the controller.

Euler angles (with their inherent singularity issues), a convenient possibility is to resort to an orientation error term directly defined on $SO(3)$ (as shown in [36] and [37]). Assume, as before, that $\boldsymbol{R}_d(t) \in \mathcal{C}^3$ and let $\boldsymbol{\omega}_d = [\boldsymbol{R}_d^T \dot{\boldsymbol{R}}_d]_\vee$, where $[\cdot]_\vee$ represents the inverse map from $so(3)$ to $\mathbb{R}^3$. By defining the orientation error as

$$\boldsymbol{e}_R = \frac{1}{2}\left[{}^W\boldsymbol{R}_B^T \boldsymbol{R}_d - \boldsymbol{R}_d^T {}^W\boldsymbol{R}_B\right]_\vee \tag{19}$$

the choice

$$\ddot{\boldsymbol{\omega}}_r = \ddot{\boldsymbol{\omega}}_d + \boldsymbol{K}_{\omega_1}(\dot{\boldsymbol{\omega}}_d - \dot{\boldsymbol{\omega}}_B) + \boldsymbol{K}_{\omega_2}(\boldsymbol{\omega}_d - \boldsymbol{\omega}_B) + \boldsymbol{K}_{\omega_3}\boldsymbol{e}_R \tag{20}$$

in (17) yields an exponential convergence for the orientation-tracking error to $\boldsymbol{0}$ as desired, provided that the (diagonal) gain matrixes $\boldsymbol{K}_{\omega_1}$, $\boldsymbol{K}_{\omega_2}$, $\boldsymbol{K}_{\omega_3}$ define a Hurwitz polynomial.

C. Optimization of Additional Criteria

As the final step, we now discuss how to exploit the 2-D actuation redundancy of the holocopter by exploiting vector $\boldsymbol{z}$ in (16).

Being projected onto the null space of $\boldsymbol{A}$, vector $\boldsymbol{z}$ does not produce actions interfering with the output-tracking objective and can thus be exploited to fulfill additional tasks. In our case, a first mandatory requirement is to keep $\rho_A = 6$ at all times for avoiding singularities of the decoupling matrix $\boldsymbol{A}$ in (16). As explained, this objective can be easily met by ensuring $\boldsymbol{w} \neq \boldsymbol{0}$. Likewise another important requirement is to minimize the norm of $\boldsymbol{w}$ to reduce the energy consumption during flight since, for instance, the air drag torques $\boldsymbol{\tau}_{\text{ext}_i}$ in (2) are always performing a dissipative work against w_i.

A possible cost function $H(\boldsymbol{w})$ considering these two competing objectives is

$$H(\boldsymbol{w}) = \sum_{i=1}^{4} h(w_i)$$

with

$$h(w_i) = \begin{cases} k_{h_1}\tan^2(\gamma_1|w_i| + \gamma_2) & w_{\min} < |w_i| \leq w_{\text{rest}} \\ k_{h_2}(|w_i| - w_{\text{rest}})^2 & |w_i| > w_{\text{rest}} \end{cases} \tag{21}$$

$$\gamma_1 = \frac{\pi}{2(w_{\text{rest}} - w_{\min})}, \quad \gamma_2 = -\gamma_1 w_{\text{rest}}, \quad k_{h_1} > 0, \quad k_{h_2} > 0$$

suitable scalar gains. Here, $w_{\min} > 0$ represents a minimum value for the propeller spinning velocities and $w_{\text{rest}} > w_{\min}$ a suitable rest speed. Furthermore, functions $h_i(w_i)$ are such that $h_i(w_i) \to \infty$ if either $|w_i| \to w_{\min}$ or $|w_i| \to \infty$, and have a unique minimum (with continuous derivative) at w_{rest}. For the placement of w_{rest}, in our implementation we chose

$$w_{\text{rest}} = \frac{mg}{4k_f} \tag{22}$$

that is, the (squared) spinning velocity needed by each propeller to hover.

An illustrative example for $h_i(w_i)$ with $w_{\min} = 126^2$ [rad^2/s^2] and $w_{\text{rest}} = 450^2$ [rad^2/s^2] is shown in Fig. 6 (these values correspond to the ones used for the motor propeller combination of our holocopter prototype).

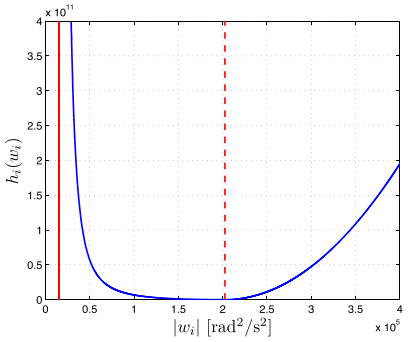


Fig. 6. Example of a function $h_i(w_i)$ with $\bar{w}_{\min} = 126$ [rad/s] (solid red vertical line), $\bar{w}_{\text{rest}} = 450$ [rad/s] (dashed red vertical line). Note that $h_i(w_i) \to \infty$ as $|w_i| \to w_{\min}$ or $|w_i| \to \infty$, and that $h_i(w_i)$ has a unique minimum at w_{rest} with continuous derivative.

Minimization of $H(\boldsymbol{w})$, compatibly with the output-tracking task, is then obtained by setting in (16)

$$\boldsymbol{z} = -k_H \begin{bmatrix} \nabla_{\boldsymbol{w}} H(\boldsymbol{w}) \\ \boldsymbol{0} \end{bmatrix} \tag{23}$$

where $k_H > 0$ is the suitable step size. Note that, as a byproduct, this choice will also result in a beneficial velocity damping-like action on the states $\boldsymbol{w}$ as, for example, described in [38]. We finally note that additional optimization actions could be embedded in vector $\boldsymbol{z}$, for instance by concurrently minimizing the second cost function $H_\alpha(\boldsymbol{\alpha})$ representing constraints on the range of the tilting angles $\boldsymbol{\alpha}$.

D. Final Considerations

We conclude the section by noting that the position/orientation feedback terms in (18)–(20) require a measurement of the holocopter:

1) position $\boldsymbol{p}$ and orientation ${}^W\boldsymbol{R}_B$;
2) linear velocity $\dot{\boldsymbol{p}}$ and angular velocity $\boldsymbol{\omega}_B$;
3) linear acceleration $\ddot{\boldsymbol{p}}$ and angular acceleration $\dot{\boldsymbol{\omega}}_B$.

Availability of the quantities in items 1 and 2 is a standard requirement for any UAV flight control scheme (see also the next Section IV-B), whereas measurement of the linear/angular accelerations of item 3 can pose some challenges in real-world scenarios because of the typical high noise level of these signals when obtained from onboard sensors (e.g., accelerometers) or numerical differentiation of velocity-like quantities.

As an alternative, one can also exploit the holocopter model to evaluate the linear/angular acceleration $(\ddot{\boldsymbol{p}}, \dot{\boldsymbol{\omega}}_B)$ in terms of sole velocity measurements (vector $\boldsymbol{w}$) since, from (12), it follows that:

$$\begin{bmatrix} \ddot{\boldsymbol{p}} \\ \dot{\boldsymbol{\omega}}_B \end{bmatrix} = \boldsymbol{f} + \boldsymbol{J}_R \bar{\boldsymbol{J}}_\alpha(\boldsymbol{\alpha})\boldsymbol{w}. \tag{24}$$

Obviously this possibility assumes a good knowledge of the model parameters and of the system state: the simulation and experimental results given in Sections V and VI are nevertheless conducted by relying on (24) for obtaining $(\ddot{\boldsymbol{p}}, \dot{\boldsymbol{\omega}}_B)$, and will thus confirm the appropriateness of this assumption for our setup.

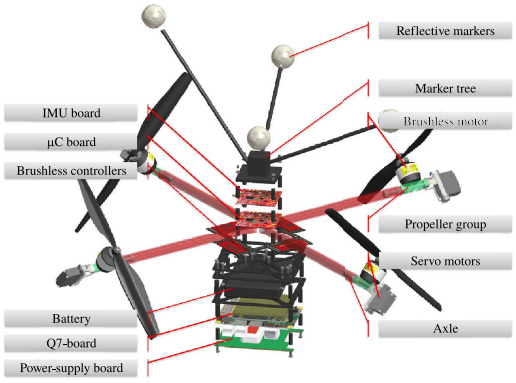


Fig. 7. Exploded view of the various components of the holocopter. All the important parts are properly labeled.

IV. Holocopter Prototype and System Architecture

A. Prototype

For the first prototype, we opted for a very low-cost solution with all parts available off-the-shelf. The overall cost including all mechanical and electrical parts and actuators is below 1000 €. The mechanical main frame of the holocopter is based on the MikroKopter[9] module, including the propeller (EPP1045 CF) and the brushless propeller motors (Roxxy 2827-35). At the end of every arm of the holocopter body, a rigidly connected axle allows rotation of the propeller groups containing the propeller motor and the servo motor for the tilting actuation (Robbe S3150 Digital) (see Fig. 7). This has a maximum torque $\tau_{\alpha_{\max}} = 0.37$ Nm and a maximum rotation speed $\dot{\alpha}_{\max} = 4.1$ rad/s. The propeller group is designed to have its barycenter as close as possible to the axle, as assumed in the dynamical model developed in Section II.

Furthermore, two microcontroller boards are mounted on the top of the holocopter. The first contains the gyroscopes measuring $\boldsymbol{\omega}_B$, and is also in charge of reading the tilting angles α_i of the servo motors and the spinning velocities $\bar{\omega}_i$ of the propellers. The second microcontroller board sends the desired spinning velocities $\bar{\omega}_{\mathrm{Des}_i}$ to the brushless controller and the desired angles $\alpha_{\mathrm{Des}i}$ to the servo motors.

The trajectory tracking controller described in Section III is implemented in MATLAB/Simulink and, via the Real-Time Workshop toolbox, deployed and executed in real-time on an Intel Atom board (Quadmo747, from now on Q7-board) running the Linux Ubuntu10.10 real-time environment. The Q7-board is mounted below the battery and is equipped with a Wifi USB-dongle for communication. As only one RS-232 port (TTL level) is available on the Q7-board, the second microcontroller board is connected via USB-port and an USBToSerial converter. The Q7-board is powered by a battery, with the necessary voltage conversion and stabilization performed by a power-supply board containing a 12 V dc/dc power converter.

The nominal mass of the full holocopter is 1.32 kg. From a high-detail CAD model of the body and propeller groups, we also obtained the following inertia matrixes:

$$\boldsymbol{I}_{P_i} = \begin{bmatrix} 8.450e^{-5} & 0 & 0 \\ 0 & 8.450e^{-5} & 0 \\ 0 & 0 & 4.580e^{-5} \end{bmatrix} \left[\mathrm{kg\,m^2}\right]$$

and

$$\boldsymbol{I}_B = \begin{bmatrix} 0.0154 & 0 & 0 \\ 0 & 0.0154 & 0 \\ 0 & 0 & 0.0263 \end{bmatrix} \left[\mathrm{kg\,m^2}\right].$$

In the current setup, the servo motors are limited in their rotation by mechanical end stops in the range of $-90° < \alpha_i < 90°$. For our particular prototype, these limits translate into a maximum achievable rotation (in hover) of $\approx \pm 55°$ around the roll or pitch axes for the body frame B (this value was experimentally determined).

To obtain accurate values of k_f and k_m for our motor–propeller combination, we made use of a testbed equipped with a 6-dof torque/force sensor (Nano17-E, see Fig. 8) for identifying the mappings between the propeller spinning velocity and the generated thrust $\|\boldsymbol{T}_{P_i}\|$ and torque $\|\boldsymbol{\tau}_{\mathrm{ext}_i}\|$ [see (2) and (3)]. This resulted in the following polynomial models (shown in Fig. 9):

$$\|\boldsymbol{T}_{P_i}\| = 4.94e^{-18}|w_i|^3 + 9.62e^{-13}|w_i|^2 + 1.56e^{-5}|w_i| \quad (25)$$

and

$$\|\boldsymbol{\tau}_{\mathrm{ext}_i}\| = -5.41e^{-19}|w_i|^3 + 2.50e^{-13}|w_i|^2 - 2.53e^{-7}|w_i| \quad (26)$$

[9]http://www.mikrokopter.de

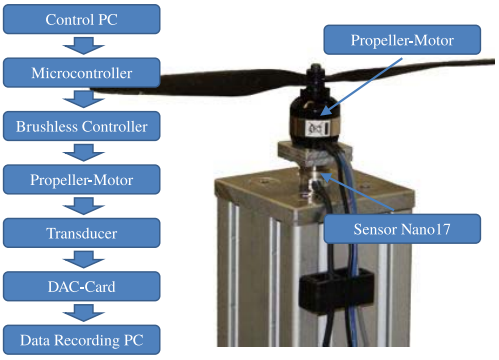


Fig. 8. Left: scheme of the measurement chain. Right: motor testbed including Propeller motor combination and Nano17 sensor mounted at a height of 0.45 m.

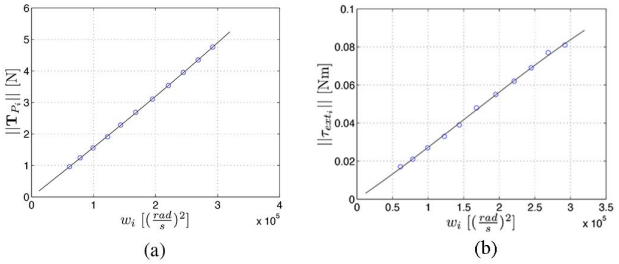


Fig. 9. (a) and (b) Dots: measured values of the thrust $\|\boldsymbol{T}_{P_i}\|$ and torque $\|\boldsymbol{\tau}_{\mathrm{ext}_i}\|$ versus the signed squared spinning velocity w_i. Black lines: identified polynomial model (25) and (26).

where $w_i = \bar{w}_i|\bar{w}_i|$ is the signed square of the propeller spinning velocity as previously explained. The controller (16) was then implemented by directly exploiting the mappings (25) and (26) for obtaining $(\|\boldsymbol{T}_{P_i}\|, \|\boldsymbol{\tau}_{\text{ext}_i}\|)$, and by replacing

$$k_f = \left.\frac{\partial\|\boldsymbol{T}_{P_i}\|}{\partial w_i}\right|_{w_i} \quad \text{and} \quad k_m = \left.\frac{\partial\|\boldsymbol{\tau}_{\text{ext}_i}\|}{\partial w_i}\right|_{w_i}$$

both evaluated upon the measured w_i.

B. System Architecture

The Q7-board runs a GNU-Linux Ubuntu 10.10 real-time OS and executes the MATLAB-generated code. The controller runs at 500 Hz and takes as inputs: 1) the desired trajectory $(\boldsymbol{p}_d(t), \boldsymbol{R}_d(t))$ and needed derivatives $(\dot{\boldsymbol{p}}_d(t), \ddot{\boldsymbol{p}}_d(t), \dddot{\boldsymbol{p}}_d(t))$ and $(\boldsymbol{\omega}_d(t), \dot{\boldsymbol{\omega}}_d(t), \ddot{\boldsymbol{\omega}}_d(t))$; 2) the current position/orientation of the holocopter $(\boldsymbol{p}, {}^W\boldsymbol{R}_B)$ and its linear/angular velocity $(\dot{\boldsymbol{p}}, \boldsymbol{\omega}_B)$; 3) the spinning velocities of the propellers w_i; and 4) the tilting angles α_i.

The position $\boldsymbol{p}$ and orientation ${}^W\boldsymbol{R}_B$ of the holocopter are directly obtained from an external motion capture system[10] (MoCap) at 200 Hz. A marker tree consisting of five infrared markers is mounted on top of the holocopter for this purpose. Knowing $\boldsymbol{p}$, the linear velocity $\dot{\boldsymbol{p}}$ is then obtained via numerical differentiation, while the angular velocity $\boldsymbol{\omega}_B$ is measured by the onboard Inertial Measurement Unit (IMU) (three ADXRS610 gyroscopes).

Because of performance reasons (bottleneck in serial communication), the sending of the desired motor speeds and tilting angles, and the reading of the IMU-data, of the actual spinning velocities, and of tilting angles is split among two communication channels and two microcontrollers (called, from now on, μC-Board and IMU-Board). The desired motor spinning velocities w_{Des_i} are sent from the Q7-board to the μC-Board via a serial connection at the frequency of 250 Hz and 8 b resolution, and from the μC-Board to the brushless controllers via I^{2}C-bus at again 250 Hz. In the same manner, the actual spinning velocities w_i of the four propellers can be read back at a frequency of 250 Hz and a resolution of 8 b. The brushless controllers implement a PID-controller for regulating the spinning velocity. The desired tilting angles α_{Des_i} are sent from the Q7-board to the μC-Board via the same serial connection at a frequency of 55 Hz and 10 b resolution, and from the μC-Board to the servo motors via PWM (signal length 15 ms). We note that the trajectory tracking controller described in Section III assumes availability of the tilting velocities w_{α_i} as inputs [see (16)], whereas the current architecture only allows for sending desired angles commands $\alpha_{\text{Des}_i}(t)$. This is addressed by numerically integrating over time the controller commands w_{α_i}, that is, by implementing

$$\alpha_{\text{Des}_i}(t) = \int_{t_0}^{t} w_{\alpha_i}(\tau)\mathrm{d}\tau + \alpha_i(t_0) \tag{27}$$

where $\alpha_i(t_0)$ is the ith measured tilting angle at the beginning of motion.[11]

[10] http://www.vicon.com/products/bonita.html

[11] We note that, in order to avoid possible numerical drifts, one could also exploit the measured $\alpha_i(t)$ for resetting (27) when needed.

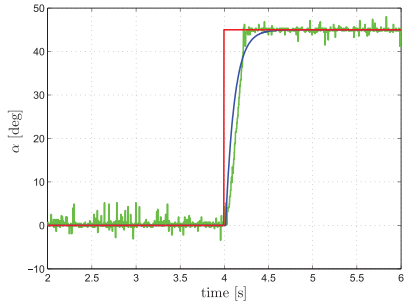


Fig. 10. Modeling of the servo motor. Behavior of the real servo motor (green) and the model (blue) following a step input (red) of 45° after compensating for the (known) transport delay $T = 18$ ms.

The IMU-Board reads the current angles α_i of the propeller groups P_i by a direct connection between the servo motor potentiometer and the A/D-converter of the microcontroller (10 b resolution at 250 Hz). It also retrieves the current spinning velocities $\bar{w}_i$ of the propellers via the I^{2}C-Bus (8 b resolution and 250 Hz). The gyro data are read at 250 Hz and converted with 10 b resolution. Finally, the values of α_i, $\bar{w}_i$, and the gyro data are transmitted from the IMU-Board to the Q7-board via the RS232-port at 250 Hz. All values of the controller can be monitored on a remote Windows PC which mirrors the running controller in real-time using the MATLAB/simulink external mode. This simplifies the development as most of the gains and settings can be changed online during flight tests.

The communication architecture for the tilting angles α_{Des_i} (in particular, the PWM modulation) unfortunately introduces a nonnegligible roundtrip delay of about 18 ms form sent commands to read values. We experimentally found this delay to significantly degrade the closed-loop performance of the controller, and therefore propose in Section IV-C a simple prediction scheme for mitigating its adverse effects.

C. Coping With the Nonidealities of the Servo Motors

The ith servo motor for the tilting angles can be approximately modeled as a linear transfer function $G(s)$ with, in series, a transport delay of $T = 18$ ms, that is, as the delayed linear system $\alpha_i(s) = G(s)e^{-Ts}\alpha_{\text{Des}_i}(s)$. A model of the undelayed $G(s)$ was experimentally obtained by measuring the step response of the servo motors while having the propellers spinning at $\bar{w}_i = 450$ rad/s (the velocity corresponding to hovering), and by compensating offline for the known delay T (see Fig. 10). This resulted into the estimated transfer function

$$G_{\text{est}}(s) = \frac{0.4s + 6}{0.06s^2 + s + 6}. \tag{28}$$

The performance degradation of the cartesian trajectory controller (16)–(23) can then be due to two main effects, namely presence of the transport delay T and slow dynamic response of $G_{\text{est}}(s)$ to fast changing inputs. To mitigate these shortcomings, we resorted to the following simple strategy (see Fig. 11); instead of feeding back the measured (i.e., delayed) angles α_i to the cartesian controller (16)–(23), we replaced them with the (undelayed) desired angles α_{Des_i} from (27). In parallel, we aimed at improving the servo

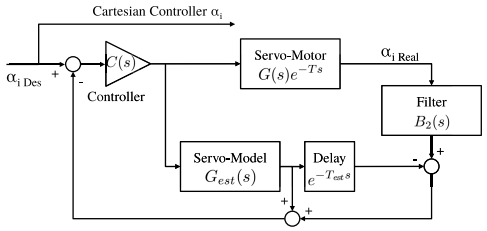


Fig. 11. Scheme of the Smith predictor for α_i including the controller $C(s)$, the servo motor $G(s)e^{-Ts}$, the model of the servo motor $G_{\mathrm{est}}(s)e^{-Ts}$, and the Butterworth filter $B_2(s)$.

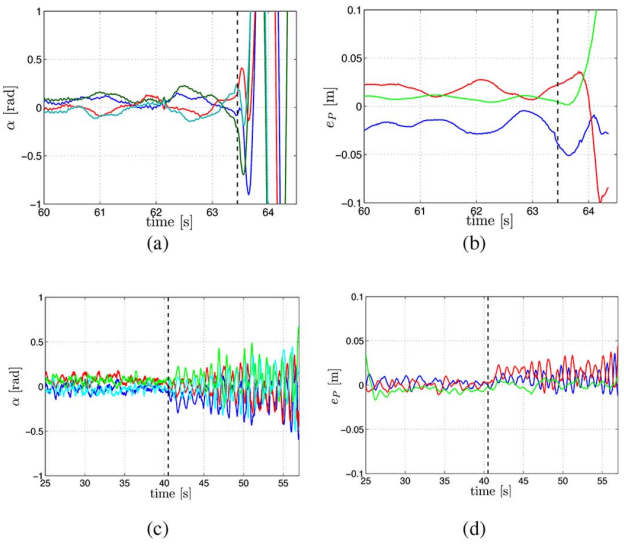


Fig. 12. Results of the experiments assessing the effectiveness of the scheme in Fig. 11. In all the plots, the vertical dashed black line indicates the time t_{off} after which the scheme is switched off. (a) Behavior of the tilting angles $\boldsymbol{\alpha}(t)$ during case I. (b) Position-tracking error $\boldsymbol{e}_P(t)$ during case I. (c) Behavior of tilting angles $\boldsymbol{\alpha}(t)$ during case II. (d) Position-tracking error $\boldsymbol{e}_P(t)$ during case II. Note how in both cases the holocopter becomes unstable for $t \geq t_{\mathrm{off}}$, thus confirming the validity of the scheme in Fig. 11 in coping with the nonidealities of the employed servo motors.

motor performance (i.e., making $G_{\mathrm{est}}(s)$ more responsive) by resorting to a Smith predictor scheme [39]. In fact, as well known from classical control theory, the Smith predictor is an effective tool for coping with known delays affecting known stable linear systems. In our case, an additional outer PID controller $C(s)$ plugged into the Smith predictor loop (as shown in Fig. 11), allowed to improve the rising time of the servo controller. Finally, because we found the measured angles α_i to be affected by significant noise, we filtered their readings with the second-order Butterworth filter with a cutoff frequency of 20 Hz. The location of this cutoff frequency was experimentally determined by analyzing offline the power spectrum of the angles α_i recorded during a hovering flight of 40 s.

As experimental proof of the effectiveness of the aforementioned strategy in coping with the nonidealities of the holocopter actuation system, we ran the following two illustrative experiments involving a simple hovering on the spot task. In both experiments, the scheme shown in Fig. 11 was activated for an initial period $0 \leq t \leq t_{\mathrm{off}}$ after which the scheme was instead switched off: for $t \geq t_{\mathrm{off}}$ the trajectory controller was then fed back with the measured (and thus delayed) angles $\alpha_i(t)$ (case I), and with the desired angles $\alpha_{\mathrm{Des}_i}(t)$ (case II).

The results are shown in Fig. 12(a)–(d). We first note how, in both cases, the hovering task is correctly realized during $0 \leq t \leq t_{\mathrm{off}}$, that is, when employing the scheme shown in Fig. 11. Then, in case I the holocopter becomes unstable almost instantaneously for $t > t_{\mathrm{off}}$, while in case II the servo motors start to slowly oscillate to then reach practical instability at about $t > t_{\mathrm{off}} + 15$ s. These results allow us to then conclude the ability of the proposed strategy to cope with the shortcomings of the holocopter actuation system.

TABLE III
MEAN SPINNING VELOCITY $\bar{\boldsymbol{w}}$ OF THE FOUR PROPELLERS FOR DIFFERENT HOVERING ORIENTATIONS

Pose	Mean spinning velocity $Avg(\bar{\boldsymbol{w}})$
$\theta = 0\ rad, \phi = 0\ rad, \psi = 0\ rad$	$450\ \frac{rad}{s}$
$\theta = \frac{\pi}{4}\ rad, \phi = 0\ rad, \psi = 0\ rad$	$504\ \frac{rad}{s}$
$\theta = \frac{\pi}{2}\ rad, \phi = 0\ rad, \psi = 0\ rad$	$465\ \frac{rad}{s}$
$\theta = \frac{\pi}{2}\ rad, \phi = 0\ rad, \psi = \frac{\pi}{4}\ rad$	$530\ \frac{rad}{s}$

D. Energetic Efficiency of the Holocopter in Hovering

As a final consideration, we briefly discuss the energetic efficiency of the holocopter in a hovering condition. Indeed, we note that the holocopter energetic efficiency for arbitrary hovering orientations can be less than in the (standard) horizontal case ($\phi = 0,\ \theta = 0$), and also despite the optimization action (23). This is due to the adopted mechanical design which allows each propeller to only rotate about one tilting angle (angles α_i): therefore, there will exist hovering orientations at which the thrust vectors $\boldsymbol{T}_{P_i}$ cannot be aligned against gravity, with thus some of the thrust lost in internal forces. This reduced efficiency cannot be completely avoided with the adopted design, but only partially mitigated via optimization actions such as (23). In this sense, Table III gives an illustration of the energetic efficiency of our prototype (in terms of the mean spinning velocity of the four propellers) obtained at several hovering conditions (including the horizontal one).

V. SIMULATION RESULTS

We now report some simulation results aimed at validating the proposed control design. We first consider in Section V-A an ideal case in which controller (16) is tested on the complete holocopter dynamical model described in Section II, that is, by including all the inertial/gyroscopic effects neglected at the control design stage. These results are meant to illustrate the flying performance of the holocopter in ideal conditions, that is, when not considering all the limitations and nonidealities affecting the real prototype as in Section IV-B.

Subsequently, we present in Section V-B an additional set of realistic simulations that explicitly include the prototype main nonidealities (data-exchange rates, control frequencies, and actuation delays). In these simulations, the controller (16) is also complemented with the prediction scheme described in Section IV-C so as to replicate, as much as possible, the control

architecture of the real prototype. The results are intended to show the robustness of the adopted control approach as the holocopter is still able to execute complex trajectories although with a poorer flight performance compared with the ideal case (as expected). Furthermore, a comparison between both cases clearly shows the margin left for improving the actuation system of our prototype in its second generation (see Section VII).

A. Ideal Simulations

The aim of the following simulations is twofold: on one side, we want to highlight the tracking capabilities of the proposed controller and the beneficial action of the null-space term (19) in avoiding singularities for the decoupling matrix $\boldsymbol{A}(\boldsymbol{\alpha}, \boldsymbol{\omega})$. On the other side, we also want to show the robustness of the controller against all the inertial/gyroscopic effects neglected at the control design stage but included in the quadrotor dynamic model $(1-5)$.

1) Rotation on Spot: In this first simulation, we tested a simple trajectory involving a rotation of π rad on the spot along the $\boldsymbol{Y}_B$ axis.[12] The initial conditions were set to $\boldsymbol{p}(t_0) = \boldsymbol{0}$, $\dot{\boldsymbol{p}}(t_0) = \boldsymbol{0}$, $\boldsymbol{R}(t_0) = \mathbf{I}_3$, $\boldsymbol{\omega}_B(t_0) = \boldsymbol{0}$, $\boldsymbol{\alpha}(t_0) = \boldsymbol{0}$, $\dot{\boldsymbol{\alpha}}(t_0) = \boldsymbol{0}$, and $\boldsymbol{w}(t_0) = w_{\text{rest}}$. The desired trajectory was chosen as $\boldsymbol{p}_d(t) \equiv 0$ and $\boldsymbol{R}_d(t) = \boldsymbol{R}_X(\phi(t))$ with $\phi(t)$ following a smooth profile with maximum velocity $\dot{\phi}_{\max} = 0.49$ rad/s and maximum acceleration $\ddot{\phi}_{\max} = 0.16$ rad/s^2. The trajectory was executed twice by: 1) including and 2) not including the null-space term $\boldsymbol{z}$ (23) into (16) ($k_H = 1$ or $k_H = 0$). The gains in (18)–(20) were set to $\boldsymbol{K}_{p_1} = 28.5\mathbf{I}_3$, $\boldsymbol{K}_{p_2} = 271\mathbf{I}_3$, $\boldsymbol{K}_{p_3} = 857\mathbf{I}_3$ and $\boldsymbol{K}_{\omega_1} = 45\,\mathbf{I}_3$, $\boldsymbol{K}_{\omega_2} = 675\,\mathbf{I}_3$, and $\boldsymbol{K}_{\omega_3} = 3375\mathbf{I}_3$.

Fig. 13(a)–(f) shows the results of the simulation in these two cases. In particular, Fig. 13(a) shows the superimposition of $H(\boldsymbol{w})$ when including $\boldsymbol{z}$ [red dashed line, case (i)] and not including $\boldsymbol{z}$ [blue solid line, case (ii)]. It is clear that, in the first case, $H(\boldsymbol{w})$ attains a lower value over time thanks to the optimization action in (23). As a consequence, this results in a lower value for $\|\boldsymbol{w}\|$ over time as shown in Fig. 13(b) (same color pattern), showing that the given task (rotation on the spot) can be realized in a more energy-efficient way when properly shaping the cost function $H(\boldsymbol{w})$. Note that, as a byproduct, the better performance of case (i) comes at the expense of a more complex reorientation of the propeller groups during the motion. This is shown in Fig. 13(c) and (d) which report the behavior of the four tilt angles α_i in cases (i) (left) and (ii) (right): compared with Fig. 13(d), note the rotation of two propellers starting from $t \approx 25.8$ $[s]$ in Fig. 13(c).

Finally, Fig. 13(e) and (f) shows, for case (i) only, the position-tracking error $\boldsymbol{e}_P(t) = \boldsymbol{p}_d(t) - \boldsymbol{p}(t)$ and orientation-tracking error $\boldsymbol{e}_R(t)$ as defined in (19). Despite the fast reorientation of two propellers shown in Fig. 13(c), the tracking errors stay small (note the scales) and eventually converge to zero as the desired trajectory comes to a full stop. Also, as one would expect, the largest peaks for the errors

[12]This upside-down flip motion would be clearly unfeasible for a standard quadrotor. It is (unfortunately) also unfeasible for the current experimental prototype because of the mechanical end stops in the propeller tilting actuation (see Section IV-A).

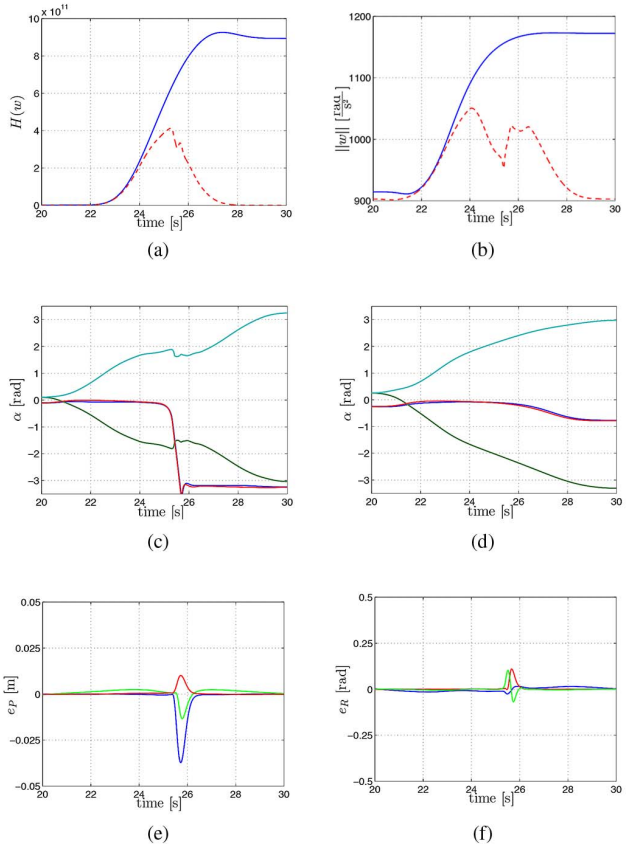


Fig. 13. Results of the first ideal simulation with (i) and without (ii) exploiting the null-space term (23). (a) Behavior of $H(\boldsymbol{w})$ for cases (i) (red dashed line) and (ii) (blue solid line). (b) Behavior of $\|\boldsymbol{w}\|$ for cases (i) (red dashed line) and (ii) (blue solid line). (c) and (d) Behavior of the tilt angles $\boldsymbol{\alpha}$ for cases (i) (left) and (ii) (right). (e) and (f) Behavior of the position-/orientation-tracking errors $(\boldsymbol{e}_P, \boldsymbol{e}_R)$ for case (ii).

$\boldsymbol{e}_P(t)$ and $\boldsymbol{e}_R(t)$ occur at about $t \approx 25.8$ $[s]$, that is, during the fast reorientation of the two propellers because of the internal gyroscopic effects treated as external disturbances by the controller.

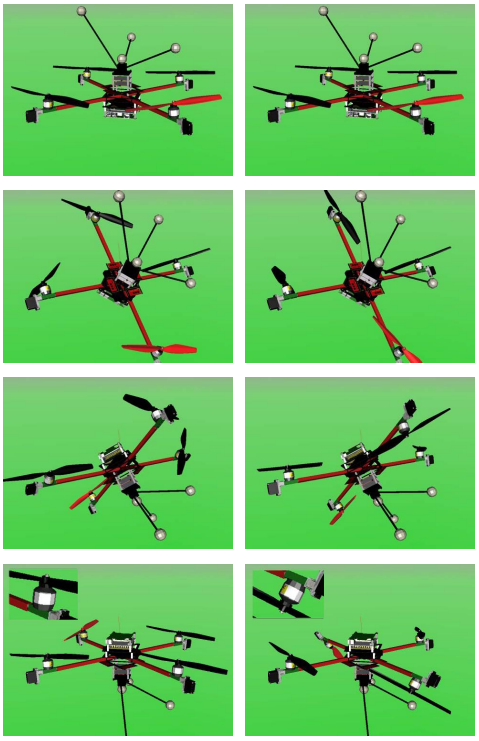
These results then provide a first confirmation of the validity of our assumptions in Section III, that is, robustness of the controller with respect to the gyroscopic/inertial effect due to the internal relative motion of the different bodies composing the quadrotor. For the reader's convenience, we also report in Fig. 14(a) and (b) a series of snapshots illustrating the quadrotor motion in these two cases [note the very different final configuration of the propeller group in cases (i) and (ii)].

2) Eight-Shape Trajectory: In this second simulation, the holocopter task is to track a planar eight-shape trajectory $\boldsymbol{p}_d(t)$ while, at the same time, performing a sinusoidal rotation around the $\boldsymbol{Y}_B$ axis. The chosen desired trajectory $\boldsymbol{p}_d(t)$ is a horizontal eight-shape with size of 1.0 m by 1.4 m and lying at a height of $z = 1.0$ m from ground, that is

$$\boldsymbol{p}_d(t) = \begin{bmatrix} 0.5\sin(0.135t) \\ 0.7\sin(0.27t) \\ 1 \end{bmatrix} [m] \tag{29}$$

[see Fig. 15(a)].

As for the rotation about $\boldsymbol{Y}_B$, Fig. 15(b) shows the chosen profile for the pitch angle $\theta(t)$. The main quantities of interest



(a) (b)

Fig. 14. Results of the first ideal simulation. Left column: holocopter motion while minimizing $H(\boldsymbol{w})$ [case (i)]. Right column: holocopter motion without minimizing $H(\boldsymbol{w})$ [case (ii)]. Note the large reorientation of the propeller groups in case (i) which, thanks to the action of the optimization term (23), end up in a full upright position (minimum energy consumption) with respect to case (ii).

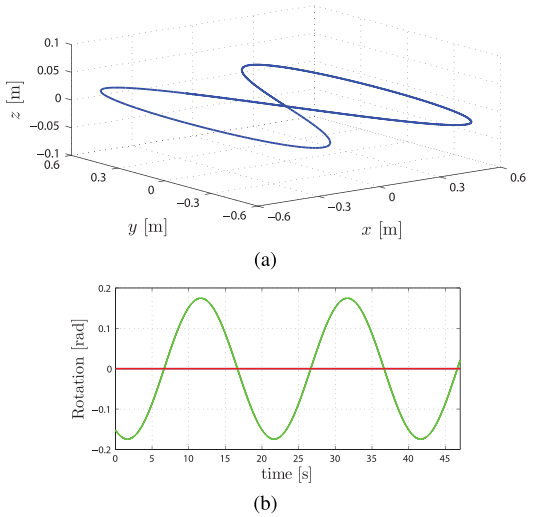
are in this case:

1) maximum speed along the trajectory: $v_{\max} = 0.20\ m/s$;
2) maximum acceleration along the trajectory: $a_{\max} = 0.05\ m/s^2$;
3) amplitude of the sinusoidal rotation: $\theta_{\max} = 0.17$ rad;
4) maximum rotational velocity: $\dot{\theta}_{\max} = 0.05$ rad$/s$;
5) maximum rotational acceleration: $\ddot{\theta}_{\max} = 0.02$ rad$/s^2$.

Fig. 16(a) and (b) shows the position- and orientation-tracking errors ($\boldsymbol{e}_P(t)$, $\boldsymbol{e}_R(t)$) while performing the task, with maximum values of $e_{P\max} \simeq 0.28$ mm and $e_{R\max} \simeq 0.0035$ rad. Again, these errors can be ascribed to the unmodeled internal gyroscopic and inertial forces caused by the propeller group rotations with respect to the body frame. The behavior of $\boldsymbol{\alpha}(t)$ is also shown in Fig. 16(c). The results of the simulation demonstrate again the tracking abilities of the proposed controller as the position and orientation errors keep very small values while following this more complex trajectory.

B. Realistic Simulations

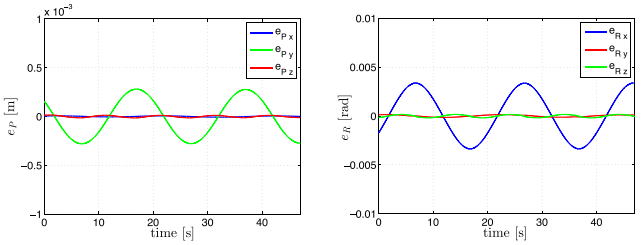
As explained, the realistic simulations of this section have been obtained by including in the holocopter model all the nonidealities of our real prototype, in particular by replicating the various signal and control frequencies (e.g., motion capture frequency at 200 Hz, servo motor frequency at 70 Hz) and delays (e.g., servo motor delay of 18 ms). In addition, we also



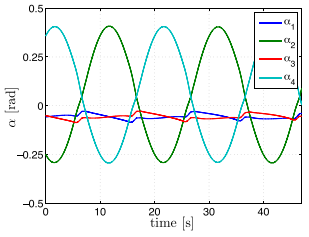
(a)

(b)

Fig. 15. Results of the second ideal simulation. (a) Desired eight-shape trajectory. (b) Desired sinusoidal orientation over time, red line indicates desired pitch angle θ.



(a) (b)



(c)

Fig. 16. Results of the second ideal simulation. (a) Position-tracking error $\boldsymbol{e}_P(t)$. (b) Orientation-tracking error $\boldsymbol{e}_R(t)$. (c) Behavior of the tilt angles $\boldsymbol{\alpha}(t)$ while tracking the trajectory.

considered the noise of onboard sensor readings (gyroscopes, accelerometers, and measurement of angles $\boldsymbol{\alpha}$), of the actuators (servo and propeller motors), and of the motion capture system by either obtaining the noise characteristics from data sheets, or via a preliminary off-line identification. Finally, the prediction scheme described in Section IV-C was implemented in the control loop.

Despite assessing the effects of these nonidealities on the overall flight performance, the results of this section are also helpful for saving development time as they allow a pretuning of the various control gains for the deployment on the real prototype, and the identification of the most influential parameters to be optimized in view of the second-generation prototype.

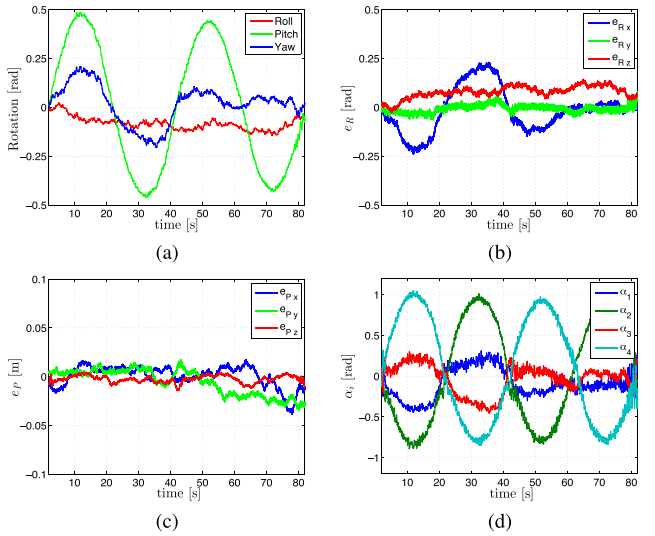


Fig. 17. Results of the first realistic simulation. Rotation on the spot around the $\boldsymbol{Y}_B$-axis. (a) Orientation of the main body B. (b) and (c) Orientation-tracking error $\boldsymbol{e}_R(t)$ and position-tracking error $\boldsymbol{e}_P(t)$. (d) Behavior of the tilting angles $\boldsymbol{\alpha}(t)$.

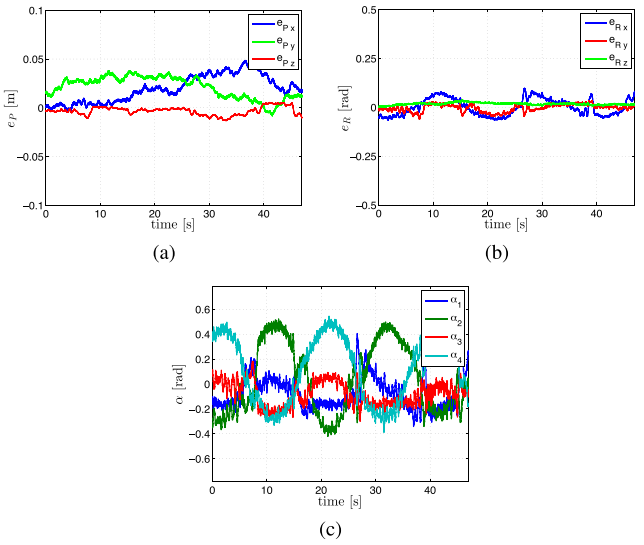


Fig. 18. Results of the second realistic simulation. (a) Position-tracking error $\boldsymbol{e}_P(t)$. (b) Orientation-tracking error $\boldsymbol{e}_R(t)$. (c) Behavior of the tilting angles $\boldsymbol{\alpha}(t)$.

1) Rotation on Spot: Because of the limited range of the tilting angles $\boldsymbol{\alpha}$ in the real prototype, it is not possible to perform a full rotation on the spot as in the previous ideal case explained in Section V-A 1. Therefore, we opted for a (more feasible) sinusoidal rotation around the $\boldsymbol{Y}_B$-axis (pitch), that is, with $\boldsymbol{R}_d(t) = \boldsymbol{R}_Y(\theta(t))$ with $\theta_{\max} = 0.436$ rad and $\dot{\theta}_{\max} = 0.07$ rad/s. The initial conditions were set to hovering ($\boldsymbol{p}(t_0) = \boldsymbol{0}$, $\dot{\boldsymbol{p}}(t_0) = \boldsymbol{0}$, $\boldsymbol{R}(t_0) = \mathbf{I}_3$, $\boldsymbol{\omega}_B(t_0) = \boldsymbol{0}$, $\boldsymbol{\alpha}(t_0) = \boldsymbol{0}$, and $\boldsymbol{w}(t_0) = w_{\text{rest}}$), and the controller gains were chosen as $\boldsymbol{K}_{p_1} = 30\mathbf{I}_3$, $\boldsymbol{K}_{\omega_1} = 55.5\mathbf{I}_3$, $\boldsymbol{K}_{p_2} = 300\mathbf{I}_3$, $\boldsymbol{K}_{\omega_2} = 1027\mathbf{I}_3$, $\boldsymbol{K}_{p_3} = 1000\mathbf{I}_3$, $\boldsymbol{K}_{\omega_3} = 6331\mathbf{I}_3$.

The results are shown in Fig. 17(a)–(d); Fig. 17(a) shows the behavior of the quadrotor orientation during flight (blue: roll, green: pitch, red: yaw) and Fig. 17(b) the behavior of the orientation error $\boldsymbol{e}_R(t)$. The maximum rotation errors are 0.240 rad (roll), 0.079 rad (pitch), and 0.144 rad (yaw). Fig. 17(c) shows the behavior of the position-tracking error $\boldsymbol{e}_P(t)$ characterized by a mean value $\text{avg}(\|\boldsymbol{e}_P(t)\|) \approx 1.6$ cm and a maximum value $\max(\|\boldsymbol{e}_P(t)\|) \approx 4.4$ cm. Finally, Fig. 17(d) shows the behavior of the tilting angles $\boldsymbol{\alpha}(t)$ over time.

Therefore, despite the (expected) worse overall performance with respect to the ideal case, in this realistic case the holocopter is still able to fulfill the assigned motion task with a good enough accuracy.

2) Eight-Shape Trajectory: In this simulation, we considered the same eight-shape trajectory illustrated in Section V-A 2. Fig. 18(a) and (b) shows the behavior of the position and orientation error vectors ($\boldsymbol{e}_P(t)$, $\boldsymbol{e}_R(t)$) over time. The average position-tracking error results about 0.034 m with a maximum of 0.050 m. The maximum rotation errors are 0.097 rad (roll), 0.061 rad (pitch), and 0.039 rad (yaw). Finally, Fig. 18(c) shows the behavior of the tilting angles $\alpha_i(t)$ while following the trajectory.

It is interesting to compare these results with those of the ideal simulation described in Section V-A.2 [Fig. 16(a)–(c)]; again, the overall tracking performance results degraded with respect to the ideal case, although the holocopter can still realize the task with a sufficient accuracy. Also, note how the angles $\alpha_i(t)$ shown in Fig. 18(c) follow essentially the same behavior as those of the ideal case shown in Fig. 16(c) despite the higher noise level present in the system.

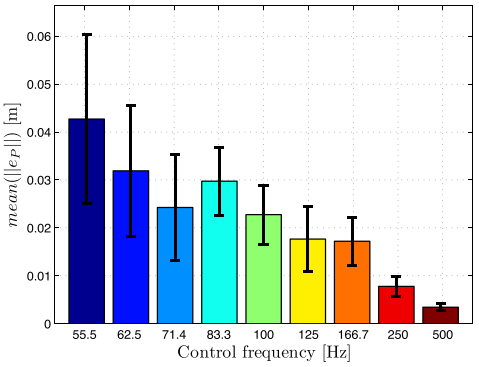


Fig. 19. Mean position error $\text{avg}(\|\boldsymbol{e}_P\|)$ and standard deviation of the position error $\text{stdev}(\|\boldsymbol{e}_P\|)$ for different PWM control frequencies of the servo motors while following the eight-shape trajectory reported in Section V-A 2.

3) Effect of Servo Control Frequency: As previously stated, the low control rate and the delayed response of the employed servo motors are expected to be the main cause of the flight performance degradation in the realistic case (and, of course, in the real prototype). The servo motors are controlled via a pulse-width-modulated (PWM) signal with a signal length of 14 ms and a control frequency of 55 Hz. To assess the effects of these parameters, we ran several instances of the previous realistic simulation described in Section V-B.2 by employing increasing control frequencies and correspondingly decreasing delays for the servo motor PWM while keeping all the other parameters (e.g., control gains) constant.

As flight performance measure, we considered the mean position error $\text{avg}(\|\boldsymbol{e}_P(t)\|)$ and the standard deviation of the position error $\text{stdev}(\|\boldsymbol{e}_P(t)\|)$ during the trajectory. Fig. 19 shows the results: with an increasing control frequency,

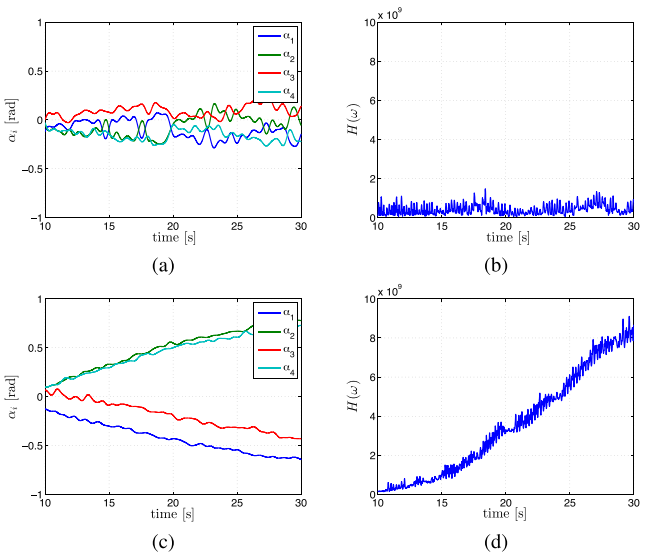


Fig. 20. First experiment. Results for hovering on the spot with (i) and without (ii) including the null-space term (23). (a) α_i for case (i) while hovering. (b) $H(\boldsymbol{w})$ for case (i) while hovering. (c) α_i for case (ii) while hovering. (d) $H(\boldsymbol{w})$ for case (ii) while hovering.

the mean position error and the standard deviation are clearly decreasing from, for example, $\text{avg}(\|\boldsymbol{e}_P\|)_{55\text{ Hz}} = 0.042$ m to $\text{avg}(\|\boldsymbol{e}_P\|)_{500\text{ Hz}} = 0.003$ m, thus approaching the performance of the ideal case.

VI. Experimental Results

In this last section, we finally present results from three experiments conducted with the holocopter prototype. The first experiment is a hovering task meant to show the overall performance in the simplest scenario, and also to highlight again the importance of having included the null-space optimization term (23) in the control strategy. The other two experiments involve instead the more complex trajectories illustrated in Sections V-B.1 and V-B.2. Finally we encourage the reader to watch the video clip attached to this paper where several holocopter maneuvers are shown along with additional details on the employed prototype.

A. Hovering on the Spot

In the first experiment, following what was done in the ideal case explained in Section V-A.1, we show the importance of having included the minimization of the cost function $H(\boldsymbol{w})$ in the proposed controller. To this end, we report the results of a simple hovering on the spot by (i) including and (ii) not including the null-space optimization term (23). The quadrotor starts from the initial state of $\boldsymbol{p}(t_0) = \boldsymbol{0}$, $\dot{\boldsymbol{p}}(t_0) = \boldsymbol{0}$, $\boldsymbol{R}(t_0) = \mathbf{I}_3$, $\boldsymbol{\omega}_B(t_0) = \boldsymbol{0}$, $\boldsymbol{\alpha}(t_0) = \boldsymbol{0}$, and $\boldsymbol{w}(t_0) = w_{\text{rest}}$, and is commanded to stay still while maintaining the desired attitude $\boldsymbol{R}_d = \mathbf{I}_3$. The gains in (18) and (20) were set to $\boldsymbol{K}_{p_1} = 30\mathbf{I}_3$, $\boldsymbol{K}_{\omega_1} = 55.5\mathbf{I}_3$, $\boldsymbol{K}_{p_2} = 300\mathbf{I}_3$, $\boldsymbol{K}_{\omega_2} = 1027\mathbf{I}_3$, $\boldsymbol{K}_{p_3} = 1000\mathbf{I}_3$, $\boldsymbol{K}_{\omega_3} = 6331\mathbf{I}_3$ (these values were experimentally tuned).

Fig. 20(a) and (b) shows the results for case (i): the angles α_i stay close to 0 rad over time, as expected for such a hovering maneuver, and $H(\boldsymbol{w})$ keeps a constant and low value as the propellers spin with a speed close to the allowed minimum. In case (ii), however, the situation looks completely different: the lack of any minimization action on $H(\boldsymbol{w})$, coupled with the presence of noise and nonidealities, makes the angles α_i to eventually diverge over time from their (expected) vertical direction and, accordingly, the value of $H(\boldsymbol{w})$ to increase as the propellers need to accelerate to keep the quadrotor still in place [Fig. 20(c) and (d)].

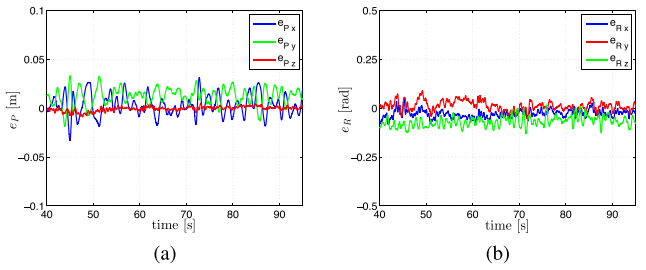


Fig. 21. First experiment. Tracking error while hovering (a) position-tracking error $\boldsymbol{e}_P(t)$ and (b) orientation-tracking error $\boldsymbol{e}_R(t)$.

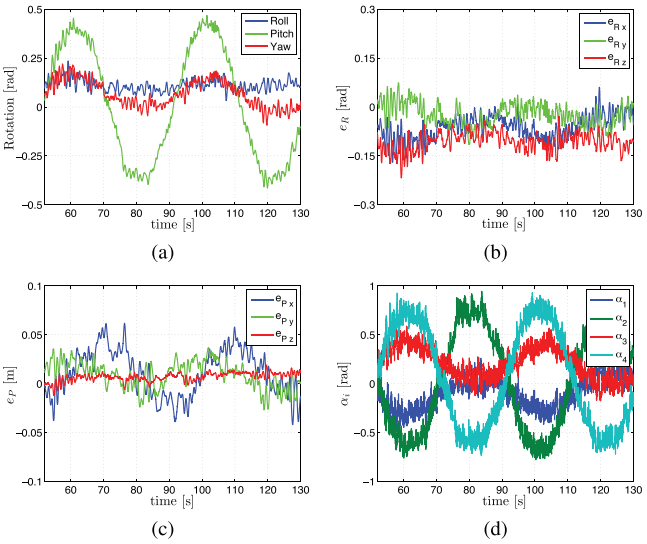


Fig. 22. Second experiment. Rotation on the spot around the $\boldsymbol{Y}_B$-axis. (a) Orientation of the main body B. (b) and (c) Orientation error vector $\boldsymbol{e}_R(t)$ and position error vector $\boldsymbol{e}_P(t)$. (d) Behavior of the tilting angles α_i.

Finally, Fig. 21(a) and (b) shows the position error $\boldsymbol{e}_P(t)$ and orientation error $\boldsymbol{e}_R(t)$ during the experiment. The average position-tracking error is about 0.017 m with a maximum of 0.047 m. The maximum rotation errors are 0.082 rad (roll), 0.131 rad (pitch), and 0.089 rad (yaw).

B. Rotation on the Spot

In this second experiment, we replicate the realistic simulation case explained in Section V-B.1 by commanding the holocopter to follow a given orientation profile $\boldsymbol{R}_d(t)$ while remaining still in space. The initial conditions, control gains, and trajectory parameters are the same as those reported in Section V-B.1.

Fig. 22(a)–(d) shows the results of the flight: in particular, Fig. 18(a) reports the quadrotor orientation during

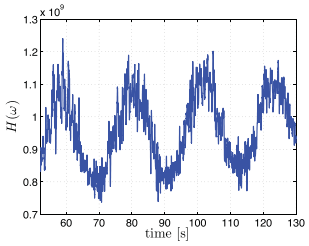


Fig. 23. Second experiment. Behavior of $H(\boldsymbol{w})$ while rotating on the spot.

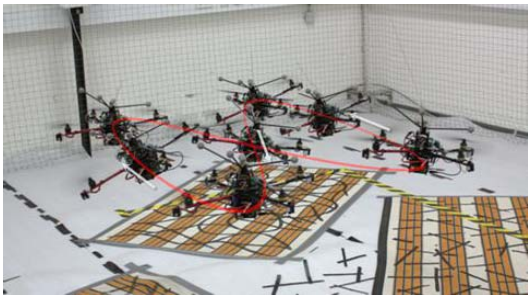


Fig. 24. Third experiment. Overlay of several snapshots of the holocopter while performing eight-shape trajectory.

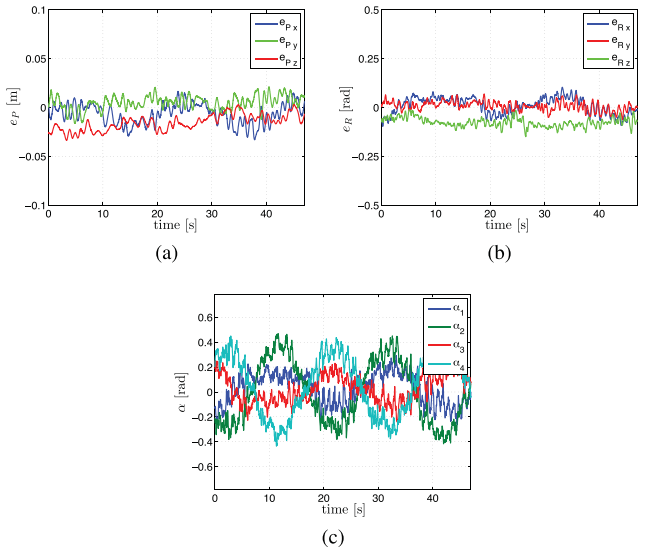


Fig. 25. Third experiment. (a) Position-tracking error e_P: x (blue), y (green), and z (red). (b) Orientation-tracking error e_R: roll (blue), pitch (red), and yaw (green). (c) Behavior of the tilting angles α_i.

flight (blue: roll, green: pitch, red: yaw), and Fig. 18(b) the orientation-tracking error $\boldsymbol{e}_R(t)$. The position-tracking error $\boldsymbol{e}_P(t)$ is shown in Fig. 18(c), with a maximum of $\max(\|\boldsymbol{e}_P(t)\|) = 0.062$ m. Finally, Fig. 18(d) shows the behavior of the tilting angles $\alpha_i(t)$ during the maneuver, and Fig. 23 shows the behavior of $H(\boldsymbol{w})$. As clear from the plots, this experiment involving a rotation on the spot still confirms the capabilities of the holocopter and the robustness of the proposed control strategy in coping with all the nonidealities of real-world conditions.

Fig. 26. CAD image of the new prototype with improved mechanics, actuation system, electronics, and an overall reduced mass and inertia.

C. Eight-Shape Trajectory

This last experiment shows the performance of the holocopter in tracking the same eight-shape trajectory with superimposed sinusoidal rotation described in Sections V-A.2 and V-B.2. Fig. 24 shows an overlay of several snapshots taken during flight.

Fig. 25(a) shows the position-tracking error $\boldsymbol{e}_P(t)$ of the holocopter while following the trajectory, while Fig. 25(b) shows the orientation-tracking error $\boldsymbol{e}_R(t)$. The maximum position error $\max(\|\boldsymbol{e}_P(t)\|)$ while following the path was approximately 3.9 cm, with $\text{avg}(\|\boldsymbol{e}_P(t)\|) \approx 2.2$ cm. The maximum orientation errors were 0.10 rad for roll, 0.06 rad for pitch, and 0.15 rad for yaw. Fig. 25(b) shows the behavior of the tilting angles $\alpha_i(t)$. Note how these experimental results match very well those of the realistic simulation described in Section V-B.2, thus also confirming the validity of the employed holocopter model. The interested reader is also appreciated to watch the execution of this task in the video attached to this paper.

VII. Conclusion

In this paper, we have presented a novel overactuated quadrotor UAV called holocopter able to achieve full controllability over its 6-dof body pose in space. This design overcomes the typical limitations of standard (underactuated) quadrotor UAVs where only 4 DoF can be independently controlled, namely the body position and yaw angle. The holocopter design, in contrast, allows to actively and individually control the tilting angles of its four propellers, thus granting a total of $4+4$ available control inputs which (as proven in this paper) yields full controllability over its pose. Several ideal/realistic simulations and experimental results have been reported and discussed to assess the effectiveness of the control strategy developed for the holocopter; despite the various nonidealities of the real prototype with respect to the modeling assumptions, a satisfactory performance has nevertheless been achieved when executing complex maneuvers which would be impossible for standard quadrotor UAVs.

Our future goal is to exploit the holocopter as a flying service robot capable of advanced interaction tasks with the environment. To this end, we are currently developing the second-generation prototype shown in Fig. 26 with the aim of overcoming the limitations of the first prototype discussed in this paper. In particular, we are focusing on improving the holocopter actuation system (see Section IV-C), as well as obtaining a reduced overall weight, better onboard

sensors, higher onboard computational power, and a more robust mechanical design. This will enable better tracking performance and allow for a full exploitation of the holocopter 6-dof motion capabilities in the planned interaction tasks with the environment.

ACKNOWLEDGMENT

Part of Heinrich Bülthoff's research was supported by the Brain Korea 21 PLUS Program through the National Research Foundation of Korea funded by the Ministry of Education. Correspondence should be directed to Heinrich H. Bülthoff.

REFERENCES

[1] S. Bouabdallah, M. Becker, and R. Siegwart, "Autonomous miniature flying robots: Coming soon!—Research, development, and results," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 88–98, Sep. 2007.

[2] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Eng. Pract.*, vol. 18, no. 7, pp. 691–699, 2010.

[3] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.

[4] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1837–1853, Aug. 2009.

[5] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles," *IEEE Control Syst. Mag.*, vol. 33, no. 1, pp. 61–75, Feb. 2013.

[6] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," in *Proc. Int. Symp. Experim. Robot.*, 2010.

[7] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 57–68, Sep. 2012.

[8] A. Kushleyev, D. Mellinger, and V. Kumar, "Towards a swarm of agile micro quadrotors," in *Proc. Robot., Sci. Syst.*, Jul. 2012.

[9] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor," in *Proc. Robot., Sci. Syst.*, Jun. 2013.

[10] *EU Collaborative Project ICT-248669 AIRobots* [Online]. Available: http://www.airobots.eu

[11] *EU Collaborative Project ICT-287617 ARCAS* [Online]. Available: http://www.arcas-project.eu

[12] L. Gentili, R. Naldi, and L. Marconi, "Modeling and control of VTOL UAVs interacting with the environment," in *Proc. 47th IEEE Conf. Decision Control (CDC)*, Dec. 2008, pp. 1231–1236.

[13] R. Naldi and L. Marconi, "Modeling and control of the interaction between flying robots and the environment," in *Proc. 8th Int. Fed. Autom. Control (IFAC) Symp. Nonlinear Control Syst. (NOLCOS)*, 2010, pp. 975–980.

[14] L. Marconi and R. Naldi, "Control of aerial robots: Hybrid force and position feedback for a ducted fan," *IEEE Control Syst. Mag.*, vol. 32, no. 4, pp. 43–65, Aug. 2012.

[15] P. E. I. Pound, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 2491–2498.

[16] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," in *Proc. Robot., Sci. Syst.*, Jun. 2011.

[17] R. Spica, A. Franchi, G. Oriolo, H. H. Bülthoff, and P. R. Giordano, "Aerial grasping of a moving target with a quadrotor UAV," in *Proc. IEEE Int. Conf. Intell. Robot. Syst. (IROS)*, Jul. 2012, pp. 4985–4992.

[18] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Proc. Robot., Sci. Syst.*, Jun. 2013.

[19] M. Manubens, D. Devaurs, L. Ros, and J. Cortés, "Motion planning for 6-D manipulation with aerial towed-cable systems," in *Proc. Robot., Sci. Syst.*, Berlin, Germany, Jun. 2013.

[20] M.-D. Hua, T. Hamel, and C. Samson, "Control of VTOL vehicles with thrust-direction tilting," in *Proc. 19th Int. Fed. Autom. Control (IFAC) World Congr.*, 2014.

[21] Y. Long and D. J. Cappelleri, "Linear control design, allocation, and implementation for the omnicopter MAV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 289–294.

[22] K. T. Oner, E. Cetinsoy, M. Unel, M. F. Aksit, I. Kandemir, and K. Gulez, "Dynamic model and control of a new quadrotor unmanned aerial vehicle with tilt-wing mechanism," in *Proc. World Acad. Sci., Eng. Technol.*, 2008, pp. 58–63.

[23] K. T. Oner, E. Cetinsoy, E. Sirimoglu, C. Hancer, T. Ayken, and M. Unel, "LQR and SMC stabilization of a new unmanned aerial vehicle," in *Proc. World Acad. Sci., Eng. Technol.*, Oct. 2009, pp. 554–559.

[24] R. Voyles and G. Jiang, "Hexrotor UAV platform enabling dextrous interaction with structures—Preliminary work," in *Proc. IEEE Int. Symp. Safety, Security, Rescue Robot. (SSRR)*, Nov. 2012, pp. 1–7.

[25] F. Kendoul, I. Fantoni, and R. Lozano, "Modeling and control of a small autonomous aircraft having two tilting rotors," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1297–1302, Dec. 2006.

[26] A. Sanchez, J. Escareño, O. Garcia, and R. Lozano, "Autonomous hovering of a noncyclic tiltrotor UAV: Modeling, control and implementation," in *Proc. 17th Int. Fed. Autom. Control (IFAC) Wold Congr.*, Jul. 2008, pp. 803–808.

[27] F. Forte, R. Naldi, A. Serrani, and L. Marconi, "Control of modular aerial robots: Combining under- and fully-actuated behaviors," in *Proc. IEEE 51st Annu. Conf. Decision Control (CDC)*, Dec. 2012, pp. 1160–1165.

[28] M. Ryll, H. H. Bülthoff, and P. R. Giordano, "Modeling and control of a quadrotor UAV with tilting propellers," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 4606–4613.

[29] R. Falconi and C. Melchiorri, "Dynamic model and control of an over-actuated quadrotor UAV," in *Proc. 10th Int. Fed. Autom. Control (IFAC) Symp. Robot. Control*, 2012, pp. 192–197.

[30] M. Ryll, H. H. Bülthoff, and P. R. Giordano, "First flight tests for a quadrotor UAV with tilting propellers," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 295–302.

[31] K. P. Valavanis, Ed., *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*. New York, NY, USA: Springer-Verlag, 2007.

[32] W. Stepniewski and C. Keys, Eds., *Rotary-Wing Aerodynamics*. New York, NY, USA: Dover, 1984.

[33] G. Fay, *Derivation of the Aerodynamic Forces for the Mesicopter Simulation*. Stanford, CA, USA: Stanford Univ. Press, 2001.

[34] A. Isidori, *Nonlinear Control Systems*, 3rd ed. New York, NY, USA: Springer-Verlag, 1995.

[35] A. De Luca, G. Oriolo, and P. R. Giordano, "Kinematic control of nonholonomic mobile manipulators in the presence of steering wheels," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2010, pp. 1792–1798.

[36] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 2520–2525.

[37] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. 49th IEEE Conf. Decision Control (CDC)*, Dec. 2010, pp. 5420–5425.

[38] A. De Luca, G. Oriolo, and B. Siciliano, "Robot redundancy resolution at the acceleration level," *Robotica*, vol. 4, no. 2, pp. 97–106, 1992.

[39] W. A. Wolovich, *Automatic Control Systems*. London, U.K.: Oxford Univ. Press, 1994.

Markus Ryll received the B.Sc. degree in mechatronics from Cooperative State University, Karlsruhe, Germany, in 2008, and the M.Eng. degree in medical engineering from the University of Applied Sciences, Ulm, Germany, in 2010, with the master's thesis carried out at the Karolinska Institute, Stockholm, Sweden. He is currently pursuing the Ph.D. degree with the Max Planck Institute for Biological Cybernetics, Tübingen, Germany.

His current research interests include unmanned aerial vehicles, robotics, and human–robot interaction tasks.

Heinrich H. Bülthoff (M'96) received the Ph.D. degree in biology from Eberhard Karls University, Tübingen, Germany, in 1980.

He was a Research Scientist with the Max Planck Institute for Biological Cybernetics, Tübingen, and the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, from 1980 to 1988. He was an Assistant, Associate, and Full Professor of Cognitive Science with Brown University, Providence, RI, USA, from 1988 to 1993, before becoming the Director of the Department for Human Perception, Cognition and Action at the Max Planck Institute for Biological Cybernetics and a Scientific Member of the Max Planck Society in 1993. He has been an Honorary Professor with Eberhard Karls University since 1996, and is an Adjunct Professor with the Korea University, Seoul, Korea. His current research interests include object recognition and categorization, perception and action in virtual environments, and human–robot interaction and perception.

Paolo Robuffo Giordano (M'08) received the M.Sc. degree in computer science engineering in 2001, and the Ph.D. degree in systems engineering in 2008, from the University of Rome "La Sapienza", Rome, Italy.

He spent one year as a Post-Doctoral with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany, from 2007 to 2008, and from 2008 to 2012 he was a Senior Research Scientist with the Max Planck Institute for Biological Cybernetics, Tübingen, Germany. Since 2012 he has been a CNRS Researcher with the Lagadic Group, Irisa and Inria Bretagne Atlantique, Rennes, France. His research interests span nonlinear control, robotics, planning, haptics and VR applications.

Appendix C

Annex to Chapter VIII

Active Structure From Motion: Application to Point, Sphere, and Cylinder

Riccardo Spica, *Student Member, IEEE*, Paolo Robuffo Giordano, *Member, IEEE*, and François Chaumette, *Fellow, IEEE*

***Abstract*—In this paper, we illustrate the application of a nonlinear *active* structure estimation from motion (SfM) strategy to three problems, namely 3-D structure estimation for 1) a point, 2) a sphere, and 3) a cylinder. In all three cases, an appropriate parameterization reduces the problem to the estimation of a single quantity. Knowledge of this estimated quantity and of the available measurements allows for then retrieving the full 3-D structure of the observed objects. Furthermore, in the point feature case, two different parameterizations based on either a planar or a spherical projection model are critically compared. Indeed, the two models yield, somehow unexpectedly, to different convergence properties for the SfM estimation task. The reported simulative and experimental results fully support the theoretical analysis and clearly show the benefits of the proposed active estimation strategy, which is in particular able to impose a *desired transient response* to the estimation error equivalent to that of a reference linear second-order system with assigned poles.**

***Index Terms*—Nonlinear estimation, structure from motion, visual servoing.**

I. Introduction

THE problem of structure from motion (SfM), i.e., how to recover the missing structure of the observed scene from images taken by a moving camera, is a very classical and well-studied topic in computer and robot vision. One solution is to rely on prior knowledge of the scene as, e.g., known size of a tracked object. Alternatively, one can exploit the possibility of observing the same scene from different points of view, and fuse together the acquired images with the known camera displacement among them. When processing consecutive images over time, a possibility is to treat SfM as a recursive/filtering task: images and camera motion can be elaborated *online* for obtaining an incremental estimation of the scene structure. Other approaches (e.g., bundle adjustment) rely, instead, on *global* optimization methods meant to solve SfM problems by processing altogether information acquired over an extended time period. A recent discussion about the pros/cons of both approaches in the context of visual SLAM can be found in [1].

Manuscript received November 25, 2013; revised May 26, 2014; accepted October 23, 2014. Date of publication November 18, 2014; date of current version December 3, 2014. This paper was recommended for publication by Associate Editor J. A. Castellanos and Editor D. Fox upon evaluation of the reviewers' comments.

R. Spica is with University of Rennes 1 at Irisa and Inria Rennes Bretagne Atlantique, 35042 Rennes, France (e-mail: riccardo.spica@irisa.fr).

P. Robuffo Giordano is with CNRS at Irisa and Inria Rennes Bretagne Atlantique, 35042 Rennes, France (e-mail: prg@irisa.fr).

F. Chaumette is with Inria at Irisa and Inria Rennes Bretagne Atlantique, 35042 Rennes, France (e-mail: francois.chaumette@irisa.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2014.2365652

Within the first class of (recursive) methods, a vast literature exists for addressing SfM: for instance, as a nonexhaustive list, extended Kalman filter (EKF)-based solutions have been proposed in [2]–[5] and, along similar lines, an unscented Kalman filter was exploited in [6]. All these strategies have the considerable advantage of being "aware," to some extent, of the measurement and process noise (when modeled as Gaussian distributions). On the other hand, they require a certain level of approximation of the system dynamics, which may affect the estimation performance. Other approaches exploiting tools from (deterministic) nonlinear observation can instead be found in [7]–[14], while [15] has tackled the more challenging problem of structure *and* motion estimation, i.e., how to simultaneously recover the missing structure along with the (partially) unknown camera velocity. This class of methods does not typically involve any linearization of the system dynamics, and allows for some formal characterization of the estimation error convergence. However, the presence of noise is not explicitly taken into account, with the filter design being developed in a fully deterministic setting. A recent experimental comparison of an EKF solution versus a deterministic nonlinear filter in the context of SfM for a quadrotor UAV can also be found in [16].

While all these works study the general issue of structure estimation from motion in different contexts, much less attention has been devoted to the problem of *actively imposing* a desired (e.g., optimized) convergence behavior to an SfM estimation task by acting on the *motion imposed to the camera* and on the *employed estimation gains*. For instance, in [17], an active strategy for minimizing the effects of image noise and discretization errors was proposed and experimentally tested, but without the aim of *also* imposing a desired estimation transient response. In [18] the problem of actively selecting which features to track for improving the indoor localization of a wheeled mobile robot is successfully addressed; however, no attempt is made to actively shape the robot motion so as to optimize the SfM convergence (the robot navigates in an "uninformed" way with respect to the estimation task). In [19] an EKF-based SfM estimation scheme for a UAV is integrated with a path-planning strategy aiming at minimizing the covariance matrix of the estimated states at the end of the motion. Nevertheless, one needs to assume full preknowledge of the surrounding environment (e.g., obstacles) so as to numerically propagate the EKF filter along all the edges of a randomly constructed roadmap (the method is, thus, only amenable for an offline/planning use).

With respect to this state-of-the-art, this paper then tackles the problem of designing an *online* and *active* algorithm for structure from *known* and *controlled* motion, i.e., assuming that the camera velocity can be measured and actively modified

by the robot actuators (as it is often the case in robotic applications). The active component of the scheme makes it possible to impose an estimation error transient response equivalent to that of a reference linear second-order system with *desired* poles. The developments build upon the theoretical framework presented in [20]: in a nonlinear context, the observability properties of the states under consideration are not (in general) time-invariant but may depend on the current state and on the *current inputs* applied to the system. It is then possible to simultaneously act on the estimation gains and system inputs (i.e., the camera velocity for an SfM) in order to optimize the observation process.

The methodology proposed in [20] can be applied to all those systems in which an *invertible function* of the unknown states can appear linearly in the system dynamics, as it is indeed the case for SfM problems. We then exploit this fact and propose three concrete active SfM applications: 3-D structure estimation for 1) a point feature, 2) a spherical target, and 3) a cylindrical target. The estimation of the depth of a point feature has already been well studied in the past literature (see, e.g., [2], [7], [9], [11], [21], [22]), although by never considering the *active* perspective taken in this study. On the other hand, the machinery proposed for the spherical and cylindrical objects represents a novel contribution also in terms of the chosen parameterization. Indeed, we show that a suitable transformation of the observed features allows us to express the 3-D sphere/cylinder structures in terms of image measurements and of only one unknown *constant* parameter (the sphere/cylinder radius) rather than the classical (and time-varying)-scaled orientation of the limb surface in the camera frame. This, of course, significantly simplifies the SfM task.

We conclude by highlighting that the ability of both characterizing and optimizing the transient response of the estimation error brings several added values compared with more classical inactive estimation strategies: for instance, it allows obtaining the "best" estimation error convergence when subject to real-world constraints, such as limited camera velocity or upper bounds on the estimation gains due to noise, discretization, or other typical nonidealities. Furthermore, from a more theoretical perspective, the proposed methodology can also be used to get insights into the optimal camera trajectories needed to estimate the scene structure for particular classes of SfM problems (e.g., when dealing with point features or specific 3-D geometrical primitives). Finally, we note the many similarities between the SfM approach adopted in this study and the notion of "sensor-based" or "ego-centric" visual SLAM (see, e.g., [23] for a recent overview). In both the cases, a robot/camera builds a 3-D model of the environment in its own body/sensor frame via a filtering technique: an EKF in [23] and similar studies, and the deterministic filter (but with a fully *characterized* and *actively optimizable* transient response) derived from [20] in this paper.

The rest of this paper is organized as follows: Section II reviews the SfM problem in the context of nonlinear state observation and briefly summarizes the methodology developed in [20] for actively imposing a desired transient behavior to the estimation error. Section III then focuses on the three SfM problems considered in this study. Subsequently, Section IV reports the corresponding simulative and experimental results obtained with a manipulator equipped with an eye-in-hand camera. Finally, Section V concludes this paper and discusses some future directions.

II. Preliminaries

In this section, we briefly summarize the active estimation framework originally proposed in [20]. This is then applied to the SfM case studies discussed in Section III.

A. Nonlinear Observation Scheme

Let $(\boldsymbol{s}, \boldsymbol{\chi}) \in \mathbb{R}^{m+p}$ be the state of a dynamical system in the form

$$\begin{cases} \dot{\boldsymbol{s}} &= \boldsymbol{f}_m(\boldsymbol{s}, \boldsymbol{u}, t) + \boldsymbol{\Omega}^T(t)\boldsymbol{\chi} \\ \dot{\boldsymbol{\chi}} &= \boldsymbol{f}_u(\boldsymbol{s}, \boldsymbol{\chi}, \boldsymbol{u}, t) \end{cases} \tag{1}$$

where $\boldsymbol{s} \in \mathbb{R}^m$ and $\boldsymbol{\chi} \in \mathbb{R}^p$ represent, respectively, a *measurable* and *unmeasurable* component of $\boldsymbol{x}$, and $\boldsymbol{u} \in \mathbb{R}^v$ is the system input vector. In formulation (1), vector $\boldsymbol{\chi}$ is required to appear *linearly* in the dynamics of $\boldsymbol{s}$ (first equation). Furthermore, matrix $\boldsymbol{\Omega}(t) \in \mathbb{R}^{p\times m}$ and vectors $\boldsymbol{f}_m(\cdot) \in \mathbb{R}^m$ and $\boldsymbol{f}_u(\cdot) \in \mathbb{R}^p$ are assumed to be generic but *known* and sufficiently smooth functions with respect to their arguments, which are all available apart from the unknown value of $\boldsymbol{\chi}$ in $\boldsymbol{f}_u(\cdot)$.

SfM problems can be recast to formulation (1) by taking $\boldsymbol{s}$ as a set of visual features measured in the image, $\boldsymbol{u} = (\boldsymbol{v}, \boldsymbol{\omega})$ as the camera linear/angular velocity in camera frame, and $\boldsymbol{\chi}$ as a suitable (and locally invertible) function of the unknown structure of the scene to be estimated. For instance, in the point feature case, $\boldsymbol{\chi}$ can be taken as the *inverse* of the feature depth [9], and, for image moments of planar scenes, $\boldsymbol{\chi}$ can be taken as the normal vector of the observed plane scaled by its distance from the camera optical center [10]. Furthermore, in SfM one has $\boldsymbol{\Omega}(t) = \boldsymbol{\Omega}(\boldsymbol{s}(t), \boldsymbol{v}(t))$ with, in particular, $\boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{0}) \equiv \boldsymbol{0}$: the camera linear velocity $\boldsymbol{v}(t)$ plays a key role for the resolution of SfM problems.[1]

For a system in form (1), a possible estimation scheme can be devised as follows [9], [20]: let $(\hat{\boldsymbol{s}}, \hat{\boldsymbol{\chi}}) \in \mathbb{R}^{m+p}$ be the estimated state, $\boldsymbol{\xi} = \boldsymbol{s} - \hat{\boldsymbol{s}}$, $\boldsymbol{z} = \boldsymbol{\chi} - \hat{\boldsymbol{\chi}}$, $\boldsymbol{e} = (\boldsymbol{\xi}, \boldsymbol{z})$ be the total error vector, and consider the following *observer*:

$$\begin{cases} \dot{\hat{\boldsymbol{s}}} &= \boldsymbol{f}_m(\boldsymbol{s}, \boldsymbol{u}, t) + \boldsymbol{\Omega}^T(t)\hat{\boldsymbol{\chi}} + \boldsymbol{H}\boldsymbol{\xi} \\ \dot{\hat{\boldsymbol{\chi}}} &= \boldsymbol{f}_u(\boldsymbol{s}, \hat{\boldsymbol{\chi}}, \boldsymbol{u}, t) + \boldsymbol{\Lambda}\boldsymbol{\Omega}(t)\boldsymbol{Q}\boldsymbol{\xi} \end{cases} \tag{2}$$

where $\boldsymbol{H} > 0$, $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T > 0$, and $\boldsymbol{Q} = \boldsymbol{Q}^T > 0$ are positive definite gain matrices. Note that observer (2) is function of only measured/known quantities, with in particular a feedback action on the measurable error component $\boldsymbol{\xi}$. The corresponding estimation error dynamics is then given by

$$\begin{cases} \dot{\boldsymbol{\xi}} &= -\boldsymbol{H}\boldsymbol{\xi} + \boldsymbol{\Omega}^T(t)\,\boldsymbol{z} \\ \dot{\boldsymbol{z}} &= -\boldsymbol{\Lambda}\boldsymbol{\Omega}(t)\,\boldsymbol{Q}\boldsymbol{\xi} + (\boldsymbol{f}_u(\boldsymbol{s}, \boldsymbol{\chi}, \boldsymbol{u}) - \boldsymbol{f}_u(\boldsymbol{s}, \hat{\boldsymbol{\chi}}, \boldsymbol{u})) \\ &= -\boldsymbol{\Lambda}\boldsymbol{\Omega}(t)\,\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{g}(\boldsymbol{e}, t) \end{cases} \tag{3}$$

[1] This is due to the well-known fact that, under perspective and spherical projections, the motion in the image induced by pure rotations of the camera (i.e., when $\mathbf{v} = \mathbf{0}$) does not depend on the structure of the scene.

with $\boldsymbol{g}(\boldsymbol{e},\, t)$ being a "perturbation term" vanishing with respect to the error vector $\boldsymbol{e}$, i.e., such that $\boldsymbol{g}(\boldsymbol{0},\, t) = \boldsymbol{0}\, \forall t$. The origin of (3) can be proved to be locally exponentially stable if and only if (iff) the following *persistency of excitation* (PE) condition holds:

$$\int_t^{t+T} \boldsymbol{\Omega}\,(\tau)\,\boldsymbol{\Omega}^T\,(\tau)\,d\tau \geq \gamma \boldsymbol{I}_p > 0, \quad \forall t \geq t_0 \tag{4}$$

for some $T > 0$ and $\gamma > 0$, with $\boldsymbol{I}_n$ representing the $n \times n$ identity matrix.[2]

Remark II.1. We note that the *local* stability properties of the error dynamics (3) are due to the perturbation term $\boldsymbol{g}(\boldsymbol{e},\, t)$, which affects an otherwise globally exponentially stable error system. Indeed, in the special case $\dot{\boldsymbol{\chi}} = \boldsymbol{0}$ (unknown but *constant* parameters), one has $\boldsymbol{g}(\boldsymbol{e},\, t) \equiv \boldsymbol{0}$ and *global* exponential convergence for the error system (3). This is, for instance, the case of the structure estimation problems for spherical and cylindrical objects considered in Sections III-B and C. We stress, however, that the estimation scheme (2) is not restricted to this particular situation but can be applied (with, in this case, only local convergence guarantees) to the more general case of *state observation* problems in which the unknown $\boldsymbol{\chi}$ is subject to a nonnegligible dynamics as in (1). The depth estimation for a point feature discussed in the following Section III-A falls in this class.

The PE condition (4) plays the role of an *observability* criterium: convergence of the estimation error $\boldsymbol{e}(t) \to \boldsymbol{0}$ is possible iff the square matrix $\boldsymbol{\Omega}\,(t)\,\boldsymbol{\Omega}^T\,(t) \in \mathbb{R}^{p\times p}$ remains full rank in the integral sense of (4). We note that if $m \geq p$, that is, if the number of independent measurements $\boldsymbol{s}$ is larger or equal to the number of estimated quantities $\boldsymbol{\chi}$, then it is possible to *instantaneously* satisfy (4) by enforcing

$$\boldsymbol{\Omega}\,(t)\,\boldsymbol{\Omega}^T\,(t) \geq \frac{\gamma}{T}\boldsymbol{I}_p, \quad \forall t. \tag{5}$$

In the rest of this study, we will only consider this (more restrictive) observability condition.

B. Active Estimation Strategy

As clear from (4) and (5), (some measure of) the norm of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ determines the convergence properties of the error system (3). Furthermore, since in the SfM case it is $\boldsymbol{\Omega}(t) = \boldsymbol{\Omega}(\boldsymbol{s}(t),\, \boldsymbol{v}(t))$, it is meaningful to study how to optimize the camera linear velocity $\boldsymbol{v}$ in order to affect matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ and, as a consequence, to shape the transient response of the error vector $\boldsymbol{e}(t)$. The *active* strategy developed in [20] and summarized hereafter shows how to achieve this goal.

Let $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{\Omega}$ be the singular value decomposition of matrix $\boldsymbol{\Omega}$, where $\boldsymbol{\Sigma} = [\boldsymbol{S}\ \ \boldsymbol{0}]$, $\boldsymbol{S} = \mathrm{diag}(\sigma_i) \in \mathbb{R}^{p\times p}$, and $0 \leq \sigma_1 \leq \ldots \leq \sigma_p$ are the p singular values of $\boldsymbol{\Omega}$. Let also $\boldsymbol{Q} = \alpha \boldsymbol{I}_m$ and $\boldsymbol{\Lambda} = \beta \boldsymbol{I}_p$, with $\alpha > 0$, $\beta > 0$ (scalar gain matrices). By designing the gain matrix $\boldsymbol{H}$ in (2) as

$$\boldsymbol{H} = \boldsymbol{V} \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_2 \end{bmatrix} \boldsymbol{V}^T \tag{6}$$

with $\boldsymbol{D}_1 \in \mathbb{R}^{p\times p} > 0$, $\boldsymbol{D}_2 \in \mathbb{R}^{(m-p)\times(m-p)} > 0$, it is possible to show that under the change of coordinates

$$\boldsymbol{\eta} = \frac{1}{\sqrt{\alpha\beta}} \boldsymbol{S}^{-1}\boldsymbol{U}^T \boldsymbol{z} \tag{7}$$

and in the approximation $\boldsymbol{S}^{-1}\boldsymbol{U}^T \approx const$, the behavior of vector $\boldsymbol{\eta}$ (and hence, of the estimation error $\boldsymbol{z} = \boldsymbol{\chi} - \hat{\boldsymbol{\chi}}$) is governed by the following linear (and almost diagonal) dynamics:

$$\ddot{\boldsymbol{\eta}} = (\boldsymbol{\Pi} - \boldsymbol{D}_1)\dot{\boldsymbol{\eta}} - \alpha\beta \boldsymbol{S}^2 \boldsymbol{\eta}. \tag{8}$$

System (8) can be interpreted as a (unit-)mass-spring-damper system with diagonal stiffness matrix $\alpha\beta\boldsymbol{S}^2$ and damping matrix $\boldsymbol{D}_1$, together with an additional "perturbing" term $\boldsymbol{\Pi}$ whose full expression can be found in [20].

The convergence rate of (8) is then related to its slowest mode dictated by the "stiffness value" $\alpha\beta\sigma_1^2$, with σ_1^2 being the smallest eigenvalue of the square matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$. Therefore, for the sake of imposing a desired transient response to vector $\boldsymbol{\eta}(t)$ [i.e., to the estimation error $\boldsymbol{z}(t) = \boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t)$], one can "*place the poles*" of (8) by 1) shaping the damping factor $\boldsymbol{D}_1$ in (6) (a free parameter), 2) regulating the value of the smallest eigenvalue σ_1^2 by acting upon vector $\boldsymbol{v}$, and 3) suitably choosing the gain $\alpha\beta$ (a free parameter).

For what concerns the design of matrix $\boldsymbol{D}_1$, we first note that, as explained in [20], matrix $\boldsymbol{\Pi}$ in (8) can be regarded as a second-order perturbation term affecting the dissipative action induced by $\boldsymbol{D}_1$. Therefore, neglecting the effects of matrix $\boldsymbol{\Pi}$ and choosing $\boldsymbol{D}_1 = \mathrm{diag}(c_i)$, $c_i > 0$, allows obtaining a completely decoupled transient behavior for (8):

$$\ddot{\eta}_i + c_i\dot{\eta}_i + \alpha\beta\sigma_i^2\eta_i = 0, \quad i = 1 \ldots p. \tag{9}$$

One can then take, for instance, $c_i = c_i^* = 2\sqrt{\alpha\beta}\sigma_i$ in order to impose a critically damped evolution to the estimation error [coincident eigenvalues for (9)].

As for the regulation of $\sigma_1(t)$, being $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{s},\, \boldsymbol{v})$, it is

$$\dot{(\sigma_i^2)} = \boldsymbol{J}_{v,i}\dot{\boldsymbol{v}} + \boldsymbol{J}_{s,i}\dot{\boldsymbol{s}} \tag{10}$$

where the Jacobian matrices $\boldsymbol{J}_{v,i} \in \mathbb{R}^{1\times v}$ and $\boldsymbol{J}_{s,i} \in \mathbb{R}^{1\times n}$ can be computed in *closed form* (see [20] for all the details). By inverting the differential mapping (10), vector $\dot{\boldsymbol{v}}$ can then be exploited so as to, e.g., asymptotically enforce $\sigma_1^2(t) \to \sigma_{1,des}^2$ for some desired value $\sigma_{1,des}^2 > 0$. We note that ensuring $\sigma_1^2(t) \to \sigma_{1,des}^2 > 0$ also automatically satisfies the observability condition (5).

Finally, the following considerations hold for the choice of gain $\alpha\beta$ in (8). In the SfM context, the norm of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ is strongly related to the norm of the camera linear velocity $\boldsymbol{v}$. Roughly speaking, the "faster" the motion ($\sim$ larger $\|\boldsymbol{v}\|$), the "larger" the value of σ_1^2 ($\sim$ larger $\|\boldsymbol{\Omega}\boldsymbol{\Omega}^T\|$). Therefore, in order to maximize the estimation convergence speed of (8) (dictated by $\alpha\beta\sigma_1^2$), one can equivalently 1) travel at a larger speed $\|\boldsymbol{v}\|$ for a given gain $\alpha\beta$, or 2) increase the gain $\alpha\beta$ for a given

[2]The stability proof requires some additional technical assumptions on the regularity of the vanishing disturbance $\boldsymbol{g}$ (locally Lipschitz in a neighbourhood of the origin), on its growth bound with respect to $\|\boldsymbol{e}\|$ [which, since $\boldsymbol{g}(\cdot) \to \boldsymbol{0}$ if $(\boldsymbol{v},\, \boldsymbol{\omega}) \to \boldsymbol{0}$, can always be made small enough by limiting $(\boldsymbol{v},\, \boldsymbol{\omega})$], and on the norm of the initial error $\|\boldsymbol{e}(t_0)\|$. The interested reader can find in [9] a detailed derivation of the proof.

$\|\boldsymbol{v}\|$. While increasing the gain $\alpha\beta$ may always appear more convenient in terms of reduced control effort, practical issues such as noise, discretization, or quantization errors, may impose an upper limit on the possible value of $\alpha\beta$, thus necessarily requiring a larger $\|\boldsymbol{v}\|$ for obtaining the desired convergence speed. Furthermore, as in all SfM problems, a $\|\boldsymbol{v}\| \neq 0$ is also mandatorily required for guaranteeing $\sigma_1^2 > 0$ (a nontranslating camera cannot estimate the scene structure).

Remark II.2. We note that the proposed strategy is an *active* one since, in the general case, inversion of (10) will result in a camera linear velocity $\boldsymbol{v}$ optimized as a function of the system-measured state $\boldsymbol{s}$ in order to attain a desired $\sigma_{1,des}^2$ over time. We also highlight the role played by the gain matrix $\boldsymbol{H}$ weighting the feedback term in observer (2): the proposed machinery in fact relies on a suitable state-dependent design of the damping matrix $\boldsymbol{H}$ via the choice of $\boldsymbol{D}_1 = \mathrm{diag}(c_i^*)$ with $c_i^* = 2\sqrt{\alpha\beta}\sigma_i$. For the interested reader, this state-dependent design is conceptually equivalent to the *shaping of the damping factor* adopted in the context of impedance control for robot manipulator arms, see, e.g., [24].

Remark II.3. We also note that, in general, it is not possible to fully compensate for the term $\boldsymbol{J}_{s,i}\dot{\boldsymbol{s}}$ when inverting (10). Indeed, the expression in (1) implies a direct dependence of $\dot{\boldsymbol{s}}$ from the unmeasurable $\boldsymbol{\chi}$ so that an exact evaluation of $\dot{\boldsymbol{s}}$ is not obtainable in practice. A possible solution could be to use an approximation $\hat{\dot{\boldsymbol{s}}}$ of $\dot{\boldsymbol{s}}$ obtained by evaluating $\boldsymbol{f}_u(\cdot)$ on the current estimate $\hat{\boldsymbol{\chi}}$. Another simple workaround is, however, to just enforce $\dot{\boldsymbol{s}} \simeq 0$ by imposing a constraint on the camera motion. A combination of both strategies is, of course, also possible. The next sections will present some examples in this sense.

Remark II.4. It can be finally shown that, in the special situation $p = 1$ (only one quantity to be estimated), if $\sigma_1(t) \equiv const$ then $\boldsymbol{S}^{-1}\boldsymbol{U}^T \equiv const$ in (7) and matrix $\boldsymbol{\Pi}$ has no disturbing effects on (8). Therefore, in this case, it is always possible to *exactly* enforce the ideal estimation error dynamics (9) by just keeping $\|\boldsymbol{\Omega}(t)\|^2 = \sigma_1^2(t) = const$ during the camera motion. This situation will apply to all the case studies discussed in the rest of this paper.

III. Applications to Structure From Motion

In this section, we illustrate the application of the proposed active estimation framework to three concrete SfM problems: 1) estimation of the 3-D coordinates of a point feature, 2) estimation of the 3-D position and radius of a spherical target, and 3) estimation of the 3-D position and radius of a cylindrical target.

In the point feature case, the effects of the adopted projection model on the estimation convergence are also explicitly considered by discussing the differences between the two popular choices of *planar* and *spherical* projection models. For the spherical and cylindrical targets, we instead propose two *novel* minimal parameterizations that allow us to express the sphere/cylinder 3-D structures in terms of measured visual features and of a single unknown parameter (the sphere/cylinder radius). This allows, in all three cases, to reduce the SfM task to the estimation of a single unknown quantity (point feature depth or sphere/cylinder radius); thus, satisfying the requirements of Remark II.4 for *exactly* imposing the ideal dynamics (9) to the estimation error.

A. Depth Estimation for a Point Feature

1) Planar Projection Model: Let $\boldsymbol{p} = (x,\, y,\, 1) = (X/Z,\, Y/Z,\, 1) \in \mathbb{R}^3$ be the perspective projection of a 3-D point $\boldsymbol{P} = (X,\, Y,\, Z)$ onto the image plane of a calibrated pinhole camera. As it is well known [25], the differential relationship between the image motion of a point feature and the camera linear/angular velocity $\boldsymbol{u} = (\boldsymbol{v},\, \boldsymbol{\omega}) \in \mathbb{R}^6$ expressed in camera frame is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{Z} & 0 & \dfrac{x}{Z} & xy & -\left(1+x^2\right) & y \\ 0 & -\dfrac{1}{Z} & \dfrac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \boldsymbol{u} \tag{11}$$

where Z is the *depth* of the feature point. The dynamics of Z is

$$\dot{Z} = \begin{bmatrix} 0 & 0 & -1 & -yZ & xZ & 0 \end{bmatrix} \boldsymbol{u}.$$

The expression in (11) is not linear in Z, but it is linear in $1/Z$. Therefore, by defining $\boldsymbol{s} = (x,\, y) \in \mathbb{R}^2$ and $\chi = 1/Z$, with then $m = 2$ and $p = 1$, we obtain for (1)

$$\begin{cases} \boldsymbol{f}_m(\boldsymbol{s},\, \boldsymbol{u},\, t) = \begin{bmatrix} xy & -\left(1+x^2\right) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\Omega}(\boldsymbol{s},\, \boldsymbol{v}) = \begin{bmatrix} xv_z - v_x & yv_z - v_y \end{bmatrix} \\ f_u(\boldsymbol{s},\, \chi,\, \boldsymbol{u},\, t) = v_z\chi^2 + (y\omega_x - x\omega_y)\,\chi \end{cases} \tag{12}$$

with the perturbation term $g(\boldsymbol{e},\, t)$ in (3) taking the expression

$$g(\boldsymbol{e},\, t) = v_z\left(\chi^2 - \hat{\chi}^2\right) + (y\omega_x - x\omega_y)\, z \tag{13}$$

so that $g(\boldsymbol{0},\, t) = 0$, as expected. Note that once χ has been estimated, one can obviously retrieve the 3-D position of the point feature as $\boldsymbol{P} = \boldsymbol{p}/\chi$.

In the point feature case, matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ reduces to its single eigenvalue that, for a planar projection model, takes the expression

$$\sigma_1^2 = \|\boldsymbol{\Omega}\|^2 = (xv_z - v_x)^2 + (yv_z - v_y)^2. \tag{14}$$

Furthermore, using (14), the Jacobian $\boldsymbol{J}_{v,1}$ in (10) is given by

$$\boldsymbol{J}_{v,1} = 2\begin{bmatrix} v_x - xv_z \\ v_y - yv_z \\ (xv_z - v_x)\,x + (yv_z - v_y)\,y \end{bmatrix}^T. \tag{15}$$

Since σ_1^2 does not depend on $\boldsymbol{\omega}$, it is then possible to freely exploit the camera angular velocity for fulfilling additional goals of interest without interfering with the regulation of $\sigma_1^2(t)$ (only affected by $\boldsymbol{v}$). For instance, as in [17], one can use $\boldsymbol{\omega}$ for keeping $\boldsymbol{s} \simeq const$ so as to make the effects of $\dot{\boldsymbol{s}}$ negligible when inverting (10) with respect to $\dot{\boldsymbol{v}}$, see Remark II.3.

We now note that σ_1^2 in (14) depends on both the camera linear velocity $\boldsymbol{v}$ and on the location $\boldsymbol{p}$ of the feature point on the image plane. Since the value of σ_1^2 directly affects the

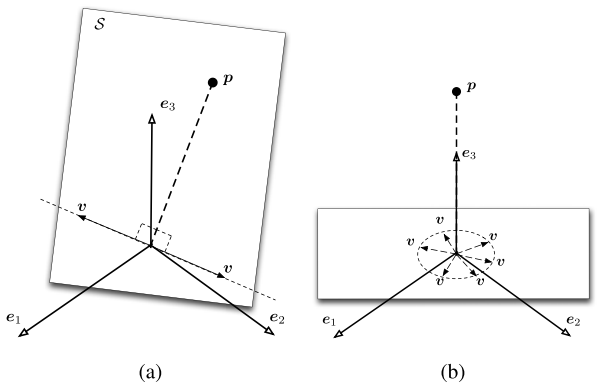


Fig. 1. Optimality conditions for the camera linear velocity $\boldsymbol{v}$ as dictated by system (17). (a) When $\boldsymbol{p} \neq \boldsymbol{e}_3$, vector $\boldsymbol{v}$ must be orthogonal to $\boldsymbol{p}$ and lie on the plane $\mathcal{S}$ spanned by $\boldsymbol{p}$ and $\boldsymbol{e}_3$ (that is, $\boldsymbol{v}$ must belong to a specific straight line). (b) When $\boldsymbol{p} = \boldsymbol{e}_3$, any $\boldsymbol{v} \perp \boldsymbol{e}_3$ is a valid solution to (17).

convergence speed of the estimation error, it is interesting to study what conditions on $\boldsymbol{p}$ and $\boldsymbol{v}$ result in the largest possible σ_1^2 (i.e., the fastest possible convergence for a given gain $\alpha\beta$). Letting $\boldsymbol{e}_3 = (0, 0, 1)$ being the camera optical axis, it is (by inspection)

$$\begin{bmatrix} \boldsymbol{\Omega}^T \\ 0 \end{bmatrix} = [\boldsymbol{e}_3]_\times [\boldsymbol{p}]_\times \boldsymbol{v}$$

where $[\boldsymbol{v}_1]_\times$ is the skew-symmetric matrix representing the cross-product operator for 3-D vectors (i.e., $[\boldsymbol{v}_1]_\times \boldsymbol{v}_2 = \boldsymbol{v}_1 \times \boldsymbol{v}_2$). Therefore,

$$\begin{aligned} \sigma_1^2 &= \begin{bmatrix} \boldsymbol{\Omega} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega}^T \\ 0 \end{bmatrix} = \| [\boldsymbol{e}_3]_\times [\boldsymbol{p}]_\times \boldsymbol{v} \|^2 \\ &= \|\boldsymbol{p}\|^2 \|\boldsymbol{v}\|^2 \sin^2(\theta_{p,v}) \sin^2\left(\theta_{e_3, [\boldsymbol{p}]_\times \boldsymbol{v}}\right) \end{aligned}$$

where $\theta_{\boldsymbol{p},\boldsymbol{v}}$ and $\theta_{\boldsymbol{e}_3, [\boldsymbol{p}]_\times \boldsymbol{v}}$ represent the angles between vectors $(\boldsymbol{p}, \boldsymbol{v})$ and vectors $(\boldsymbol{e}_3, [\boldsymbol{p}]_\times \boldsymbol{v})$, respectively. The maximum attainable value for σ_1^2 is then

$$\sigma_{\max}^2 = \max_{p,\, v} \sigma_1^2 = \|\boldsymbol{p}\|^2 \|\boldsymbol{v}\|^2. \tag{16}$$

This maximum is obtained when the camera linear velocity $\boldsymbol{v}$ is such that $\boldsymbol{p} \perp \boldsymbol{v}$ and $\boldsymbol{e}_3 \perp [\boldsymbol{p}]_\times \boldsymbol{v}$, i.e., rearranging in matrix form

$$\begin{bmatrix} \boldsymbol{p}^T \\ \boldsymbol{e}_3^T [\boldsymbol{p}]_\times \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} x & y & 1 \\ -y & x & 0 \end{bmatrix} \boldsymbol{v} = 0. \tag{17}$$

If $\boldsymbol{p} \neq \boldsymbol{e}_3$ (point feature *not* at the center of the image plane), system (17) has (full) rank 2 and admits the unique solution (up to a scalar factor)

$$\boldsymbol{v} = \delta [\boldsymbol{p}]_\times^2 \boldsymbol{e}_3, \quad \delta \in \mathbb{R}.$$

This requires the linear velocity $\boldsymbol{v}$ to be orthogonal to $\boldsymbol{p}$ and to lie on the plane defined by vectors $\boldsymbol{p}$ and $\boldsymbol{e}_3$ [i.e., $\boldsymbol{v}$ must belong to a straight line as shown in Fig. 1(a)].

If $\boldsymbol{p} = \boldsymbol{e}_3$ (point feature at the center of the image plane), system (17) loses rank and any $\boldsymbol{v} \perp \boldsymbol{e}_3$ is a valid solution [see Fig. 1(b)].

It is then possible to draw the following conclusions: for a given norm of the linear velocity $\|\boldsymbol{v}\|$ (i.e., the amount of "control effort"), system (17) determines the direction of $\boldsymbol{v}$ resulting in $\sigma_1^2 = \sigma_{\max}^2$ (maximization of σ_1^2). These conditions are summarized in Fig. 1(a) and (b). The value of $\sigma_{\max}^2$ is, however, also a function of the feature point location $\boldsymbol{p}$ that can be arbitrarily positioned on the image plane. In particular, $\sigma_{\max}^2 = \|\boldsymbol{v}\|^2$ for $\boldsymbol{p} = \boldsymbol{e}_3$ and $\sigma_{\max}^2 = \|\boldsymbol{p}\|^2 \|\boldsymbol{v}\|^2 > \|\boldsymbol{v}\|^2$ $\forall\, \boldsymbol{p} \neq \boldsymbol{e}_3$, with $\lim_{\|\boldsymbol{p}\| \to \infty} \sigma_{\max}^2(\boldsymbol{p}) = \infty$. The value of $\|\boldsymbol{p}\|$ (distance of the point feature from the image center) thus acts as an amplification factor for $\sigma_{\max}^2$. Therefore,

1) the smallest $\sigma_{\max}^2$ (i.e., the *slowest* "optimal" convergence for the depth estimation error) is obtained for the smallest value of $\|\boldsymbol{p}\|$, i.e., when $\boldsymbol{p} = \boldsymbol{e}_3 \Rightarrow \|\boldsymbol{p}\| = 1$ (feature point at the center of the image plane). It is worth noting that in this case $v_z = 0$ (from the condition $\boldsymbol{v} \perp \boldsymbol{p}$) and $\sigma_{\max}^2 = \|\boldsymbol{v}\|^2 = v_x^2 + v_y^2$: the camera moves on the surface of a sphere with a constant radius (depth) pointing at the feature point. In addition, being in this case $\dot{\chi} = \dot{Z}/Z^2 = 0$, one has $g(\boldsymbol{e}, t) \equiv 0$ and global convergence for the estimation error (see Remark II.1);
2) the largest $\sigma_{\max}^2$ (i.e., the *fastest* "optimal" convergence for the depth estimation error) is obtained for the largest possible value of $\|\boldsymbol{p}\|$. In the usual case of a rectangular image plane centered at the origin, this translates into keeping the feature point positioned at one of the four image corners. However, compared with the previous case, this results in a $\boldsymbol{g}(e, t) \neq \boldsymbol{0}$ and only local convergence for the estimation error.

2) Spherical Projection Model: We now develop the depth estimation machinery for the spherical projection model. In this case, the following quantity is taken as visual feature measured on the image plane:

$$\boldsymbol{s} = \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|} = \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|} \in \mathbb{S}^2$$

where $\mathbb{S}^2$ represents the unit sphere and, as well known [26]

$$\dot{\boldsymbol{s}} = \left[\tfrac{1}{\|\boldsymbol{P}\|} \left(\boldsymbol{s}\boldsymbol{s}^T - \boldsymbol{I}_3 \right) \tfrac{1}{\|\boldsymbol{P}\|} [\boldsymbol{s}]_\times \right] \boldsymbol{u}$$

and

$$\frac{d}{dt}\left(\frac{1}{\|\boldsymbol{P}\|}\right) = -\frac{1}{\|\boldsymbol{P}\|^2} \frac{d\|\boldsymbol{P}\|}{dt} = -\frac{\boldsymbol{s}^T \dot{\boldsymbol{P}}}{\|\boldsymbol{P}\|^2} = \frac{\boldsymbol{s}^T \boldsymbol{v}}{\|\boldsymbol{P}\|^2}.$$

Hence, by taking $\chi = 1/\|\boldsymbol{P}\|$ one obtains for (1)

$$\begin{cases} \boldsymbol{f}_m(\boldsymbol{s}, \boldsymbol{u}) = [\boldsymbol{s}]_\times \boldsymbol{\omega} \\ \boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v}) = -\boldsymbol{v}^T \left(\boldsymbol{I}_3 - \boldsymbol{s}\boldsymbol{s}^T \right) \\ f_u(\boldsymbol{s}, \chi, \boldsymbol{u}) = \chi^2 \boldsymbol{s}^T \boldsymbol{v} \end{cases} \tag{18}$$

with $m = 3$, $p = 1$, and $g(\boldsymbol{e}, t) = (\chi^2 - \hat{\chi}^2) \boldsymbol{s}^T \boldsymbol{v}$ for the perturbation term in (3). We note that, although in this case $m = 3$, vector $\boldsymbol{s}$ is subject to the constraint $\|\boldsymbol{s}\| = 1$; thus, resulting in only two independent measurements (as in the previous case of planar projection). Moreover, from the estimated χ one can easily retrieve $\boldsymbol{P} = \boldsymbol{s}/\chi$.

For the spherical projection model, the eigenvalue determining the convergence of the estimation error is

$$\sigma_1^2 = \boldsymbol{\Omega}\boldsymbol{\Omega}^T = \boldsymbol{v}^T\boldsymbol{v} - (\boldsymbol{s}^T\boldsymbol{v})^2$$

with thus

$$\boldsymbol{J}_{v,1} = 2\boldsymbol{v}^T(\boldsymbol{I}_3 - \boldsymbol{s}\boldsymbol{s}^T). \tag{19}$$

As before, σ_1^2 does not depend on $\boldsymbol{\omega}$, which can then be exploited to fulfil any additional task of interest (e.g., keeping $\boldsymbol{s} \simeq const$ during motion).

As for the conditions on $\boldsymbol{s}$ and $\boldsymbol{v}$ that yield maximization of σ_1^2, one clearly has

$$\sigma_1^2 = \sigma_{\max}^2 = \max_{\boldsymbol{s},\,\boldsymbol{v}} \sigma_1^2 = \|\boldsymbol{v}\|^2 \tag{20}$$

iff $\boldsymbol{s}^T\boldsymbol{v} = 0$ (linear velocity orthogonal to the projection ray passing through $\boldsymbol{P}$). We also note that, in this case, one has $\dot{\chi} = 0$ and $g(\boldsymbol{e},\, t) \equiv 0$ (*constant* unknown state and *global* convergence for the estimation error) *regardless* of the location of $\boldsymbol{s}$ on the image plane.

3) Comparison Between Planar and Spherical Projection Models: For a spherical projection model, maximization of the eigenvalue σ_1^2 imposes only one condition for the linear velocity $\boldsymbol{v}$ ($\boldsymbol{s}^T\boldsymbol{v} = 0$). When this condition is met, one has $\sigma_1^2 = \sigma_{\max}^2 = \|\boldsymbol{v}\|^2$ and global convergence for the estimation error *whatever* the location of the feature point $\boldsymbol{s}$. This is equivalent to what was obtained for the planar projection case in the special situation $\boldsymbol{p} = \boldsymbol{e}_3$ (indeed the two projection models coincide for $\boldsymbol{p} = \boldsymbol{s} = \boldsymbol{e}_3$). However, with a spherical projection model, one also loses the possibility of increasing the estimation convergence rate by suitably positioning the point feature $\boldsymbol{s}$ on the image plane (since in this case $\sigma_{\max}^2$ does not depend on $\boldsymbol{s}$).

It is then worth noting the complementarity of the two cases: for a given $\|\boldsymbol{v}\|$, and provided the optimal condition $\boldsymbol{p}^T\boldsymbol{v} = 0$ is satisfied, the *planar* projection allows obtaining a faster error convergence at the price of local stability (increase of the perturbation g) by suitably positioning $\boldsymbol{s} = (x, y)$ (the larger $\|\boldsymbol{s}\|$ the faster the convergence). The *spherical* projection guarantees global error convergence for *any* location of the feature point, but at the price of being always subject to the same convergence rate only function of the control effort $\|\boldsymbol{v}\|$.

B. Structure Estimation for a Spherical Target

We now detail the application of the proposed estimation machinery to the case of a spherical target. Consider a sphere $\mathcal{O}_s$ of radius R and let $\boldsymbol{P}_0 = (X_0,\, Y_0,\, Z_0)$ be the coordinates of its center in the camera frame. Let also

$$\mathcal{L}: \quad \boldsymbol{n}^T\boldsymbol{X} + d = 0$$

represent the *planar limb surface* associated with the sphere in the camera frame, where $\boldsymbol{X} \in \mathbb{R}^3$ is any 3-D point on the plane, $\boldsymbol{n} \in \mathbb{S}^2$ is the plane unit normal vector, and $d \in \mathbb{R}$ the plane distance to the camera center [27]. Fig. 2 shows the quantities of interest.

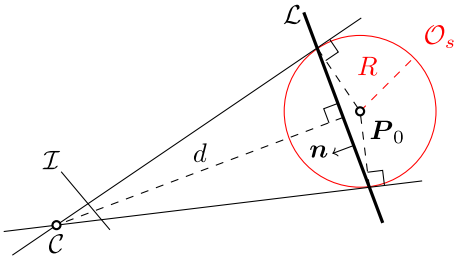


Fig. 2. Spherical target $\mathcal{O}_s$ and planar limb surface $\mathcal{L}$.

The depth Z of any point $\boldsymbol{X}$ lying on $\mathcal{L}$ can be expressed in terms of its normalized image coordinates $\boldsymbol{p} = (x,\, y,\, 1)$ as

$$\frac{1}{Z} = \frac{X_0}{K}x + \frac{Y_0}{K}y + \frac{Z_0}{K} = \boldsymbol{\chi}^T\boldsymbol{p} \tag{21}$$

where $K = \boldsymbol{P}_0^T\boldsymbol{P}_0 - R^2$ and $\boldsymbol{\chi} = \boldsymbol{P}_0/K = -\boldsymbol{n}/d \in \mathbb{R}^3$ represent *unmeasurable* quantities (analogously to Z for the point feature case), see [28] for all the details. The interaction matrix of a generic $(i,\, j)$th order moment m_{ij} evaluated on the image of $\mathcal{O}_s$ depends linearly on $\boldsymbol{\chi}$, see [10], [27]. Therefore, a first possibility of retrieving the sphere 3-D parameters $(\boldsymbol{P}_0,\, R)$ would be to implement the estimation scheme (2) with $\boldsymbol{s}$ being a suitable collection of image moments (e.g., area and barycenter). It is in fact possible to show that (see Appendix A)

$$\dot{\boldsymbol{\chi}} = -\frac{\boldsymbol{v}}{K} - [\boldsymbol{\omega}]_\times\boldsymbol{\chi} + 2\boldsymbol{\chi}\boldsymbol{\chi}^T\boldsymbol{v}$$

and that K can be expressed in terms of image moments and of vector $\boldsymbol{\chi}$ itself so that having estimated $\boldsymbol{\chi}$, one can consequently retrieve $\boldsymbol{P}_0 = \boldsymbol{\chi}K$ and $R = \sqrt{\boldsymbol{P}_0^T\boldsymbol{P}_0 - K}$.

Although conceptually valid, this solution requires the concurrent estimation of *three time-varying quantities* [vector $\boldsymbol{\chi}(t)$]. On the other hand, inspired by [29], we now describe a *novel representation* of the sphere projection on the image plane that allows us to reformulate the structure estimation task in terms of a *single unknown constant parameter*, i.e., the sphere radius R.

To this end, define vector $\boldsymbol{s} = (s_x,\, s_y,\, s_z) \in \mathbb{R}^3$ as

$$\begin{cases} s_x &= \dfrac{x_g}{s_z a_1^2} \\ s_y &= \dfrac{y_g}{s_z a_1^2} \\ s_z &= \sqrt{\dfrac{1 + a_1^2}{a_1^2}} \end{cases} \tag{22}$$

where $(x_g,\, y_g,\, n_{20},\, n_{11},\, n_{02})$ represents the barycenter and normalized centered moments of order 2 measured from the elliptical projection of the sphere $\mathcal{O}_s$ on the image plane, and a_1 is the minor axis of the observed ellipse with [27]:

$$a_1^2 = 2\left(n_{20} + n_{02} - \sqrt{(n_{20} - n_{02})^2 + 4n_{11}}\right). \tag{23}$$

We thus note that vector $\boldsymbol{s}$ can be directly evaluated in terms of measured image quantities.

From [27], [29], one also has

$$x_g = \frac{X_0 Z_0}{Z_0^2 - R^2}, \quad y_g = \frac{Y_0 Z_0}{Z_0^2 - R^2}, \quad a_1^2 = \frac{R^2}{Z_0^2 - R^2} \tag{24}$$

which when plugged in (22) and (23), result in the equivalent expression $\boldsymbol{s} = \boldsymbol{P}_0/R$. Since vector $\boldsymbol{s}$ can be computed from image measurements as in (22), estimation of the (unknown) sphere radius R allows us to recover the 3-D sphere center as $\boldsymbol{P}_0 = \boldsymbol{s}R$.

Exploiting now the results of [29], it is possible to show that

$$\dot{\boldsymbol{s}} = \left[-\frac{1}{R}\boldsymbol{I}_3 \quad [\boldsymbol{s}]_\times \right] \boldsymbol{u}. \tag{25}$$

Since (25) is linear in $1/R$, we can define $\chi = 1/R$, with then $m = 3$ and $p = 1$, and obtain for (1) and (3)

$$\begin{cases} \boldsymbol{f}_m(\boldsymbol{s}, \boldsymbol{u}) &= [\boldsymbol{s}]_\times \boldsymbol{\omega} \\ \boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{u}) &= -\boldsymbol{v}^T \\ f_u(\boldsymbol{s}, \chi, \boldsymbol{u}) &= 0 \\ g(\boldsymbol{e}, t) &= 0. \end{cases} \tag{26}$$

We note that in this case it is always possible to obtain *global* convergence for the estimation error since $\dot{\chi} = 0$ and, therefore, $g(\boldsymbol{e}, t) = 0$ by construction (see Remark II.1). Furthermore, matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ reduces again to its single eigenvalue $\sigma_1^2 = \|\boldsymbol{v}\|^2$ and, if $\sigma_1^2(t) \equiv const > 0$, the "ideal" estimation error dynamics (9) can be exactly obtained. One also has $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{v})$ and $\boldsymbol{J}_{v,1} = 2\boldsymbol{v}^T$.

We finally note the following facts: first, contrarily to the previous cases, here $\dot{\boldsymbol{s}}$ has no effect on the regulation of σ_1^2, which is only function of the camera linear velocity $\boldsymbol{v}$. It is then, of course, still possible to freely exploit the camera angular velocity $\boldsymbol{\omega}$ for, e.g., keeping the sphere at the center of the image by regulating (s_x, s_y) to zero. Second, we note the strong similarities with the previous optimal results obtained for a point feature under a *spherical* projection model [$\sigma_{\max}^2$ in (20)]: in both the cases, the maximum estimation convergence rate for a given $\|\boldsymbol{v}\|$ does not depend on the position of the observed object on the image plane.

C. Structure Estimation for a Cylindrical Target

We now finally consider the case of SfM for a 3-D cylindrical object. A cylinder $\mathcal{O}_c$ can be described by its radius $R > 0$ and by its main axis $\boldsymbol{a} \in \mathbb{S}^2$ passing through a 3-D point $\boldsymbol{P}_0 = (X_0, Y_0, Z_0)$, with $\|\boldsymbol{a}\| = 1$ and, w.l.o.g., $\boldsymbol{a}^T\boldsymbol{P}_0 = 0$ ($\boldsymbol{P}_0$ can be chosen as the closest point on $\boldsymbol{a}$ to the origin of the camera frame [17]). Moreover, analogously to the sphere, a cylinder is also associated with a planar limb surface $\mathcal{L}$ such that (21) holds for any point on $\mathcal{L}$ with projection $\boldsymbol{p} = (x, y, 1)$. Therefore, a possibility is to estimate the three unknown parameters of the limb plane $\mathcal{L}$ (vector $\boldsymbol{\chi}$) by exploiting (at least) three image measurements, see [17] and Appendix B for some details in this sense. However, following the previous developments, we now propose a *novel representation* of the cylinder projection on the image plane that, again, allows us to obtain the cylinder parameters $(\boldsymbol{P}_0, \boldsymbol{a}, R)$ in terms of image measurements and

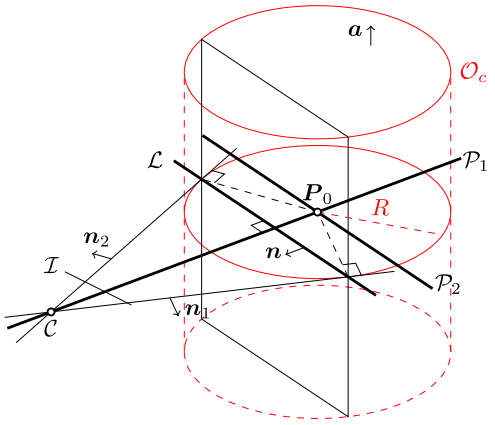


Fig. 3. Camera $\mathcal{C}$ and cylindrical target $\mathcal{O}_c$ with the planar limb surface $\mathcal{L}$ and the other planes of interest $\mathcal{P}_1$ and $\mathcal{P}_2$.

of the *unknown but constant* cylinder radius R that, therefore, represents the only quantity to be estimated.

Let (ρ_1, θ_1) and (ρ_2, θ_2) be the (*measured*) distance/angle parameters of the two straight lines resulting from the projection of the cylinder on the image plane, and

$$\boldsymbol{n}_1 = (\cos\theta_1, \sin\theta_1, -\rho_1), \ \boldsymbol{n}_2 = (\cos\theta_2, \sin\theta_2, -\rho_2) \tag{27}$$

be the normal vectors to the two planes passing through the origin of the camera frame and the two aforementioned projected lines.[3] Fig. 3 gives a graphical representation of the quantities of interest. Note that vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ can be directly evaluated from image measurements (the line parameters). We then define vector $\boldsymbol{s} \in \mathbb{R}^3$ as

$$\boldsymbol{s} = \frac{\boldsymbol{\Delta}}{\|\boldsymbol{\Delta}\|^2} \tag{28}$$

with

$$\boldsymbol{\Delta} = \frac{1}{2}\left(\frac{\boldsymbol{n}_1}{\|\boldsymbol{n}_1\|} + \frac{\boldsymbol{n}_2}{\|\boldsymbol{n}_2\|} \right). \tag{29}$$

Vector $\boldsymbol{s}$ is, thus, also directly obtainable in terms of image quantities.

We now note that, from [30], an equivalent expression for vectors $\boldsymbol{n}_1$, $\boldsymbol{n}_2$ in terms of the cylinder 3-D geometry can be obtained as

$$\boldsymbol{n}_1 = \frac{1}{N_1}\begin{bmatrix} R\frac{X_0}{\sqrt{K}} - \alpha \\ R\frac{Y_0}{\sqrt{K}} - \beta \\ R\frac{Z_0}{\sqrt{K}} - \gamma \end{bmatrix}, \quad \boldsymbol{n}_2 = \frac{1}{N_2}\begin{bmatrix} R\frac{X_0}{\sqrt{K}} + \alpha \\ R\frac{Y_0}{\sqrt{K}} + \beta \\ R\frac{Z_0}{\sqrt{K}} + \gamma \end{bmatrix} \tag{30}$$

[3]The two planes are, therefore, tangent to the surface of the cylinder.

with

$$\begin{cases} K &= \sqrt{\boldsymbol{P}_0^T\boldsymbol{P}_0 - R^2} \\ (\alpha,\,\beta,\,\gamma) &= [\boldsymbol{P}_0]_\times \boldsymbol{a} \\ N_1 &= \sqrt{\left(R\frac{X_0}{\sqrt{K}} - \alpha\right)^2 + \left(R\frac{Y_0}{\sqrt{K}} - \beta\right)^2} \\ N_2 &= \sqrt{\left(R\frac{X_0}{\sqrt{K}} + \alpha\right)^2 + \left(R\frac{Y_0}{\sqrt{K}} - \beta\right)^2} \end{cases} \tag{31}$$

thus yielding

$$\begin{cases} \dfrac{\boldsymbol{n}_1}{\|\boldsymbol{n}_1\|} &= \dfrac{1}{\boldsymbol{P}_0^T\boldsymbol{P}_0}\begin{bmatrix} RX_0 - \alpha\sqrt{K} \\ RY_0 - \beta\sqrt{K} \\ RZ_0 - \gamma\sqrt{K} \end{bmatrix} \\ \dfrac{\boldsymbol{n}_2}{\|\boldsymbol{n}_2\|} &= \dfrac{1}{\boldsymbol{P}_0^T\boldsymbol{P}_0}\begin{bmatrix} RX_0 + \alpha\sqrt{K} \\ RY_0 + \beta\sqrt{K} \\ RZ_0 + \gamma\sqrt{K} \end{bmatrix} \end{cases}. \tag{32}$$

Plugging (32) in (29) results in the equivalent expression

$$\boldsymbol{\Delta} = \frac{R^2}{\boldsymbol{P}_0^T\boldsymbol{P}_0}\boldsymbol{s}$$

which, using (28), finally yields the following relationship between image quantities and cylinder 3-D structure:

$$\boldsymbol{s} = \frac{\boldsymbol{\Delta}}{\|\boldsymbol{\Delta}\|^2} = \frac{\boldsymbol{P}_0}{R}. \tag{33}$$

As for the cylinder axis $\boldsymbol{a}$, exploiting (30) one has

$$\begin{aligned} [\boldsymbol{n}_2]_\times\boldsymbol{n}_1 &= \frac{2R}{N_1N_2\sqrt{K}}\begin{bmatrix} Z_0\beta - Y_0\gamma \\ X_0\gamma - Z_0\alpha \\ Y_0\alpha - X_0\beta \end{bmatrix} \\ &= \frac{2R}{N_1N_2\sqrt{K}}\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}_\times \boldsymbol{P}_0 \\ &= \frac{2R}{N_1N_2\sqrt{K}}\left[[\boldsymbol{P}_0]_\times\,\boldsymbol{a}\right]_\times \boldsymbol{P}_0 = \frac{2R\boldsymbol{P}_0^T\boldsymbol{P}_0}{N_1N_2\sqrt{K}}\boldsymbol{a} \end{aligned} \tag{34}$$

where, in the last step, the property $\boldsymbol{a}^T\boldsymbol{P}_0 = 0$ was used. Since $\|\boldsymbol{a}\| = 1$, from (34) it is

$$\boldsymbol{a} = \frac{[\boldsymbol{n}_2]_\times\boldsymbol{n}_1}{\left\|[\boldsymbol{n}_2]_\times\,\boldsymbol{n}_1\right\|}. \tag{35}$$

The cylinder axis $\boldsymbol{a}$ can then be directly obtained in terms of only measured quantities.

We now note that, as in the sphere case, the only unknown left is the cylinder radius R: once known, the cylinder 3-D structure can be fully recovered from image measurements as $\boldsymbol{P}_0 = R\boldsymbol{s}$ from (33) and $\boldsymbol{a}$ from (35).

An estimation scheme for R can be obtained exploiting the following differential relationship whose derivation is given in Appendix C:

$$\dot{\boldsymbol{s}} = \begin{bmatrix} -\dfrac{1}{R}\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right) & [\boldsymbol{s}]_\times \end{bmatrix}\boldsymbol{u}. \tag{36}$$

Note the similarity of (36) with (25) for the sphere case.

Being (36) linear in $1/R$, one can then apply observer 2 by choosing $\chi = 1/R$ with $m = 3$ and $p = 1$, and obtaining

$$\begin{cases} \boldsymbol{f}_m\,(\boldsymbol{s},\,\boldsymbol{u}) &= [\boldsymbol{s}]_\times\boldsymbol{\omega} \\ \boldsymbol{\Omega}\,(\boldsymbol{s},\,\boldsymbol{u}) &= -\boldsymbol{v}^T\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right) \\ f_u\,(\boldsymbol{s},\,\chi,\,\boldsymbol{u}) &= 0 \\ g(\boldsymbol{e},\,t) &= 0. \end{cases} \tag{37}$$

Note how, again, being $\dot{\chi} = 0$ it is $g(\boldsymbol{e},\,t) = 0$ [global convergence for the error system (3) as in the sphere case].

Matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ reduces to its single eigenvalue

$$\sigma_1^2 = \boldsymbol{\Omega}\boldsymbol{\Omega}^T = \|\boldsymbol{v}\|^2 - (\boldsymbol{a}^T\boldsymbol{v})^2. \tag{38}$$

It is worth comparing (38) with the result obtained for the sphere ($\sigma_1^2 = \|\boldsymbol{v}\|^2$). In the cylinder case, the convergence rate of the estimation error is affected by both the *norm* and the *direction* of the linear velocity $\boldsymbol{v}$. In particular, for a given $\|\boldsymbol{v}\| = const$, the maximum value for σ_1^2 is obtained when $\boldsymbol{v}$ has a null component along the cylinder axis $\boldsymbol{a}$ ($\boldsymbol{a}^T\boldsymbol{v} = 0$) with, in this case, $\sigma_1^2 = \sigma_{1\max}^2 = \|\boldsymbol{v}\|^2$. Intuitively, any camera motion along the cylinder axis does not provide any useful information to the estimation task. Furthermore, as in all previous cases, keeping a $\sigma_1^2(t) = const$ allows us to exactly enforce the ideal estimation error dynamics (9), see Remark II.4.

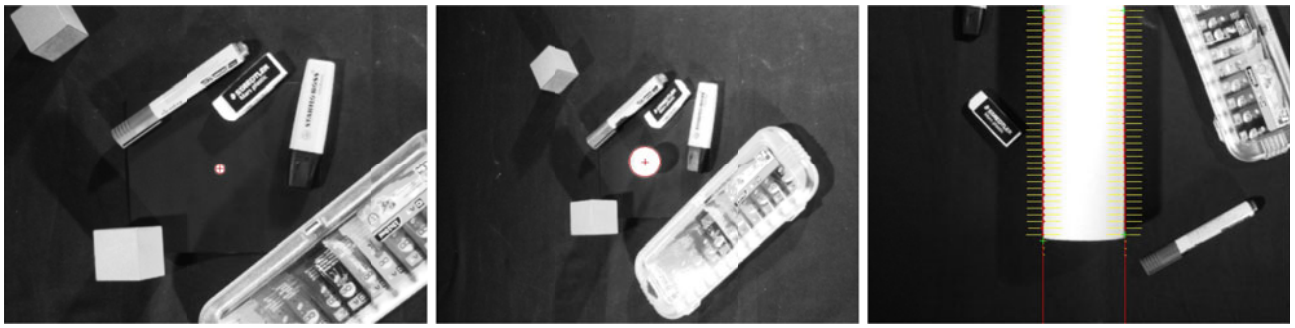
Finally, from (38) one has

$$\dot{(\sigma_1^2)} = \boldsymbol{J}_{v,1}\dot{\boldsymbol{v}} + \boldsymbol{J}_{a,1}\dot{\boldsymbol{a}} = \boldsymbol{J}_{v,1}\dot{\boldsymbol{v}} + \boldsymbol{J}_{a,1}[\boldsymbol{a}]_\times\boldsymbol{\omega} \tag{39}$$

with $\boldsymbol{J}_{v,1} = 2\boldsymbol{v}^T\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right)$ and $\boldsymbol{J}_{a,1} = 2\boldsymbol{v}^T\boldsymbol{a}\boldsymbol{v}^T$. Although (39) also depends on the angular velocity $\boldsymbol{\omega}$, it is possible to fully compensate for the effects of $\boldsymbol{J}_{a,1}[\boldsymbol{a}]_\times\boldsymbol{\omega}$ (a known quantity) when inverting (39) with respect to $\dot{\boldsymbol{v}}$ as discussed in Section IV-D. Therefore, one can act on $\dot{\boldsymbol{v}}$ to regulate the value of $\sigma_1^2(t)$ and, at the same time and in a decoupled way, exploit the camera angular velocity $\boldsymbol{\omega}$ for implementing additional tasks of interest, such as keeping the cylinder axis $\boldsymbol{a}$ at the center of the image plane by enforcing $(s_x,\,s_y) = \boldsymbol{0}$.

IV. Simulation and Experimental Results

In this section, we show some experimental and simulation results meant to validate the theoretical developments of the previous sections. The experiments were run by employing a grayscale camera with a resolution of 640×480 px and a frame rate of 30 fps. The camera was mounted on the end-effector of a 6-dofs Gantry robot commanded in velocity at a frequency of 100 Hz. All the image processing and feature tracking were implemented via the open-source ViSP library [31]. Some snapshots of the three experiments are shown in Fig. 4, where the result of the image processing is highlighted in red.



(a) (b) (c)

Fig. 4. Camera snapshots for the point feature. (a) Sphere. (b) Cylinder. (c) Experiment.

A video of the reported experiments (including the three cases of point, sphere, and cylinder) is also attached to the paper for the reader's convenience.

A. Comparison of Planar and Spherical Projection Models

We start by comparing via simulation results the effects of adopting a planar and spherical projection model for the depth estimation of a point feature as extensively discussed in Sections III-A1) and III-A2). We considered three cases differing for the location on the image plane at which the point feature was (purposely) kept exploiting the camera angular velocity $\boldsymbol{\omega}$:

1) in case I, the point feature was kept at the center of the image plane (red line in the following plots);
2) in case II, the point feature was kept at one of the corners of an image plane with the same size of the camera used in the experiments (green line in the following plots);
3) in case III, the point feature was kept at one of the corners of an image plane with a size five times larger than case II (blue line in the following plots).

In all the cases, a constant camera velocity $\boldsymbol{v}(t) \equiv \boldsymbol{v}(t_0) = const$ was kept during motion, with the initial condition $\boldsymbol{v}(t_0)$ chosen so as to comply with the optimality conditions discussed in Sections III-A1) and III-A2) for letting $\sigma_1^2 = \sigma_{\max}^2$ [e.g., with $\boldsymbol{v}(t_0)$ being a solution of (17) in the planar projection case].

Fig. 5(a) shows the behavior of $z(t)$ for the three cases when using a *planar* projection model. We can then note how the convergence rate of the estimation error increases from case I (slowest convergence) to case III (fastest convergence) as predicted by the theory (for the same $\|\boldsymbol{v}\|$ a larger $\|\boldsymbol{p}\|$ results in a larger $\sigma_{\max}^2$). Similarly, Fig. 5(b) reports the behavior of $\sigma_1^2(t)$ for the three cases: as expected, $\sigma_1^2(t)$ results the largest for case III. Note also how $\sigma_1^2(t)$ for case II (green line) is only *slightly* larger than case I (red line). This is due to relatively small size of the image plane of case II whose dimensions were set as those of the real camera used for the experiments. Finally, Fig. 5(c) shows the behavior of the perturbation term $g(\boldsymbol{e}, t)$ in the three cases: here, one can verify how $g = 0$ for case I, with then an increasing $|g|$ for cases II and III. Indeed, as discussed in Section III-A1), the "amplification" effect on $\sigma_{\max}^2$ obtained by increasing $\|\boldsymbol{p}\|$ comes at the price of an increased magnitude of the perturbation g. This is also evident in Fig. 5(a), where the ideal response of (9) is plotted with the dashed lines for the three considered cases. We can, thus, note how $z(t)$ in case I presents a perfect match with its corresponding ideal response, with then an increasing (albeit very limited) mismatch in the other two cases due to the increased effect of the perturbation g.

As for the spherical projection model, Fig. 5(d) reports the behavior of the estimation error $z(t)$ for the three cases under consideration, together with the ideal response (9). Here, the symbol $z_s(t)$ is used to denote the estimation error in the spherical projection case in order to distinguish it from the error obtained with the planar projection model. All the plots result perfectly superimposed as expected from the analysis of Section III-A2. Indeed, in the spherical projection case, $\sigma_{\max}^2 = \|\boldsymbol{v}\|^2$ regardless of the location of $\boldsymbol{p}$ and $g(t) \equiv 0$. However, the absence of perturbation terms is obtained at the expense of the convergence rate of $z_s(t)$, which indeed results slower or equal to that of $z(t)$ in the planar projection case. This is shown in Fig. 5(e), where the behavior of $z(t) - z_s(t)$ is reported for the three cases. We can then note how $z(t) - z_s(t) = 0$ only in case I, as the planar and spherical models coincide when the feature point is at the center of the image plane.

These results then fully confirm the validity of the theoretical analysis reported in Section III-A1 and III-A2. However, we also note the marginal effects of the two projection models on the estimation performance when applied to an image plane of size comparable with that of the real camera used in our experimental setup. Therefore, in the following experimental results, we will only consider the case of planar projection model.

B. Depth Estimation for a Point Feature

We report here some experimental results for the depth estimation of a point feature under a planar projection model (see Section III-A1). The following experiments are meant to demonstrate how the proposed active estimation framework can be exploited to select online the "best" camera motion. As visual target, we made use of a circular white dot of $5\,mm$ radius painted on a planar black surface and sufficiently far from the camera in order to safely consider it as a "point feature."

Fig. 6(a) shows the evolution of the estimation error $z(t) = 1/Z(t) - 1/\hat{Z}(t)$ for two experiments[4] in which $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}_0\|$, but with its direction being either optimized in order to maximize the estimation convergence rate (case I, red line)

[4]The ground truth $Z_0(t)$ was obtained from a previous offline estimation of the 3-D position $\boldsymbol{P}_0$ in the world frame, and by then using the information on the camera position provided by the robot forward kinematics.

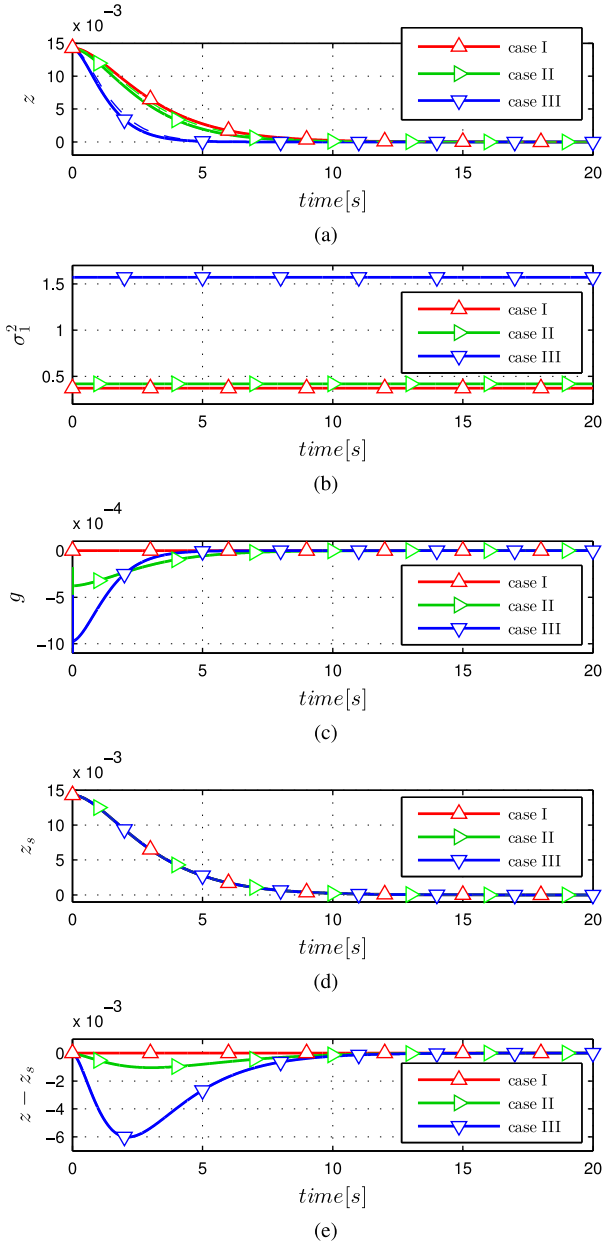


Fig. 5. Simulation results comparing the planar and spherical projection models for the depth estimation of a point feature. The following color coding is adopted for the three considered cases: red–case I, green–case II, blue–case III. (a) Behavior of the estimation error $z(t)$ in the planar projection case (solid lines) with superimposed the corresponding ideal response (9) (dashed lines). (b) Behavior of $\sigma_1^2(t)$ for the three cases with, again, the largest $\sigma_1^2(t)$ in case III. (c) Behavior of the perturbation term $g(\boldsymbol{e},\, t)$ for the three cases. (d) Behavior of the estimation error $z_s(t)$ for the spherical projection model in the three cases. (e) Behavior of $z(t) - z_s(t)$.

or kept constant so that $\boldsymbol{v}(t) = \boldsymbol{v}_0 = const$ (case II, blue line). This effect was obtained by using the following control law[5]:

$$\dot{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|^2} k_1 \left(\kappa_{des} - \kappa\right) + k_2 \left(\boldsymbol{I}_3 - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^2}\right) \boldsymbol{J}_{v,1}^T \qquad (40)$$

[5]The value of $\dot{\boldsymbol{v}}$ resulting from this (and following) optimizations was numerically integrated so as to obtain the commanded $\boldsymbol{v}(t)$ sent to the robot low-level controller.

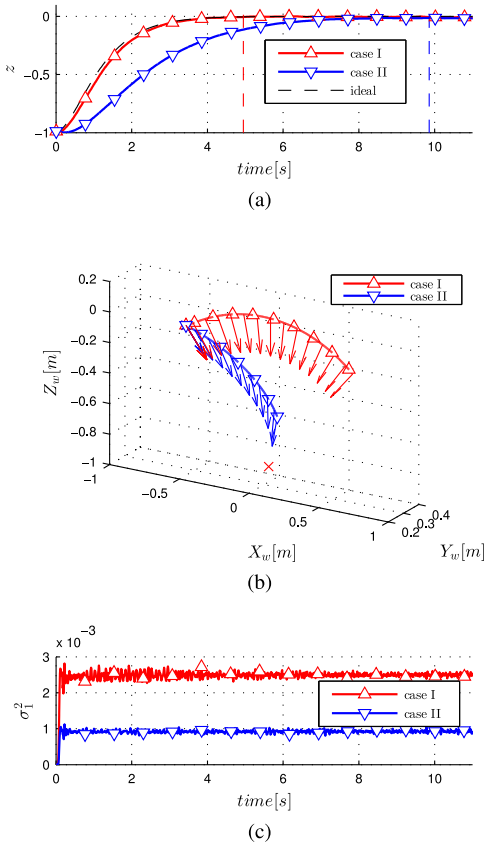


Fig. 6. Experimental results for the point feature case. (a) Behavior of the estimation error for case I (solid red line) and case II (solid blue line), and for an "ideal" second order system 9 with desired poles at $\sigma_{1\max}^2$ (dashed black line). The two vertical dashed lines indicate the times $T_1 = 4.95$ s and $T_2 = 9.85$ s at which the estimation error drops below the threshold of 5 mm. (b) Camera trajectories for case I (red line) and case II (blue line) with arrows indicating the direction of the camera optical axis. (c) Behavior of $\sigma_1^2(t)$ for case I (red line) and case II (blue line).

with $k_1 > 0$, $k_2 \geq 0$, $\kappa = \frac{1}{2}\boldsymbol{v}^T\boldsymbol{v}$, $\kappa_{des} = \frac{1}{2}\boldsymbol{v}_0^T\boldsymbol{v}_0$, and $\boldsymbol{J}_{v,1}$ given by (15). In fact, the first term in (40) enforces the constraint $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}_0\|$ (same control effort in both cases), while the second term allows us to implement either case I ($k_2 > 0$) or case II ($k_2 = 0$) (maximization of σ_1^2) within the null space of the first constraint. In both the cases, the angular velocity $\boldsymbol{\omega}$ was exploited for keeping the point feature at the center of the image plane $(x,\, y) \to (0,\, 0)$. We note that, as discussed in Section III-A1, when $(x,\, y) = (0,\, 0)$ one has $\sigma_{1\max}^2 = v_x^2 + v_y^2$ from 16 and $\sigma_1^2 = \sigma_{1\max}^2$ iff $v_z = 0$ (circular motion around the point feature). The experiments were run with the following parameters: $\alpha\beta = 10^3$ for gains $\boldsymbol{Q}$ and $\boldsymbol{\Lambda}$, $c_1 = c_1^*$ for $\boldsymbol{D}_1$ in (6), $\boldsymbol{v}(t_0) = \boldsymbol{v}_0 = (0.03, 0, -0.04)$ m/s, $k_1 = 5$ and $k_2 = 10^4$, thus resulting in the maximum value $\sigma_{1\max}^2 = 0.0025$ for the eigenvalue σ_1^2.

As clear from Fig. 6(b), while in case II the camera gets closer to the point feature, the use of the active strategy of case I results in a null component of $\boldsymbol{v}$ along the projection ray of the point feature (i.e., $v_z = 0$) and in an associated circular trajectory centered on the tracked point as predicted by the theoretical analysis of Section IV-A. This then allows us to move faster in the "useful" directions (while keeping the same

constant $\|\boldsymbol{v}\|$), and, thus, to increase the value of σ_1^2 toward its theoretical maximum $\sigma^2_{\max} = 0.0025$ [see Fig. 6(c)], resulting in an overall faster convergence for the estimation error [see Fig. 6(a)]. Furthermore, Fig. 6(a) also reports the ideal response of (9) with desired poles at $\sigma^2_{\max}$ (dashed black line). We can then note the almost perfect match with case I (solid red line): indeed, as explained in Remark II.4, imposing a $\sigma_1^2(t) = const$ allows us to exactly obtain the ideal behavior governed by (9). It is finally worth noting the accuracy of the reconstructed depth: Fig. 6(a) reports two vertical-dashed lines indicating, for the two cases under consideration, the times $T_1 = 4.95$ s and $T_2 = 9.85$ s at which the estimation error $z(t)$ becomes smaller than 5 mm. We then obtained a standard deviation of approximately 0.8 and 0.3 mm evaluated on a time window of 1 s after the times T_1 and T_2, respectively. These results then also confirm the robustness of the proposed estimation approach despite the unavoidable presence of noise and discretization in the image acquisition. Note also that, as expected, the estimation error in the (active) case I reaches "convergence" (i.e., drops below the threshold of 5 mm) significantly faster than case II ($T_1 < T_2$).

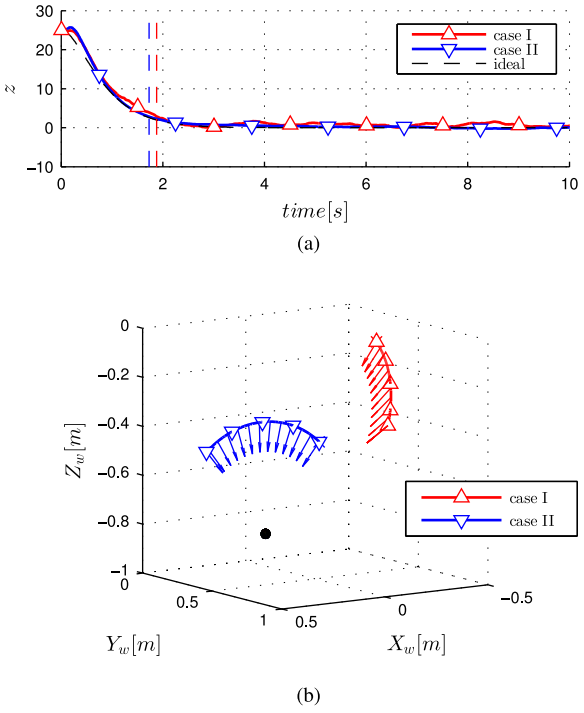


Fig. 7. Experimental results for the estimation of the radius of a sphere using different constant camera velocities with the same norm. (a) Behavior of the estimation error $z(t)$ for the two cases (solid blue and red lines), and for an "ideal" second-order system with poles at the desired locations (dashed black line). The vertical-dashed lines indicate the times at which the estimation error $z(t)$ drops below the threshold of 1 mm. (b) Camera trajectories for case I (blue line) and case II (red line) with arrows indicating the direction of the camera optical axis.

C. Structure Estimation for a Spherical Target

We now discuss some experimental results concerning the estimation of the radius of a spherical target: indeed, as explained in Section III-B, estimation of R allows us to fully recover the sphere 3-D position $\boldsymbol{P}_0 = \boldsymbol{s}R$, where vector $\boldsymbol{s}$ is directly obtainable from image measurements, see (22). As object to be tracked, we made use of a white table tennis ball placed on a black table and with a radius of 1.9 cm. As explained in Section III-B, the convergence rate of the estimation error for the sphere case only depends on the norm of the linear velocity $\|\boldsymbol{v}\|$ and not on its direction. This fact is proved by the first experiment, where the estimation task is run twice starting from two different positions and imposing two different camera velocities but with same norm. These values were used during the experiments: $\alpha\beta = 2 \cdot 10^3$ for gains $\boldsymbol{Q}$ and $\boldsymbol{\Lambda}$, $c_1 = c_1^* = 2\sqrt{\alpha\beta}\sigma_1$ for $\boldsymbol{D}_1$ in 6, and $\boldsymbol{v} = (-0.05,\, 0,\, 0)$ m/s for case I and $\boldsymbol{v} = (0,\, 0.045,\, 0.02)$ m/s for case II, with $\|\boldsymbol{v}\| = 0.05$ m/sin both cases. The camera angular velocity $\boldsymbol{\omega}$ was exploited to keep $(s_x,\, s_y) \simeq (0,\, 0)$ (centered sphere).

Fig. 7(a) shows the behavior of the estimation errors (solid blue and red lines): note how the error transient response for the two cases is essentially coincident, as well as equivalent to that of the *reference second order system* (9) with the desired poles, i.e., by setting $\sigma_1^2 = \|\boldsymbol{v}\|^2 = const$ and $c_1 = c_1^*$ in (9) (dashed black line). The higher noise level in case II (red line) is due to the larger distance between the camera and the spherical target [see Fig. 7(b)], which negatively affects the estimation task. The standard deviation of the radius estimation error, computed on a time window of 1 s after $z(t)$ has become smaller than 1 mm (vertical-dashed lines in the plot), is 0.3 mm for case I and 0.2 mm for case II: we can note, again, the very satisfactory results obtained with the proposed estimation scheme in terms of accuracy of the reconstructed sphere radius. Note also how, in the two cases, the estimation error $z(t)$ drops below the threshold of 1 mm at essentially the same time, as expected (same error transient response).

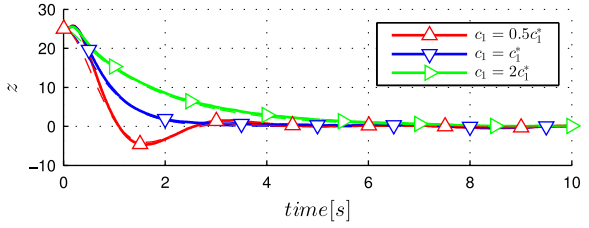


Fig. 8. Experimental results for the estimation of the radius of a sphere with $c_1 = c_1^*$ (blue line), $c_1 = 2c_1^*$ (green line) and $c_1 = 0.5c_1^*$ (red line). The dashed lines represent the response of an "ideal" second-order system with the corresponding poles. Note again the almost perfect match between the plots.

Since the direction of the velocity does not play any role in this case, no optimization of σ_1^2 can be performed under the constraint $\|\boldsymbol{v}\| = const$. On the other hand, the analysis of Section II-B clearly indicates the importance of choosing a proper value of c_1 for the damping matrix $\boldsymbol{D}_1$ in (6). To show this fact, we report here three experiments characterized by the same camera trajectory of the previous case I, but by employing three different values for c_1, that is, c_1^*, $2c_1^*$, and $0.5c_1^*$. These correspond to a critically damped, overdamped, and underdamped response for the ideal system (9), respectively. The experimental results reported in Fig. 8 show that the behavior of the estimation error z (solid lines) has an excellent match with that of (9) (represented

by dashed lines), thus fully confirming 1) the validity of the proposed theoretical analysis, and 2) the importance of choosing the "right" damping matrix $\boldsymbol{D}_1$ for optimizing the convergence speed in addition to a proper regulation of σ_1^2.

D. Structure Estimation for a Cylindrical Target

In this final section, we report some experimental results concerning the active estimation of the radius of a cylindrical object. Indeed, as in the sphere case, knowledge of R allows us to fully recover the 3-D point $\boldsymbol{P}_0 = R\boldsymbol{s}$, with vector $\boldsymbol{s}$ from (33) and the cylinder axis $\boldsymbol{a}$ in (35) being directly obtainable from image measurements. For these experiments, we used a white cardboard cylinder placed on a black table. The radius of the cylinder was approximately 4.2 cm.

In the cylinder case, the convergence rate of the estimation error depends both on the norm of the camera linear velocity $\boldsymbol{v}$ and on its direction with respect to the cylinder axis $\boldsymbol{a}$, see (38). It is then interesting to optimize the direction of $\boldsymbol{v}$ under the constraint $\|\boldsymbol{v}\| = const$ for maximizing the eigenvalue σ_1^2 (i.e., so as to obtain the fastest convergence rate for a given "control effort" $\|\boldsymbol{v}\|$).

From (39), maximization of $\sigma_1^2(t)$ with respect to vector $\boldsymbol{v}$ can be obtained by choosing

$$\dot{\boldsymbol{v}} = \boldsymbol{J}_{v,1}^T - \boldsymbol{J}_{v,1}^{\dagger}\boldsymbol{J}_{a,1}[\boldsymbol{a}]_{\times}\boldsymbol{\omega} \tag{41}$$

with $\boldsymbol{A}^{\dagger}$ being the pseudoinverse of a matrix $\boldsymbol{A}$, i.e., by following the gradient of σ_1^2 with respect to $\boldsymbol{v}$ and by compensating for the (known) effects of input $\boldsymbol{\omega}$. In order to additionally enforce the constraint $\|\boldsymbol{v}\| = const$ during the eigenvalue maximization, (41) can be modified as

$$\dot{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|^2}k_1\left(\kappa_{des} - \kappa\right) + k_2\left(\boldsymbol{I}_3 - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^2}\right) \times \left(\boldsymbol{J}_{v,1}^T - \boldsymbol{J}_{v,1}^{\dagger}\boldsymbol{J}_{a,1}[\boldsymbol{a}]_{\times}\boldsymbol{\omega}\right) \tag{42}$$

with $k_1 > 0$ and $k_2 > 0$. Analogously to the point feature case, the first term in (42) asymptotically guarantees $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}_0\|$, while the second term projects (41) onto the null space of the constraint $\|\boldsymbol{v}(t)\| = const$. As for the angular velocity $\boldsymbol{\omega}$, we exploited it for keeping the axis of the cylinder at the center of the image plane by regulating $(s_x,\, s_y)$ to $(0,\, 0)$.

We now present three experimental results structured as follows: in the first experiment (case I), the update rule (42) is fully implemented ($k_1 > 0, k_2 > 0$) for actively optimizing the direction of $\boldsymbol{v}$. In the second experiment (case II), the camera starts from the same initial pose and velocity as in case I, but (42) is implemented with $k_1 > 0$ and $k_2 = 0$, i.e., without performing any optimization of σ_1^2. Finally, in the third experiment (case III), the camera starts from a different initial pose and with a different velocity direction (but same norm) with respect to the previous two cases, and (42) is again fully implemented. This last case is meant to show how the convergence properties of the estimator are not affected by the direction of the camera linear velocity as long as it stays orthogonal to the cylinder axis $\boldsymbol{a}$.

The experiments were run with the following conditions: $\alpha\beta = 500$ for gains $\boldsymbol{Q}$ and $\boldsymbol{\Lambda}$, $c_1 = c_1^*$ for $\boldsymbol{D}_1$ in (6), $k_1 = 10$, $k_2 = 1$ for cases I and III, and $k_2 = 0$ for case II. As for the linear velocity, we set $\boldsymbol{v}(t_0) = \boldsymbol{v}_0 = (-0.01,\, 0.05,\, 0.05)$ m/s for cases I and II, and $\boldsymbol{v}(t_0) = \boldsymbol{v}_0 = (-0.05,\, 0.05,\, 0.01)$ m/s for case III (note how $\|\boldsymbol{v}_0\|^2 = 5.1 \times 10^{-3}$ m^2/s^2 in all three cases).

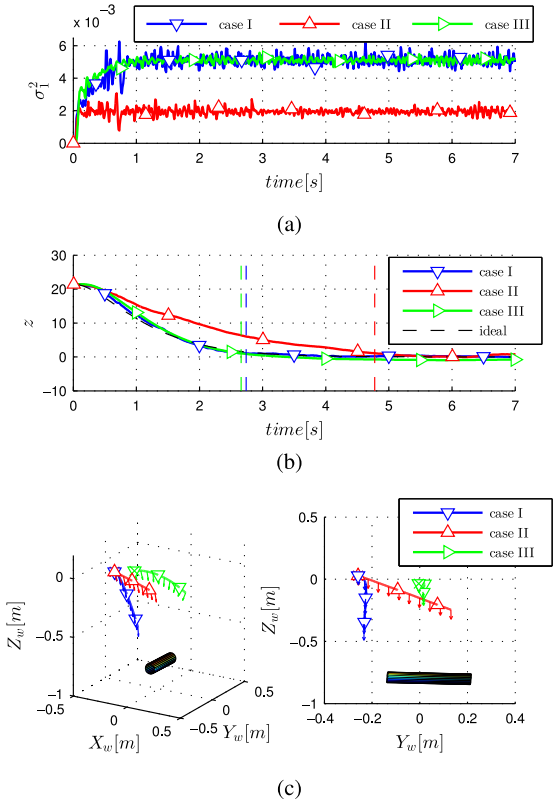


Fig. 9. Experimental results for the estimation of the radius of a cylinder with the following color coding: blue–case I, red–case II, green–case III. (a) Behavior of $\sigma_1^2(t)$ for the three cases (coincident for cases I and III and larger than in case II). (b) Behavior of $z(t)$. The three vertical-dashed lines indicate the times $T_1 = 2.74$ s, $T_2 = 4.78$ s and $T_3 = 2.66$ s at which the estimation error drops below the threshold of 2 mm. Note how $T_1 \approx T_3$ and $T_1 < T_2$ as expected. (c) Two views of the camera trajectories for the three cases with arrows indicating the direction of the camera optical axis.

The behavior of $\sigma_1^2(t)$ is shown in Fig. 9(a): as explained at the end of Section III-C, under the constraint $\|\boldsymbol{v}\| = const$, one has $\max_{\boldsymbol{v}} \sigma_1^2 = \|\boldsymbol{v}\|^2$ as the largest possible value for σ_1^2 (obtained when $\boldsymbol{v}^T\boldsymbol{a} = 0$). It is then possible to verify that, indeed, $\sigma_1^2(t) \to \|\boldsymbol{v}_0\|^2$ in cases I and III despite the different initial conditions of the experiments (different camera pose and direction of $\boldsymbol{v}$). The optimization in (42) results in a null component of $\boldsymbol{v}$ along $\boldsymbol{a}$, thus allowing us to move faster in the "useful" directions (while keeping a constant $\|\boldsymbol{v}\|$), and to increase the value of σ_1^2 to its maximum possible value. In addition, note how the value of $\sigma_1^2(t)$ for case II results smaller than in the other two cases (as expected) since no optimization is present in this case.

The behavior of the estimation error $z(t)$ is shown in Fig. 9(b): again, we can note that the transient response for cases I and III results essentially coincident and in almost perfect agreement with that of the reference system (9) with desired poles (dashed black line). As expected, the response for case II (red line) is

slower than in cases I and III. As in the point feature case, Fig. 9(b) reports, for the three cases under consideration, the times $T_1 = 2.74$ s, $T_2 = 4.78$ s, and $T_3 = 2.66$ s at which the estimation error drops below the threshold 2 mm (vertical-dashed lines). The standard deviation of the error evaluated on a time window of 1 s after convergence has been "reached" resulting in the values of 0.4, 0.6, and 0.7 mm, respectively. We can then appreciate, again, the high accuracy of the proposed approach in estimating the cylinder radius R, while also optimizing online for the camera motion. The higher estimation error in case III can be ascribed to the larger distance between the camera and the observed target, which increases the effect of discretization errors. Note also how $T_1 \approx T_3 < T_2$ thanks to the active optimization of the error convergence rate. Finally, Fig. 9(c) depicts the camera trajectories for the three experiments with arrows indicating the direction of the optical axis. In case II, the camera simply travels along a straight line ($\boldsymbol{v}(t) \equiv \boldsymbol{v}_0$), while in cases I and III the direction of $\boldsymbol{v}$ is suitably modified resulting in a trajectory lying on a plane orthogonal to $\boldsymbol{a}$.

V. Conclusion

In this paper, we have addressed the problem of active SfM for recovering the 3-D structure of a point feature and of spherical and cylindrical objects by exploiting a novel active estimation strategy tailored to the three cases under consideration. For the depth estimation of a point feature, two possibilities differing in the adopted projection model (planar or spherical) were proposed and critically compared. The results showed the complementarity of the two models in terms of attainable convergence rates and basin of attraction for the estimation error. In the spherical and cylindrical cases, we instead showed how an adequate choice of the measured visual features allows us to reduce the SfM task to the estimation of a single unknown constant quantity (the sphere/cylinder radius R) in place of the classical (and time-varying) three parameters (scaled normal vector of the planar limb surface). Availability of this quantity allows us to then retrieve the full 3-D structure of the observed targets. The reported experimental results fully confirmed the validity of the theoretical analysis and, in particular, the ability of the proposed *active* estimation strategy to impose, in all three cases, a *desired transient response* to the estimation error equivalent to that of a reference linear second-order system with desired poles.

We are currently investigating the use of similar active strategies for dealing with more complex 3-D scenes. A possibility in this sense could be to decompose the SfM problem in two phases by 1) extracting and classifying, possibly from an initial measurement in the form of a point cloud, a set of primitive shapes belonging to the classes described in our work (points, spheres, cylinders or other 3-D geometries), or also to other classes, such as 2-D planar patches made of discrete/dense sets of points; 2) performing an (active) estimation of the whole scene structure by applying the same strategy presented here and by either sequencing the estimation of single scene components, or considering an "extended" system obtained by concatenating the observable and unobservable components corresponding to each of the basic shapes/classes.

We are also investigating how to extend our solution to problems involving the estimation of more parameters than the number of available measurements (i.e., with $m < p$), thus requiring to fulfil the more general observability condition 4. Finally, we are also interested in the use of the proposed active strategy in the context of vision-based manipulation tasks. Some preliminary results in this context are reported in [32] where it is shown, and experimentally proven, that an online optimization of the estimation convergence rate can improve the performance in executing visual servoing tasks.

Appendix A
Time-Derivative of the Limb Surface Parameters for a Spherical Target

Differentiation of $\boldsymbol{\chi}$ from (21) with respect to time yields

$$\dot{\boldsymbol{\chi}} = \frac{\dot{\boldsymbol{P}}_0 K - \boldsymbol{P}_0 \dot{K}}{K^2} = \frac{\dot{\boldsymbol{P}}_0 K - 2\boldsymbol{P}_0 \boldsymbol{P}_0^T \dot{\boldsymbol{P}}_0}{K^2} \tag{43}$$

which being $\dot{\boldsymbol{P}}_0 = -\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{P}_0$ and exploiting the property $\boldsymbol{P}_0^T [\boldsymbol{\omega}]_\times \boldsymbol{P}_0 = 0$, can be rewritten as

$$\begin{aligned} \dot{\boldsymbol{\chi}} &= -\frac{\boldsymbol{v}}{K} - \frac{[\boldsymbol{\omega}]_\times \boldsymbol{P}_0}{K} + 2\frac{\boldsymbol{P}_0 \boldsymbol{P}_0^T \boldsymbol{v}}{K^2} \\ &= -\frac{\boldsymbol{v}}{K} - [\boldsymbol{\omega}]_\times \boldsymbol{\chi} + 2\boldsymbol{\chi}\boldsymbol{\chi}^T \boldsymbol{v}. \end{aligned} \tag{44}$$

Letting $s_z = Z_0/R$ ($s_z > 1$), one then has

$$\boldsymbol{\chi}^T \boldsymbol{\chi} - \frac{1}{s_z^2}\chi_z^2 = \frac{X_0^2 + Y_0^2 + Z_0^2}{K^2} - \frac{R^2}{Z_0^2}\frac{Z_0^2}{K^2} = \frac{1}{K}. \tag{45}$$

This then shows how $1/K$ can be expressed in terms of $\boldsymbol{\chi}$ and of s_z^2, with s_z being directly obtainable from image measurements, see (22).

Appendix B
Estimation of the Limb Surface Parameter for a Cylindrical Target

In order to estimate the parameters of the limb surface associated with a cylindrical object, one could consider as measurement the $2+2$ angle-distance parameters (θ_i, ρ_i) of the straight lines resulting from the projection of the cylinder on the image plane. From [17], [27], the interaction matrix in this case is given by

$$\boldsymbol{L} = \begin{bmatrix} \lambda_{\rho_1} c_1 & \lambda_{\rho_1} s_1 & -\lambda_{\rho_1}\rho_1 & (1+\rho_1^2)s_1 & -(1+\rho_1^2)c_1 & 0 \\ \lambda_{\theta_1} c_1 & \lambda_{\theta_1} s_1 & -\lambda_{\theta_1}\rho_1 & -\rho_1 c_1 & -\rho_1 s_1 & -1 \\ \lambda_{\rho_2} c_2 & \lambda_{\rho_2} s_2 & -\lambda_{\rho_2}\rho_2 & (1+\rho_2^2)s_2 & -(1+\rho_2^2)c_2 & 0 \\ \lambda_{\theta_2} c_2 & \lambda_{\theta_2} s_2 & -\lambda_{\theta_2}\rho_2 & -\rho_2 c_2 & -\rho_2 s_2 & -1 \end{bmatrix} \tag{46}$$

with $s_i = \sin\theta_i$, $c_i = \cos\theta_i$, and

$$\begin{cases} \lambda_{\rho_i} &= -(\chi_x \rho_i c_i + \chi_y \rho_i s_i + \chi_z) \\ \lambda_{\theta_i} &= \chi_y c_i - \chi_x s_i. \end{cases} \tag{47}$$

Therefore, being (46), (47) linear in the unknown $\boldsymbol{\chi}$, one can again apply the estimation scheme (2) with $\boldsymbol{s}$ taken as

the vector of measured quantities on the image plane, i.e., $\boldsymbol{s} = (\rho_1, \theta_1, \rho_2, \theta_2)$.

As for the dynamics of $\boldsymbol{\chi}$, since (21) still holds for a cylindrical object (see [27]), one can again exploit (43) with, however, in this case

$$\dot{\boldsymbol{P}}_0 = -\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right)\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{P}_0$$

and thus

$$\dot{\boldsymbol{\chi}} = -\left(\frac{1}{K}\boldsymbol{I}_3 - 2\boldsymbol{\chi}\boldsymbol{\chi}^T\right)\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right)\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{\chi}.$$

Finally, one can invoke (45) in order to express $1/K$ as a function of $\boldsymbol{\chi}$ and s_z^2, with s_z being the third element of vector $\boldsymbol{s}$ in (33).

APPENDIX C
DERIVATION OF EQUATION (36)

We note that the cylinder axis $\boldsymbol{a}$ can be determined by the intersection of two planes $\mathcal{P}_i : \boldsymbol{r}_i^T \boldsymbol{X} - d_i = 0, i = 1, 2$, with

$$\boldsymbol{r}_1 = \frac{[\boldsymbol{a}]_\times \boldsymbol{P}_0}{\|\boldsymbol{P}_0\|},\ d_1 = 0, \quad \boldsymbol{r}_2 = -\frac{\boldsymbol{P}_0}{\|\boldsymbol{P}_0\|},\ d_2 = \|\boldsymbol{P}_0\| \quad (48)$$

(see Fig. 3). In particular, plane $\mathcal{P}_1$ passes through the camera optical center, it is orthogonal to plane $\mathcal{P}_2$, and both planes contain the axis $\boldsymbol{a}$ passing through $\boldsymbol{P}_0$ (by construction).

Since $R\boldsymbol{s} = \boldsymbol{P}_0$ and $\boldsymbol{P}_0$ belongs to the cylinder axis $\boldsymbol{a}$, we have $R\boldsymbol{r}_i^T \boldsymbol{s} - d_i = 0$, $i = 1, 2$ (the point $R\boldsymbol{s}$ belongs to both planes $\mathcal{P}_i$). Taking the time derivative of these latter constraints (with $R = const$), one has

$$\boldsymbol{r}_i^T \dot{\boldsymbol{s}} = \frac{1}{R}\dot{d}_i - \boldsymbol{s}^T \dot{\boldsymbol{r}}_i, \quad i = 1, 2. \quad (49)$$

Since $\dot{\boldsymbol{r}}_i = [\boldsymbol{r}_i]_\times \boldsymbol{\omega}$ and $\dot{d}_i = \boldsymbol{r}_i^T \boldsymbol{v}$ (see [10]), (49) can be rewritten as

$$\boldsymbol{r}_i^T \dot{\boldsymbol{s}} = \frac{1}{R}\boldsymbol{r}_i^T \boldsymbol{v} - \boldsymbol{s}^T [\boldsymbol{r}_i]_\times \boldsymbol{\omega}, \quad i = 1, 2. \quad (50)$$

Finally, from $\boldsymbol{a}^T \boldsymbol{P}_0 = 0$ and $\boldsymbol{P}_0 = R\boldsymbol{s}$, we have $\boldsymbol{a}^T \boldsymbol{s} = 0$ implying that

$$\boldsymbol{a}^T \dot{\boldsymbol{s}} = -\boldsymbol{s}^T \dot{\boldsymbol{a}} = -\boldsymbol{s}^T [\boldsymbol{a}]_\times \boldsymbol{\omega}. \quad (51)$$

We now note that (50), (51) provide three linear constraints for $\dot{\boldsymbol{s}}$ that, by using (48), can be rearranged in matrix form as the following linear system:

$$\begin{bmatrix} \frac{\boldsymbol{P}_0^T}{\|\boldsymbol{P}_0\|} \\ \boldsymbol{a}^T \\ \frac{\left([\boldsymbol{a}]_\times \boldsymbol{P}_0\right)^T}{\|\boldsymbol{P}_0\|} \end{bmatrix} \dot{\boldsymbol{s}} = \frac{1}{R}\begin{bmatrix} \frac{\boldsymbol{P}_0^T}{\|\boldsymbol{P}_0\|}\boldsymbol{v} \\ -\boldsymbol{P}_0^T [\boldsymbol{a}]_\times \boldsymbol{\omega} \\ \|\boldsymbol{P}_0\| \boldsymbol{a}^T \boldsymbol{\omega} + \frac{\left([\boldsymbol{a}]_\times \boldsymbol{P}_0\right)^T}{\|\boldsymbol{P}_0\|}\boldsymbol{v} \end{bmatrix}. \quad (52)$$

It is easy to verify that the 3×3 matrix on the left-hand side of (52) is orthonormal: by then solving (52) for $\dot{\boldsymbol{s}}$ and performing some simplifications, we finally obtain the sought result

$$\dot{\boldsymbol{s}} = \begin{bmatrix} -\frac{1}{R}\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right) & [\boldsymbol{s}]_\times \end{bmatrix} \boldsymbol{u}. \quad (53)$$

REFERENCES

[1] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *J. Image Vis. Comput.*, vol. 30, no. 2, pp. 65–77, 2012.

[2] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *Int. J. Comput. Vis.*, vol. 3, no. 3, pp. 209–238, 1989.

[3] S. Soatto, R. Frezza, and P. Perona, "Motion estimation via dynamic vision," *IEEE Trans. Autom. Control*, vol. 41, no. 3, pp. 393–413, Mar. 1996.

[4] J. Civera, A. J. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.

[5] J. Civera, Ó. Grasa, A. J. Davison, and J. M. M. Montiel, "1-Point RANSAC for EKF filtering: Application to real-time structure from motion and visual odometry," *J. Field Robot.*, vol. 27, no. 5, pp. 609–631, 2010.

[6] S. Omari and G. Ducard, "Metric visual-inertial navigation system using single optical flow feature," in *Proc. Eur. Control Conf.*, 2013, pp. 1310–1316.

[7] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, "Range identification for perspective vision systems," *IEEE Trans. Autom. Control*, vol. 48, no. 12, pp. 2232–2238, Dec. 2003.

[8] N. Metni and T. Hamel, "Visual tracking control of aerial robotic systems with adaptive depth estimation," *Int. J. Control, Autom., Syst.*, vol. 1, no. 5, pp. 51–60, 2007.

[9] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int. J. Robot. Res.*, vol. 27, no. 10, pp. 1093–1116, 2008.

[10] P. Robuffo Giordano, A. De Luca, and G. Oriolo, "3D structure identification from image moments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 93–100.

[11] F. Morbidi, G. L. Mariottini, and D. Prattichizzo, "Observer design via immersion and invariance for vision-based leader-follower formation control," *Automatica*, vol. 46, no. 1, pp. 148–154, 2010.

[12] M. Sassano, D. Carnevale, and A. Astolfi, "Observer design for range and orientation identification," *Automatica*, vol. 46, no. 8, pp. 1369–1375, 2010.

[13] A. Durand Petiteville, M. Courdesses, and V. Cadenat, "A new predictor/corrector pair to estimate the visual features depth during a vision-based navigation task in an unknown environment," in *Proc. Int. Conf. Informat. Control, Autom. Robot.*, 2010, pp. 268–274.

[14] A. Martinelli, "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 44–60, Feb. 2012.

[15] A. P. Dani, N. R. Fischer, and W. E. Dixon, "Single camera structure and motion," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 238–243, Jan. 2012.

[16] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano, "A comparison of scale estimation schemes for a quadrotor UAV based on optical flow and IMU measurements," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2013, pp. 5193–5200.

[17] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 5, pp. 492–504, May 1996.

[18] A. J. Davison and D. W. Murray, "Mobile robot localisation using active vision," in *Proc. Eur. Conf. Comput. Vis.*, 1998, pp. 809–825.

[19] M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "Path planning for motion dependent state estimation on micro aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3926–3932.

[20] R. Spica and P. Robuffo Giordano, "A framework for active estimation: Application to structure from motion," in *Proc. 52nd IEEE Conf. Decision Control*, 2013, pp. 7647–7653.

[21] R. Mahony, A. von Brasch, P. Corke, and T. Hamel, "Adaptive depth estimation in image based visual servo control of dynamic systems," in *Proc. IEEE Conf. Decision Control*, 2005, pp. 5372–5378.

[22] P. Corke, "Spherical image-based visual servo and structure estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 5550–5555.

[23] B. Guerreiro, P. Batista, C. Silvestre, and P. Oliveira, "Globally asymptotically stable sensor-based simultaneous localization and mapping," *IEEE Trans. Robot.*, vol. 29, no. 6, pp. 1380–1395, Dec. 2013.

[24] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: Recent results with the

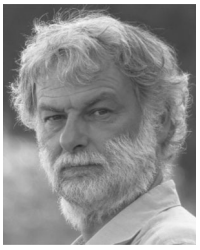
DLR-light-weight-arms," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 3704–3709.
[25] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
[26] T. Hamel and R. Mahony, "Visual servoing of an under-actuated rigid body system: An image based approach," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 187–198, Apr. 2002.
[27] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, Aug. 2004.
[28] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.
[29] R. T. Fomena and F. Chaumette, "Improvements on visual servoing from spherical targets using a spherical projection model," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 874–886, Aug. 2009.
[30] F. Chaumette, "Visual servoing using image features defined upon geometrical primitives," in *Proc. IEEE Conf. Decision Control*, 1994, pp. 3782–3787.
[31] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: A generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.
[32] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Coupling visual servoing with active structure from motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 3090–3095.

Riccardo Spica (M'12) received the M.Sc. degree in electronic engineering from University of Rome "La Sapienza," Rome, Italy, in 2012. Since December 2013, he has been working toward the Ph.D. degree in signal processing at University of Rennes, Rennes, France, within the Lagadic Group of Irisa and Inria.

He was a Master's Student and later a Graduate Research Assistant with Max Planck Institute for Biological Cybernetics, Tuebingen, Germany, for one year between 2012 and 2013. His research interests include planning and control for robotics applications, particularly visual servoing and active structure from motion.

Paolo Robuffo Giordano (M'08) received the M.Sc. degree in computer science engineering in 2001 and the Ph.D. degree in systems engineering in 2008, both from University of Rome "La Sapienza," Rome, Italy.

In 2007 and 2008, he spent one year as a Postdoctoral Researcher with Institute of Robotics and Mechatronics of the German Aerospace Center, and from 2008 to 2012 he was a Senior Research Scientist with Max Planck Institute for Biological Cybernetics, Tübingen, Germany. Since 2012 he has been a CNRS Researcher with Lagadic Group, Irisa and Inria, Rennes, France. He is an Associate Editor of IEEE TRANSACTIONS ON ROBOTICS. His research interests include nonlinear control, robotics, planning, haptics, and VR applications.



François Chaumette (M'02–SM'09–F'13) graduated from École Nationale Supérieure de Mécanique, Nantes, France, in 1987. He received the Ph.D. degree in computer science from University of Rennes, Rennes, France, in 1990.

Since 1990 he has been with Inria, Rennes, where he is currently a Senior Research Scientist and Head of the Lagadic Group. His research interests include robotics and computer vision, especially visual servoing and active perception. He has been an Associate Editor of IEEE TRANSACTIONS ON ROBOTICS from 2001 to 2005 and is currently on the Editorial Board of *International Journal of Robotics Research*.

Dr. Chaumette received the AFCET/CNRS Prize for the best French thesis in automatic control in 1991. He also received the 2002 King-Sun Fu Memorial Best IEEE Transactions on Robotics and Automation Paper Award.

Journal Publications

[J1] P. Robuffo Giordano, M. Vendittelli, J.-P. Laumond, and P. Souères. Nonholonomic distance to polygonal obstacles for a car-like robot of polygonal shape. *IEEE Transactions on Robotics*, 22(5):1040–1047, 2006.

[J2] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Image-based visual servoing schemes for nonholonomic mobile manipulators. *Robotica*, 25(2):131–145, 2007.

[J3] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Feature Depth Observation for Image-based Visual Servoing: Theory and Experiments. *The International Journal of Robotics Research*, 27(10):1093–1116, 2008.

[J4] J. L. Souman, P. Robuffo Giordano, I. Frissen, A. De Luca, and M. O. Ernst. Making virtual walking real: perceptual evaluation of a new treadmill control algorithm. *ACM Transactions on Applied Perception*, 7(2), 2009.

[J5] P. Robuffo Giordano and M. Vendittelli. Shortest paths to obstacles for a polygonal Dubins car. *IEEE Transactions on Robotics*, 25(5):1184–1191, 2009.

[J6] F. Soyka, P. Robuffo Giordano, K. Beykirch, and H. H. Bülthoff. Predicting Direction Detection Thresholds for Arbitrary Translational Acceleration Profiles in the Horizontal Plane. *Experimental Brain Research*, 209(1):95–107, 2011.

[J7] J. L. Souman, P. Robuffo Giordano, M. Schwaiger, I. Frissen, T. Thümmel, H. Ulbrich, A. De Luca, H. H. Bülthoff, and M. O. Ernst. CyberWalk: Enabling unconstrained omnidirectional walking through virtual environments. *ACM Transactions on Applied Perception*, 8(4), 2011.

[J8] F. Soyka, P. Robuffo Giordano, M. Barnett-Cowan, and H. H. Bülthoff. Modeling direction discrimination thresholds for yaw rotations around an earth-vertical axis for arbitrary motion profiles. *Experimental Brain Research*, 220(1):89–99, 2012.

[J9] S. Haddadin, A. Peer, and P. Robuffo Giordano. Guest Editorial: **Special Issue on Autonomous Physical Human-Robot Interaction**. *The International Journal of Robotics Research*, 31(13):1529–1530, 2012.

[J10] A. Franchi, C. Secchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of Groups of Mobile Robots with Time-Varying Topology. *IEEE Transactions on Robotics*, 28(5):1019–1033, 2012.

[J11] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Modeling and Control of UAV Bearing-Formations with Bilateral High-Level Steering. *The International*

Journal of Robotics Research, **Special Issue on 3D Exploration, Mapping and Surveillance**, 31(12):1504–1525, 2012.

[J12] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Shared Control: Balancing Autonomy and Human Assistance with a Group of Quadrotor UAVs. *IEEE Robotics and Automation Magazine,* **Special Issue on Aerial Robotics and the Quadrotor Platform**, 19(3):57–68, 2012.

[J13] A. De Luca, R. Mattone, P. Robuffo Giordano, H. Ulbrich, M. Schwaiger, M. Van den Bergh, E. Koller-Meier, and L. Van Gool. Motion Control of the CyberCarpet Platform. *IEEE Transactions on Control Systems Technology*, 21(2):410–427, 2013.

[J14] H. I. Son, A. Franchi, L. L. Chuang, J. Kim, H. H. Bülthoff, and P. Robuffo Giordano. Human-Centered Design and Evaluation of Haptic Cueing for Teleoperation of Multiple Mobile Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 43(2):597–609, 2013.

[J15] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. A Passivity-Based Decentralized Strategy for Generalized Connectivity Maintenance. *The International Journal of Robotics Research*, 32(3):299–323, 2013.

[J16] D. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano. Semi-Autonomous Haptic Teleoperation Control Architecture of Multiple Unmanned Aerial Vehicles. *IEEE/ASME Transactions on Mechatronics,* **Focused Section on Aerospace Mechatronics**, 18(4):1334–1345, 2013.

[J17] R. Spica, P. Robuffo Giordano, and F. Chaumette. Active Structure from Motion: Application to Point, Sphere and Cylinder. *IEEE Transactions on Robotics*, 30(6):1499–1513, 2014.

[J18] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Decentralized Rigidity Maintenance Control with Range Measurements for Multi-Robot Systems. *The International Journal of Robotics Research*, 34(1):105–128, 2015.

[J19] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. A Novel Overactuated Quadrotor UAV: Modeling, Control and Experimental Validation. *IEEE Transactions on Control Systems Technology*, 23(2):540–556, 2015.

[J20] V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. Robuffo Giordano. Nonlinear Ego-Motion Estimation from Optical Flow for Online Control of a Quadrotor UAV. *The International Journal of Robotics Research*, 34(8):1114–1135, 2015.

[J21] G. Antonelli, E. Cataldi, F. Arrichiello, P. Robuffo Giordano, S. Chiaverini, and A. Franchi. Adaptive Trajectory Tracking for Quadrotors MAVs in Presence of Parameter Uncertainties and External Disturbances. *submitted to the IEEE/ASME Transactions on Mechatronics*, 2015.

[J22] F. Flacco, P. Robuffo Giordano, and A. De Luca. A Reverse Priority Approach to Smooth Task Transitions. *submitted to the IEEE Transactions on Robotics*, 2015.

[J23] R. Spica, P. Robuffo Giordano, and F. Chaumette. Bridging Visual Control and Active Perception via a Large Projector Operator. *submitted to the IEEE Transactions on Robotics*, 2015.

[J24] A. Franchi and P. Robuffo Giordano. Online Leader Selection for Improved Collective Tracking and Formation Maintenance. *submitted to the IEEE Transactions on Control of Network Systems*, 2015.

[J25] C. Masone, P. Robuffo Giordano, and A. Franchi. A Framework for Shared Planning and Control with Integral Haptic Feedback. *submitted to the The International Journal of Robotics Research*, 2015.

[J26] S. Bista, P. Robuffo Giordano, and F. Chaumette. Appearance-based Indoor Navigation by IBVS using Line Segments. *submitted to the IEEE Robotics and Automation Letters*, 2015.

[J27] T. Nestmeyer, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi. Decentralized Simultaneous Multi-target Exploration using a Connected Network of Multiple Robots. *submitted to Autonomous Robots*, 2015.

Conference Publications

[C1] A. De Luca, G. Oriolo, L. Paone, and P. Robuffo Giordano. Experiments in visual feedback control of a wheeled mobile robot. In *Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation (ICRA 2002)*, pages 2073–2078, 2002.

[C2] A. De Luca, G. Oriolo, L. Paone, P. Robuffo Giordano, and M. Vendittelli. Visual-based planning and control for nonholonomic mobile robots. In *Proc. of the 10th IEEE Mediterranean Conference on Control and Automation (MED 2002)*, 2002.

[C3] A. De Luca, R. Mattone, and P. Robuffo Giordano. The motion control problem for the Cyber-Carpet. In *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation (ICRA 2006)*, pages 3532–3537, 2006.

[C4] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Kinematic Modeling and Redundancy Resolution for Nonholonomic Mobile Manipulators. In *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation (ICRA 2006)*, pages 1867–1873, 2006.

[C5] P. Robuffo Giordano and M. Vendittelli. The Minimum-Time Crashing Problem for the Dubin's Car. In *Proc. of the 8th Int. IFAC Symposium on Robot Control (SYROCO 2006)*, 2006.

[C6] A. De Luca, R. Mattone, and P. Robuffo Giordano. Feedback/Feedforward Schemes for Motion Control of the CyberCarpet. In *Proc. of the 8th Int. IFAC Symposium on Robot Control (SYROCO 2006)*, 2006.

[C7] A. De Luca, R. Mattone, and P. Robuffo Giordano. Acceleration-level control of the CyberCarpet. In *Proc. of the 2007 IEEE Int. Conf. on Robotics and Automation (ICRA 2007)*, pages 2330–2335, 2007.

[C8] A. De Luca, G. Oriolo, and P. Robuffo Giordano. On-Line Estimation of Feature Depth for Image-Based Visual Servoing Schemes. In *Proc. of the 2007 IEEE Int. Conf. on Robotics and Automation (ICRA 2007)*, pages 2823–2828, 2007.

[C9] P. Robuffo Giordano, A. De Luca, and G. Oriolo. 3D Structure Identification from Image Moments. In *Proc. of the 2008 IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*, pages 93–100, 2008.

[C10] A. De Luca, M. Ferri, G. Oriolo, and P. Robuffo Giordano. Visual Servoing with Exploitation of Redundancy: An Experimental Study. In *Proc. of the 2008 IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*, pages 3231–3237, 2008.

[C11] P. Robuffo Giordano, A. Stemmer, K. Arbter, and A. Albu-Schäffer. Robotic assembly of complex planar parts: An experimental evaluation. In *Proc. of the 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2008)*, pages 3775–3782, 2008.

[C12] P. Robuffo Giordano, M. Fuchs, A. Albu-Schäffer, and G. Hirzinger. On the Kinematic Modeling and Control of a Mobile Platform Equipped with Steering Wheels and Movable Legs. In *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation (ICRA 2009)*, pages 4080–4087, 2009.

[C13] M. Fuchs, C. Borst, P. Robuffo Giordano, A. Baumann, E. Krämer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze, R. Burger, F. Schmidt, T. Wimböck, and G. Hirzinger. Rollin' Justin – Design Considerations and Realization of a Mobile Platform for a Humanoid Upper Body. In *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation (ICRA 2009)*, pages 4131–4137, 2009.

[C14] C. Borst, T. Wimböck, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. Robuffo Giordano, R. Konietschke, W. Sepp, S. Fuchs, C. Rink, A. Albu-Schäffer, and G. Hirzinger. Rollin' Justin – Mobile Platform with Variable Base. In *Proc. of the 2009 IEEE Int. Conf. on Robotics and Automation,* **Best Video Award** *(ICRA 2009)*, pages 1597–1598, 2009.

[C15] F. Soyka, H. Teufel, K. Beykirch, P. Robuffo Giordano, J. Butler, F. M. Nieuwenhuizen, and H. H. Bülthoff. Does jerk have to be considered in linear motion simulation? In *Proc. of the 2009 AIAA Modeling and Simulation Technologies Conference*, 2009.

[C16] M. Niccolini, L. Pollini, M. Innocenti, P. Robuffo Giordano, H. Teufel, and H. H. Bülthoff. Towards Real-Time Aircraft Simulation with the MPI Motion Simulator. In *Proc. of the 2009 AIAA Modeling and Simulation Technologies Conference*, 2009.

[C17] A. De Luca, R. Mattone, P. Robuffo Giordano, and H. H. Bülthoff. Control Design and Experimental Evaluation of the 2D CyberWalk Platform. In *Proc. of the 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2009)*, pages 5051–5058, 2009.

[C18] P. Robuffo Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini, and H. H. Bülthoff. A Novel Framework for Closed-Loop Robotic Motion Simulation - Part II: Motion Cueing Design and Experimental Validation. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 3896–3903, 2010.

[C19] P. Robuffo Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini, and H. H. Bülthoff. A Novel Framework for Closed-Loop Robotic Motion Simulation - Part I: Inverse Kinematics Design. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 3876–3883, 2010.

[C20] A. De Luca, G. Oriolo, and P. Robuffo Giordano. Kinematic Control of Nonholonomic Mobile Manipulators in the Presence of Steering Wheels. In *Proc. of the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010)*, pages 1792–1798, 2010.

[C21] P. Robuffo Giordano, H. Deusch, J. Lächele, and H. H. Bülthoff. Visual-Vestibular Feedback for Enhanced Situational Awareness in Teleoperation of UAVs. In *Proc. of the AHS International 66th Annual Forum & Technological Display*, 2010.

[C22] S. M. C. Alaimo, L. Pollini, A. Magazzù, J.-P. Bresciani, P. Robuffo Giordano, M. Innocenti, and H. H. Bülthoff. Preliminary Evaluation of a Haptic Aiding Concept for Remotely Piloted Vehicles. In *Proc. of EuroHaptics 2010*, 2010.

[C23] V. Grabe, P. Pretto, P. Robuffo Giordano, and H. H. Bülthoff. Influence of Display Type on Drivers' Performance in a motion-based Driving Simulator. In *Proc. of the Driving Simulation Conference 2010 Europe (DSC 2010)*, 2010.

[C24] D. Lee, A. Franchi, P. Robuffo Giordano, H. I. Son, and H. H. Bülthoff. Haptic Teleoperation of Multiple Unmanned Aerial Vehicles over the Internet. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 1341–1347, Shanghai, China, May 2011.

[C25] A. Franchi, P. Robuffo Giordano, C. Secchi, H. I. Son, and H. H. Bülthoff. A Passivity-Based Decentralized Approach for the Bilateral Teleoperation of a Group of UAVs with Switching Topology. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 898–905, Shanghai, China, May 2011.

[C26] C. Masone, P. Robuffo Giordano, and H. H. Bülthoff. Mechanical Design and Control of the new 7-DOF CyberMotion Simulator. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation (ICRA 2011)*, pages 4935–4942, Shanghai, China, May 2011.

[C27] H. I. Son and J. Kim and L. Chuang and A. Franchi and P. Robuffo Giordano and D. Lee and H. H. Bülthoff. An evaluation of haptic cues on the tele-operator's perceptual awareness of multiple UAVs' environments. In *Proc. of the 2011 IEEE World Haptics Conference (WHC 2011)*, pages 149–154, Istanbul, Turkey, Jun. 2011.

[C28] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. Bilateral Teleoperation of Groups of UAVs with Decentralized Connectivity Maintenance. In *Proc. of the 2011 Robotics: Science and Systems Conference (RSS 2011)*, Los Angeles, CA, Jun. 2011.

[C29] H. I. Son, L. L. Chuang, A. Franchi, J. Kim, D. J. Lee, S. W. Lee, H. H. Bülthoff, and P. Robuffo Giordano. Measuring an Operator's Maneuverability Performance in the Haptic Teleoperation of Multiple Robots. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2011)*, pages 3039–3046, San Francisco, CA, Sep. 2011.

[C30] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff. Experiments of Passivity-Based Bilateral Aerial Teleoperation of a Group of UAVs with Decentralized Velocity Synchronization. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* **Invited paper** *(IROS 2011)*, pages 163–170, San Francisco, CA, Sep. 2011.

[C31] A. Franchi, C. Masone, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of Multiple UAVs with Decentralized Bearing-only Formation Control. In *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* **Invited paper** *(IROS 2011)*, pages 2215–2222, San Francisco, CA, Sep. 2011.

[C32] A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Distributed Online Leader Selection in the Bilateral Teleoperation of Multiple UAVs. In *Proc. of the 50th IEEE Conference on Decision and Control,* **Invited paper** *(CDC 2011)*, pages 3559–3565, Orlando, FL, Dec. 2011.

[C33] C. Secchi, P. Robuffo Giordano, and A. Franchi. Bilateral Teleoperation of Groups of Mobile Robots with Time-Varying Topology. In *Automatica.it 2011*, Pisa, Italy, Sep. 2011.

[C34] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 491–497, St. Paul, MN, May 2012.

[C35] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Modeling and Control of a Quadrotor UAV with Tilting Propellers. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 4606–4613, St. Paul, MN, May 2012.

[C36] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Teleoperation of a Group of UAVs with Communication Delays and Switching Topology. In *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation (ICRA 2012)*, pages 4307–4314, St. Paul, MN, May 2012.

[C37] D. Zelazo, A. Franchi, F. Allgöwer, H. H. Bülthoff, and P. Robuffo Giordano. Rigidity Maintenance Control for Multi-Robot Systems. In *Proc. of the 2012 Robotics: Science and Systems Conference (RSS 2012)*, Sydney, Australia, Jul. 2012.

[C38] M. Riedel, A. Franchi, H. H. Bülthoff, P. Robuffo Giordano, and H. I. Son. Experiments on Intercontinental Haptic Control of Multiple UAVs. In *Proc. of the 12th International Conference on Intelligent Autonomous System (IAS 2012)*, pages 227–238, Jeju Island, Korea, Jun. 2012.

[C39] R. Spica, A. Franchi, G. Oriolo, H. H. Bülthoff, and P. Robuffo Giordano. Aerial Grasping of a Moving Target with a Quadrotor UAV. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, pages 4985–4992, Vilamoura, Portugal, Oct. 2012.

[C40] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. Robust Optical-Flow Based Self-Motion Estimation for a Quadrotor UAV. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Portugal, Oct. 2012.

[C41] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Interactive Planning of Persistent Trajectories for Human-Assisted Navigation of Mobile Robots. In *Proc. of the 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2012)*, pages 2641–2648, Vilamoura, Portugal, Oct. 2012.

[C42] A. Nesti, C. Masone, M. Barnett-Cowan, P. Robuffo Giordano, H. H. Bülthoff, and P. Pretto. Roll rate thresholds and perceived realism in driving simulation. In *Proc. of the Driving Simulation Conference 2012 Europe (DSC 2012)*, 2012.

[C43] J. Lächele, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. SwarmSimX: Real-time Simulation Environment for Multi-robot Systems. In *Proc. of the 3rd Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2012)*, Tsukuba, Japan, Nov. 2012.

[C44] A. Franchi and P. Robuffo Giordano. Decentralized Control of Parallel Rigid Formations with Direction Constraints and Bearing Measurements. In *Proc. of the 51st IEEE Conference on Decision and Control (CDC 2012)*, pages 5310–5317, Maui, HI, Dec. 2012.

[C45] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Bilateral Control of the Degree of Connectivity in Multiple Mobile-robot Teleoperation. In *Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*, 2013.

[C46] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. First Flight Tests for a Quadrotor UAV with Tilting Propellers. In *Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation (ICRA 2013)*, 2013.

[C47] G. Antonelli, F. Arrichiello, S. Chiaverini, and P. Robuffo Giordano. Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances. In *Proc. of the 2013 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM 2013)*, 2013.

[C48] V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi. The TeleKyb Framework for a Modular and Extendible ROS-based Quadrotor Control. In *Proc. of the 6th European Conf. on Mobile Robots (ECMR 2013)*, 2013.

[C49] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. A Comparison of Scale Estimation Schemes for a Quadrotor UAV based on Optical Flow and IMU Measurements. In *Proc. of the 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2013)*, 2013.

[C50] G. Antonelli, E. Cataldi, P. Robuffo Giordano, S. Chiaverini, and A. Franchi. Experimental validation of a new adaptive control scheme for quadrotors MAVs. In *Proc. of the 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2013)*, 2013.

[C51] R. Spica and P. Robuffo Giordano. A Framework for Active Estimation: Application to Structure from Motion. In *Proc. of the 2013 IEEE Conf. on Decision and Control (CDC 2013)*, 2013.

[C52] R. Spica, P. Robuffo Giordano, M. Ryll, H. H. Bülthoff, and A. Franchi. An Open-Source Hardware/Software Architecture for Quadrotor UAVs. In *2nd Workshop on Research, Education and Development of Unmanned Aerial System*, 2013.

[C53] P. Robuffo Giordano, R. Spica, and F. Chaumette. An Active Strategy for Plane Detection and Estimation for a Monocular Camera. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C54] R. Spica, P. Robuffo Giordano, and F. Chaumette. Active Structure from Motion for Spherical and Cylindrical Targets. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C55] R. Spica, P. Robuffo Giordano, and F. Chaumette. Coupling Image-Based Visual Servoing with Active Structure from Motion. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C56] C. Masone, P. Robuffo Giordano, H. H. Bülthoff, and A. Franchi. Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control. In *2014 IEEE Int. Conf. on Robotics and Automation (ICRA 2014)*, 2014.

[C57] D. Zelazo, A. Franchi, and P. Robuffo Giordano. Rigidity Theory in $SE(2)$ for Unscaled Relative Position Estimation using only Bearing Measurements. In *Proc. of the 2014 European Control Conference (ECC 2014)*, 2014.

[C58] N. Cazy, C. Dune, P.-B. Wieber, P. Robuffo Giordano, and F. Chaumette. Pose Error Correction For Visual Features Prediction. In *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2014)*, 2014.

[C59] N. Cazy, P.-B. Wieber, P. Robuffo Giordano, and F. Chaumette. Visual Servoing when visual information is missing: experimental comparison of Visual Feature Prediction Schemes. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015.

[C60] R. Spica, P. Robuffo Giordano, and F. Chaumette. Plane Estimation by Active Vision from Point Features and Image Moments. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015.

[C61] P. Robuffo Giordano, R. Spica, and F. Chaumette. Learning the Shape of Image Moments for Optimal 3D Structure Estimation. In *2015 IEEE Int. Conf. on Robotics and Automation (ICRA 2015)*, 2015.

[C62] Omar Tahri, Paolo Robuffo Giordano, and Youcef Mezouar. Rotation Free Active Vision. In *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2015)*, 2015.

[C63] D. Zelazo, P. Robuffo Giordano, and A. Franchi. Bearing-Only Formation Control Using an $SE(2)$ Rigidity Theory. In *2015 IEEE Conf. on Decision and Control (CDC 2015)*, 2015.

[C64] F. Schiano, A. Franchi, D. Zelazo, and P. Robuffo Giordano. A Rigidity-Based Decentralized Bearing Formation Controller for Groups of Quadrotor UAVs. In *submitted to the 2016 IEEE Int. Conf. on Robotics and Automation (ICRA 2016)*, 2016.

[C65] S. R. Bista, P. Robuffo Giordano, and F. Chaumette. Appearance based Indoor Navigation by IBVS using Mutual Information. In *submitted to the 2016 IEEE Int. Conf. on Robotics and Automation (ICRA 2016)*, 2016.

Conference Talks and Workshops

[W1] P. Robuffo Giordano, J. L. Souman, R. Mattone, A. De Luca, M. O. Ernst, and H. H. Bülthoff. The CyberWalk Platform: Human-Machine Interaction Enabling Unconstrained Walking through VR. In *1st Int. Work. on Human-Friendly Robotics (HFR 2008)*, 2008.

[W2] P. Robuffo Giordano. Providing vestibular cues to a human operator for a new generation of human-machine interfaces. In *2nd Int. Work. on Human-Friendly Robotics (HFR 2009)*, 2009.

[W3] P. Robuffo Giordano, A. Franchi, H. I. Son, C. Secchi, D. Lee, and H. H. Bülthoff. Towards Bilateral Teleoperation of Multi-Robot Systems (hfr 2010). In *3rd Int. Work. on Human-Friendly Robotics*, Tübingen, Germany, Oct. 2010.

[W4] P. Robuffo Giordano. Decentralized Bilateral Aerial Teleoperation of Multiple UAVs - Part II: a Bottom-Up Perspective. In *RSS 2011 Work. on 3D Exploration, Mapping, and Surveillance with Aerial Robots (RSS 2011)*, Los Angeles, CA, Jun. 2011.

[W5] P. Robuffo Giordano. Towards Aerial Telerobotics: Enabling human operators to bilaterally control single/multiple UAVs for accomplishing remote tasks. In *euRobotics Forum: UAV Workshop*, Västeras, Sweden, April 2011.

[W6] A. Franchi, C. Masone, and P. Robuffo Giordano. A Synergetic High-level/Reactive Planning Framework with Application to Human-Assisted Navigation. In *2012 IEEE IROS Work. on Real-time Motion Planning: Online, Reactive, and in Real-time (IROS 2012)*, Vilamoura, Portugal, Oct. 2012.

[W7] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Shared Trajectory Planning for Human-in-the-loop Navigation of Mobile Robots in Cluttered Environments. In *5th Int. Work. on Human-Friendly Robotics (HFR 2012)*, Bruxelles, Belgium, Oct. 2012.

[W8] M. Riedel, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. Intercontinental Haptic Control and Advanced Supervisory Interfaces for Groups of Multiple UAVs. In *5th Int. Work. on Human-Friendly Robotics (HFR 2012)*, Bruxelles, Belgium, Oct. 2012.

[W9] F. Soyka, M. Barnett Cowan, P. Robuffo Giordano, and H. H. Bülthoff. Temporal processing of self-motion: Translations are processed slower than rotations. In *13th International Multisensory Research Forum (IMRF 2012)*, Oxford, UK, June 2012.

[W10] F. Soyka, M. Barnett Cowan, P. Robuffo Giordano, and H. H. Bülthoff. Translations are processed slower than rotations: reaction times for self-motion stimuli predicted by vestibular organ dynamics. In *27th Bárány Society Meeting*, Uppsala, Sweden, June 2012.

[W11] P. Robuffo Giordano. Shared Control of Multi-Robot Systems: Passivity, Decentralization, and Connectivity Maintenance. In *Workshop on Haptic Teleoperation of Mobile Robots: Theory, Applications and Perspectives at ICRA 2012*, St. Paul, MN, USA, May 2012.

[W12] T. Nestmeyer, P. Robuffo Giordano, and A. Franchi. Multi-target Simultaneous Exploration with Continual Connectivity. In *2th International Workshop on Crossing the Reality Gap: From Single to Multi- to Many Robot Systems at ICRA 2013*, Karlsruhe, Germany, May 2013.

[W13] T. Nestmeyer, M. Riedel, J. Lächele, S. Hartmann, F. Botschen, P. Robuffo Giordano, and A. Franchi. Interactive Demo: Haptic Remote Control of Multiple UAVs with Autonomous Cohesive Behavior. In *Int. Work. on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration at ICRA 2013*, Karlsruhe, Germany, May 2013.

[W14] J. Lächele, M. Riedel, P. Robuffo Giordano, and A. Franchi. SwarmSimX and TeleKyb: Two ROS-integrated Software Frameworks for Single- and Multi-Robot Applications. In *Int. Work. on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration at ICRA 2013*, Karlsruhe, Germany, May 2013.

[W15] T. Nestmeyer, P. Robuffo Giordano, and A. Franchi. Human-assisted Parallel Multi-target Visiting in a Connected Topology. In *6th Int. Work. on Human-Friendly Robotics (HFR 2013)*, Rome, Italy, September 2013.

[W16] M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Overactuation in UAVs for Enhanced Aerial Manipulation: A Novel Quadrotor Concept with Tilting. In *6th Int. Work. on Human-Friendly Robotics (HFR 2013)*, Rome, Italy, September 2013.

[W17] R. Spica, G. Claudio, F. Spindler, and P. Robuffo Giordano. Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms. In *ICRA 2014 Workshop: MATLAB/Simulink for Robotics Education and Research*, Hong Kong, China, July 2014.

[W18] R. Spica, P. Robuffo Giordano, and F. Chaumette. A Framework for Coupling Visual Control and Active Structure from Motion. In *ICRA 2015 Workshop: Scaling Up Active Perception*, Seattle, USA, May 2015.

Invited Talks and Seminars

[T1] P. Robuffo Giordano. **Invited Seminar**: Shortest Path Synthesis for a Reeds-Shepp Car. *Robotics Group, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Roma, Italy*, June 2005.

[T2] P. Robuffo Giordano. **Invited Seminar**: Introduction to Visual Servoing: Basic and Advanced Methods. *Institute for Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany*, May 2007.

[T3] P. Robuffo Giordano. **Invited Talk**: A control theory approach for modeling and control of robotic systems. *Max Planck Institute for Biological Cybernetics, Tübingen, Germany*, May 2008.

[T4] P. Robuffo Giordano. **Invited Talk**: Modeling and Control of a Wheeled Mobile Platform with Movable Legs. *Institute for Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany*, October 2008.

[T5] P. Robuffo Giordano. **Invited Seminar**: Introduction to the state estimation for dynamical systems. *Max Planck Institute for Biological Cybernetics, Tübingen, Germany*, November 2009.

[T6] P. Robuffo Giordano. **Invited Talk**: Towards a New Generation of Multisensory Human-Machine Interfaces for VR and Real-World Applications. *Kolloquium Technische Kybernetik, Institut für Systemtheorie und Regelungstechnik, Stuttgart, Germany*, January 2010.

[T7] P. Robuffo Giordano. **Invited Talk**: Human-Machine Interfaces for VR and Real World Applications. *Autonomous Systems Laboratory, Eidgenössische Technische Hochschule Zürich (ETH), Switzerland*, April 2010.

[T8] P. Robuffo Giordano. **Invited Talk**: Realizing a Closed-Loop Robotic Motion Simulator. *GRASP Lab: University of Pennsylvania, Philadelphia, USA*, May 2010.

[T9] P. Robuffo Giordano. **Invited Talk**: Towards Bilateral Teleoperation of Multi-Robot Systems. *Autonomous Intelligent Systems Lab, University of Freiburg, Germany*, November 2010.

[T10] P. Robuffo Giordano. **Invited Seminar**: An Introduction to Passivity and port-Hamiltonian Systems. *Robotics Group, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Roma, Italy*, December 2010.

[T11] P. Robuffo Giordano. **Invited Seminar**: Bilateral Teleoperation of Groups of Mobile Robots. *SIRSLab, Dipartimento di Ingegneria dell'Informazione Università di Siena, Siena, Italy*, November 2011.

[T12] P. Robuffo Giordano. **Invited Seminar**: Bilateral Teleoperation of Multi-Robot Systems: Passivity, Decentralization, and Connectivity Maintenance. *Oberseminar Intelligent Autonomous Systems Group, Technische Universität Darmstadt, Germany*, December 2011.

[T13] P. Robuffo Giordano. **Invited Talk**: Shared Control of Remote Multi-Robot Systems: State-of-the-art and Future Perspectives. *University of Würzburg–Institute of Computer Science, Würzburg, Germany*, March 2012.

[T14] P. Robuffo Giordano. **Invited Talk**: Shared Control of Remote Multi-Robot Systems: State-of-the-art and Future Perspectives. *IRISA/INRIA, Rennes, France*, March 2012.

[T15] P. Robuffo Giordano. **Invited Talk**: Passive and Decentralized Shared Control of Multi-Robot Systems. *Boston University, Boston, MA, USA*, May 2012.

[T16] P. Robuffo Giordano. **Invited Talk**: Shared control of remote multi-robot aerial systems: state-of-the-art and future perspectives. *"UAVs in perspective" Symposium, Università di Bologna, Bologna, Italy*, May 2012.

[T17] P. Robuffo Giordano. **Invited Seminar**: Passive and Decentralized Shared Control of Multi-Robot Systems. *Università di Modena e Reggio Emilia, Reggio Emilia, Italy*, June 2012.

[T18] P. Robuffo Giordano. **Invited Talk**: Multi-UAV Bilateral Shared Control and Decentralization. *Summer School Airobots: Aerial Service Robotics, Eidgenössische Technische Hochschule, Zürich, Switzerland*, July 2012.

[T19] P. Robuffo Giordano. **Invited Talk**: Passive and Decentralized Shared Control of Multi-Robot Systems. *Australian National University, Canberra, Australia*, July 2012.

[T20] P. Robuffo Giordano. **Invited Talk**: Passive and Decentralized Shared Control of Multi-Robot Systems. *I3S, Sophia-Antipolis, France*, February 2013.

[T21] P. Robuffo Giordano. **Invited Talk**: An overview of recent results in the bilateral shared control of multiple mobile robots. *INRIA, Grenoble, France*, April 2013.

[T22] P. Robuffo Giordano. **Invited Seminar**: A Framework For Nonlinear Active Estimation With Applications To Structure From Motion. *DIAG, University of Rome "La Sapienza", Rome, Italy*, May 2013.

[T23] P. Robuffo Giordano. **Invited Talk**: An overview of recent results in the bilateral shared control of multiple mobile robots. *ISIR, Université Pierre et Marie Curie, Paris, France*, June 2013.

[T24] P. Robuffo Giordano. **Invited Talk**: Overactuation, state estimation and human interaction for quadrotor UAVs. *UAV French day, Ensam, Paris, France*, June 2013.

[T25] P. Robuffo Giordano. **Invited Talk**: Vision-Based Bilateral Shared Control and Autonomous Flight for Single and Multiple Quadrotor UAVs. *IEEE/RSJ IROS Int. Work. on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments*, November 2013.

[T26] P. Robuffo Giordano. **Invited Talk**: Collective Control, State Estimation and Human Interaction for Teams of Mobile Robots. *Symposium "The Very Best of Robotics Today", Department of Computer Science, Rice University, Houston, USA*, April 2014.

[T27] P. Robuffo Giordano. **Invited Talk**: Collective Control, State Estimation and Human Interaction for Teams of Mobile Robots. *ACROSS Workshop on Cooperative Systems, Dubrovnik, Croatia*, September 2014.

[T28] P. Robuffo Giordano. **Invited Talk**: An Overview on Collective Coordination of Multiple Robots. *Computer Science and Telecommunications Department, Ecole Normale Supèrieure, Rennes, France*, October 2014.

[T29] P. Robuffo Giordano. **Invited Talk**: Control of Quadrotors in Unstructured Environments. *RSS 2015 Symposium: Frontiers of Robotics*, April 2015.

Bibliography

[AEROARMS, 2015] AEROARMS (2015). http://www.iri.upc.edu/project/show/148. EU Collaborative Project H2020-ICT-2014-1-644271.

[AIRobots, 2013] AIRobots (2010–2013). www.airobots.eu. EU Collaborative Project ICT-248669 AIRobots.

[Anderson et al., 2008] Anderson, B. D. O., Changbin, Y., Fidan, B., and Hendrickx, J. M. (2008). Rigid graph control architectures for autonomous formations. *IEEE Control Systems Magazine*, 6(28):48–63.

[ARCAS, 2011] ARCAS (2011). www.arcas-project.eu. EU Collaborative Project ICT-287617 ARCAS.

[Ariel and Sivan, 1984] Ariel, D. and Sivan, R. (1984). False cue reduction in moving flight simulators. *IEEE Trans. on Systems, Man, and Cybernetics*, 14(4):665–671.

[Avizzano et al., 2000] Avizzano, C. A., Barbagli, F., and Bergamasco, M. (2000). Washout filter design for a motorcycle simulator. In *2000 IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 2, pages 995–1000.

[Avizzano et al., 2010] Avizzano, C. A., Tripicchio, P., Joale, L., and Bergamasco, M. (2010). Design of a motion based sailing simulator. In *2010 IEEE IEEE Int. Symp. on Robots and Human Interactive Communications*, pages 1–7.

[Bellmann et al., 2007] Bellmann, T., Otter, M., Heindl, J., and Hirzinger, G. (2007). Real-time path planning for an interactive and industrial robot-based motion simulator. *Proc. of the 2nd Motion Simulator Conference*.

[Bemporad and Rocchi, 2011] Bemporad, A. and Rocchi, C. (2011). Decentralized Linear Time-Varying Model Predictive Control of a Formation of Unmanned Aerial Vehicles. In *2011 IEEE Conf. on Decision and Control*, pages 7488–7493.

[Benson et al., 1986] Benson, A. J., Spencer, M., and Stott, J. (1986). Thresholds for the detection of the direction of whole-body, linear movement in the horizontal plane. *Aviat Space Environ Med*, 11(57):1088–1096.

[Besançon and Hammouri, 1996] Besançon, G. and Hammouri, H. (1996). On uniform observation of nonuniformity observable systems. *Systems & Control Letters*, 29(1):9–19.

[Beykirch et al., 2007] Beykirch, K., Nieuwenhuizen, F. M., Teufel, H., Nusseck, H.-G., Butler, J., and Bülthoff, H. H. (2007). Control of a lateral helicopter side-step maneuver on an anthropomorphic robot. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, pages 1–11.

[Biagiotti and Melchiorri, 2008] Biagiotti, L. and Melchiorri, C. (2008). *Trajectory Planning for Automatic Machines and Robots*. Springer.

[Bles and Groen, 2009] Bles, W. and Groen, E. (2009). *The DESDEMONA Motion Facility: Applications for Space Research*. Springer.

[Brock and Khatib, 2002] Brock, O. and Khatib, O. (2002). Elastic strips: A framework for motion generation in human environments. *The International Journal of Robotics Research*, 21(12):1031–1052.

[Bullo et al., 2011] Bullo, F., Frazzoli, E., Pavone, M., Savla, K., and Smith, S. L. (2011). Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482–1504.

[Burki-Cohen et al., 2001] Burki-Cohen, J., Go, T. H., and Longridge, T. (2001). Flight simulator fidelity for total air line pilot training and evaluation. In *Proc. of the AIAA Modeling and Simulation Technologies Conference*.

[Chaumette and Hutchinson, 2006] Chaumette, F. and Hutchinson, S. (2006). Visual servo control, Part I: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90.

[Chaumette and Hutchinson, 2007] Chaumette, F. and Hutchinson, S. (2007). Visual servo control, Part II: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118.

[Chiacchio et al., 1991] Chiacchio, P., Chiaverini, S., Sciavicco, L., and Siciliano, B. (1991). Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *Int. J. of Robotics Research*, 10(4):410–425.

[Chiaverini, 1997] Chiaverini, S. (1997). Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Trans. on Robotics and Automation*, 13(4):398–410.

[Chopra et al., 2008] Chopra, N., Spong, M. W., and Lozano, R. (2008). Synchronization of bilateral teleoperators with time delay. *Automatica*, 44(8):2142–2148.

[Civera et al., 2008] Civera, J., Davison, A. J., and Montiel, J. (2008). Inverse Depth Parametrization for Monocular SLAM. *IEEE Trans. on Robotics*, 24(5):932–945.

[Civera et al., 2010] Civera, J., Grasa, Ó., Davison, A. J., and Montiel, J. M. M. (2010). 1-Point RANSAC for EKF Filtering: Application to Real-Time Structure from Motion and Visual Odometry. *Journal of Field Robotics*, 27(5):609–631.

[Conrad and Schmidt, 1971] Conrad, B. and Schmidt, S. F. (1971). A study of techniques for calculating motion drive signals for flight simulators. Technical Report NASA CR-114345, NASA.

[De Luca et al., 2008] De Luca, A., Oriolo, G., and Robuffo Giordano, P. (2008). Feature depth observation for image-based visual servoing: Theory and experiments. *Int. Journal of Robotics Research*, 27(10):1093–1116.

[Dieudonne et al., 1972] Dieudonne, J. E., Parrish, R. V., and Bardusch, R. E. (1972). An actuator extension transformation for a motion simulator and an inverse transformation applying newton-raphson's method. Nasa tn d-7067, National Aeronautics and Space Administration.

[Ding et al., 2009] Ding, X. C., Powers, M., Egerstedt, M., Young, R., and Balch, T. (2009). Executive decision support: Single-agent control of multiple UAVs. *IEEE Robotics & Automation Magazine*, 16(2):73–81.

[Dixon et al., 2003] Dixon, W. E., Fang, Y., Dawson, D. M., and Flynn, T. J. (2003). Range Identification for Perspective Vision Systems. *IEEE Trans. on Automatic Control*, 48(12):2232–2238.

[Duindam et al., 2009] Duindam, V., Macchelli, A., Stramigioli, S., and Bruyninckx, H. (2009). *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*. Springer.

[Duindam and Stramigioli, 2004] Duindam, V. and Stramigioli, S. (2004). Port-based asymptotic curve tracking for mechanical systems. *European Journal of Control*, 5(10):411–420.

[Durand Petiteville et al., 2010] Durand Petiteville, A., Courdesses, M., and Cadenat, V. (2010). A New Predictor/Corrector Pair to Estimate the Visual Features Depth during a Vision-based Navigation Task in an Unknown Environment. In *Inte. Conf. on Informatics in Control, Automation and Robotics*, pages 268–274.

[Escande et al., 2014] Escande, A., Mansard, N., and Wieber, P.-B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 3(7):1006–1028.

[Faiz et al., 2001] Faiz, N., Agrawal, S. K., and Murray, R. M. (2001). Trajectory planning of differentially flat systems with dynamics and inequalities. *Journal of Guidance, Control, and Dynamics*, 24(2):219–227.

[Faulwasser et al., 2011] Faulwasser, T., Hagenmeyer, V., and Findeisen, R. (2011). Optimal exact path-following for constrained differentially flat systems. In *2011 IFAC World Congress*, pages 9875–9880.

[Fernandez and Goldberg, 1976] Fernandez, C. and Goldberg, J. (1976). Physiology of peripheral neurons innervating otolith organs of squirrel monkey. *Journal of Neurophysiology*, 5(39):996–1008.

[Ferrazzin et al., 1999] Ferrazzin, D., Salsedo, F., and Bergamasco, M. (1999). The MORIS simulator. In *Robot and Human Interaction, 1999. RO-MAN '99. 8th IEEE International Workshop on*, pages 136–141.

[Fink et al., 2010] Fink, J., Michael, N., Kim, S., and Kumar, V. (2010). Planning and control for cooperative manipulation and transportation with aerial robots. *The International Journal of Robotics Research*, 30(3):324–334.

[Fink et al., 2012] Fink, J., Ribeiro, A., and Kumar, V. (2012). Robust Control for Mobility and Wireless Communication in Cyber–Physical Systems With Application to Robot Teams. *Proceedings of the IEEE*, 1(100):164–178.

[Flacco and De Luca, 2013a] Flacco, F. and De Luca, A. (2013a). Fast redundancy resolution for high-dimensional robots executing prioritized tasks under hard bounds in the joint space. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2500–2506.

[Flacco and De Luca, 2013b] Flacco, F. and De Luca, A. (2013b). Optimal SNS Control of Redundant Robots under Hard Joint Constraints. In *IEEE Int. Conf. on Robotics and Automation*, pages 3969–3975.

[Flacco and De Luca, 2014] Flacco, F. and De Luca, A. (2014). A Reverse Priority Approach to Multi-Task Control of Redundant Robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.

[Franken et al., 2011] Franken, M., S.Stramigioli, Misra, S., Secchi, C., and Macchelli, A. (2011). Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency. *IEEE Trans. on Robotics*, 27(4):741–756.

[Gentili et al., 2008] Gentili, L., Naldi, R., and Marconi, L. (2008). Modelling and control of VTOL UAVs interacting with the environment. In *2008 IEEE Conf. on Decision and Control*, pages 1231–1236.

[Gescheider, 1997] Gescheider, G. (1997). *Psychophysics: the fundamentals*. L. Erlbaum Associates.

[Gioioso et al., 2014] Gioioso, G., Franchi, A., Salvietti, G., Scheggi, S., and Prattichizzo, D. (2014). The Flying Hand: a Formation of UAVs for Cooperative Aerial Tele-Manipulation. In *2014 IEEE Int. Conf. on Robotics and Automation*, pages 4335–4341.

[Goodrich and Schultz, 2007] Goodrich, M. A. and Schultz, A. C. (2007). Human-robot interaction: a survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275.

[Grant and Reid, 1997] Grant, P. R. and Reid, L. D. (1997). Motion washout filter tuning: Rules and requirements. *J. of Aircraft*, 34(2):145–151.

[Greig, 1988] Greig, G. L. (1988). Masking of motion cues by random motion: Comparison of human performance with a signal detection model. Technical Report 313, Univ. of Toronto.

[Groen and Bles, 2004] Groen, E. L. and Bles, W. (2004). How to use body tilt for the simulation of linear self motion. *Journal of Vestibular Research*, 14(5):375–385.

[Hashtrudi-Zaad and Salcudean, 2001] Hashtrudi-Zaad, K. and Salcudean, S. E. (2001). Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators. *The International Journal of Robotics Research*, 6(20):419–445.

[Heerspink et al., 2005] Heerspink, H. M., Berkouwer, W. R., Stroosma, O., van Paassen, M. M., Mulder, M., and Mulder, J. A. (2005). Evaluation of vestibular thresholds for motion detection in the simona research simulator. In *AIAA modeling and simulation technologies conference*.

[Heindl et al., 2006] Heindl, J., Otter, M., Hirschmueller, H., Fromberger, M., Siegert, F., and Heinrich, H. (2006). The robocoaster simulation platform, path and video generation for an authentic mars flight simulation. In *Proceedings ISR-2006, Joint Conference on Robotics / ROBOTIK 2006*, München.

[Heindl et al., 2005] Heindl, J., Otter, M., Hirschmüller, H., Frommberger, M., Sporer, N., Siegert, F., and Heinrich, H. (2005). The robocoaster as simulation platform – Experiences from the "first authentic mars flight simulation". *Proc. of the 1nd Motion Simulator Conference*.

[Hing and Oh, 2009] Hing, J. T. and Oh, P. Y. (2009). Development of an unmanned aerial vehicle piloting system with integrated motion cueing for training and pilot evaluation. *Journal of Intelligent and Robotic Systems*, 54(1–3):3–19.

[Hokayem and Spong, 2006] Hokayem, P. F. and Spong, M. W. (2006). Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035–2057.

[Hosman and van der Vaart, 1978] Hosman, R. and van der Vaart, J. C. (1978). Vestibular models and thresholds of motion perception. results of tests in a flight simulator. *Techreport LR-265, TU Delft*, 11(57):1088–1096.

[Hou and Mahony, 2014] Hou, X. and Mahony, R. (2014). An intuitive multimodal haptic interface for teleoperation of aerial robots. In *Proc. of the 2014 IEEE Int. Conf. on Robotics and Automation*, pages 838–845.

[Hou et al., 2013] Hou, X., Mahony, R., and Schill, F. (2013). Representation of vehicle dynamics in haptic teleoperation of aerial robots. In *Proc. of the 2013 IEEE Int. Conf. on Robotics and Automation*, pages 1485–1491.

[Huang and Fu, 2006] Huang, C.-I. and Fu, L.-C. (2006). Human vestibular based (HVB) senseless maneuver optimal washout filter design for VR-based motion simulator. *Proc. of the 2006 IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 4451–4458.

[Kant and Zucker, 1986] Kant, K. and Zucker, S. W. (1986). Toward efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89.

[Kar and Moura, 2010] Kar, S. and Moura, J. M. F. (2010). Distributed Consensus Algorithms in Sensor Networks: Quantized Data and Random Link Failures. *IEEE Trans. on Signal Processing*, 3(58):1383–1400.

[Kim and Mesbahi, 2006] Kim, Y. and Mesbahi, M. (2006). On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. *IEEE Trans. on Robotics*, 1(51):116–120.

[Kondak et al., 2014] Kondak, K., Huber, F., Schwarzbach, M., Laiacker, M., Sommer, D., Bejar, M., and Ollero, A. (2014). Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. In *2014 IEEE Int. Conf. on Robotics and Automation*, pages 2107–2112.

[Lam et al., 2006a] Lam, T. M., D'Amelio, V., Mulder, M., and Paassen, M. V. (2006a). UAV teleoperation using haptics with a degraded visual interface. In *2006 IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 3, pages 2440–2445.

[Lam et al., 2006b] Lam, T. M., Mulder, M., and Paassen, M. M. V. (2006b). Haptic feedback for UAV tele-operation - force offset and spring load modification. In *2006 IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 2, pages 1618–1623.

[LaValle, 2006] LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press. Available at `http://planning.cs.uiuc.edu`.

[Lawrence, 1993] Lawrence, D. (1993). Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation*, 5(9):624–637.

[Lee and Spong, 2005] Lee, D. and Spong, M. W. (2005). Bilateral teleoperation of multiple cooperative robots over delayed communication network: theory. In *2005 IEEE Int. Conf. on Robotics and Automation*, pages 360–365, Barcelona, Spain.

[Lee and Xu, 2011] Lee, D. and Xu, D. (2011). Feedback r-passivity of Lagrangian systems for mobile robot teleoperation. In *Proc. of the 2011 IEEE Int. Conf. on Robotics and Automation*, pages 2118–2123.

[Lee and Huang, 2010] Lee, D. J. and Huang, K. (2010). Passive-set-position-modulation framework for interactive robotic systems. *IEEE Trans. on Robotics*, 26(2):354–369.

[Lee, 2008] Lee, E. A. (2008). Cyber physical systems: Design challenges. In *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing*.

[Lewis, 1995] Lewis, F. L. (1995). *Optimal Control*. Wiley–Interscience.

[Lindsey et al., 2011] Lindsey, Q., Mellinger, D., and Kumar, V. (2011). Construction of cubic structures with quadrotor teams. In *Robotics: Science and Systems*.

[Ma et al., 2003] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. (2003). *An invitation to 3D vision*. Springer.

[Malis et al., 2010] Malis, E., Mezouar, Y., and Rives, P. (2010). Robustness of Image-Based Visual Servoing With a Calibrated Camera in the Presence of Uncertainties in the Three-Dimensional Structure. *IEEE Trans. on Robotics*, 26(1):112–120.

[Manocha and Canny, 1992] Manocha, D. and Canny, J. F. (1992). Detecting cusps and inflection points in curves. *Comp. Aided Geom. Design*, 9:1–24.

[Manubens et al., 2013] Manubens, M., Devaurs, D., Ros, L., and Cortés, J. (2013). Motion Planning for 6-D Manipulation with Aerial Towed-cable Systems. In *2013 Robotics: Science and Systems*, Berlin, Germany.

[Marconi and Naldi, 2012] Marconi, L. and Naldi, R. (2012). Control of aerial robots. hybrid force/position feedback for a ducted-fan. *IEEE Control System Magazine*, 32(4):43–65.

[Marey and Chaumette, 2010] Marey, M. and Chaumette, F. (2010). A new large projection operator for the redundancy framework. In *2010 IEEE Int. Conf. on Robotics and Automation*, pages 3727–3732.

[Martinelli, 2012] Martinelli, A. (2012). Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Trans. on Robotics*, 28(1):44–60.

[Matthies et al., 1989] Matthies, L., Kanade, T., and Szeliski, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *Int. Journal of Computer Vision*, 3(3):209–238.

[McLean et al., 1991] McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991). A unified approach to mixed linear models. *The American Statistician*, 45(1):54–64.

[Mellinger and Kumar, 2011] Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 2520–2525.

[Mesbahi and Egerstedt, 2010] Mesbahi, M. and Egerstedt, M. (2010). *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press. Available at `http://planning.cs.uiuc.edu`.

[Metni and Hamel, 2007] Metni, N. and Hamel, T. (2007). Visual Tracking Control of Aerial Robotic Systems with Adaptive Depth Estimation. *Int. Journal of Control, Automation, and Systems*, 1(5):51–60.

[Michael et al., 2012] Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., Takeuchi, E., and Tadokoro, S. (2012). Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 5(29):832–841.

[Monza racetrack,] Monza racetrack. www.monzanet.it.

[Morbidi et al., 2010] Morbidi, F., Mariottini, G. L., and Prattichizzo, D. (2010). Observer design via immersion and invariance for vision-based leader-follower formation control. *Automatica*, 46(1):148–154.

[Murphy et al., 2009] Murphy, R., Kravitz, J., Stover, S., and Shoureshi, R. (2009). Mobile robots in mine rescue and recovery. *IEEE Robotics & Automation Magazine*, 16(2):91–103.

[Murray et al., 1994] Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

[Nahon and Reid, 1990] Nahon, M. A. and Reid, L. D. (1990). Simulator motion-drive algorithms: A designer's perspective. *J. of Guidance, Control, and Dynamics*, 13(2):356–362.

[Nahon et al., 1992] Nahon, M. A., Reid, L. D., and Kirdeikist, J. (1992). Adaptive simulator motion software with supervisory control. *J. of Guidance and Dynamics*, 15(2):376–383.

[Naldi and Marconi, 2010] Naldi, R. and Marconi, L. (2010). Modeling and control of the interaction between flying robots and the environment. In *Proc. of the 2010 IFAC NOLCOS*.

[Nehaoua et al., 2006a] Nehaoua, L., Arioui, H., Espie, S., and Mohellebi, H. (2006a). Motion cueing algorithms for small driving simulator. *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation*, pages 3189–3194.

[Nehaoua et al., 2011] Nehaoua, L., Arioui, H., and Mammar, S. (2011). Review on single track vehicle and motorcycle simulators. In *2011 19th Mediterranean Conference on Control Automation (MED)*, pages 940–945.

[Nehaoua et al., 2006b] Nehaoua, L., Arioui, H., Mohellebi, H., and Espie, S. (2006b). Restitution movement for a low cost driving simulator. *Proc. of the 2006 American Control Conference*, pages 2599–2604.

[Nelson and Khosla, 1995] Nelson, B. J. and Khosla, P. K. (1995). Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. *The International Journal of Robotics Research*, 14(3):255–269.

[Niemeyer et al., 2008] Niemeyer, G., Preusche, C., and Hirzinger, G. (2008). Telerobotics. In Siciliano, B., , and Khatib, O., editors, *Handbook of Robotics*, pages 741–757. Springer.

[Nuno et al., 2011] Nuno, E., Basanez, L., and Ortega, R. (2011). Passivity-based control for bilateral teleoperation: A tutorial. *Automatica*, 47(3):485–495.

[Omari and Ducard, 2013] Omari, S. and Ducard, G. (2013). Metric Visual-Inertial Navigation System Using Single Optical Flow Feature. In *2013 European Control Conference*, pages 1310–1316.

[Parrish et al., 1975] Parrish, R. V., Dieudonne, J. E., Bowles, R. L., and Martin, D. J. (1975). Coordinated adaptive washout for motion simulators. *J. of Aircraft*, 12(1):44–50.

[Peng and Akella, 2005] Peng, J. and Akella, S. (2005). Coordinating multiple robots with kinodynamic constraints along specified paths. *The International Journal of Robotics Research*, 24(4):295–310.

[Pollini et al., 2008] Pollini, L., Innocenti, M., and Petrone, A. (2008). Novel motion platform for flight simulators using an anthropomorphic robot. *J. of Aerospace Computing, Information, and Communication*, 5:175–196.

[Pound et al., 2011] Pound, P. E. I., Bersak, D. R., and Dollar, A. M. (2011). Grasping from the air: Hovering capture and load stability. In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 2491–2498.

[Reid and Nahon, 1985] Reid, L. D. and Nahon, M. A. (1985). Flight simulator motion-base drive algorithms: Part 1 – developing and testing the equations. Technical Report 296, Univ. of Toronto.

[Reid and Nahon, 1986a] Reid, L. D. and Nahon, M. A. (1986a). Flight simulator motion-base drive algorithms: Part 2 – selecting the system parameters. Technical Report 307, Univ. of Toronto.

[Reid and Nahon, 1986b] Reid, L. D. and Nahon, M. A. (1986b). Flight simulator motion-base drive algorithms: Part 3 – pilot evaluations. Technical Report 319, Univ. of Toronto.

[Robocoaster,] Robocoaster. www.robocoaster.com. http://www.robocoaster.com.

[Robotics in the United States of America, 2013] Robotics in the United States of America (2013). A Roadmap for U.S. Robotics: From Internet to Robotics. http://robotics-vo.us/.

[Robotics2020, 2014] Robotics2020 (2014). The Strategic Research Agenda for Robotics in Europe. http://www.robotics-platform.eu.

[Robuffo Giordano et al., 2008] Robuffo Giordano, P., De Luca, A., and Oriolo, G. (2008). 3D structure identification from image moments. In *2008 IEEE Int. Conf. on Robotics and Automation*, pages 93–100, Pasadena, CA.

[Robuffo Giordano et al., 2010] Robuffo Giordano, P., Masone, C., Tesch, J., Breidt, M., Pollini, L., and Bülthoff, H. H. (2010). A novel framework for closed-loop robotic motion simulation - part II: Motion cueing design and experimental validation. In *2010 Int. Conf. on Robotics and Automation*, pages 3896–3903.

[Rodríguez-Seda et al., 2010] Rodríguez-Seda, E. J., Troy, J. J., Erignac, C. A., Murray, P., Stipanović, D. M., and Spong, M. W. (2010). Bilateral teleoperation of multiple mobile agents: Coordinated motion and collision avoidance. *IEEE Trans. on Control Systems Technology*, 18(4):984–992.

[Saab et al., 2013] Saab, L., Ramos, O. E., Keith, F., Mansard, N., Soueres, P., and Fourquet, J. (2013). Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Trans. on Robotics*, 29(2):346–362.

[Sassano et al., 2010] Sassano, M., Carnevale, D., and Astolfi, A. (2010). Observer design for range and orientation identification. *Automatica*, 46(8):1369–1375.

[Schmidt and Conrad, 1970] Schmidt, S. F. and Conrad, B. (1970). Motion drive signals for piloted flight simulators. Technical Report CR-1601, NASA.

[Schwager et al., 2011a] Schwager, M., Julian, B., Angermann, M., and Rus, D. (2011a). Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561.

[Schwager et al., 2011b] Schwager, M., Michael, N., Kumar, V., and Rus, D. (2011b). Time scales and stability in networked multi-robot systems. In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 3855–3862, Shanghai, China.

[Secchi et al., 2007] Secchi, C., Stramigioli, S., and Fantuzzi, C. (2007). *Control of Interactive Robotic Interfaces: a port-Hamiltonian Approach*. Tracts in Advanced Robotics. Springer.

[Sepulchre et al., 1997] Sepulchre, R., Jankovic, J. M., and Kokotovic, P. V. (1997). *Constructive Nonlinear Control*. Springer.

[SHERPA, 2012] SHERPA (2012). www.sherpa-project.eu/sherpa. EU Collaborative Project ICT-600958 SHERPA.

[Shi et al., 2011] Shi, J., Wan, J., Yan, H., and Suo, H. (2011). A survey of cyber-physical systems. In *Proc. of the Int. Conf. on Wireless Communications and Signal Processing*.

[Siuikat, 2005] Siuikat, R. (2005). The new dynamic driving simulator at DLR. In *Driving Simulator Conference 2005*, pages 374–381, Orlando, FL, USA.

[Sivan et al., 1982] Sivan, R., Ish-Shalom, J., and Huang, J. (1982). An optimal approach to the design of moving flight simulators. *IEEE Trans. on Systems, Man, and Cybernetics*, 12(6):818–827.

[Smith et al., 2012] Smith, S. L., Schwager, M., and Rus, D. (2012). Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Trans. on Robotics*, 28(2):410–426.

[Soatto et al., 1996] Soatto, S., Frezza, R., and Perona, P. (1996). Motion estimation via dynamic vision. *IEEE Trans. on Automatic Control*, 41(3):393–413.

[Sreenath and Kumar, 2013] Sreenath, K. and Kumar, V. (2013). Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots. In *2013 Robotics: Science and Systems*.

[Stewart, 1965] Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 180(1):371–386.

[Stone and DeRose, 1989] Stone, M. C. and DeRose, T. D. (1989). A geometric characterization of parametric cubic curves. *ACM Trans. Graph.*, 8(3):147–163.

[Stump et al., 2011] Stump, E., Michael, N., Kumar, V., and Isler, V. (2011). Visibility-based deployment of robot formations for communication maintenance. In *2011 IEEE Int. Conf. on Robotics and Automation*, pages 4489–4505, Shanghai, China.

[Tahri and Chaumette, 2005] Tahri, O. and Chaumette, F. (2005). Point-Based and Region-Based Image Moments for Visual Servoing of Planar Objects. *IEEE Trans. on Robotics*, 21(6):1116–1127.

[Telban et al., 2002] Telban, R. J., Cardullo, F. M., and Houck, J. A. (2002). A nonlinear, human-centered approach to motion cueing with a neurocomputing solver. Technical Report 2002-4692, NASA.

[Thrun, 2004] Thrun, S. (2004). Toward a framework for human-robot interaction. *Human-Computer Interaction*, 19(1–2):9–24.

[Waibel et al., 2011] Waibel, M., Beetz, M., Civera, J., D'Andrea, R., Elfring, J., Galvez-Lopez, D., Häussermann, K., Janssen, R., Montiel, J. M. M., Perzylo, A., Schießle, B., Tenorth, M., Zweigle, O., and van de Molengraft, R. (2011). RoboEarth: A World Wide Web for Robots. *IEEE Robotics and Automation Magazine*, 18(2):69–82.

[Wentink et al., 2005] Wentink, M., Bles, W., Hosman, R., and Mayrhofer, M. (2005). Design & evaluation of spherical washout algorithm for Desdemona simulator. *Proc. of AIAA Modeling and Simulation Technologies*.

[Wentink et al., 2006] Wentink, M., Bos, J., Groen, E., and Hosman, R. (2006). Development of the Motion Perception Toolbox. In *Proc. of the AIAA Modeling And Simulation Technologies Conference*.

[Wikipedia, 2013] Wikipedia (2013). Cyber-physical system. http://en.wikipedia.org/wiki/Cyber-physical_system.

[World Robotics, 2011] World Robotics (2011). Manfred Gundel, CEO KUKA Roboter. http://www.worldrobotics.org/.

[Yang and Lee, 2014] Yang, H. and Lee, D. J. (2014). Dynamics and Control of Quadrotor with Robotic Manipulator. In *2014 IEEE Int. Conf. on Robotics and Automation*, pages 5544–5549.

[Yang and Lee, 2015] Yang, H. and Lee, D. J. (2015). Hierarchical Cooperative Control Framework of Multiple Quadrotor-Manipulator Systems. In *2015 IEEE Int. Conf. on Robotics and Automation*.

[Yang et al., 2010] Yang, P., Freeman, R. A., Gordon, G. J., Lynch, K. M., Srinivasa, S. S., and Sukthankar, R. (2010). Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 2(46):390–396.

[Young and Meiry, 1968] Young, L. and Meiry, J. (1968). A revised dynamic otolith model. *Aerosp Med*, 39(6):606–608.

[Yüksel et al., 2015] Yüksel, B., Mahboubi, S., Secchi, C., Bülthoff, H. H., and Franchi, A. (2015). Design, Identification and Experimental Testing of a Light-Weight Flexible-joint Arm for Aerial Physical Interaction. In *2015 IEEE Int. Conf. on Robotics and Automation*.

[Zaichik et al., 1999] Zaichik, L., Rodchenko, V., Rufov, I., Yashin, Y., and White, A. (1999). Acceleration perception. In *AIAA modeling and simulation technologies conference*, pages 512–520.

[Zavlanos and Pappas, 2007] Zavlanos, M. M. and Pappas, G. J. (2007). Potential fields for maintaining connectivity of mobile networks. *IEEE Trans. on Robotics*, 4(23):812–816.