



**HAL**  
open science

# Control Methods for Aerial Robotic Systems Physically Interacting with Humans

Gianluca Corsini

► **To cite this version:**

Gianluca Corsini. Control Methods for Aerial Robotic Systems Physically Interacting with Humans. Automatic. UPS Toulouse, 2023. English. NNT: . tel-04424344v1

**HAL Id: tel-04424344**

**<https://laas.hal.science/tel-04424344v1>**

Submitted on 29 Jan 2024 (v1), last revised 13 Mar 2024 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# THÈSE

En vue de l'obtention du  
**DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**  
Délivré par l'Université Toulouse 3 - Paul Sabatier

---

Présentée et soutenue par  
**Gianluca CORSINI**

Le 31 octobre 2023

**Méthodes de contrôle des systèmes robotiques aériens en  
interaction physique avec les humains**

---

Ecole doctorale : **SYSTEMES**

Spécialité : **Robotique**

Unité de recherche :

**LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes**

Thèse dirigée par

**Daniel SIDOBRE et Antonio FRANCHI**

Jury

**M. Paolo ROBUFFO GIORDANO**, Rapporteur

**M. Vincenzo LIPPIELLO**, Rapporteur

**Mme Marija POPOVI**, Examinatrice

**M. Daniel SIDOBRE**, Directeur de thèse

**M. Antonio FRANCHI**, Co-directeur de thèse

**M. Simon LACROIX**, Président



# Table of Contents

<b>Table of Contents</b>	<b>i</b>
<b>Résumé</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Acronyms</b>	<b>ix</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xix</b>
<b>List of Multimedias</b>	<b>xxi</b>
<b>I Preliminaries</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Human-Robot Interaction . . . . .	3
1.2 Aerial robotics . . . . .	5
1.2.1 Taxonomy of multi-rotor aerial vehicles . . . . .	8
1.3 Physical Aerial Robot Interaction . . . . .	11
1.3.1 From research to industries . . . . .	13
1.4 Physical Human-Aerial Robot Interaction . . . . .	14
1.5 Related works in pHARI . . . . .	15
1.6 Aim of the thesis . . . . .	17
1.7 Context and objectives . . . . .	18
1.8 Thesis contributions . . . . .	18
1.9 Publication note . . . . .	20
1.10 Thesis outline . . . . .	20
<b>2 Research problem</b>	<b>23</b>
2.1 Human-Aerial Robot Handover . . . . .	23
2.1.1 Human-robot handover . . . . .	23
2.1.2 The aerial case . . . . .	24
2.1.3 Focus of the thesis . . . . .	25
2.1.4 Assumptions . . . . .	26
2.2 Research challenges . . . . .	27
2.2.1 Physical interaction . . . . .	27
2.2.2 Robot actuation properties . . . . .	28
2.2.3 Perception . . . . .	29

2.2.4	Safety . . . . .	29
2.2.5	Ergonomics . . . . .	30
2.3	Problem decomposition . . . . .	30
2.4	Overview on the state of the art . . . . .	32
2.4.1	Control for physical interaction . . . . .	32
2.4.2	Robot actuation properties and related control architectures . . . . .	34
2.4.3	Perception-based control . . . . .	37
2.4.4	Safety . . . . .	39
2.4.5	Ergonomics . . . . .	41
2.5	Conclusion . . . . .	42
<b>3</b>	<b>Method overview</b>	<b>45</b>
3.1	Methodology . . . . .	45
3.1.1	Visual and physical control for HAR interaction . . . . .	45
3.1.2	Human-aware control for HAR handover . . . . .	47
3.2	Main results . . . . .	50
3.2.1	Physical Human-Aerial robot interaction . . . . .	50
3.2.2	Human-aware aerial robot handover . . . . .	56
3.3	Synthesis . . . . .	60
<b>II</b>	<b>Scientific work</b>	<b>63</b>
<b>4</b>	<b>Modeling</b>	<b>65</b>
4.1	Notation . . . . .	65
4.1.1	Sets . . . . .	66
4.1.2	Operators . . . . .	66
4.1.3	Reference frames . . . . .	67
4.1.4	Subscripts . . . . .	67
4.2	Rigid-body attitude representations . . . . .	67
4.2.1	Rotation matrices . . . . .	67
4.2.2	Euler angles . . . . .	68
4.2.3	Angle and axis . . . . .	70
4.2.4	Unit quaternions . . . . .	70
4.3	Overview on rigid body dynamics . . . . .	72
4.3.1	Lagrange formalism . . . . .	73
4.3.2	Newton-Euler formalism . . . . .	75
4.4	Generically-tilted multi-rotor systems . . . . .	76
4.4.1	Equations of motion . . . . .	76
4.4.2	Center of mass . . . . .	78
4.4.3	Model assumptions . . . . .	79
4.5	Aerial manipulator systems . . . . .	80
4.5.1	A flying base and a robotic arm . . . . .	80
4.5.2	Recursive Newton-Euler algorithm . . . . .	81
4.5.3	Equations of motion . . . . .	83
4.6	Actuators of aerial robots . . . . .	83
4.6.1	Motor-propeller actuation units . . . . .	84
4.6.2	Joint actuators . . . . .	88
4.6.3	Actuation limitations . . . . .	88

4.7	State-space representation of aerial robotic agents . . . . .	90
4.7.1	The GTMR case . . . . .	90
4.7.2	The AM case . . . . .	91
4.8	Perceptive sensors on-board aerial robots . . . . .	92
4.9	Human agent . . . . .	93
4.9.1	Trunk . . . . .	93
4.9.2	Arm . . . . .	94
4.10	Conclusion . . . . .	95
<b>5</b>	<b>Control methodology for visual and physical Human-Aerial Robot Interaction</b>	<b>97</b>
5.1	Hierarchical control architecture . . . . .	97
5.1.1	Vision-based control . . . . .	99
5.1.2	Physical interaction control . . . . .	101
5.1.3	Motion control . . . . .	103
5.2	Interaction with the environment . . . . .	106
5.2.1	Validation setup . . . . .	106
5.2.2	Experiments . . . . .	109
5.3	Extension to aerial manipulators . . . . .	113
5.3.1	Model of the aerial manipulator . . . . .	113
5.3.2	Vision-based control . . . . .	114
5.3.3	Physical interaction control . . . . .	115
5.3.4	Motion control . . . . .	116
5.4	Interaction with humans . . . . .	117
5.4.1	Validation setup . . . . .	117
5.4.2	Experiments . . . . .	119
5.5	Conclusion . . . . .	123
5.5.1	Synthesis . . . . .	123
5.5.2	Limitations . . . . .	124
5.5.3	Perspectives . . . . .	124
<b>6</b>	<b>Predictive human-aware control</b>	<b>127</b>
6.1	Model Predictive Control . . . . .	127
6.1.1	Optimal Control Problem formulation . . . . .	129
6.1.2	Nonlinear Programming Problem . . . . .	130
6.2	Human-aware NMPC formulation . . . . .	131
6.2.1	Definition of system state, input and parameter variables . . . . .	132
6.2.2	Objective function . . . . .	133
6.2.3	Equality constraints . . . . .	142
6.2.4	Inequality constraints . . . . .	142
6.2.5	Human-aware NLP . . . . .	144
6.3	Simulation validation . . . . .	146
6.3.1	Motion reference generation . . . . .	146
6.3.2	Simulation setup . . . . .	147
6.3.3	Results analysis and discussion . . . . .	148
6.4	Improvements . . . . .	153
6.4.1	Comments from the reviewers . . . . .	153
6.4.2	Generalization . . . . .	154
6.4.3	Physical interaction . . . . .	155

6.5	Experimental validation . . . . .	156
6.5.1	Experimental setup . . . . .	156
6.5.2	Experimental results . . . . .	157
6.6	Conclusion . . . . .	159
6.6.1	Limitations and perspectives . . . . .	159
<b>III</b>	<b>Conclusions</b>	<b>161</b>
<b>7</b>	<b>Conclusion</b>	<b>163</b>
7.1	Overall contributions . . . . .	163
7.2	Limitations and perspectives . . . . .	165
	<b>Bibliography</b>	<b>169</b>
	<b>Résumé - Abstract</b>	<b>199</b>

# Résumé

Les robots aériens (ARs) connaissent un large intérêt pour diverses applications qui vont des opérations sans contact telles que la surveillance de zones ou les missions de recherche et de sauvetage, aux scénarios nécessitant un contact tels que l'inspection de surface ou l'installation de capteurs. Les véhicules aériens à rotors multiples (multi-rotor aerial vehicles (MRAVs)) suscitent une attention particulière de la communauté robotique en raison de leur conception personnalisable, de leur facilité de déploiement et de leur construction à base des composants standards disponible sur le marché. Leur agilité et leurs capacités de manoeuvre leur permettent d'atteindre des zones difficiles d'accès et d'accomplir une grande variété de tâches à l'aide de capteurs embarqués et d'effecteurs terminaux spécialisés.

Ces dernières années, le montage d'outils fixes ou de bras articulés à plusieurs degrés de liberté sur des robots aériens a permis à ces robots d'interagir physiquement avec leur environnement et d'accomplir des tâches qui nécessitent d'exercer des forces et des couples. Cette nouvelle approche ouvre la voie vers l'intégration de ces robots dans des environnements quotidiens où ils pourront collaborer avec les humains ou les assister dans diverses tâches. Pour des applications en hauteur, par exemple, les robots aériens peuvent être utilisés pour surveiller les opérations à l'aide de capteurs embarqués et fournir des outils nécessaires aux opérateurs humains.

Toutefois, le déploiement de robots aériens pour l'interaction humain-robot représente de nouveaux défis qui nécessitent des solutions appropriées. Du point de vue du contrôle, des algorithmes novateurs sont nécessaires pour que l'interaction physique entre l'opérateur humain et le robot soit sûre. Ces algorithmes de contrôle doivent garantir la sécurité des agents humains tout en les assistant de la manière la plus ergonomique possible, par exemple en réduisant la charge articulaire.

Cette thèse aborde ces défis et propose différentes architectures de contrôle pour les robots aériens à rotors multiples afin de réaliser des interactions physiques avec les humains. L'accent principal est mis sur le problème du transfert d'un objet ou d'un outil entre un robot aérien et un opérateur humain.

La méthodologie proposée combine des techniques de contrôle et d'estimation classiques issues des connaissances bien établies en interactions humain-robot dans le cas des manipulateurs terrestres et des algorithmes de contrôle basés sur l'optimisation prédictive. Les robots aériens posent des difficultés particulières par rapport aux bras manipulateurs industriels, dues notamment à des capacités d'actionnement limitées, à l'absence d'une base fixe, à l'interaction entre le mouvement et la vision, ainsi qu'à une dynamique rapide et hautement non linéaire.

Cette thèse propose une formulation du problème de transfert d'objet basée sur



une décomposition en sous-problèmes et l'élaboration de méthodes de contrôle pour aborder un sous-ensemble de ces sous-problèmes, puis elle fournit des éléments pour aborder le problème dans sa globalité.

Bien que des recherches supplémentaires soient nécessaires, les résultats analytiques et expérimentaux présentés dans cette thèse démontrent la validité et la pertinence pratique de la méthodologie proposée.

Ce travail a donné lieu à trois publications dans des conférences internationales évaluées par les pairs, et la plupart des logiciels associés ont été publiés en open source pour la communauté robotique.

**Mots clefs** - Robotique aérienne, Contrôle des robots, Manipulation aérienne, Interactions physiques entre humain et robot, Échange d'objet et co-manipulation, Robotique collaborative

# Abstract

Aerial robots (ARs) are experiencing a large interest for various applications ranging from contact-less operations, like area monitoring and search-and-rescue missions, to contact-based scenarios, such as surface inspection, sensor installation, and pick-and-place tasks. Multi-rotor aerial vehicles (MRAVs) are attracting the attention of roboticists due to their customizable design, scalability, ease of deployment, and assembling from off-the-shelf materials. Their agility and maneuvering capabilities, combined with a larger workspace compared to wheeled and ground manipulators, enable them to reach inaccessible areas and perform a broad variety of tasks relying on onboard sensors and specialized end effectors.

In recent years, mounting fixed tools or multi-degree-of-freedom robotic arms on aerial robots has allowed physical interaction with the environment, empowering them to accomplish tasks that require exerting forces and torques. This new trend paves the way for the integration of these robots into everyday-life settings, collaborating with humans, and assisting them in various assignments. In high-from-ground applications, for instance, ARs can be employed to monitor the operations using onboard sensors and provide the necessary tools to the human operators. Similarly, in industrial scenarios, aerial robots can swiftly move between locations to fetch the requested materials and tools.

However, deploying aerial robots in human-robot interaction settings opens new challenges that require safe and appropriate solutions. From a control perspective, novel control algorithms are needed to enable aerial vehicles to interact physically and safely with human operators. These control algorithms must ensure the safety of the human agents while assisting them in an ergonomic manner, such as reducing their joint stress.

This thesis addresses these challenges by proposing different control architectures for multi-rotor aerial robots to achieve physical and safe interaction with humans. The primary focus is on the handover problem consisting in an aerial robot handing over an object or a tool to a human operator.

The proposed methodology combines classical control and estimation techniques derived from the well-established knowledge of Human-Robot Interaction (HRI) with ground manipulators and the exploitation of predictive optimization-based algorithms. Aerial robots pose unique difficulties compared to industrial manipulator arms, due to their limited actuation capabilities, the absence of a fixed base, the interplay between motion and vision, and they own a rapid and highly nonlinear dynamics.

This thesis provides a formulation of the handover problem as a decomposition

into subproblems and it develops suitable control methods addressing a subset of these subproblems. Lastly, it sketches some considerations aimed at tackling the problem as a whole.

While further research is necessary, the analytical and experimental results presented in this thesis demonstrate the validity and practical relevance of the proposed methodology.

This work has resulted in three publications in international peer-reviewed conferences, and most of the related software productions have been published open-source for the robotics community.

**Keywords** - Aerial Robotics, Control of Robots, Aerial Manipulation, Physical Human-Robot Interaction, Handover and co-manipulation, Collaborative Robotics

# List of Acronyms

**ACW** aerial co-worker

**Aerial-CORE** Aerial COgnitive integrated multi-task Robotic system with Extended operation range and safety

**AF** Admittance filter

**AM** aerial manipulator

**AR** aerial robot

**CAD** Computer Aided Design

**CCW** counter-clockwise

**CNRS** Centre National de la Recherche Scientifique

**CoM** center of mass

**CW** clockwise

**DH** Denavit-Hartenberg

**DoF** Degree of Freedom

**DP** Dynamic Programming

**e.g.** *exempli gratia*

**EE** end effector

**EoM** Equation of Motion

**ESC** Electronic Speed Controller

**FCW** The Flying Co-Worker

**FoV** Field of View

**GenOM** Generator of modules

**GTMR** Generically-Tilted Multi-Rotor

**H2R** human-to-robot

**HAR** Human-Aerial Robot

- HARH** Human-Aerial robot handover
- HDI** Human-Drone Interaction
- HRI** Human-Robot Interaction
- HVS** Hybrid Visual Servoing
- i.e.** id est
- IBVS** Image-Based Visual Servoing
- IMU** Inertial Measurement Unit
- INRIA** Institut National de Recherche en Informatique et Automatique
- IROS** IEEE/RSJ International Conference on Intelligent Robots and Systems
- LAAS** Laboratoire d'analyse et d'architecture des systèmes
- LBF** laterally-bounded force
- MATLAB** MATrix LABoratory
- MDT** Multi-Directional Thrust
- Mocap** Motion capture
- MPC** Model Predictive Control
- MRAV** multi-rotor aerial vehicle
- NLP** Nonlinear Programming
- NMPC** Nonlinear Model Predictive Control
- OCP** Optimal Control Problem
- ODE** Ordinary Differential Equation
- ODT** Omni-Directional Thrust
- pARI** physical Aerial Robot Interaction
- PBVS** Position-based Visual Servoing
- PD** Proportional-Derivative
- pHARI** physical Human-Aerial Robot Interaction
- pHRI** physical Human-Robot Interaction
- PID** Proportional-Integral-Derivative
- QP** Quadratic Programming
- R2H** robot-to-human

**RA-L** IEEE Robotics and Automation Letters

**REBA** rapid entire body assessment

**RGBD** RGB+Depth

**RHC** Receding Horizon Control

**RNEA** Recursive Newton-Euler Algorithm

**ROS** Robot Operating System

**RPY** Roll-Pitch-Yaw

**RULA** rapid upper limb assessment

**UAV** Unmanned Aerial Vehicle

**UDT** Uni-directional Thrust

**UKF** Unscented Kalman Filter

**VS** Visual Servoing

**VTOL** Vertical Take-Off and Landing

**w.r.t.** with respect to

**WO** Wrench observer



# List of Figures

1.1	First historical prototypes of automatic aircraft. . . . .	6
1.2	The three main categories of aerial robots based on the typed of wing. . . . .	6
1.3	Comparison of the thrust generation between an UDT (left) and a MDT (right) multi-rotor aerial vehicle. The actuators of the former generate thrust forces ( $f_i$ ) all parallel to the same axis with respect to (w.r.t.) the body. Hence, the Uni-directional Thrust (UDT) quad-rotor has to apply a moment ( $\boldsymbol{\tau}$ ) in order to steer the total thrust vector ( $\mathbf{f}$ ) along another direction. Contrary, the latter, id est (i.e.) the Multi-Directional Thrust (MDT) hexa-rotor, can generate forces along multiple axes as summation of the individual actuation contributions, while a null total moment can be achieved. . . . .	9
1.4	Examples of popular collinear multi-rotor aerial vehicles (MRAVs). All motor-propeller pairs are parallel to a common axis which is perpendicular to the robot body. . . . .	10
1.5	Examples of Multi-Directional Thrust and Omni-Directional Thrust MRAVs. . . . .	11
1.6	Examples of Multi-Directional Thrust and Omni-Directional Thrust MRAVs. . . . .	12
2.1	Pictures showing two human-robot handovers. In the image on the left, the robot is handing over the object to the human, while in the image on the right the inverse situation is taking place. . . . .	24
2.2	Schematic representation of the handover task. It comprises activities at the physical and social-cognitive level which cover all the phases involved in the process. The handover consists mainly of two phases: a prehandover part and a physical handover phase. In the former, the two agents approach and reach themselves while agreeing on mutual aspects of the collaboration. In the latter, the object transfer takes place. Later, in a fourth phase, the two agents retract. . . . .	25
3.1	Generic control architecture for vision-based physical interaction with fully-actuated platforms. In green the vision-based control, in orange the module in charge of the physical interaction, in blue the geometric controller, and in gray the aerial robotic platform. . . . .	46



3.2	A visual description of the Optimal Control Problem (OCP) formulation employed in the second subproblem (human-aware handover). We highlight the terms related to the reference trajectory, the perception and actuation requirements in red, green and blue, respectively. Then, in orange and purple we denote, respectively, the terms related to the safety and the ergonomics. The solution to this minimization problem provides the optimal robot states and inputs. . . . .	49
3.3	First part of the pick-and-place operation designed to experimentally validate the visual and physical control architecture allowing the robot to physically interact with the environment. The frames acquired by the onboard camera are reported on the right of each image. The second half of the experimental validation is reported in Figure 3.4. . . . .	51
3.4	Second part of the pick-and-place operation. . . . .	52
3.5	Experimental validation of the hierarchical control architecture allowing the robot to physically interact with a human worker. The experiments involve a Human-Aerial robot handover scenario, where an operator is working on a scaffolding and needs to return a tool to the robot. These images depict the approach and reach phases of the human-to-robot (H2R) handover process. Pictures of the next part of the experiments are reported in Figure 3.6. . . . .	54
3.6	Photos of the interaction phase of the human-to-robot (H2R) handover. The last part of this experimental validation is reported in Figure 3.7. . . . .	55
3.7	Last part of the human-to-robot (H2R) handover. Here, the images show the last portion of the interaction between the human and the aerial robot, and the retraction phase where the robot returns to the initial position. . . . .	56
3.8	First part of the simulations of the robot-to-human (R2H) handover we carried out to validate the human-aware Nonlinear Model Predictive Control (NMPC) framework. For each subfigure, the current frame acquired by the onboard camera is reported on the bottom left corner. The tiny red dots over the fiducial markers (AruCo tags) denote their correct detection. The second half of the simulation validation is reported in Figure 3.9. . . . .	57
3.9	Snapshots related to the second part of the simulation validation campaign of the “human-aware” controller. . . . .	58
3.10	Experimental validation of the human-aware Nonlinear Model Predictive Control (NMPC)-based controller which achieves a safe and ergonomic robot-to-human handover. The controller is informed that a virtual object is frontally attached to the robot body. This virtual item is visualized in 3.10c as a green dot attached to a virtual rigid arm (the dashed green line) at the robot frame. . . . .	59
4.1	Here, a schematic representation of a Generically-Tilted Multi-Rotor (GTMR) is provided as well as the depiction of the main frames used to describe its motion. The $\mathbf{x}$ , $\mathbf{y}$ and $\mathbf{z}$ of each frame are respectively drawn using the RGB convention. . . . .	77

4.2	This figure presents a schematic drawing of an aerial manipulator, being a composition of a Generically-Tilted Multi-Rotor (GTMR) and a robotic arm. . . . .	80
4.3	This pictures show the angle of attack $\mu_i$ and the value of $\bar{c}_{s,i}$ of the $i$ -th counter-clockwise (CCW) or clockwise (CW) propeller for positive (counter-clockwise (CCW)) and negative (clockwise (CW)) spinning velocities $\Omega_i$ . Accordingly, the thrust force $\gamma_i$ along the propeller axis $\mathbf{z}_{A_i}$ and the drag torque $\tau_{d,i}$ are drawn. . . . .	85
4.4	This figure illustrates the model used to describe the human worker as a composition of two parts: a trunk and an arm. The second arm is drawn just for visual purposes. . . . .	94
5.1	Generic visual and physical control architecture enhancing fully-actuated multi-rotor aerial vehicles (MRAVs) with physical interaction capabilities. In green, the vision-based control constituted of a Hybrid Visual Servoing. In orange, the physical interaction layer comprising an external wrench observer and admittance filter. In blue, the motion controller and, lastly, the aerial robotic platform and the robot state estimation module in gray. . . . .	98
5.2	Internal structure of the hybrid Wrench observer. . . . .	102
5.3	Schematic representation of the geometric controller adopted to stabilize the translational and rotational dynamics of the fully-actuated multi-rotor aerial vehicle (MRAV). . . . .	104
5.4	On the left, a picture of the fully-actuated hexa-rotor used in the experimental validation, while on the right a picture taken from the onboard camera. In the latter image, the bricks and the placing location are shown. Their successful detection is marked with a red dot displayed over their fiducial markers. . . . .	107
5.5	Schematic drawing representing the AR closing the distance with the object to pick, on the left, and during the contact phase, on the right. The origins of the main frames are drawn. For instance, $O_F$ is the origin of the feature frame $\mathcal{F}_F$ . The vectors ${}^S\mathbf{p}_F$ and ${}^S\mathbf{p}_E$ are the positions of the feature point and the robot end effector (EE) tip expressed in $\mathcal{F}_S$ , respectively. . . . .	109
5.6	Position tracking during the two experiments. In dashed lines, the position references of the robot trajectory, while in continuous lines, the current robot coordinates. The blue area shows a phase of interest, as detailed in Section 5.2.2. . . . .	110
5.7	Plots related to the first experiment (“Full”) only. In the top, the robot height from the ground in $\mathcal{F}_W$ , while in the bottom, the estimated contact force ${}^B\hat{\mathbf{f}}_{\text{ext}z}$ along $\mathbf{z}_B$ . . . . .	112
5.8	Figure related to the first experiment (“Full”) only. The displacement and heading of the detected brick in the image plane of the onboard camera. . . . .	112
5.9	A picture of the aerial manipulator adopted in the human-to-robot (H2R) handover experiment. . . . .	117

5.10	A picture of the aerial manipulator adopted in the human-to-robot (H2R) handover experiment. For visualization purposes, a hand gripper is mounted as end effector. . . . .	118
5.11	A picture of the human-to-robot (H2R) handover taken during the interaction phase. The direction along which the robot is compliant and the human can guide its end effector (EE) is shown in red. It is parallel to $\mathbf{x}_W$ . Along the other axes, the robot exhibits a stiff behavior.	119
5.12	The plot is divided into three different highlighted regions representing the three phases of the experiment. Here, the actual position of the aerial manipulator (AM) and its reference trajectory are displayed. . . . .	120
5.13	Here, we report the position errors of the aerial robot tracking the reference trajectory throughout the experiment. In general, the motion controller provides acceptable and satisfying tracking performance in all phases. . . . .	120
5.14	In these plots, we provide the estimated force along $\mathbf{x}_W$ , which is the direction excited by the human during the interaction phase. Due to real system uncertainties, a non-zero force is estimated even when there is no contact, specifically during the approach-and-reach phases and the retraction part of the handover. This constitutes one downside of using a model-based wrench estimator. This problem is generally solved by using thresholds within a deadzone strategy. Here, we show the output of the wrench observer without (top) and with (bottom) using such a technique. The chosen deadzone thresholds for the force output are $\bar{\mathbf{f}}_{\text{ext}} = \mathbf{f}_{\text{ext}} = [1 \ 0 \ 0]^\top$ . As we consider only the forces along $\mathbf{x}_W$ , we set only the first element different from zero in the vectors containing the threshold parameters. . . . .	122
6.1	The visual description of the Optimal Control Problem (OCP) already presented in Chapter 6, and specifically in Figure 3.2. . . . .	132
6.2	This picture shows the vector representing the robot position expressed in the human-trunk frame $\mathcal{F}_{H_t}$ , id est (i.e.), the quantity ${}^{H_t}\mathbf{p}_B$ . . . . .	134
6.3	A schematic drawing showing the bearing vector going from the camera origin to the feature $F$ , and the related angular distance $\beta$ . . . . .	135
6.4	This picture shows the vector representing the human hand position expressed in the shoulder frame $\mathcal{F}_{H_s}$ , id est (i.e.), the quantity ${}^{H_s}\mathbf{p}_{H_h}$ . . . . .	136
6.5	3D (left) and 2D (right) representations of the shoulder-frame $\mathbf{x}_{H_s}$ , $\mathbf{y}_{H_s}$ plane. . . . .	137
6.6	Schematic top view representation of the human-aerial robot handover. The human trunk is the larger purple circle, while the smaller one represents the user shoulder. The robot is depicted in yellow, while the carried object as a red circle. . . . .	140
6.7	Two trajectories generated by the reference motion algorithm are provided for two different starting robot positions, which are marked in blue and red, respectively. Each trajectory is composed of five waypoints: $\text{WP}_1$ to $\text{WP}_5$ . . . . .	146

6.8	On the left, a snapshot of the simulated platform during the approaching phase. On the right, a frame of the robot onboard camera taken during that motion. Some Aruco markers are attached to the human body to allow the robot perception system to detect and estimate the human pose from the camera images. . . . .	148
6.9	Block diagram showing the implementation of the “human-aware” control framework. . . . .	149
6.10	Top view of the approaching phase, id est (i.e.), in the $\mathbf{x}_W, \mathbf{y}_W$ plane. In light blue, the initial reference motion task generated by our motion planning strategy. In blue and orange, we draw the executed trajectory of the robot in two simulations, (1) and (2), while the trajectory of the human is illustrated in red. The human motion is the same one for both cases. . . . .	150
6.11	Estimated and ground-truth values of the human position and yaw angle during the approaching phase. . . . .	151
6.12	Visibility constraint over time during the approaching phase. The quantity $c_{\beta}$ synthetically summarizes the Field of View (FoV) constraints Equations (6.50j) and (6.50k) in a 1D representation. . . . .	152
6.13	Side view of the human arm plane. The color gradient shows the sum of the absolute values of the human arm joint torques. $p_O$ denotes the projection of the object point $O$ in the shoulder plane, during the reaching phase. In particular, the carried item starts from the initial position, being the black square, and it moves to the final handover location, which is marked as a black triangle. . . . .	153
6.14	In this picture, we show the testbed setup used in the experiments related to the “human-aware” predictive controller. . . . .	156
6.15	In the top, the perception objective is shown. In the middle, the $x, y$ and $z$ coordinates related to the ground-truth position of the human mockup panel, respectively in red, green, and blue. In the bottom, the yaw angle of the human avatar board. . . . .	158
6.16	In the top, the estimated and the ground-truth position coordinates of the human (trunk) board, respectively as dotted and dashed lines. In the bottom, and in a similar fashion, its estimated and ground-truth yaw angles. . . . .	158



# List of Tables

1.1	The work detailed in this manuscript is based on the three publications listed here. For each one, we provide a brief summary of the article content, a list of its main elements and the international conference where it has been published. . . . .	21
2.1	This table shows the decomposition of the Human-Aerial robot handover (HARH) problem into two complementary subproblems (SP1 and SP2) based on the research challenges we discussed in this chapter. It shows the sections of the next chapter providing an introduction to the methodology addressing each subproblem, and the chapters with a more in-depth treatment. The symbols ✓ and ✗ denote, respectively, the analysis or the disregard of a certain aspect. “C.EE.P.” and “Visib.” denote the use of perception to control the robot end effector (EE) pose and the problem of maintaining visibility of the human partner, respectively. Likewise, “6D” is the exploitation of fully-actuated aerial robots, while “Limit.” is the consideration of the actuation limitations.	32
5.1	Means and standard deviations of the distance error $e_d$ , and rotation-tracking ones $\mathbf{e}_\eta$ . The two columns refer to the two conducted experimental cases: using our entire proposed control architecture (“Full”) and the same one, but with the physical interaction module disabled (“No phy.”). . . . .	109



# List of Multimedias

3.1 QR code to the video showcasing the pick-and-place validation campaign. Link: <a href="https://peertube.laas.fr/w/eEY4Q8nWdj1z5vs4JBaMpt">https://peertube.laas.fr/w/eEY4Q8nWdj1z5vs4JBaMpt</a> . . . . .	53
3.2 QR code to the video presenting the experiments conducted at the University of Twente in the scope of the Aerial COgnitive integrated multi-task Robotic system with Extended operation range and safety (Aerial-CORE) project. Link: <a href="https://youtu.be/LrQxXbQ5IHc">https://youtu.be/LrQxXbQ5IHc</a> . . . . .	55
3.3 QR code to the video showing the simulations realized in the Gazebo simulator for validating the “human-aware” controller. Link: <a href="https://peertube.laas.fr/w/fvDnY4ZCaMhd5ztT2aUhS5">https://peertube.laas.fr/w/fvDnY4ZCaMhd5ztT2aUhS5</a> . . . . .	59
3.4 QR code to the video presenting the preliminary experiments realized at Laboratoire d’analyse et d’architecture des systèmes (LAAS) for testing the “human-aware” controller in real-world settings. Link: <a href="https://peertube.laas.fr/w/5aCzSTsi5BMhgXHSUB9Kg4">https://peertube.laas.fr/w/5aCzSTsi5BMhgXHSUB9Kg4</a> . . . . .	60





**Part I**  
**Preliminaries**



# Chapter 1

## Introduction

In this manuscript, we investigate control solutions aimed at enabling aerial robots (ARs) to interact effectively and safely with humans. In this chapter, we present the main topics covered by our research. Specifically, in Section 1.1, we provide an overview of the broad landscape of robotic systems that interact and collaborate with human partners. In Section 1.2, we narrow our focus to autonomous robots capable of flight, which constitute the subject of study in the field of *aerial robotics*. Among the variety of existing designs, our work considers uniquely those ARs that can fly by means of multiple rotors and propellers, known as *multi-rotor aerial vehicles* (MRAVs). Anticipating the next analysis, the usage of MRAVs is rapidly expanding across many applications, with a particularly interesting and promising area being the deployment of these robots for tasks involving physical interaction with both the environment and humans. Henceforth, in Sections 1.3 and 1.4, we delve into these domains, introducing the reader to the fields of *physical Aerial Robot Interaction* (pARI) and *physical Human-Aerial Robot Interaction* (pHARI). As pHARI represents the primary focus of our research, Section 1.5 offers an overview of the existing literature on that topic. This review of the state of the art paves the way to Section 1.6, where we outline the motivations behind the choice of the research problem we explored and the following investigation. Next, Section 1.7 provides insights into the project context incorporating our work, and the related research objectives. We then detail the key contributions and the research outcomes of the conducted analysis, respectively, in Sections 1.8 and 1.9. Lastly, in Section 1.10, we conclude this chapter with a summary of the content structure of this thesis.

### 1.1 Human-Robot Interaction

The idea of machines autonomously performing complex tasks, reasoning on the surrounding environment and working aside humans, was originally born in the fictional imagination<sup>1</sup>. In that context, for the first time, the term *robot*<sup>2</sup> appeared and the idea of *automata* started to spread out as the mental model of an automatic

---

<sup>1</sup>Notions of human-like creatures and machines dates back of centuries along with religion, mythology and philosophy. The interested reader is referred to [Goodrich, 2008] for a historical tour of the concept origins.

<sup>2</sup>The word robot comes from the Czech word “robota” meaning “labor”.

and human-like-behaving machine. This gave birth to subsequent literary works which found their emblem with the visionary novels of Isaac Asimov and the statement of his famous and iconic *Three Fundamental Laws of Robotics* [Asimov, 1950]. Likewise, preliminary attempts of realizing functioning prototypes of machines autonomously performing a simple assigned task followed in the course of centuries, which contributed to the diffusion of the term and paved the way to modern robotics.

Today, that futuristic vision that seemed possible only in a distant future is a concrete part of our everyday life. Industrial robots represent a first evident and convincing case: while experiencing a large widespread that started almost 40 years ago, these machines have been equipped with various sensors and tools to accomplish a variety of tasks [Angerer, 2013; Djuric, 2016]. As a matter of fact, robots are nowadays employed in a wide range of industrial fields, including assembly lines, quality inspection, packaging, fabrication of electronic components, chemical and pharmaceutical processes, to name a few. Thanks to their capability of executing repetitive, tedious and hazardous motions and actions in a precise and rapid manner, robotic systems have contributed to improve the productivity and the efficiency in many sectors [Hentout, 2019]. Although industrial automation existed before the appearance of manipulator arms within factories, the use of robots has relieved human workers of carrying out alienating manual activities and avoided exposing many operators to dangerous working conditions [Heyer, 2010].

Due to the limited flexibility and adaptability, industrial robotic systems require a precise knowledge of the surrounding environment and a re-programming phase to be re-adjusted to a new working setup [Heyer, 2010]. Therefore, robots were initially meant to operate apart from the human partners employed in the same factory. Working cells and cages have been designed to restrain the workspace of the deployed robotic systems, although offering a fixed and structured environment for the robots to safely and efficiently perform their activities. Indeed, no particular restrictions in terms of movements, speed and force profiles are necessary, unless those imposed by the specific task and the avoidance of potential collisions. In these conditions, the robot primary focus is on the task execution and the performance maximization.

Recently, robotic systems have gained increased motor capabilities, reasoning skills and autonomy thanks to the huge research efforts made in the field of control, motion planning, perception and artificial intelligence [Hirzinger, 1999]. This has been accompanied by the exponential and favourable technological advances in processing and computing hardware, software architectures, and the availability of high-performance sensors and actuators [Bekey, 2008]. Robots are now capable of performing accurate and rapid movements, process large volumes of data, perceive the surrounding environment, and take complex decisions. Nevertheless, they are still not capable of working fully independently from their human partners, since they lack the high-level reasoning skills and the degree of adaptability to fast-varying environmental conditions that is typical of humans. This brought the idea of leveraging the strength and the accurate motor capabilities of robots, and combine them with the versatility, the decision-making abilities and the cognitive skills of humans [Djuric, 2016]. To facilitate the collaboration between human operators and robotic systems, the physical barriers dividing their workspace are now removed. Therefore, the two agents move and act in the same environment, so that the robots can assist their partners in various tasks, which would be otherwise rather complex

and costly to fully automatize. As a result, the paradigm of *Human-Robot Interaction* (HRI) started to emerge in the mid 1990s and early 2000s. This new field aims at reducing the physical and mental burden of human workers by successfully deploying *collaborative robots*, or simply *cobots*, as useful human collaborators. Nevertheless, this has been also accompanied by the arising of new challenges, such as guarantee the user safety by avoiding potential collisions, increase the operator ergonomics, and allow robots to better understand the human current activities and needs [Kirsch, 2010].

Currently, within factories, HRI is becoming easier and safer with the emergence of industrial cobots, such as the UR5 of Universal Robots<sup>3</sup>, the Baxter of Rethink Robotics [Soratana, 2018], the LBR iiwa of Kuka<sup>4</sup>, or the robotic arms from Franka Emika<sup>5</sup>. Likewise, in the research panorama, the HRI trend brought forth the development and the investigation of new types of robots in collaborative tasks with humans, like assistive humanoids [Goodrich, 2013; Vianello, 2021], quadrupeds [Xiao, 2021; Morlando, 2023], and exoskeletons [Zhou, 2021], to name a few. This innovating push is focusing on developing robotic assistants that can possibly help humans not only in industrial settings, but also in domestic and office environments in a near future. In fact, the goal of the robotics community is to improve the overall quality of human life by not only reducing fatigue and stress but also increasing human capabilities in terms of force, speed and precision [Santis, 2008].

The idea of robotic systems in collaboration with humans is crossing many other research topics, such as the medical and the care-giving sectors, space and underwater exploration, automotive, and also sociology and ethics [Goodrich, 2008]. As a matter of fact, HRI is not only meant to be physical and in closed proximity, but it can be remote and it can encompass socio-cognitive aspects. Therefore, Human-Robot Interaction is a multi-disciplinary domain which also comprises applications like tele-operation of mobile robots in challenging terrains and hard-to-reach areas, tele-manipulation of robotic arms, and robots assisting persons in everyday scenarios [Darvish, 2023]. The interaction can take place either by means of a remote device, which allows controlling a distant robot and receive sensory feedback from its operational environment, or through emotive and cognitive computing, while exploiting natural and body languages, *exempli gratia* (e.g.), speech and gestures.

As it will be clear later, one promising field where it is permeating the idea of having humans interact with robotic agents is the one of aerial robotics.

## 1.2 Aerial robotics

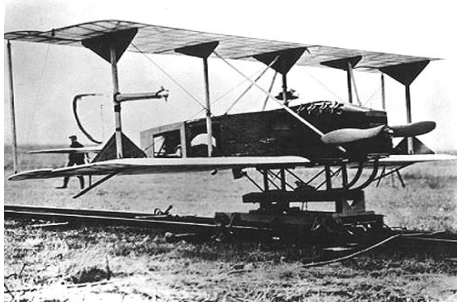
In the wide panorama of robotic systems, aerial robots (ARs) have attracted exponential attention in the last decades. The term *aerial robot* refers to *a system capable of sustained flight with no direct human control and able to perform a specific task* [Feron, 2008]. Although this definition encompasses remotely controlled aircraft, aerial robotics mainly focuses on the problem of increasing the system autonomy

---

<sup>3</sup><https://www.universal-robots.com/products/ur5-robot/>.

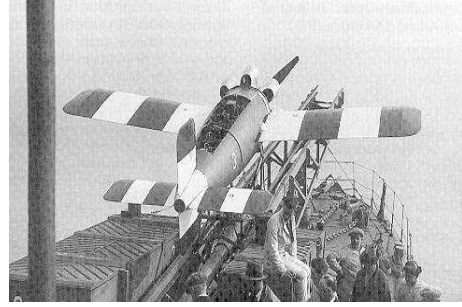
<sup>4</sup><https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/lbr-iiwa>.

<sup>5</sup><https://www.franka.de/>.



Courtesy of Wikipedia.

(a) The Hewitt-Sperry Automatic Airplane prototype developed in 1918, being the first automatic flying vehicle.



Courtesy of Wikipedia.

(b) The Royal Aircraft Establishment (RAE) Larynx is an unmanned aircraft whose first tests were carried out starting from 1927 in Great Britain.

Figure 1.1: First historical prototypes of automatic aircraft.



Courtesy of [Balaram, 2021].

(a) The NASA “Ingenuity” helicopter deployed on the planet Mars within the scope of the “Perseverance” exploratory mission.



Courtesy of [Hattenberger, 2022].

(b) Fixed-wing aircraft featuring an onboard sensor for cloud sampling and monitoring.



Courtesy of [Zufferey, 2021].

(c) The autonomous flapping-wing robot “E-Flap” featuring onboard sensing, planning and control capabilities.

Figure 1.2: The three main categories of aerial robots based on the typed of wing.

ideally aiming for zero human intervention or, more realistically, to minimal one. In the aeronautic jargon, ARs are also popularly known as Unmanned Aerial Vehicles (UAVs) since they lack a human pilot [Newcome, 2004].

The first prototypes of aerial robots, such as the Hewitt-Sperry Automatic Airplane and the Larynx shown in Figures 1.1a and 1.1b, were consisting mainly of flying vehicles with restricted autonomy and limited sensing capabilities. Over the course of the last century, the technological advances in robotics allowed the conception of more advanced and autonomous systems, capable not only of receiving and executing specific tasks, but also reasoning and taking automatic decisions. This was further encouraged by the integration of onboard sensors and powerful computing units, which lead to the deployment of these aerial robots in more complex, unknown or hazardous environments. Modern examples of ARs comprise, for instance, autonomous helicopters, as the one in Figure 1.2a, the robotic airplane in Figure 1.2b, or the bio-inspired robot in Figure 1.2c.

As evident from the examples reported in Figure 1.2, aerial robots encompass diversified and heterogeneous designs. According to the type of *wings* utilized

for flying, ARs can be categorized into three main classes<sup>6</sup>. The first category comprises *flapping-wing* ARs (Figure 1.2c), which typically own a lightweight airframe and a complex actuation system which imitates the flight mechanism employed by living creatures like birds and insects [Croon, 2009]. Due to their intrinsic mechanical complexity, a limited payload capacity, and the challenges in modeling the aerodynamic phenomena induced by the flapping phenomenon [Hoeijmakers, 2022], they represent the less commonly adopted type of UAVs. With a simpler structure, we have instead *fixed-wing* UAVs (Figure 1.2b). These aircraft were the first ones to be historically realized as their similarity to classical airplanes allowed for a smooth technological transfer [Newcome, 2004]. Compared to other aerial vehicles, this class of flying robots is well suited for long-range cruising missions thanks to their significant energy efficiency, since they exploit the effective wing-lifting mechanism for flying. The third category includes *rotary-wings* ARs, also known as *rotorcraft*, which feature one or more rotary wings capable of generating lift by accelerating the surrounding air when spun (Figure 1.2a).

Mono-rotary rotorcraft, namely helicopters, own just one main rotor for the generation of the lifting thrust, and they usually feature variable-pitch propellers and a complex rotor mechanism. While the first prototypes of helicopters were realized at the end of the 19<sup>th</sup> century, multi-rotor aerial vehicles (MRAVs) faced a slower spread and made their appearance only in the second half of the 20<sup>th</sup> century. The reasons behind their late development are mainly due to the scarcity of feasible technological solutions in terms of actuators and computing units. Consequently, the recent improvements in electrical motors and driving circuitry, combined with the advent of powerful embedded processors, boosted the realization of numerous designs and prototypes of MRAVs, mainly of small and medium sizes [Michael, 2010]. Compared to helicopters, MRAVs own a simple rotor mechanism as they use more than one, usually at least two, fixed-pitch propellers to fly. In this case, the flight stabilization is performed by regulating the speed of each individual actuator, which allows generating a moment about any axis.

Differently from fixed and flapping-wing aerial vehicles, multi-rotor aerial vehicles can perform precise maneuvers and stable flight at a given given location (id est (i.e.), *hover*), which makes them more effective for those applications requiring accurate positioning. This comes at the expenses of a higher power consumption, since more energy is needed to sustain flight and cancel out gravity. Nevertheless, besides hovering, they own Vertical Take-Off and Landing (VTOL) capabilities [Bicego, 2020], and they can perform agile and high-speed maneuvers [Hehn, 2014; Mellinger, 2014; Hanover, 2023]. To overcome the power efficiency issue, hybrid designs have been proposed to combine the VTOL motion capabilities of MRAVs and the long-range energy autonomy of fixed-wing aircraft, thus addressing the shortcomings of both types of platforms [Morin, 2015; Anglade, 2019].

In recent years, the market flooded with affordable off-the-shelf materials and spare parts favouring a rapid and affordable prototyping of rotorcraft designs [Cutler,

---

<sup>6</sup>Actually, there are *four* classes of aerial robots. Among the three discussed in this manuscript, the fourth one includes *lighter-than-air* UAVs [Ruggiero, 2018]. Hereby, this category is not treated since it usually comprises aircraft of considerable sizes and payloads, such as autonomous blimps. Conversely, the present work focuses on small and medium-scale robots that can closely and physically interact with humans.



2015]. By exploiting this large hardware availability, the research community and enthusiasts roboticists have released several open-hardware projects and open-source software architectures which fostered the spread of MRAsVs [Abeywardena, 2015; Kotarski, 2021; Foehn, 2022].

Nowadays, the use of MRAsVs is rapidly proliferating to a broad spectrum of applications, thanks to their theoretical unlimited workspace<sup>7</sup> [Yüksel, 2019], their maneuvering capabilities and agility, and the possibility to transport different types of payload, such as sensors [Lindsey, 2012; de Angelis, 2019]. By looking at the research panorama, the scenarios where MRAsVs have been employed do not only include indoor and cluttered environments, but also outdoor scenarios and hard-to-reach locations. To name a few, we can list applications involving aerial photography and mapping [Trujillo, 2016; Petráček, 2020], environmental and urban monitoring [Ristorto, 2015], search-and-rescue missions and area patrolling [Scherer, 2015].

Additionally, in the last decade, MRAsVs have also got the attention from the civil society and companies. As a matter of fact, commercial solutions have started to experience a fascinating outbreak. Along this line, clear examples are the cases of *Parrot*<sup>8</sup> and, more recently, *DJI*<sup>9</sup>, which propose commercial solutions in the segment of personal and professional entertainment (mostly photography and video making). Moreover, one may recall the statement from one of the most prominent e-commerce websites in the world about the future plan of delivering packages to its clients by means of autonomous aerial robots<sup>10</sup>. Recently, also other companies started to envision the deployment of MRAsVs for mass mobility and goods shipping. For instance, the companies *Jetson*<sup>11</sup> and *Volocopter*<sup>12</sup> are actively developing and testing multi-rotor-based aircraft for the private and business transportation sector. Lastly, also governments have started to deploy MRAsVs for several civil applications and they are currently evaluating the social and economic impacts derived from their use. For instance, the Australian government has founded a research program with the objective of evaluating the possible economic and social benefits provided by the adoption of ARs in applications like agriculture, fire monitoring, disaster response and management<sup>13</sup>.

### 1.2.1 Taxonomy of multi-rotor aerial vehicles

As a wide array of heterogeneous multi-rotor designs is available, it is important to introduce the reader to some important taxonomy in order to facilitate the analysis carried out in the rest of this manuscript. In the following, we provide a brief taxonomy of multi-rotor aerial vehicles based on their actuation capabilities, i.e. their ability to generate forces along one or more directions in space.

---

<sup>7</sup>However, when deployed in the real world, MRAsVs operate in a *limited* workspace as, for instance, they cannot fly outside the atmosphere and only a limited amount of energy can be stored on-board with the current battery technology.

<sup>8</sup><https://www.parrot.com/us/drones>.

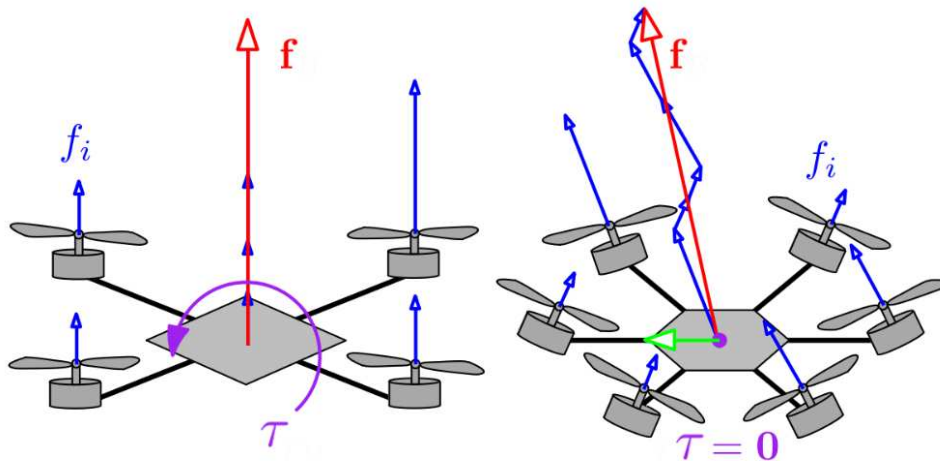
<sup>9</sup><https://www.dji.com/fr>.

<sup>10</sup><https://www.aboutamazon.com/news/transportation/a-drone-program-taking-flight>

<sup>11</sup><https://www.jetsonaero.com/>.

<sup>12</sup><https://www.volocopter.com/>.

<sup>13</sup><https://www.infrastructure.gov.au/department/media/news/drone-benefits-soar-new-report-shows>



Courtesy of [Bicego, 2019].

Figure 1.3: Comparison of the thrust generation between an UDT (left) and a MDT (right) multi-rotor aerial vehicle. The actuators of the former generate thrust forces ( $f_i$ ) all parallel to the same axis with respect to (w.r.t.) the body. Hence, the Uni-directional Thrust (UDT) quad-rotor has to apply a moment ( $\tau$ ) in order to steer the total thrust vector ( $\mathbf{f}$ ) along another direction. Contrary, the latter, i.e. the Multi-Directional Thrust (MDT) hexa-rotor, can generate forces along multiple axes as summation of the individual actuation contributions, while a null total moment can be achieved.

The typical MRV design features all the rotors parallel to just one single axis specified with respect to (w.r.t.) its body airframe. This configuration is said to be *collinear*, and it is clearly the simplest in terms of assembly and, as all the thrust force is usually oriented in such a way to compensate for gravity, the most energy efficient one. In turn, this causes the total thrust force, given by the sum of the individual contributions produced by each actuator, to be applied only along that axis, as shown on the left of Figure 1.3. This brought up the definition of *Uni-directional Thrust* (UDT) robots. If a lateral displacement is desired, the platform needs to change its body attitude in order to steer the thrust force vector towards the direction of motion, and produce the necessary lateral accelerations. Consequently, this rotor arrangement creates a coupling within the translational and the rotational dynamics. To produce a lateral movement the robot needs to vary its orientation first, and accordingly a change in the vehicle attitude generates a displacement in a certain direction. For this reason, collinear MRVs fall into the category of *under-actuated* robots. Common examples of UDT MRV are the quad-rotor and the collinear hexa-rotor as the ones shown in Figures 1.4a and 1.4b, respectively.

Collinear MRVs can achieve arbitrary 3D positioning at the price of sacrificing the attitude regulation, as they cannot attain desired orientations while performing a longitudinal or a lateral motion. Consequently, under actuation may constitute a serious problem in those applications where it is required to navigate in narrow and cluttered environments, or to resist wind gusts and other external disturbances while maintaining a desired orientation [Bicego, 2019]. Special treatment is necessary at the control level to account for the coupled dynamics, since its negligence can deteriorate the system performance or possibly result in the task failure [Fantoni, 2001].

To solve this issue, the research community has recently proposed a different



Courtesy of © Cyril FRESILLON / LAAS / CNRS Photothèque.

(a) UDT quad-rotor: 4 motor-propeller pairs.



Courtesy of [Liu, 2013].

(b) UDT hexa-rotor: 6 motor-propeller pairs.

Figure 1.4: Examples of popular collinear MRAVs. All motor-propeller pairs are parallel to a common axis which is perpendicular to the robot body.

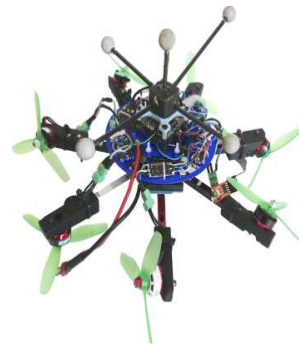
rotor arrangement, where the propellers are mounted along different axes, as shown in Figure 1.5. This allows generating forces along multiple directions, not only perpendicular to the robot body but also laterally, in turn allowing the platform to perform lateral displacements without the need to vary its body attitude. As a result, these particular rotorcraft constitute *fully-actuated* robots, and they feature a fully-decoupled translational and rotational dynamics. Interestingly, it has been proven that MRAVs need to be equipped with at least 6 propellers in a non-collinear (tilted) configuration to achieve full actuation [Michieletto, 2018]. Additionally, since fully-actuated ARs can apply forces in all or more than one direction, as schematically depicted on the right of Figure 1.3, the term *Multi-Directional Thrust* (MDT) is used in the literature [Franchi, 2018; Bicego, 2019; Hamandi, 2021]. However, in order to minimize the energy waste, tilted-rotor aerial vehicles are characterized by a principal direction of thrust along which most of the thrust can be applied. Whereas, the maximum thrust allowed along non-principal (lateral) directions is limited, and it is typically much lower than the one along the vertical body axis. For this reason, MDT robots are also termed *laterally-bounded force* (LBF) platforms by the research community [Franchi, 2018; Bicego, 2019]. The larger the amount of lateral force they can produce, the more closely ARs are proven to be capable of tracking an arbitrary 6D trajectory, i.e., any combination of position and orientation references [Franchi, 2018]. Thanks to the dynamical decoupling, fully-actuated MRAVs are well suited for moving across obstacles, narrow gaps in cluttered environments, and they can hover or take-off at non-flat attitude configurations.

The inclusion of an additional propeller in a fully-actuated hexa-rotor, thus obtaining a hepta-rotor, is proven to be a necessary condition for obtaining *omni directionality*, i.e. the ability to generate a force in any direction within a sphere centered in the robot body [Tognon, 2018; Hamandi, 2020]. As a result, this type of flying vehicles can arbitrarily fly in any direction and attaining any body orientation, thus they have received the appellation of *Omni-Directional Thrust* (ODT) aircraft. In recent years, prototypes of omni-directional ARs have been realized, as the one reported in Figure 1.5b, and they have been shown to offer incredible maneuvering capabilities [Allenspach, 2023]. Nevertheless, this is paid with a more challenging modeling of their flight dynamics, the need of advanced control



Courtesy of © Cyril FRESILLON / LAAS / CNRS Photothèque.

(a) Example of MDT hexa-rotor built at LAAS.



Courtesy of [Hamandi, 2020].

(b) Example of ODT hepta-rotor built at LAAS.

Figure 1.5: Examples of Multi-Directional Thrust and Omni-Directional Thrust MRAVs.

algorithms to exploit the omni directionality, and a non-trivial mechanical design and realization process [Hamandi, 2020]. When the platform has more than 6 propellers and it is fully-actuated, it results in an *over-actuated*<sup>14</sup> robot [Hamandi, 2021]. Indeed, the platform can exploit the redundancy to perform secondary tasks, e.g., remain operative in case of a rotor failure or better reject an external disturbance.

A straightforward approach to obtain a fully-actuated MRAV is by *fixedly* arranging the propellers along non-coplanar axes, as in the case of the platform shown in Figure 1.5a, which results in a *passive* rotor configuration [Bicego, 2019]. This design choice does not require additional components which would otherwise increase the take-off mass, consequently reducing the transportable payload and the flight time. Alternatively, the research community has recently proposed the adoption of an *active* rotor arrangement by employing additional actuators (e.g., servo motors) to control the orientation of the propellers, or just a subset of them [Ryll, 2012; Odelga, 2016; Kamel, 2018; Zheng, 2020; Ryll, 2022; Aboudorra, 2023]. Clearly, this ends up in a far more complex and heavier mechanical assembly, providing a less energy efficient aircraft but granting increased motion capabilities to the robotic system.

For an in-depth discussion on the actuation capabilities of ARs, their taxonomy and more details about existing designs, the interested reader is referred to [Bicego, 2019; Rashad, 2020; Hamandi, 2021].

### 1.3 Physical Aerial Robot Interaction

Given the recent advancements in control, sensing, and actuation of ARs, the robotic community shifted its interest from contact-less to *contact-based* applications [Ollero, 2021]. This has been possible by integrating an end effector (EE)<sup>15</sup> on-board the aircraft with the objective of enabling the robot to perform work on the environment. The EE choice encompasses rigid tools fixedly mounted to the robot body [Nguyen,

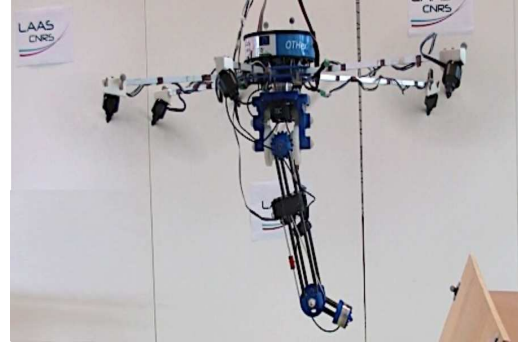
<sup>14</sup>The over actuation is expressed w.r.t. the task of following a 6D pose trajectory.

<sup>15</sup>The end effector (EE) of a robot is usually the terminating part of its structure which performs a task on the environment by being in contact and exchanging forces and torques with it.



Courtesy of [Sanalidro, 2022].

(a) Example of MDT hexa-rotor built at LAAS.



Courtesy of [Nava, 2020].

(b) Example of ODT hepta-rotor built at LAAS.

Figure 1.6: Examples of Multi-Directional Thrust and Omni-Directional Thrust MRAVs.

2013; Gioioso, 2014; Bodie, 2021a], poly-articulated robotic arms [Kim, 2013; Baizid, 2017; Tognon, 2017; Nava, 2020], and even the use of one or more cables attached to a payload to be transported [Sanalidro, 2022; Gabellieri, 2023].

On the one hand, MRAVs are typically under-actuated and the use of a robotic arm aims at overcoming such a limitation and increasing the dexterity of the platform. Thus, the system redundancy provided by the Degrees of Freedom (DoFs) of the additional manipulator can be exploited to accomplish secondary tasks, such as avoiding potential collisions, or performing postural adjustments in case of rotor failure or external disturbances applied to the AR (e.g., wind gusts) [Affi, 2022; Pose, 2022]. However, this solution does not come without any drawback: the weight of the attached manipulator arm decreases the available payload and reduces the flight time, while increasing the overall mechanical complexity. Moreover, the arm motion affects the dynamics of its flying base, thus leading to a more challenging control problem [Ruggiero, 2018; Ryll, 2019].

On the other hand, the use of a rigid tool fixedly-attached to the robot airframe requires to independently control the EE 6D pose. In the case of under-actuated ARs, this requirement is unfeasible due to the coupled rotational and translation dynamics. In this direction, recent research works have shown that the application of forces and torques on under-actuated MRAV might create stability issues and complexify the control problem. For instance, a force applied not at the center of mass (CoM) of the robot body produces a dynamics which is neither guaranteed to be stable nor easy to stabilize and even not practical for real applications [Nguyen, 2013].

A work-around solution could be the integration of one or more cables for controlling the pose of a carried object and tool. While this simple and cost-effective solution allows decoupling (partially) the rotational vehicle dynamics from the one of the payload, it requires particular attention from the planning and control point of view [Liang, 2018; Sanalidro, 2022]. Indeed, an imprecise load positioning might result into undesired oscillations potentially affecting the platform stability and possibly causing collisions with the environment, which clearly result into the task failure.

Henceforth, the research community has proposed the use of fully-actuated MRAVs with tilted propellers, as they can apply forces along any spatial direction and torques about any axis, i.e., a 6D wrench. This property makes them perfect candidates for tasks requiring physical interaction, and besides they do not require

particular extra components than a simple adjustment of their structural design. In the literature, this idea has been proposed as “*The Flying End-Effector*” paradigm [Ryll, 2019]. Specifically, it consists in the use of fully-actuated aerial vehicles which are capable of controlling the 6D pose of a rigidly-attached EE and consequently perform physical interaction in the environment. This has largely encouraged the research community to adopt fully-actuated aerial robots in heterogeneous contact-based applications. Examples are visual and physical inspection of hard-to-reach, dangerous and industrial sites and structures [Jimenez-Cano, 2017; Sanchez-Cuevas, 2017; Jiang, 2018; Ollero, 2018; Tognon, 2019; Paneque, 2022; González-Morgado, 2023], payload transportation and manipulation [Fink, 2011; Villa, 2020; Cacace, 2021; Sanalidro, 2022; Gabellieri, 2023] object grasping [Augugliaro, 2014], pick-and-place and assembly tasks [Lindsey, 2012].

These research efforts contributed to the growth of the recent and flourishing field of *physical Aerial Robot Interaction* (pARI), whose main challenge comprises obtaining stable flight while the AR is physically interacting with the environment [Yüksel, 2019]. From the physics point of view, this is performed by applying meaningful forces and torques (wrenches) from the robot actuators to the environment, while the flying vehicle is accepting the contact reactions in a secure and controlled manner.

Nowadays, also the concept of *Aerial Manipulation* is emerging, i.e. of ARs with manipulation capabilities similar to those owned by robotic arms [Ruggiero, 2018]. In order to properly manipulate objects, robots need more dexterity and possibly a larger number of DoFs. In this context, *aerial manipulators* (AMs), which are obtained by the combination of a MRVAV (as flying base) and a robotic arm, are excellent candidates. Therefore, the research community started to develop new light-weight manipulators to be embedded on-board aerial vehicles, and to address the mechanical and control issues which limited the diffusion of poly-articulated AMs [Ruggiero, 2018; Ollero, 2021].

In the literature, this class of ARs has not only been equipped with a single open-chain serial robotic arm, but also with dual and parallel manipulators. For instance, in [Korpela, 2014], a dual-arm AM is employed for physical interaction tasks with the environment, such as valve turning in an industrial setting. In [Cataldi, 2019], two robotic arms are attached to a collinear hexa-rotor to increase the system redundancy and make the system achieve multi-objective tasks involving EE positioning, camera orientation and manipulation. Instead, in [Zhang, 2022], a delta manipulator<sup>16</sup> is mounted on a multi-rotor aircraft for aerial construction applications exploiting additive technology. Similarly, the works in [Bodie, 2021b; Clark, 2022; Stephens, 2022] feature a parallel robotic arm mounted on-board a MRVAV to increase the system redundancy and perform precise tracking of the end effector pose.

### 1.3.1 From research to industries

After the initial interest from the research community, also several industries started to invest in the field of physical aerial robotics motivated by the opportunities in performing tasks in challenging and inaccessible areas with autonomous and cost-

<sup>16</sup>A delta manipulator, or more generically a parallel robot, is a closed-chain robot whose geometry is composed of smaller manipulators which connect a base to an end effector [Siciliano, 2009].

effective systems. This would allow enterprises to avoid the deployment of large human teams in potentially dangerous working conditions and the use of expensive machinery. An aerial robot capable of carrying out a simple manipulation job, and featuring sensing capabilities, could assist human operators in hazardous and critical situations [Ruggiero, 2018]. Consequently, in the last decade, new companies and start-ups were founded to propose solutions in this direction.

In the field of contact inspection, *Voliro*<sup>17</sup> and *Donecle*<sup>18</sup> are experimenting different technological approaches. The former focuses on surface scanning of industrial plants and inaccessible structures (e.g., transmission lines and wind turbines) for monitoring purposes, while the latter on inspecting the fuselage of airplanes to find manufacturing and welding defects. Similarly, the *Skydio* company<sup>19</sup> aims at deploying MRAVs for several applications, ranging from area monitoring and surveillance, to inspection of power plants and bridges.

The efforts of the enterprise sector are contributing to the advancement of the technology required for increased real-world applications involving aerial robots in physical interaction. However, for an effective deployment of MRAVs in everyday environments, also social acceptability and environmental concerns shall be taken into consideration and addressed.

## 1.4 Physical Human-Aerial Robot Interaction

As seen so far, ARs that can interact physically with the environment are increasingly considered in the aerial robotics literature with emergent industrial use cases. From an operational perspective, these flying robots have the potential to automate several tasks in challenging scenarios. One prominent example of real-world use case, where the deployment of an aerial robot is advantageous, is *work at height*. Representative applications include non-destructive testing [Tognon, 2019], and inspection of wind turbines and power lines [Cacace, 2021; Paneque, 2022].

These settings usually require specialized and trained personnel employing expensive equipment and bulky machinery. Carrying and accessing different tools in these circumstances would require uncomfortable postures and a loss of focus from the current activity. Therefore, we envision that the MRAVs can provide assistance to human operators to perform their task, e.g. to retrieve the necessary tools and provide useful information about the user activities [Affi, 2022]. For instance, AR can easily fly to a target location while transporting the essential tools, relieving the human operator from carrying extra equipment. Similarly, multiple ARs could be employed to help human operators to collaboratively handle bulky and long objects, and move them from the ground to elevated locations.

Due to the task complexity, the dexterity and the decision-making skills of the human workers are invaluable. Thus, for ARs to provide useful functionality in these applications, they must be able to coexist and collaborate with their human partners. In other words, the AR should become an *aerial co-worker (ACW)*, i.e.

---

<sup>17</sup><https://voliro.com/>.

<sup>18</sup><https://www.donecle.com/>.

<sup>19</sup><https://www.skydio.com/>.

a useful collaborator capable of performing the assigned task. Within this context, AMs possess the required maneuvering, sensing and manipulation capabilities that allow them to swiftly move in unstructured environments, reach inaccessible areas, acquire useful data on the surrounding environment, handle and manipulate objects. Furthermore, when interacting with humans, the larger dexterity offered by the additional DoFs of a robotic arm can be used to easily reach the human worker without jeopardizing the user safety, e.g., preventing the operator from staying too close to the flying base and its sharp propellers.

This opens a new panorama of opportunities and paves the way to the field of physical Human-Aerial Robot Interaction (pHARI), i.e., of aerial robots that can safely and physically interact with their human partners.

## 1.5 Related works in pHARI

Motivated by the growing interest in having aerial robots interacting with humans, the research community has started to investigate the problem of achieving safe, comfortable and natural human-UAV interaction. Hereby, we provide a panorama on the related works in this direction.

Early articles have been mostly encompassing the so-called domain of *proxemics* [Lieser, 2021], i.e. identifying suitable approaching motions and engaging modalities for aerial robots sharing the same workspace of humans. Along this investigation line, Duncan et al. [Duncan, 2013] performed the first known user study to evaluate the level of comfort experienced by several participants when a small-scale quad-rotor is approaching at different distances and heights. Similarly, later works analyzed the same problem with the objective of understanding how to improve the quality of interaction between MRAVs and humans in everyday life and working scenarios [Acharya, 2017; Yeh, 2017; Auda, 2021; Lieser, 2021].

The results obtained by the conducted user studies have highlighted how the close proximity to an aerial robot is provoking more mental discomfort and distress among the participants than a ground robot entering their personal space [Acharya, 2017]. In most cases, the main emerging factors are the unpredictability related to the trajectory executed by the AR, the difficulty in understanding its intention and the approaching direction, the unpleasant influence of the noise and the airflow generated by the rotating propellers [Acharya, 2017; Yeh, 2017; Lieser, 2021]. Additionally, the absence of human-like features in the design (e.g., face and limbs) and the erratic behavior of ARs make the interaction with this type of robots more challenging than in the case of ground mobile-based manipulators [Acharya, 2017].

Henceforth, the research community in the field of *Human-Drone Interaction* (HDI)<sup>20</sup> have explored the use of several sensory means (visual, acoustic) to convey information about the robot state and intentions to the involved users, driven by the goal of addressing these issues and improve the quality of the interaction [Naseer, 2013; Cauchard, 2015; Cauchard, 2016; Gio, 2021; Wilson-Small, 2023]. In this context,

---

<sup>20</sup>The denomination *drone* originates from the military sector. It was used to denote the autonomous aerial robots used in remote surveillance, training, and combat engagements. Then, it permeated to the public and press domains to refer to the same class of vehicles and, in particular, to indicate MRAVs [Bicego, 2019].



new multi-modal interfaces have been designed to assign tasks to multiple ARs, monitor their status and gather useful data from their onboard sensors [Cacace, 2016; Fernández, 2016]. These tools allow the operator to utilize body gestures and voice commands to naturally communicate with the MRAVs, since they represent the common means adopted in human-human interactions [Nagi, 2014; Krishna, 2015].

Inspired by the insights gathered by the HDI community, roboticists in the fields of HRI and pARI have realized that to achieve an effective and safe collaboration with aerial robots new control and planning frameworks are needed. In particular, proper methods have to be conceived to include the human state, such as ergonomics and safety, and to allow a physical contact between the two agents.

Despite an extensive literature exists on MRAVs physically interacting with the surrounding environment, the same cannot be said on physical collaboration with human partners. The idea of direct and contact-based interaction between MRAVs and human operators has only been considered just a few times in the literature.

In [Augugliaro, 2013], the authors investigate the physical interaction between a collinear quad-rotor and a potential user. Similarly, in [Rajappa, 2017], they consider the case of a human operator in direct contact with an under-actuated AR. Rajappa et al. develop a control architecture that allows rejecting possible external disturbances (e.g., wind and obstacle collisions) and, at the same time, it provides physical compliance to the operator actions. However, in these two works, the user is treated as a mere external disturbance acting on the robot dynamics, and the problem of guaranteeing a safe and comfortable collaboration is neglected. In [Tognon, 2021; Allenspach, 2022; Hallworth, 2023], the authors address a human-guiding use case consisting of a MRAV pulling a human to a location via the tension force transmitted along the cables connecting the robot and the user. Similarly to the previous works, the employed platform does not include any robotic arm and there is no direct close physical interaction between the aerial robot and the human operator. Differently, in [Li, 2023], the physical collaboration between a team of MRAVs and a human is investigated. In that paper, the authors propose a system that allows a human worker to guide a tethered multi-aerial-robot system transporting a payload to a desired location. At the same time, the developed architecture assures the avoidance of collisions with the potential obstacles in the environment and between each robotic agent. In [Suarez, 2022], the authors present a position-controlled delivery system, composed an aerial vehicle and a cable-suspended manipulator, that is capable of transporting a tool to the location where a human operator is standing.

In these works, neither human-centric metrics nor the state of the human are fully considered from the robot control and planning standpoints. Moreover, the operator is not in direct contact with the AR, since the interaction occurs by means of cables, EE devices or robotic arms connected through cables to the flying base. The use of tethers, although providing some degree of compliance, prevents a direct exchange of forces and torques between the human and the robot, which could be exploited to convey intentions during the interaction. With this idea in mind, Afifi et al. [Afifi, 2022] propose a control methodology that allows an aerial manipulator, equipped with a 3-DoFs robotic arm, to be in physical contact and in close proximity to a human operator in a simulated environment. Despite the control architecture allows the robot to be driven by the user actions, the method does not directly account for the human safety and ergonomics during the generation of the actuator

commands, as the robot may undertake motions that can injure the operator.

The authors in [Truc, 2022; Truc, 2023] propose an approach to consider the human state at the planning level, in order to generate kinematic-feasible trajectories for an ACW that considers the user safety, ergonomics and field of view. However, the latter methodology does not include the dynamics of the aerial robot. Hence, the resulting trajectory might be unfeasible to be tracked by the robot low-level motion controller. Additionally, the method has not been tested and evaluated with an AM in real-world experiments.

This brief overview of the literature in the field of pHARI evidences a lack of suitable control methods addressing the problem of achieving a *physical, safe and comfortable* human-aerial robot interaction.

## 1.6 Aim of the thesis

In the realm of HRI, the adoption of a robotic system has demonstrated its ability to increase efficiency in various applications. These include tasks as handing parts and tools to the workers in factories [Ortenzi, 2021], aiding in the manipulation of large objects [Maroger, 2022; Rapetti, 2023], and supporting elderly persons or physical-impaired patients at home by fetching essential items for daily activities [Goodrich, 2013]. Among the myriad of human-robot collaborative scenarios, a growing community of roboticists have tried to address the challenges in enabling seamless object handovers [Yang, 2021], as evidenced by the large volume of articles in this direction [Aleotti, 2012; Strabala, 2013; Medina, 2016; Yang, 2020; Ortenzi, 2021; Yang, 2021; Yang, 2022]. The capability for a smooth and safe object transfer constitutes a pivotal skill for a successful and effective deployment of a robotic agent in human-centric environments and manipulation tasks. When working at height, the ability to exchange objects with human workers is critical in these challenging and dangerous settings [Suarez, 2022].

As a preliminary step towards realizing a safe and effective physical Human-Aerial Robot Interaction (pHARI), our research concentrates on the still unexplored and challenging domain of Human-Aerial robot handover (HARH). Moreover, the distinctive characteristics of ARs, such as rapid and complex dynamics and limited actuation power, introduce novel challenges to the established domains of safe physical Human-Robot Interaction (pHRI) and human-robot handovers. Henceforth, this thesis aims to fill this research gap by investigating the HARH problem and proposing innovative control architecture to take a significant step towards achieving a safe and effective collaboration between humans and aerial robots. Our approach combines insights from the well-established field of HRI with the emerging domain of pARI to develop novel control methods tailored to the HARH case.

We specifically focus on small-scale MRVs, capitalizing on their maneuverability and suitable dimensions for cooperative use cases. Additionally, we limit our scope to aerial manipulators equipped with a single open-chain serial manipulator, excluding other existing designs and variants from our analysis<sup>21</sup>.

---

<sup>21</sup>For a comprehensive review of the existing designs of aerial manipulators, readers are pointed to these articles [Ruggiero, 2018; Ollero, 2021].

## 1.7 Context and objectives

The work conducted throughout this thesis takes place mainly in the context of the French ANR project<sup>22</sup> “The Flying Co-Worker” (FCW)<sup>23</sup>. This research project results from the collaboration between the French laboratories Laboratoire d’analyse et d’architecture des systèmes (LAAS) of Centre National de la Recherche Scientifique (CNRS), and the Institut National de Recherche en Informatique et Automatique (INRIA). These two laboratories are, respectively, affiliated with the Université de Toulouse and the Université de Lorraine.

The FCW project aims at combining the recent advances in the fields of physical Human-Robot Interaction (pHRI) and physical Aerial Robot Interaction (pARI) with the final objective of developing an *aerial co-worker*, i.e. an aerial manipulator capable of interacting with human partners in an industrial setting, e.g. in work-at-height scenario or in the workspace of a factory. The envisaged collaborative scenarios encompass (i) the Human-Aerial Robot (HAR) co-transportation and co-manipulation of long and bulky objects (e.g., a bar), (ii) and the HAR handover of a tool. Despite these problems have found several solutions for mobile-based manipulators, the methods are not easily transferable to aerial robots, due to the instability of their base, their complex and fast dynamics, and their onboard limited energy and payload. From the research standpoint, novel planning and control methods are necessary to guarantee a safe, effective and physical collaboration between the two agents.

This thesis tackles the development of novel control approaches to achieve safe and ergonomic physical Human-Aerial Robot Interaction (pHARI), in particular focusing on the Human-Aerial robot handover (HARH) problem. The perception and planning problems, such as a robust detection of the human pose and activity, or the generation of suitable trajectories, are relegated to the other parts involved in the FCW project.

Lastly, the activities presented in this manuscript are also partially comprised by the European project “Aerial COgnitive integrated multi-task Robotic system with Extended operation range and safety” (Aerial-CORE)<sup>24</sup>. Similarly to the FCW, it aims at developing aerial cognitive robots capable of interacting with the environment and assist human workers in high-from-ground and challenging environments, such as the inspection of power lines and other large infrastructures.

## 1.8 Thesis contributions

The previous sections support the significant relevance of the problem of letting an aerial robot collaborate with a human partner. As clear from the literature, the field of pHARI opens promising opportunities to automatize several tasks and assist human operators in several challenging and hazardous applications. However, the

---

<sup>22</sup>It is a public administrative institution under the authority of the French Ministry of Higher Education, Research and Innovation. The agency funds project-based research carried out by public operators cooperating with each other or with private companies. Source: <https://anr.fr/en/>.

<sup>23</sup><https://anr.fr/Project-ANR-18-CE33-0001>.

<sup>24</sup><https://aerial-core.eu/>.

presence of very few works on this topic suggests (i) the need of further exploring this subject, and (ii) the lack of effective solutions enabling the HAR collaboration. Moreover, from the authors' knowledge, the problem of HARH has still not been fully addressed, and no control methods are currently available within the research domain of aerial robotics to achieve a safe and comfortable HAR interaction.

Therefore, with this work, we take a preliminary and pioneer step towards the realization of aerial co-workers (ACWs) that safely and physically collaborate with human partners. In particular, the contributions of this work are fourfold:

1. Formally define and analyze the HARH problem according to the literature on human-robot handovers related to ground mobile-based robots.
2. Propose a valid decomposition of the considered problem into two subparts to facilitate its tractability, and highlight the associated challenges.
3. Propose two control architectures respectively tackling the key aspects of the HARH process, namely (i) the physical interaction between the two agents, (ii) the inclusion of user state awareness and human-centric metrics within the control framework.
4. Sketch an approach combining the two proposed methodologies in order to thoroughly achieve a HARH.

In particular, the proposed methods are inspired from the control strategies currently conceived in the fields of HRI and pARI. By exploiting our decomposition of the problem under analysis, we propose two control approaches.

The first control method is based on a hierarchical control architecture, which is a typical choice for controlling ARs and aerial manipulators (AMs), as we will discuss in Chapter 2. Then, in Chapter 3, we will show that it enables the aerial robotic agent to physically interact with the environment first, and with a human worker later. Specifically, we perform experiments involving a pick-and-place task and a tool-delivery scenario at height, respectively. In the former experimental case (pick-and-place), the AR is a fully-actuated MRAV and it exchanges forces with the environment through its end-effector. In the latter application (tool-delivery at height), the human and the robotic partner, namely an AM equipped with a poly-articulated robotic arm, directly interact without resorting to any intermediate element such as cables or passive joints. Additionally, a small item is exchanged. The collected results demonstrate the effectiveness of the proposed control solution to achieve physical human-aerial robot interaction by means of an aerial manipulator.

The second control architecture adopts a state-of-the-art optimization-based predictive control technique to conceive a “human-aware” controller. As outlined in Chapter 3, the proposed method accounts for the human state and human-centric metrics, such as safety and ergonomics, while taking into consideration the platform dynamics and some other task-related constraints. We validate the control architecture first in a safe and controlled, but yet realistic, simulation environment, which allows proving the soundness of the control formulation. Then, we test the controller in a set of experiments to showcase the effective applicability of such a method in a real-world scenario. Despite the method being applicable to any mobile-base robot, as we will show later in Chapter 6, we carry out the validation by employing a collinear quad-rotor. This type of AR is chosen because the under

actuation makes the achievement of the task more challenging, and the satisfaction of the related constraints harder. Therefore, the successful experimental results demonstrate the capabilities of the derived controller to cope with the platform actuation properties and limits.

Our methodology is inspired from the state of the art in human-robot interaction and aerial physical interaction, and it is tailored to this new use case, namely the HARH. The achieved results prove the feasibility in achieving safe and physical interaction between a human and heterogeneous aerial robotic platforms. The work presented in this manuscript and the conducted experiments go beyond the state of the art in the field of pHARI. We believe that this thesis can stimulate future research in the same direction, and subsequently bring forth the proposition of new control solutions tailored to collaborative ACWs.

## 1.9 Publication note

This thesis is based on three international conference articles, one of which has been submitted also to a journal, but it has been rejected due to some considerations raised by the reviewing committee. In this manuscript, and precisely in Chapter 6, we will report those valuable comments and we will show how we can take them into consideration to conceive improvements of that control method. Hence, we seek to present the improved and extended version to a journal venue in future work. Finally, a systematic view of the publications is reported in chronological order in Table 1.1.

## 1.10 Thesis outline

This section outlines the structure of the manuscript, making it easier for the reader to navigate through its three parts. For each part, we provide a list of chapters along with a brief description of the covered topics.

**Part I Preliminaries** has a fourfold objective. First, it introduces the main research fields that provide the context for the work presented in the rest of the document. Secondly, it familiarizes the reader with the problem under investigation, the underlying assumptions, and the research challenges addressed. Thirdly, it offers an overview of the literature related to the challenges associated with the analyzed problem. Fourthly, we briefly present the key features behind the proposed control methods and we summarize the main results obtained from the validation campaign of our methodology.

**Chapter 1 Introduction** provides an overview of the research landscape in the fields of Human-Robot Interaction (HRI), aerial robotics, and the interaction of flying robots with both the environment and humans. This background serves at setting the context for our work. Next, it outlines the aims of our investigation, the objectives, and the related research project. Lastly, it presents the contributions, academic outcomes, and provides an outline of this manuscript.

Publication	Contribution
<p>[Corsini, 2021] 2021 Workshop on Aerial Robotic Systems Physically Interacting with the Environment.</p>	<p><b>Summary:</b> Hierarchical control architecture for fully-actuated ARs performing tasks in the environment requiring perception and physical interaction capabilities.</p> <p><b>Key aspects:</b></p> <ul style="list-style-type: none"> <li>• Control architecture suitable for any fully-actuated MRAV.</li> <li>• Modular architecture featuring (i) a visual servoing scheme, (ii) a wrench observer and an admittance filter, (iii) a geometric motion controller.</li> <li>• Experimental validation with a real aerial robot in a pick-and-place scenario.</li> </ul>
<p>[Corsini, 2022] 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Joint submission also to IEEE Robotics and Automation Letters (RA-L), but rejected.</p>	<p><b>Summary:</b> Human-aware optimization-based and predictive controller for human-to-aerial robot handovers.</p> <p><b>Key aspects:</b></p> <ul style="list-style-type: none"> <li>• Control method suitable for any MRAV design.</li> <li>• Based on Nonlinear Model Predictive Control.</li> <li>• Inclusion of (i) human state, (ii) human-relative motion reference, (iii) human safety, (iv) human ergonomics, (v) human visibility.</li> <li>• Validation in simulations performed in the Gazebo simulator.</li> </ul>
<p>[Affi, 2023] 2023 International Conference on Unmanned Aircraft Systems. Collaborative work.</p>	<p><b>Summary:</b> Hierarchical control architecture for aerial manipulators physically interacting with human workers.</p> <p><b>Key aspects:</b></p> <ul style="list-style-type: none"> <li>• Control framework suitable for any AM equipped with a single open-chain serial manipulator.</li> <li>• Modular architecture comprising (i) a wrench observer and an admittance filter, (ii) a feedback-linearization-based motion controller.</li> <li>• Experiments with a real robot in a human-to-aerial robot handover scenario.</li> </ul> <p><b>Personal participation:</b></p> <ul style="list-style-type: none"> <li>• Assisting the preparation of the experiments.</li> <li>• Preparing and testing the robot hardware.</li> <li>• Co-writing the manuscript.</li> </ul>

Table 1.1: The work detailed in this manuscript is based on the three publications listed here. For each one, we provide a brief summary of the article content, a list of its main elements and the international conference where it has been published.

**Chapter 2 Research problem** introduces the Human-Aerial robot handover scenario. It also emphasizes the addressed challenges and the underlying assumptions. Furthermore, in this chapter, the considered problem is decomposed into two parts to facilitate its investigation and the development of effective control methods. Lastly, it provides a literature review of the related works, which will serve as motivation for our methodology.

**Chapter 3 Method overview** contains an overview of our proposed methodology, introducing two control architectures that address the two subproblems defined in the previous chapter. Finally, it offers a qualitative summary of the achieved results.

**Part II Scientific work** encompasses the technical content of our study. Henceforth, it provides the reader with the mathematical tools essential for deriving our control methods. Next, it elaborates on the formulation of each control architecture and presents quantitative results obtained from the validation campaign.

**Chapter 4 Modeling** collects the mathematical tools we employ to describe the agents involved in the HARH problem and the models used in formulating our control methodology.

**Chapter 5 Control methodology for visual and physical Human-Aerial Robot Interaction** provides a detailed explanation of the first control method. In particular, the considered subproblem involves an aerial robot interacting with the environment first, and with a human worker later. Lastly, this chapter presents quantitative results collected from the conducted real-world experiments.

**Chapter 6 Predictive human-aware control** presents the second control architecture: a “human-aware” optimization-based and predictive controller. This method considers the human state and human-centric metrics, such as safety and ergonomics, during the second subproblem, which involves a human-to-aerial robot handover. Additionally, the proposed controller incorporates task-related constraints, robot actuation limits, and a human-visibility requirement. Moreover, this chapter outlines possible future improvements of this method. Lastly, quantitative results on the controller performance are provided.

**Part III Conclusions** is the final part of this thesis, consisting of a single chapter that summarizes the conclusions and the perspectives related to the presented work.

**Chapter 7 Conclusion** terminates this manuscript. It synthesizes our work, discusses our findings, acknowledges the limitations related to our methodology. Additionally, it outlines the idea of combining the two proposed control frameworks to address the handover problem comprehensively. Lastly, it highlights the challenges not considered in this work and suggests potential future research directions.

# Chapter 2

## Research problem

This chapter defines the problem investigated in our studies, specifically the Human-Aerial robot handover (HARH). Henceforth, in Section 2.1, we provide a precise description of the problem, emphasizing the characteristics of the aerial case. Additionally, we outline the aspects of the problem that will receive our attention, and we list the underlying assumptions. Next, in Section 2.2, we present the research challenges that our work aims to address, laying the foundation for the problem decomposition introduced in Section 2.3. There, we divide the handover problem into two complementary parts to facilitate our subsequent analysis. Then, in Section 2.4, we offer a literature overview of the main methodologies addressing the challenges considered. Lastly, in Section 2.5, we conclude this chapter by providing the motivation for our proposed methodology, which we will present in the following chapter.

### 2.1 Human-Aerial Robot Handover

In this section, we delve into the *Human-Aerial robot handover* (HARH) problem, which is the central focus of this manuscript. We start by illustrating a general picture of the handover scenario, and subsequently we tailor it to the specific aerial case considered.

#### 2.1.1 Human-robot handover

In general terms, the *handover process* is a collaborative joint action where an agent delivers an object to a second agent, as shown in Figure 2.1. The former is defined as *giver*, while the latter as *receiver* [Medina, 2016; Ortenzi, 2021]. In this collaborative operation, the two agents shall perform joint actions in space and time to effectively achieve the task, i.e. exchanging the object. Clearly, the item can be transferred in both directions: from the robot to the human, *robot-to-human* (R2H) handover, or vice versa, i.e. *human-to-robot* (H2R), as respectively depicted in Figures 2.1a and 2.1b. In many works, the focus is only on an individual object exchange direction. For instance, in [Aleotti, 2012], the authors propose a planning methodology that successfully drive a robot to give an item to a user in a comfortable way, but they do not address the other transfer case. Contrary, the research in [Yang, 2021; Yang, 2022]





A robot is handing over a tool to a human.  
Courtesy of [Aleotti, 2012].

(a) Robot-to-human: the robot is the giver, while the human the receiver.



A human is handing a small cube to a robot.  
Courtesy of [Yang, 2020].

(b) Human-to-robot: the human is the giver, while the robot the receiver.

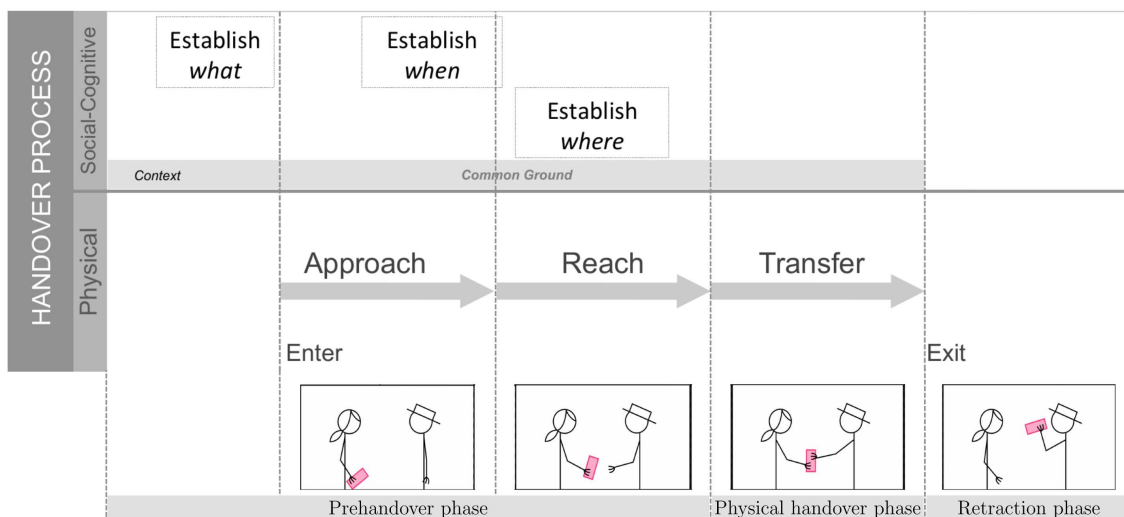
Figure 2.1: Pictures showing two human-robot handovers. In the image on the left, the robot is handing over the object to the human, while in the image on the right the inverse situation is taking place.

focuses on the inverse situation as they propose a vision-based system architecture to enable reactive H2R handovers for unknown objects.

As emerging from the investigation of human-to-human collaboration, the handover process involves coordinated behaviors of both agents at a physical and a social-cognitive level [Strabala, 2013]. The social-cognitive activities pivot on the decision about three main points, the *what*, the *when* and the *where/how*. To give a practical example, the agents must agree on the exchanged item (the *what*), decide the moment at which start transferring the object (the *when*), and lastly on where and how to meet for passing the object (the *where* and *how*). The physical level consists simply in the implementation of what the agents have agreed on during the social-cognitive coordination. Furthermore, the physical part comprises three main phases, namely the *approach*, the *reach*, and the *transfer*. The approach phase sees the giver moving towards the receiver while carrying the object to be exchanged. When the two agents are at a convenient distance, the reach phase occurs and it concludes with both agents being in contact through the object being passed. At this point, the actual transfer is starting and, in this moment, the two agents are usually maintaining a constant relative pose with respect to one another [Strabala, 2013]. After these three phases, the two agents retreat themselves. During this *retraction* phase, the receiver is holding the grasped object passed by the giver. In [Ortenzi, 2021], the first two phases, approach and reach, constitute the *prehandover* part of the process, while the third one (transfer) comprises the *physical* stage of the handover, as schematically depicted in Figure 2.2.

### 2.1.2 The aerial case

In the particular case of the HARH, one agent is an aerial robot, and the second one is a human worker who shares the same workspace of the robot and can move in the environment. The aerial robot shall be capable of giving or receiving the object to or from the partner in order to successfully perform the handover. In doing so, the robot end effector (EE) shall be moved to a reachable and convenient position



This figure has been realized by combining together two figures taken from [Ortenzi, 2021] and [Strabala, 2013].

Figure 2.2: Schematic representation of the handover task. It comprises activities at the physical and social-cognitive level which cover all the phases involved in the process. The handover consists mainly of two phases: a prehandover part and a physical handover phase. In the former, the two agents approach and reach themselves while agreeing on mutual aspects of the collaboration. In the latter, the object transfer takes place. Later, in a fourth phase, the two agents retract.

w.r.t. the human perspective. The aerial robot must rely solely on its onboard perceptive sensors (e.g. a camera) to detect the human presence in the surrounding environment and estimate the partner body pose. Additionally, the following set of task requirements are assigned to the robot while performing the handover:

1. The visibility of the operator has to be maintained at all time. Indeed, if lost, the human pose estimation quality will deteriorate and it may in turn jeopardize the human safety, by possibly leading to a collision between the two agents.
2. The object should be handed over in the most ergonomic configuration from the human standpoint. Thus, that location shall prevent the user from attaining body configurations which may result uncomfortable and unnatural during the object transfer.
3. Yet, such a location shall be re-adjusted according to the human motion. In this regard, the robot controller can exploit some visual clues in order to properly move its EE towards the human, to receive or deliver the handed-over object, and modify its position according to the partner movement.
4. The human can apply actions on the robot body or its EE to move the partner to a more convenient position when at close distance.

### 2.1.3 Focus of the thesis

First of all, in our analysis, we do not fix a particular direction for the object transfer: we seek for control algorithms that could be potentially applicable to both the H2R

and the R2H cases. Moreover, we are interested in the physical layer of the considered problem, while the social-cognitive part remains out of the scope of this work: we assume that the object to be transferred is known and agreed beforehand. As for the *when* and the *where*, we assume that the giver is taking these decisions. Hence, during the R2H handover, we task the aerial robot to propose a handover location, while in the opposite case (H2R) the human decides when handing over the tool to the robotic partner and where. This is motivated by the choice of the application we focus on. In high-from-ground applications, the human workers operate in a limited workspace which can constrain their movement ability. In this context, the aerial robot (AR) shall approach and reach the human thanks to its maneuvering capabilities. Then, in a R2H handover, the robot carries the object and it must deliver it in a comfortable way to the worker. Instead, in the H2R case, once the AR is at close distance, it shall wait for the human to initiate the interaction and pass the tool. As being in charge of the *where*, the human giver can possibly need to move the robot to a more convenient location prior expressing the intention of performing the exchange. For instance, the user can apply a force on the flying receiver, e.g. by pulling and pushing its EE, to change its postural and spatial configurations.

As we aim at proposing motion control algorithms for a safe and effective HARH, we consider both the prehandover and the physical parts of the problem. Thus, we investigate only the approach, reach and transfer phases, while leaving the retraction out from our analysis, as the former three phases constitute the core for a successful handover process. To further simplify this complex and challenging problem, we do not address the selection of a proper grasping pose in the H2R handover and, in the R2H case, the issue of delivering the object by offering a comfortable object grasp for the receiver. In a similar way, we leave for future work the problem of modulating the gripping force during the transfer phase, which can be exploited to obtain seamless and natural handovers [Ortenzi, 2021].

The objective of the current work is to pave the way to successful Human-Aerial Robot handovers, and enhance ARs with reasoning capabilities on the human state while coping with the physical interaction arising during this process.

### 2.1.4 Assumptions

Based on the previous problem statement and objectives, we can make a summary of all the assumptions made on the investigated problem.

1. The robot can estimate its pose by means of a proper estimation pipeline which integrates the measures available in the robot architecture. Therefore, in the present work, we do not focus on the problem of guaranteeing a robust pose estimation. Consequently, a *sufficiently accurate* pose estimation of the robot state is always available to the motion controller.
2. The robot uses its onboard perceptive sensors to localize the operator in the environment and to estimate the pose of the partner. This estimation algorithm is assumed to detect sufficient features on the human body to provide a reliable pose estimation as soon as the human trunk is visible. Accordingly, we neglect the problem of selecting the number and the type of features on the human body, and the task of maintaining them above a minimum amount inside the

sensor Field of View (FoV).

3. We neglect the problem of visual occlusion as no other agents, except the giver and receiver, act in the considered environment. If the camera is pointed towards the human, we assume that the operator is visible, the pose estimation is successfully carried out, and no other entity can obstruct the scene.
4. The human worker is *cooperative*, in the sense that it is not acting in an unexpected way and does not intentionally and deliberately perform actions aimed at provoking the instability of the platform.
5. As a result, the exchanged forces and torques can be assumed to be varying slowly over time.
6. The geometrical and inertial properties of the exchanged objects are known. In many structured environments, such as industrial working places, the robot is interacting with entities whose properties are known beforehand.

## 2.2 Research challenges

The human-aerial robot handover is an interesting problem which encompasses several research areas while offering various challenges from the control standpoint. Hereby, we detail those among them that we intend to address.

### 2.2.1 Physical interaction

The main research area covered by the HARH problem is the *physical interaction*, as it constitutes an essential part of the HAR collaboration. During a human-robot handover, the robot has to exchange forces and torques first with the environment, for instance, when picking up the tool requested by the human worker. Secondly, during the transfer phase, the two agents are in contact through the passed item, which results in a mutual exchange of an interaction wrench at the contact location. On the one hand, the user might convey personal preferences to the robotic agent by using intentionally physical clues. For instance, the worker could push or pull the robot body or its EE to communicate the partner to move to a more convenient location or to change configuration. On the other hand, the user might accidentally enter in contact with the aerial robot resulting in the application of an unwanted wrench on its body. Therefore, the robot motion controller shall take into account this sensorial feedback and adapt accordingly the vehicle state. In the former case, embedding a certain degree of compliance within the planning and control framework can improve the physical interaction between the two agents, resulting in a better and safer user experience [Afifi, 2022]. In the latter case, i.e. in the presence of undesired interactions, the robotic co-worker must reject the additional forces and torques that arise during the contact phase.

All these actions can perturb the platform stability, e.g. by overly tilting the robot, and hence they must be properly treated. This is not a trivial problem since ARs do not own a fixed base which can sustain their weight and they do not feature brakes allowing an emergency stop, as in the case of ground and wheeled manipulators. Besides, most of the actuation power is used to cancel out gravity, leaving a small

portion to be employed for the physical interaction task and eventually counteract external actions which could destabilize the platform. For the aforementioned reasons, the control algorithms must be capable to compensate for the external wrenches applied on the platform. At the same time, during the contact phase, the controller should also exhibit a desired dynamics in response to user-predefined and intentional interactions, while rejecting undesired ones.

## 2.2.2 Robot actuation properties

As discussed in Chapter 1, the capability of exerting a wrench on the environment is strictly intertwined with the actuation capabilities of the aerial robot. To successfully perform physical interaction and reject external disturbances, we saw that the ability to exert a 6D wrench is fundamental [Ryll, 2017]. Therefore, fully-actuated multi-rotor aerial vehicles (MRAVs) are suitable candidates for physically interacting with humans, as the controller can exploit their full actuation in order to apply any necessary combination of forces and torques. Conversely, under actuation makes the robot dynamics difficult to stabilize when external actions are applied on the system, which could jeopardize the safety of the potential users involved in the collaboration.

Aerial manipulators (AMs), which feature the same aerial base and a poly-articulated robotic arm, can use their larger dexterity and the additional Degrees of Freedom (DoFs) to improve the quality of the interaction when collaborating with humans. As an example, they can employ their redundancy<sup>1</sup> to optimize the human comfort, or again to move the EE closer to the worker. As a result, the robot can prevent exposing the human to potential risks, such as its rotating propellers and sharp body geometries which could endanger the user. As we discussed in Chapter 1, the motion of the attached arm induces disturbances in the flight dynamics, thus introducing an additional challenge from the control point of view.

The actuators of aerial robots generate bounded forces and torques due to the limited on-board energy and power. Besides, the thrusters cannot reach the requested amount instantaneously, as the actuators undergo a dynamical transient prior to reach the reference commands. When interacting with the environment or humans, a wrench arises either at the robot airframe or at its EE which, as discussed earlier, can affect the robot dynamics. Due to the leverage created by the kinematic chain connecting the EE to the base, the torques acting at the arm tip location can produce a considerably large effect on the robot body. In the effort of compensating for these actions, the controller might saturate the motor commands, which in turn may lead to the platform instability. Loosing stability is clearly unacceptable in Human-Robot Interaction (HRI) settings and it must be prevented, since it would result in the task failure and a threat to the human worker.

Hence, the robot controller must satisfy the following requirements:

1. Adapt to the different robot *actuation capabilities* and exploit them at best for achieving the task. In the case of fully-actuated ARs, the controller shall

---

<sup>1</sup>Usually, in a handover setting, a gripper is mounted at the robot EE location to grasp and then release objects. Therefore, the robot shall be able to control the full 6D pose (position and orientation) of its end effector. Consequently, the robot degree of redundancy is evaluated w.r.t. this task.

be able to generate an arbitrary 6D wrench. If the robot is featuring an additional robotic arm, it shall compensate for the dynamic coupling effects while stabilizing the whole platform.

2. At the same time, the generated motor commands shall respect the *limitations* of the low-level actuation hardware.

### 2.2.3 Perception

The aerial robot must perceive the external world in order to safely and physically interact with it and the other agent(s) sharing the same workspace. Therefore, *perception* plays a crucial role. For this reason, the robot is equipped with onboard visual sensors allowing it to perceive the human worker acting in the surrounding environment. Specifically, a perception pipeline is running on-board the robot and it provides the control architecture with a body pose estimate of the partner. To properly operate, these algorithms usually require the measurements of a set of features belonging to the observed entity [Sarafianos, 2016]. If no features are visible after the execution of the requested motion, then the estimation will diverge, no matter of the algorithm robustness [Jacquet, 2022a]. As we assume that the necessary features are located on the human body, this translates into the requirement of maintaining *visibility* of the human trunk at all time during the handover. This task is of the greatest importance, as neglecting it can lead to unacceptable collisions and potentially injure the operator.

The estimation of the human pose is a crucial information not only from the safety point of view but also for the successful task execution. Indeed, during the approach and reach phases, and especially in the case where the human cannot move towards the robotic agent, the latter is supposed to close the distance. In this situation, the AR shall move its body and particularly its end effector towards the human, either to give or receive the handed-over object. The robot architecture can exploit some visual clues that are already available on-board in order to control the EE pose towards the user and modify its position according to the partner movements.

To conclude, the robot must observe the environment and the human, reasoning on the perceived context, and perform accordingly its actions. This implies that the control architecture must not only visually drive the robot towards the perceived goal and user, but also account for the visibility requirements necessary for a robust perception.

### 2.2.4 Safety

When designing motion controllers for close and physical human-robot interaction, it is of paramount importance to account for the user safety [Santis, 2008; Haddadin, 2017; Zacharaki, 2020]. The controller shall indeed guarantee the avoidance and the prevention of any collision which may injure and endanger the human worker. Therefore, it is essential to embed the notion of *safety* in the control architecture, by letting the controller be aware of the operator motion and the relative distance. For instance, the robot can forbid reducing excessively the gap to the human in accordance to some minimal safety boundaries beyond which a collision is hardly

unavoidable [Kulić, 2005]. Besides, other metrics can be defined to account for safety. For example, it is possible to limit the amount of relative velocity with the aim of reducing the amount of energy transferred in case of a potential collision. Alternatively, one can prevent the robot from attaining dangerous human-relative positions, e.g., flying above or behind the human partner.

### 2.2.5 Ergonomics

During the handover task, the human should not undertake prolonged articular stress which may cause discomfort and injuries in the long run [Lorenzini, 2023]. Therefore, the control architecture should include a metric or index to quantify the current *comfort level* the human is experiencing in a given postural configuration. In the case of a R2H handover, the computed location for the transfer phase must appear natural and ergonomic to the operator while being compatible with the user current activity and motion. Similar considerations can apply also in the inverse situation where the robot shall receive the object from the human (H2R handover).

## 2.3 Problem decomposition

In this section, we present how we decompose the Human-Aerial robot handover problem into two complementary subproblems to facilitate its tractability.

The first subproblem comprises the development of a visual and physical interaction control architecture enabling the collaboration between aerial robots and humans. This choice is motivated by the fact that the physical interaction and the perception constitute two fundamental parts of the problem under analysis. Indeed, during the contact phase, the human and the robot interactions can have a significant impact on the platform stability and on successfully achieving a safe handover. Instead, the robot perception capabilities shall be used to control the EE pose with the aim of moving it towards a goal location (e.g., close to the human), and potentially modifying its final position according to the changes in the environment. To further simplify this first subproblem, we divide it into two milestones.

We start by designing a control architecture that can endow aerial robots with the capability of applying forces and torques on the environment. Hence, in our study, we adopt the “Flying End-Effector” paradigm, and we use fully-actuated ARs as they can exert an arbitrary 6D wrench. Therefore, at the control level, we will exploit the full actuation property to perform physical interaction tasks. Additionally, we explore the inclusion of vision in the control framework. Specifically, the onboard perception system will control the end effector pose and steer it towards the desired location.

After reaching this first milestone, we extend it to the case of ARs physically interacting with humans. Here, we equip the robot with a robotic arm, thus obtaining an AM whose larger dexterity and redundancy can be used to achieve a better-quality and safer interaction with a human worker. Once the robot shares the user workspace, we assume that the reference trajectory for the robot body and EE is produced by a more sophisticated planner accounting for the human state, as the one presented

in [Truc, 2022], or known a priori. The robot controller tracks this reference motion driving the platform close to the human agent. At this point, we expect that the user can move the robotic partner towards a more convenient position through a physical clue, e.g., a force applied on the robot end effector. For this reason, we will not make use of the previous vision system to control the EE pose, as the human decides its final location. Therefore, we focus essentially on extending the physical interaction part of the previous control architecture to (i) a more complex robot dynamics and (ii) to the *physical human-aerial* collaborative case. Thus, with the second milestone, we aim at realizing physical Human-Aerial Robot Interaction (pHARI). It is clear at this point that, in this first piece of work, the focus will be entirely on the transfer (physical) phase shown in Figure 2.2.

The second subproblem comprises the inclusion of safety and ergonomic metrics within the control framework, while accounting for the actuation limitations and the visibility constraints required by the task. In the previous subproblem, the perception is used to control the robot motion by driving the robotic platform towards the goal. There, the control algorithm may not respect the constraints of the actuators, as well as it may jeopardize the human detection. Since it does not account for the robot dynamics, it may induce robot motions that cause the observed features to move away from the onboard sensor FoV. Accordingly, in that case, we assume that the robot can always perceive the entities of interest. In the second subproblem, instead, we drop such a hypothesis and we focus on the requirement of guaranteeing constant observability of the human during the whole handover. As stated in Section 2.1.4, this assures in turn to have always enough features observable by the robot camera and a reliable human-body pose estimation. As under actuation makes the perception task even more challenging due to the coupling with the robot dynamics [Jacquet, 2021], we consider the use of an under-actuated MRAV to perform the handover task. Lastly, differently from the first subproblem, here we focus on the approach and reach phases reported in Figure 2.2, and we stop the analysis of the handover process just before the start of the interaction phase.

As evident at this stage, we obtain two problems that are complementary which can be summarized in the form of two main questions, as follows.

1. How we can visually control aerial robots towards the desired goal and enable them with physical interaction capabilities in order to achieve pHARI?
2. How we can let an aerial robot be “aware” of the human during the handover process, i.e. (i) accounting for safety, (ii) maximizing the human ergonomics, (iii) satisfying the perception constraints and (iv) considering the robot actuation limitations?

In Chapter 3, we introduce the reader to the replies addressing those two questions, and we present an overview of the designed control architectures tackling the corresponding subproblems. Table 2.1 visually summarizes this problem decomposition.



		Physical interaction	Perception		Actuation		Safety	Ergonomics
			C.EE.P.	Visib.	6D	Limit.		
SP1	Section 3.1.1 Chapter 5	✓	✓	✗	✓	✗	✗	✗
SP2	Section 3.1.2 Chapter 6	✗	✗	✓	✗	✓	✓	✓

Table 2.1: This table shows the decomposition of the HARH problem into two complementary subproblems (SP1 and SP2) based on the research challenges we discussed in this chapter. It shows the sections of the next chapter providing an introduction to the methodology addressing each subproblem, and the chapters with a more in-depth treatment. The symbols ✓ and ✗ denote, respectively, the analysis or the disregard of a certain aspect. “C.EE.P.” and “Visib.” denote the use of perception to control the robot EE pose and the problem of maintaining visibility of the human partner, respectively. Likewise, “6D” is the exploitation of fully-actuated aerial robots, while “Limit.” is the consideration of the actuation limitations.

## 2.4 Overview on the state of the art

After describing the problem at hand, outlining its associated challenges, and our approach to investigate it, we offer an overview of the relevant methods found in the literature. This overview serves as a motivation for the methodologies we introduce in the subsequent chapter. It is important to note that this thesis does not aim to provide an exhaustive review of all existing methods and published articles, but rather a summary of the main ideas behind the primary approaches. Given our focus on MRAVs, we mostly direct our attention to the application of these methods to this particular class of aerial robots. Furthermore, our analysis of the state of the art aligns with the aforementioned challenges that are encompassed by the HARH problem.

### 2.4.1 Control for physical interaction

The problem of controlling the robot position and, at the same time, the interaction force at the contact location, while preserving the system stability, has already been addressed by the research community both in the fields of HRI and physical Aerial Robot Interaction (pARI) [Santis, 2008; Ollero, 2021]. Typically, robots are position-controlled. However, managing the interaction of a robot with the environment by adopting a purely motion control strategy turns out to be inadequate [Santis, 2008]. In the case of unstructured environments, inaccurate planned tasks and unforeseen contacts with the environment or other agents, a pure motion control may cause the rise of undesired interaction forces. High values of these forces do not only deteriorate the quality of the human-robot collaboration, but also they may excessively stress the robot and the manipulated objects [Siciliano, 2009].

A very common approach adopted by roboticists is to employ an *impedance* (or *admittance*) *control* scheme, which comprises reshaping the mechanical robot response to match a virtual dynamic behavior, e.g. as a mass-damper-spring system [Dimeas, 2016]. This usually occurs by establishing the desired dynamic relationship between

the robot EE coordinates and the interaction forces applied at that location. In particular, in the impedance control, the system input is the external wrench applied at the robot EE, while the output its displacement [Ollero, 2021]. The admittance control is the dual of the previous scheme: the input is the robot EE displacement and the output the interaction forces. These schemes belong to the category of *indirect force control methods* as the interaction wrench is controlled by indirectly regulating the robot motion.

For what concern the domain of aerial robotics, the admittance (or impedance) control approach found various applications both for under-actuated and fully-actuated aerial manipulators [Ollero, 2021]. Ruggiero et al. [Ruggiero, 2014] have applied it to a collinear quad-rotor in combination with a momentum-based wrench observer to let the robot withstand external disturbances while performing simple maneuvering task, such as hovering in place or following a pre-defined trajectory. Later, in [Ryll, 2019], this control paradigm has been applied to fully-actuated MRVs in order to achieve tasks requiring physical interaction, namely touching and sliding over a surface, or inserting the tip of the robot EE in a narrow space for inspection operations.

In [Augugliaro, 2013; Rajappa, 2017], it has been exploited to achieve pHARI. Specifically, in [Augugliaro, 2013], the authors employed an admittance scheme to let the aerial robot be compliant to the external actions which could be potentially be applied by a user interacting with the platform. Similarly, in [Rajappa, 2017], the admittance control has been combined with a wrench observer and contact-based sensors to design a control architecture allowing the AR to physically interact with a human operator. By exploiting model and sensor-based estimates of the contact wrench, the platform can discriminate between human and environment actions, and it enables the robot to withstand possible external disturbances, e.g., wind or collisions with obstacles. More recently, in [Afifi, 2022], an admittance-based control architecture has been used in a simulated environment to let a fully-actuated hexa-rotor equipped with a poly-articulated robotic arm physically interact with an operator. Particularly, the authors show that the parameters of the admittance control loop can be tuned in such a way to provide comfortable and safe robot responses during the human-aerial robot collaborative task. Instead, if the robot controller is let directly respond to the human inputs, stiff and abrupt behaviors may emerge which could jeopardize the user experience. Hence, by adopting this control strategy, the potential human operator can apply external actions on the robot without destabilizing its dynamics.

Nevertheless, also *direct force control methods* exist, where the interaction force is precisely controlled by an explicit closure of a force feedback loop [Siciliano, 2009]. In aerial robotics, this latter control approach originated schemes featuring both motion and force control, which in turn have them granted the denomination of *hybrid position/force control methods*. A first example is the work in [Nguyen, 2013], where the authors propose a hybrid force/motion control for a quad-rotor equipped with a rigid tool. The proposed control methodology allows controlling the under-actuated robot dynamics in tasks requiring the application of forces over a surface and the precise control of the EE tip position. In [Nava, 2020], an aerial manipulator similar to the one in [Afifi, 2022] has been equipped with a force-torque sensor to perform push-and-slide tasks. Thanks to the availability of the force measurements

on-board the platform, the authors could close a direct force control loop. Therefore, they designed a control architecture where an optimization-based reactive controller can generate the low-level motor commands necessary for the robot to achieve the contact-based task, e.g., push and slide over a surface. Similarly, in [Peric, 2021], another optimization-based control algorithm has been used to achieve force and motion control for inspection tasks on flat surfaces by means of an over-actuated aerial robot.

In order to implement any force control method, it is necessary to estimate the robot state and the contact wrench [Ollero, 2021]. For the latter, among the different possibilities, model-based wrench observers are widely adopted, since they can exploit the robot dynamic model and the onboard proprioceptive (inertial) sensors [Tomić, 2017; Peringal, 2022].

On the one hand, tasks like contact inspection, sensor installation and object manipulation requires an accurate control over the EE pose and of the interaction forces. This can be achieved by employing the direct force control methods previously described. However, the latter category require precise measurements of the interaction forces, which could be hard to obtain with model-based observer. Modeling mismatches, unpredicted effects (e.g., aerodynamic drag and inter-propeller interactions), combined with unknown parameters and external actions can be detrimental for methods that require an accurate knowledge of the robot model to estimate the external forces applied on its body. A possible solution can be the integration of force-torque sensors on-board the robotic system, but this type of hardware is generally expensive. Moreover, their inclusion contributes to increase the robot mass and, in turn, decrease the available payload and flight time.

On the other hand, applications involving human-robot interaction do not require the fulfillment of an accurate regulation of the exchanged forces and torques. In most cases, the adoption of an indirect force control method has appeared to be much easier and intuitive from the implementation standpoint. This is particularly relevant if we consider that most of the commercial Unmanned Aerial Vehicles (UAVs) feature low-level position or velocity-based controllers, which makes the closure of a force feedback loop unpractical or even impossible. Therefore, the application of model-based observer is the typical go-to approach in these cases. Furthermore, an impedance or admittance control scheme can handle the transition from contact-less to contact-based flight operations by means of a smooth variation of the gains [Ollero, 2021].

## 2.4.2 Robot actuation properties and related control architectures

When it comes to control an aerial robot, its actuation properties and the presence of an attached manipulator determine the nature of the control architecture.

*Feedback linearization* constitutes a common approach for controlling nonlinear systems, such as ARs. This technique consists in the exploitation of the system dynamic model and its inversion to cancel out the nonlinearities and obtain in turn a linear system [Luca, 1998; Oriolo, 2002; Isidori, 2013]. Hence, the resulting dynamics is much easier to control and classical control methods from the domain of linear control theory can be applied, for instance, Proportional-Integral-Derivative (PID)

controllers [Bicego, 2019].

A fully-actuated system can be directly feedback linearized as it is possible to control independently its position and attitude [Ryll, 2019]. In [Franchi, 2018], an inverse dynamics approach has been proposed to track an arbitrary 6D (position and orientation) trajectory with fully-actuated aerial vehicles, while considering virtual bounds on the amount of lateral forces they can produce. Later, the authors in [Ryll, 2019] utilized the same control approach developed by Franchi et al. to perform tasks requiring the application of a 6D wrench (forces and torques) on the environment.

The application of feedback linearization to a Uni-directional Thrust (UDT) AR is not directly possible, as an input transformation is first necessary which considers the total thrust and moment as the new system inputs [Ollero, 2021]. By adopting this change of coordinates the system can be feedback linearized, and the resulting system is proven to be differentially-flat [Rajappa, 2015]. Differential flatness allows writing the dynamic model of a nonlinear system by means of algebraic relationships between a particular set of coordinates (the so-called *flat outputs*) and their derivatives. For a collinear quad-rotor the flat outputs are proven to be the position of its center of mass (CoM) and the heading (yaw) angle [Spica, 2012]. Fully-actuated aerial vehicles are intrinsically differentially-flat systems. Differential flatness is particularly attracting as it allows deriving control laws that compute the inputs of under-actuated ARs such that any trajectory in Cartesian space, with a desired yaw angle, can be tracked [Ruggiero, 2018].

For differential flatness-based approaches, identifying the appropriate flat output is crucial and not always feasible. There are cases where the flat outputs are successfully identified and they allow simplifying the control problem, such as quad-rotors and UDT ARs with attached arms [Welde, 2021]. However, there are also cases where differential flatness is less applicable, like controlling the tool-tip EE [Nguyen, 2013], or requiring additional design constraints, e.g., having the arm attached to the center of mass [Yüksel, 2016]. Therefore, due to these practical issues, the use of differential-flat-based controllers is limited and, in most cases, unfeasible.

Alternatively, to address the under actuation of many MRAVs, *hierarchical control architectures* are employed, which exploit the time-scale separation between the translational and rotational dynamics [Mahony, 2004]. This allows formulating a control architecture where a slower outer positional control loop is in charge of computing the orientation reference signals for the faster inner loop stabilizing the attitude dynamics. Thanks to the easiness of integrating several other modules within the same framework, and the possibility of controlling both fully-actuated and under-actuated aerial vehicles, hierarchical control architectures are very adopted in the field of aerial robotics [Ruggiero, 2018]. As an example, Ruggiero et al. [Ruggiero, 2014] build a hierarchical control architecture which decouples the translational and rotation dynamics of a Vertical Take-Off and Landing (VTOL) UAV, and it shapes its closed-loop mechanical properties according to an impedance control law. In a validation campaign, the AR controlled by means of the proposed framework is shown to perform hovering and tracking tasks without requiring a precise knowledge of the vehicle dynamics and in presence of external disturbances.

If we consider the presence of a poly-articulated robotic arm featuring several DoFs

attached to an aerial robot, the control of its dynamics becomes more challenging. As mentioned earlier, differential flatness is hardly applicable. A possible approach comprises the use of separated controllers for the flying base and the robotic arm. Control architectures of this type are categorized as *decentralized control methods*, and their accuracy relies in the robustness of the position-and-attitude control loops for the AR in compensating the coupling disturbances induced by the attached manipulator [Ruggiero, 2018]. For the flying base the previously discussed control strategies can be employed [Ruggiero, 2018], and the robotic arm features the well-established control architectures developed for industrial robots [Siciliano, 2009]. An example of those control architectures is the work in [Ruggiero, 2015]. There, the controller exploits a wrench observer to estimate the disturbances produced by a servo-driven robot attached to a flying base, and compensate for them when stabilizing the MRAV dynamics.

In opposition to decentralized architectures, *centralized control methods* allow considering the AM (flying base and robotic arm) as a unique entity [Ruggiero, 2018]. Again, feedback linearization constitutes an example of centralized approaches. However, among the other alternatives [Ruggiero, 2018], optimization-based controllers are particularly interesting and worth mentioning. These methods compute the system inputs, and possibly the states, by solving an optimization problem. The research community found them very attracting as they can not only consider the full dynamic model of the AM, but also the platform actuation limitations and other task-related constraints.

Optimization-based approaches can be further differentiated into *reactive* and *predictive* methods, depending on the fact that the control action is computed based, respectively, only on the information available at the current time, or also on the one related to a near future [Bicego, 2019]. An example of reactive controller is the *Quadratic Programming* (QP) regulator formulated in [Nava, 2020], which allows to compute feasible motor commands for a fully-actuated hexa-rotor featuring a 3-DoFs robotic arm in physical contact with the environment. Instead, in the class of predictive controllers, *Model Predictive Control* (MPC) is a promising and powerful approach that is finding large adoption in many robotic systems [Bicego, 2020].

MPC consists in the formulation of a *constrained Optimal Control Problem* (OCP), which is composed of a cost function to be optimized and a set of constraints to be satisfied, and whose solution generates a stabilizing sequence of system inputs [Raković, 2018]. In particular, MPC exploits the knowledge of the dynamic model to anticipate the future behavior of the system over a finite time horizon, and to compute proper stabilizing optimal inputs. Thanks to the recent advances in the solution of nonlinear OCPs [Moritz, 2001], it can be applied to the control of nonlinear dynamic systems in real-time, and it can naturally include the task constraints and the bounds of the robot actuators.

For its predictive nature and the possibility to include constraints, it has been widely adopted in the field of robotics, and particularly for controlling aerial robots. In [Darivianakis, 2014], a hybrid MPC is employed for accurate trajectory tracking and force control of a quad-rotor for contact-based inspection applications. Similarly, in [Kocer, 2018], a similar optimization-based technique is used for ARs in close proximity to ceilings again to perform surface inspection. Moreover, Hofer et al. [Hofer, 2016] show that it is possible to use a similar predictive controller to stabilize a small

rotorcraft featuring limited onboard computational resources. In [Bicego, 2020], a nonlinear MPC-based controller has been derived for controlling both under and fully-actuated MRAVs in real-time, while taking into consideration also the real actuation dynamics of the considered platform. Lastly, thanks to the possibility to include any system model as part of the optimization process, it has been used to control much more complex aerial systems. In this direction, in [Martí-Saumell, 2021], the authors provide simulations to validate their Nonlinear Model Predictive Control (NMPC) formulation capable of controlling the entire coupled dynamics of a poly-articulated AM, both in free flight and in physical interaction with the environment.

### 2.4.3 Perception-based control

As we mentioned in Section 2.2.3, during the HARH, the aerial robot shall drive the EE towards the goal location and maintain visibility over the other agent for successfully perform the collaborative task. Hereby, we present two vision-based control techniques available in the state of the art that allows to obtain these two objectives. For a review of the methods addressing the problem of detecting the pose of the object in the image acquired by an onboard camera, we refer the reader to [Jacquet, 2022b].

Once the knowledge of the environment is retrieved from the sensors, a *vision-based control* can be formulated to generate the motion toward the fulfillment of the task [Hutchinson, 1996]. One of the most classical and well-known method is *Visual Servoing* (VS), which consists in a control law generating motion commands (usually at the velocity level) that can nullify a vision-defined error. Depending on the definition of this error metric, VS can be divided into two main classes [Chaumette, 2006]: *Image-Based Visual Servoing* (IBVS) and *Position-based Visual Servoing* (PBVS).

IBVS defines the vision-based error in the image plane of the camera. Practically speaking, this technique tries to position the camera w.r.t. a reference image. Conversely, PBVS estimates the 3D pose of the features in the Cartesian camera frame and minimizes the tracking error w.r.t. a reference position. Due to the kinematic relationships between the visual clues in the image plane and the motion of the camera, PBVS is usually considered harder to apply than IBVS. Moreover, as PBVS uses a pose error, it requires the measure of the depth information, which for a monocular camera is challenging and subject to uncertainty. Furthermore, PBVS requires an accurate extrinsic camera calibration to nullify precisely the error, while IBVS is more robust to such imprecision since it directly controls the motion of the features in the image plane [Jacquet, 2022b]. Nevertheless, PBVS produces camera motions that are more natural and legible compared to IBVS, since the Cartesian motion of the camera is directly controlled by the former method (PBVS).

An intermediate solution is offered by *Hybrid Visual Servoing* (HVS) schemes [Malis, 1999; Conticelli, 1999], which use features both from the image space and the Cartesian space. The latter type of approaches addresses the aforementioned drawbacks of classical techniques: hence HVS is less sensitive to calibration uncertainties than PBVS, and it is more robust to singularities than IBVS. In addition, HVS relies only on partial pose estimates of the tracked visual clues.

For their general practicality, VS schemes have found large adoption not only in industrial manipulators [Kragic, 2002], but also for aerial robots and aerial manipulators to perform a variety of tasks [Santamaria-Navarro, 2019]. For instance, in [Zheng, 2018], the authors utilize an IBVS-based motion planning algorithm to compute smooth trajectories in the camera image plane letting a collinear quad-rotor to move from an initial to a desired location. The method is validated both in simulations and real-world experiments. Similarly, in [Keipour, 2022], an IBVS scheme is used to make a collinear quad-rotor land on moving targets. However, in the last two works, the considered robot does not feature any EE and it does not perform any physical interaction task. Later, in [Lippiello, 2018], it is shown how to employ a VS technique to drive the EE towards the goal. In particular, Lippiello et al. propose a control scheme combining IBVS and impedance control for an aerial manipulator comprising two robotic arms, one equipped with a camera and a second one with a force-torque sensor. The first arm is used to observe the scene and help the second one during the execution of manipulation tasks. Starting from the collected features, a vision-based control module generates the velocity commands to be fed to an inner motion control loop, which in turn produces the system control inputs. The latter signals, being the robot accelerations, are then converted to motor commands by means of a feedback linearization approach. The proposed method is validated in simulations showing the system capabilities to visually drive the AM EE into narrow gaps and achieve consequently complex peg-in-hole tasks. Very recently, He et al. [He, 2023] adopt another IBVS scheme to control the EE of a fully-actuated hexa-rotor in contact-based applications, namely bridge painting. This method, combined with a hybrid motion and impedance force controller, is shown to let the robot drive the EE to the desired location and apply a constant force on the environment.

The motion control commands generated by the approaches presented so far do not guarantee to maintain the tracked visual features always within the FoV of the onboard sensors. Under particular conditions, a VS scheme may generate commands violating this objective [Zheng, 2018]. For this reason, despite being a good candidate for visually driving the robot EE, it may lose visibility of the human partner during the handover.

In aerial robotics, we have seen that MPC-based controllers have found particular attraction mainly for the possibility to include robot and task constraints. This originated a specific branch denominated *vision-based* or *perception-aware* MPC schemes. This variant of MPC allows including the visibility requirement as part of the system constraints and within the cost function of the OCP. Then, the predictive nature of this approach is used to “look ahead” and check that certain planned actions do not jeopardize the perception-based task. Exemplary applications of such a scheme are the works of Penin et al. [Penin, 2017; Penin, 2018], Falanga et al. [Falanga, 2018], and Jacquet et al. [Jacquet, 2021]. In [Penin, 2017], a NMPC-based motion planning algorithm is formulated that allows generating minimum-time trajectories for a quad-rotor equipped with a monocular camera and tasked to maintain visibility over a set of visual features in the environment. Later, in [Penin, 2018], the same authors show that the same method can additionally include collision and occlusion constraints. Falanga et al. [Falanga, 2018] present a unified control and planning perception-based NMPC including perception objectives for quad-rotors. Lastly,

in [Jacquet, 2021], a perception-constrained NMPC is derived for generic designs of MRAVs equipped with limited-FoV visual sensors. The proposed method allows controlling in real-time the robot dynamics by generating feasible motor commands, and satisfying the perception-based requirements while tracking a given motion reference.

For a more detailed literature review in the field of perception-aware NMPC, we refer the reader to the work of Jacquet [Jacquet, 2022b].

#### 2.4.4 Safety

When addressing the second subproblem, the objective is the formulation of a “human-aware” controller. The desired control law should consider both the human safety and the user ergonomics when generating the motor commands enabling the aerial robot to achieve a safe and comfortable handover. Hereby, we delve with the problem of including human safety at the control level, while the user ergonomics is later discussed in Section 2.4.5.

In the literature, safety can have different connotations. As a matter of fact, it can be related to the subjective feeling of security perceived by the user during the collaboration [Akalin, 2022], it can be expressed as function of the amount of energy transferred from the robot to the human (or vice versa), or it can concern the potential collisions between the two agents. For this reason, safety is usually distinguished into *perceived* and *physical* safety.

In the former case, perceived safety requires a multi-disciplinary perspective and comprises several factors, such as comfort, predictability, sense of control, and trust [Akalin, 2022]. Henceforth, as it is typically hard to numerically quantify these aspects by means of metrics, perceived safety is usually evaluated through user studies involving several participants, whose personal feedback and opinions are collected to validate the robotic system under analysis. Conversely, physical safety is related to the occurrence of a contact between the two agents and, in such a case, to the amount of exchanged energy and power [Santis, 2008]. Being a physics-based concept, the latter type (physical safety) can be quantified. For instance, in [Haddadin, 2007; Haddadin, 2009], a series of crash tests are performed involving an industrial robotic manipulator which is made impacting several parts of a human mannequin. The aim of these experiments is the quantification of the potential injury risks emanating from the manipulator during an unexpected collision with a possible user. Although Haddadin et al. focused only on certain types of impacts and EE geometries (frontal collisions with non-sharp objects), valuable insights have been drawn. In particular, they have shown that, for different robot masses, the larger is the robot EE velocity, the higher is the injury risk expected after the impact. By means of the collected data, they have been able to derive metrics and obtain information which could be exploited in the design of planning and control algorithms. In this way, it would be possible to generate robot motions not leading to high-speed profiles in close proximity to sensitive human body parts, and that minimize the risk of severe injuries in case of unavoidable human-robot collisions [Haddadin, 2017; Zacharaki, 2020].

As we aim at integrating the notion of safety within a control law, our attention is focused only on its physical side, as mathematical tools can be derived for its



evaluation and quantification. To achieve collision-free motions, and possibly limit the maximum transferable energy in case of an unexpected collision, a suitable motion planner could be derived which includes part of those insights and metrics. Along this line, one of the earliest works in this area is [Sisbot, 2012], which develops a manipulation planner for an assistive robot that takes into account safety among other human aspects, such as ergonomics and field of view, to name a few. For what concerns the safety metrics, Sisbot et al. propose to use a simple function based on the relative human-robot distance, which decreases as the robot is farther away from the user. Similarly, we can mention the works of Truc et al. [Truc, 2022; Truc, 2023], where a sampling-based motion planner is derived to compute safe and comfortable trajectories letting an aerial manipulator approach and reach a human operator in a R2H handover scenario. Also in this case, the safety metric is based on the human-robot relative distance and velocity. Hence, the safety cost increases not only if the relative distance decreases, but also if the velocity vector is oriented towards the human, i.e. the robot is approaching the user. Therefore, in very proximity of the human operator, the robot should perform motions at slower speeds which are in turn related to smaller risks in the possibility of unexpected collisions.

The generation of a collision-free motion plan alone does not guarantee eliminating any safety risk. Indeed, if the trajectory is not accurately tracked by the low-level controller, a collision might anyway occur. In a similar way, if the plan is not updated sufficiently fast on-line, a sudden motion of the operator or a change in the environment might jeopardize the security of the human partner involved in the collaboration [Li, 2021]. Therefore, it is important to re-plan locally the global trajectory, while accounting for the changes in the surroundings.

As seen so far, optimization-based planning and control algorithms, mainly based on MPC, are becoming quite popular when generating robot trajectories and actuator commands to avoid collisions with dynamic entities in the environment. In the field of aerial robotics, among all the existing articles, we can mention the works in [Kamel, 2017; Castillo-Lopez, 2018; Dmytruk, 2022], where several NMPC-based algorithms are designed to avoid collisions with static and/or dynamic obstacles in the environment. The main idea pivots around the inclusion of either an expression representing the obstacle-collision avoidance requirement within the set of constraints to be satisfied, or a term to be minimized within the cost function. Henceforth, in [Dmytruk, 2022], the authors introduce a constraint in the OCP bounding the Euclidean norm of the relative distance between a moving obstacle in the environment and the robot. This allows the MRV to track an assigned trajectory while avoiding the moving entities and achieving secondary perception-based tasks. Similarly, in [Castillo-Lopez, 2018], the authors exploit the introduction of an additional variable in the optimization process to relax the collision-avoidance constraints. In this way, the NMPC-based controller can soften the constraints to facilitate the resolution of the optimization problem, which is usually computationally hard to solve in real-time. Additionally, in [Kamel, 2017], the authors propose a combination of the previous two ideas. Inside the OCP formulation, they introduce both a collision-avoidance constraint and a term in the cost function which allows two MRVs to avoid impacts between each other when sharing the same workspace as part of a team. Furthermore, these works show that the predictive nature of MPC-based methods can be used to predict the future displacements of the dynamic obstacles

and anticipate potential collisions.

### 2.4.5 Ergonomics

As for human safety, ergonomics comprehends both *physical* and *mental* factors [Lorenzini, 2023]. The former are related to the physical load applied on the human worker body while performing the assigned task, e.g., when manipulating heavy materials, in prolonged manual activities, or in attaining uncomfortable body postures. The effects of this physical burden may provoke the so-called *musculoskeletal disorders*, whose most common examples are backache and upper limb pain. Conversely, the mental factors affecting ergonomics are connected to the mental health problems originated from an uncomfortable and unpleasant workplace. Typical examples are stress and mental fatigue.

In this thesis, our interest is in the quantification and the evaluation of human ergonomics, and the exploitation of this information to produce robot commands, and in turn motions that result in a comfortable handover for the user. In the literature, a way to quantify human ergonomics is using the popular *rapid upper limb assessment* (RULA) [McAtamney, 1993] and *rapid entire body assessment* (REBA) [Hignett, 2000]. Both methods are developed to systematically evaluate the comfort level of human workers in their workplaces, and they are conceived on the basis on the International Organization for Standardization (ISO) normative 11228 and the European Standards (EN) 1005 [Lorenzini, 2023]. Specifically, the former (RULA) consists of a series of diagrams of body postures and scoring tables describing the physical burden that workers undergo during their activities. Similarly, the latter (REBA) provides a scoring system evaluating and ranking muscle activity caused by static, dynamic, rapidly-changing and unstable postures in a variety of tasks.

Due to its simplicity and the possibility to numerically evaluate human ergonomics, these methods have been applied in several planning algorithms to generate robot trajectories leading to comfortable human-robot interactions. An example is constituted by the work in [Busch, 2017], where the authors integrate the REBA in a planning framework to allow a dual-manipulator-arm robot to estimate the human ergonomics and propose more comfortable poses in an assembly-like collaborative task. In [Zanchettin, 2019], a control strategy is presented for an industrial robotic manipulator to minimize the muscular fatigue of a human operator during the manipulation of large and heavy objects. In particular, the controller makes the robot move the work-piece so that the human is always operating close to the most ergonomic posture, which is computed by means of the REBA metric. Later, in [Makrini, 2022], an algorithm is derived that proposes ergonomic human body postures in assembly and human-robot collaborative tasks. Also in this case, the determination of the comfortable joint positions is based on an optimization algorithm which exploits a kinematic human model whose parameters are derived from the REBA guidelines. In a similar fashion, other control and planning algorithms have integrated RULA-based metrics [Shafti, 2019; Ferraguti, 2020].

Although these tabular ergonomic assessment methods are simple and intuitive, their practical application within an optimization algorithm becomes challenging and cumbersome. This difficulty arises because they employ linear step-wise functions rather than differentiable expressions [Nejadasl, 2022]. To overcome this issue, in

recent years, differentiable versions [Yazdani, 2022] or learning-based approximations [Nejadasl, 2022] of RULA and REBA-based metrics have been proposed by the research community. For example, in the work of Busch et al. [Busch, 2018], the authors develop a differentiable polynomial approximation for the REBA metric. This enables them to seamlessly incorporate it into an optimization-based task and motion planning framework, which computes the joint actions and the corresponding motion of the human and robotic agents involved in a collaborative task. With the application of the proposed method, the user is shown to attain more comfortable body postures.

An alternative to employing these differentiable approximations involves modeling the human agent in a similar manner to robotic systems. [Parastegari, 2017]. In particular, as we will also show later in Chapter 4, it is possible to consider the human body as a poly-articulated multi-body system. Henceforth, it is possible to compute the human joint torques and forces necessary to achieve a given task by exploiting the same kinematic and dynamic modeling tools originally derived for robots. This approach facilitates the derivation of quantitative metrics that can be readily incorporated into an optimization framework. Then, these metrics can be used to calculate robot commands and trajectories leading to both comfortable human body postures and a reduction of the user physical burden. [Katayama, 2003].

Along this direction, in the literature, we can find the work of Peternel et al. [Peternel, 2017], where the idea of minimizing the *human overloading joint torques* is sketched. The latter method consists in reducing the effects that an external load is producing on the human body joints (e.g., the back, neck, shoulder and elbow articulations). To do so, Peternel et al. propose to compute the joint torques based on a whole-body model of the human, and to integrate this estimation technique within an optimization algorithm. Based on the computed solution, the work-piece held by the robot is moved to a more comfortable location from the human standpoint, which in turn minimizes the user joint torques.

A similar idea is later exploited in other works [Lorenzini, 2018; Kim, 2018; Kim, 2019]. Specifically Lorenzini et al. [Lorenzini, 2018] proposes an online estimation of the overloading joint torques which, once integrated in an optimization problem, allows generating robot trajectories that minimize the human joint torques. Again, in the work of Kim et al. [Kim, 2018], a real-time technique for reducing the overloading joint torques of the human body is proposed and applied to tasks involving the human-robot co-manipulation of heavy objects. Later, the same authors utilize the same idea to develop a control framework for a multi-human and mobile-robot collaborative team [Kim, 2019]. In particular, the proposed optimization-based methodology enables the mobile-base robot to assist several humans by reducing their articular stress in collaborative tasks.

## 2.5 Conclusion

In this chapter, we described the Human-Aerial robot handover problem, we highlighted the characteristics of the considered aerial case, and we listed the underlying assumptions. Subsequently, we divided the problem into two distinct parts. The first one involves, respectively, a fully-actuated MRAV and an AM equipped with a robotic

arm physically interacting with both the environment and humans. This subproblem also addresses the challenge of controlling the EE to a desired location based on visual clues. The second part of the problem deals with the integration of human safety and ergonomics into the control architecture, leading to the development of a “human-aware” controller. Simultaneously, this method must account for the requirement of maintaining visibility of the human during the handover process, and consider the actuation limitations of the AR. Lastly, we provided an overview of the related works available in the literature.

From the research panorama, particularly relevant are *indirect force control* methods which enable both position and force control of the robot EE. When the AR is equipped with visual sensors (e.g., cameras), *visual servoing* has been shown to be an effective and intuitive vision-based control capable of successfully driving the robot EE towards the desired location. Furthermore, *centralized* and *hierarchical* approaches, featuring controllers based on *feedback linearization* and *optimization-based* algorithms, can be both applied to the control of any MRV, and the flight stabilization of a generic AM with a poly-articulated robotic arm attached to its body. Specifically, among optimization-based methods, MPC is gaining particular attention in the robotics community thanks to the natural integration of constraints and objectives. For these advantageous features, it has been recently employed to consider both actuation and perception-based constraints.

Therefore, in the next chapters, we base on the previous considerations to derive our control methods enabling a physical and safe HARH.



# Chapter 3

## Method overview

The purpose of this chapter is to provide the reader with a succinct overview of the methods proposed in this manuscript, and a summary of the main achievements. We open this chapter in Section 3.1 by outlining the structure of the two control methods developed to tackle the associated parts of the Human-Aerial robot handover (HARH) problem. This serves to familiarize with the main concepts before presenting the detailed formulations in the next chapters, namely Chapters 5 and 6. Afterwards, in Section 3.2, we offer a panorama over the results obtained from the experimental campaigns carried out to validate our methodology. Lastly, in Section 3.3, we synthesize the content covered hereafter.

### 3.1 Methodology

To derive our methods, we take inspiration from (i) the rich and mature literature in physical Human-Robot Interaction (pHRI), mostly involving ground manipulators, and (ii) the emerging and greatly-inspiring efforts made in the field of physical Aerial Robot Interaction (pARI). In this work, we want to combine the methodologies emerging from both research domains to develop novel control methods empowering aerial robots to physically and safely interact with humans, while including perception requirements and accounting for their level of ergonomics.

In this section, we introduce the reader with the main key elements composing the control architectures we propose for addressing the two subproblems defined earlier. Consequently, this section is divided into two parts, each one associated to one portion of the problem considered. The detailed derivations and formulations are provided in Chapters 5 and 6.

#### 3.1.1 Visual and physical control for HAR interaction

According to our problem decomposition discussed in Section 2.3, we aim here at presenting the main concepts of the control architecture designed to tackle the first subproblem. Precisely, we start by presenting a methodology addressing the first milestone, i.e. the development of a control framework for fully-actuated aerial vehicles in vision-based physical interaction tasks. First, the physical engagement is

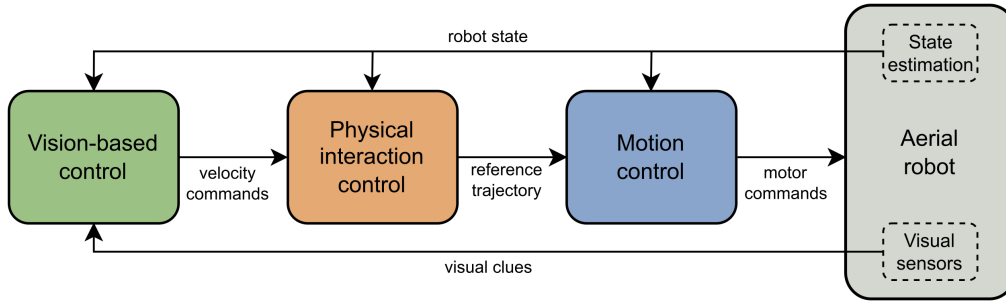


Figure 3.1: Generic control architecture for vision-based physical interaction with fully-actuated platforms. In green the vision-based control, in orange the module in charge of the physical interaction, in blue the geometric controller, and in gray the aerial robotic platform.

limited to static objects in the environment, as we neglect the presence of an other agent. Then, we extend the approach to the case of physical interaction between an aerial manipulator and a human worker, thus accomplishing the second milestone.

### Interaction with the environment

As analyzed in Chapter 2, several control strategies have been proposed in the literature to let aerial robots perceive the surrounding environment and, at the same time, accomplish physical tasks. A common approach is the use of a *hierarchical architecture* which features several independent modules. We opt for this solution not only for its intrinsic modularity and simplicity, which allow tailoring it to the individual needs, but also for the possibility to integrate and compare different control and planning algorithms [Pierri, 2018]. In Figure 3.1, we provide a block diagram scheme of our proposed hierarchical control architecture which comprises three modules.

A *vision-based* control loop generates the commands to control the robot end effector (EE) pose and drive it towards the desired location. If we consider an aerial robot (AR) in an activity requiring physical interaction, the robot tool tip usually needs to reach a set of points located on a surface of an entity in order to achieve the task. When considering a human-aerial robot handover, instead, the EE goal position is a location at a convenient distance to the human to permit the human-to-robot (H2R) or robot-to-human (R2H) object transfer. This module relies on visual clues of the observed entities to control the position of the robot end effector. The vision-based control is taking place at the kinematic level thus it provides a desired velocity as output.

Through numerical integration, a desired trajectory is obtained from the previous module, which is fed to a *physical interaction* layer. The latter (physical) component provides a twofold functionality. First, it allows estimating the presence of an external action, applied on the robot body. Then, if any force and torque is applied on the platform, it shapes the robot behavior during the interaction according to a user-defined virtual mass-damper-spring dynamics. This second functionality is achieved by means of an admittance filter scheme, as depicted in Figure 3.1.

The output of such a module is a new reference trajectory that the *motion controller* must closely track in order to achieve the dynamics imposed by the previous (physical) component. This motion control module generates the low-level motor commands to control the robot actuators given a motion to follow. More details about the internal structure and the mathematical formulation behind each control module will be later provided in Chapter 5.

### Interaction with humans

In the second case, i.e. in collaboration with a human worker, we rely on the same hierarchical control framework presented in Figure 3.1. In this setting, the interaction is intrinsically more challenging than in the case of static objects. Consequently, we assume that a more sophisticated planner is generating the reference trajectory to reach the human which is tracked by the robot controller. The human is moving in the environment and performing other tasks while the robot approaches. Therefore, it becomes necessary to review and re-adapt the original plan according to the user actions. A possible solution could be employing a planner like the one discussed in [Truc, 2022]. However, as the focus is not on designing the planning algorithm, hereafter, we assume that the motion controller receives the reference trajectory from an external module.

Contrary to the previous case, in this second subpart, we consider the use of an aerial manipulator (AM). This type of AR can exploit its arm to move the EE further away from the flying base and closer to the operator increasing the user safety and comfort. As the previous approach is limited to fully-actuated ARs without robotic arms, we employ a different control law which can take into account the larger system dynamics. Precisely, we use a feedback-linearization-based controller which allows controlling a poly-articulated platform and driving both the motor-propeller actuators of the flying base and the joint motors of the attached manipulator. As discussed in Chapter 2, this control method belongs to the category of *centralized control approaches* as it considers the robot structure as a whole. Thus, it can compensate for the coupling effects arising between the two subsystems.

Lastly, we will employ the same physical interaction layer introduced earlier in the first milestone. However, it will make use of the model of the aerial manipulator to estimate the external actions applied on the whole system, while accounting for the disturbances created by the presence of the attached robotic arm.

### 3.1.2 Human-aware control for HAR handover

The control framework derived in Section 3.1.1 is “unaware” of the human presence, since it treats the user as a source of an external disturbance influencing the robot dynamics. As a matter of fact, the control framework does not include the human comfort, and it does not consider either any real safety concern or the current user activity.

The goal of this work is to derive a control strategy that can be “aware” of the human presence, while accounting for other task requirements, thus improving the overall quality of the handover. As a result, in the following part, we address the



second half of the problem mentioned in Chapter 2. Contrary to the first subproblem, hereafter we neglect the physical interaction occurring between the two agents, thus we will focus our attention on the approach and reach phases. The objective is the inclusion of human “awareness” within the control architecture in order to achieve a safer and more ergonomic handover. Lastly, we validate our control framework with an under-actuated robot, as this makes the task more challenging and the proposed method applicable to a broader spectrum of aerial vehicles (even under-actuated).

### Nonlinear model predictive control

We adopt a control technique based on Nonlinear Model Predictive Control (NMPC) to design a “human-aware” controller, which addresses the second part of the Human-Aerial robot handover (HARH) problem.

Based on the discussion in Chapter 2, hereby, we list the considerations that motivate this choice.

1. NMPC can predict the future evolution of the system as soon as a model is available. Hence, we can include a mathematical description of the human within the Optimal Control Problem (OCP) to anticipate the user motion and actions.
2. Since it is based on the formulation of an optimization problem, it is well suited to account for different objectives. Different terms can be included within the cost function as a weighted summation where the weights are proportional to the priority of the considered task. The final solution will constitute a trade-off between the different goals.
3. As a successful handover comprises the satisfaction of a set of task requirements, we can naturally include them within the OCP to constrain the feasible state space where to search for the optimal solution.
4. Nowadays, efficient solvers are available which allow finding sub-optimal solutions at high frequency rates enabling real-time control of the robotic platform [Yutao, 2019]. This empowers researchers to develop NMPC-based controllers to control the robot dynamics in real time.

### Optimal control problem formulation

The essential part of any NMPC-based controller is the formulation of its OCP. Our proposed formulation is visually summarized in Figure 3.2, and briefly discussed hereafter. The full derivation is detailed in Chapter 6.

In the following, we list the terms composing the objective function in the OCP reported in Figure 3.2.

- First, we have a *human-relative motion* part representing the task of approaching and reaching the human during the handover. The reference trajectory is specified w.r.t. a frame attached to the human (which justifies the name “human-relative”) allowing the robot to adapt the computed solution to the operator motion.

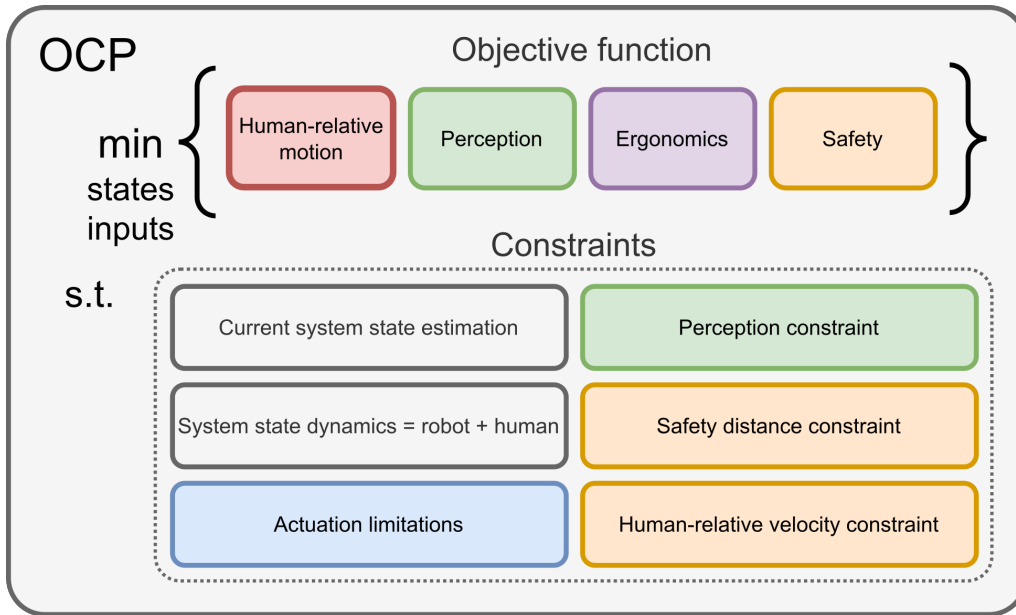


Figure 3.2: A visual description of the OCP formulation employed in the second subproblem (human-aware handover). We highlight the terms related to the reference trajectory, the perception and actuation requirements in red, green and blue, respectively. Then, in orange and purple we denote, respectively, the terms related to the safety and the ergonomics. The solution to this minimization problem provides the optimal robot states and inputs.

- Then, we have a *perception* task consisting in maintaining visibility of the human during the whole operation.
- Thirdly, the cost function accounts for the *ergonomics* of the human. As we will show in more detail in Chapter 6, this term accounts for the joint articular stress that the user undertakes during the handover. Therefore, the controller, while generating the proper commands to let the robot perform the handover, is tasked to minimize the physical burden required by the operator.
- Finally, a fourth term is added to guarantee the human *safety* and to prevent collisions between the two agents during the handover.

The constraints are as follows.

- An initialization of the OCP with the latest available system state measure, that allows coping with the real system deviations w.r.t. its nominal model.
- The system dynamics composed of the robot and the human models. The inclusion of the human model allows the robot to anticipate the user motion by exploiting the predictive nature of the controller.
- The robot actuation limitations to ensure the generation of feasible motor commands.
- A constraint on the perception to maintain visibility of the human body during the entire handover.
- A minimal safety distance to enforce the collision avoidance requirement.
- A constraint on the human-relative velocities. We want to limit the velocities in the human direction in order to constrain the amount of energy transferred in case of an unavoidable collision [Villani, 2018].

*Remark. Perception and safety requirements.* In the OCP shown in Figure 3.2, the perception and safety requirements are not only included as constraints but also as terms in the cost function. The reason behind this choice is related to the applicability of this control approach to a real system. As the optimal solution may be found at the feasibility boundaries of the search space, any uncertainty in the robot and human states and models may cause the violation of any of those constraints. Therefore, to avoid jeopardizing the satisfaction of those conditions, an additional term is added to the cost function allowing maintaining a certain degree of robustness to the uncertainties affecting the real system.

## 3.2 Main results

Here, we briefly present the results achieved with our proposed methodology. We divide this section into two parts, each one introducing the validation setup and the outcomes of the control architectures proposed for each subproblem. Therefore, in Sections 3.2.1 and 3.2.2, we show the results related to the hierarchical control architecture achieving physical Human-Aerial Robot Interaction and the human-aware Nonlinear Model Predictive Control, respectively. The detailed discussion of the quantitative results collected throughout the validation process is provided later in Chapters 5 and 6.

### 3.2.1 Physical Human-Aerial robot interaction

We validate the hierarchical control architecture presented in Section 3.1.1 in two scenarios. First, we test the developed framework in a task requiring physical interaction with the environment and, later on, physical collaboration with a human worker. As first validation setup we choose a *pick-and-place* operation, where one or more objects shall be collected by the robot and transported to a different location. We choose this application as first validation scenario because the ability to pick and place an object represents a requisite to achieve any R2H handover. Conversely, for the Human-Aerial Robot (HAR) collaborative scenario, we select a *H2R handover* in a high-from-ground setting. Specifically, a human worker is located on a scaffold while performing an activity with a certain tool. After completing the assigned task, the operator wants to return the item which is not used anymore. Thus, the robot shall retrieve the object from the partner.

#### Pick-and-place application

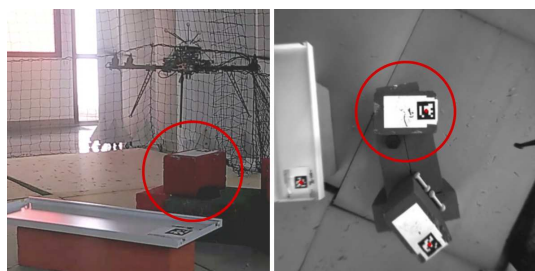
The validation setup comprises several bricks located in the environment which shall be placed at a different location. The robot is a custom-designed fully-actuated hexa-rotor equipped with a down-facing monocular camera and a gripper allowing picking only one brick at the time, as shown in Figure 3.3. The pickable objects and the placing location are marked with fiducial marker, namely Aruco tags [Garrido-Jurado, 2014], which can be detected by the onboard camera. Both the placing position and those of the bricks are not known a priori, thus the multi-rotor aerial vehicle (MRAV) has to scan the area to find them. In Figures 3.3 and 3.4, we show



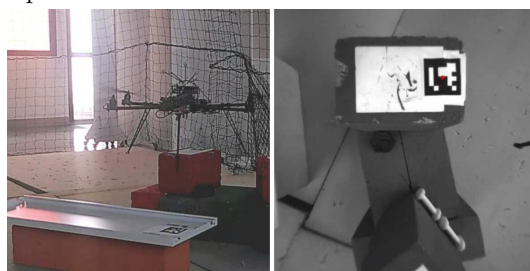
(a) The robot scans the area searching a brick to pick. On the right, no bricks appear within the frame acquired by the onboard camera.



(b) The robot has detected a brick, which is highlighted with a red circle. A successful object detection is marked through a small red dot on top of the identified fiducial marker.



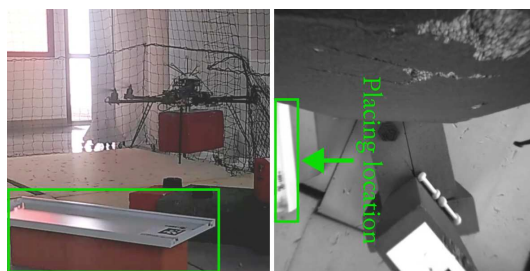
(c) The robot arrives on top of the detected object driven by the velocity commands generated by the visual servoing control loop.



(d) In that position, the robot starts a descending maneuver to collect the brick, which is still visible in the sensor FoV.



(e) The robot is in contact with the object. The interaction is regulated by means of the physical interaction module.

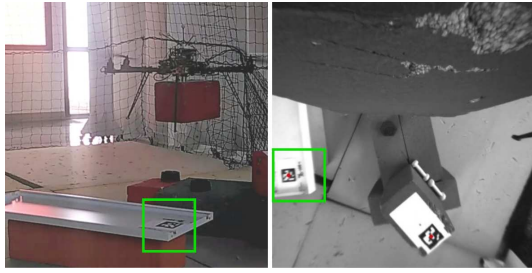


(f) The robot takes off with the collected brick. The placing location, marked in green, starts to move within the sensor image, but it is not detectable yet.

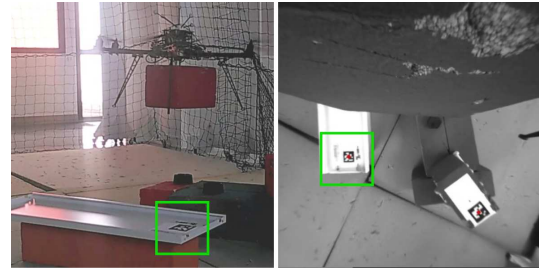
Figure 3.3: First part of the pick-and-place operation designed to experimentally validate the visual and physical control architecture allowing the robot to physically interact with the environment. The frames acquired by the onboard camera are reported on the right of each image. The second half of the experimental validation is reported in Figure 3.4.

some snapshots of the experiments conducted in the indoor arena at LAAS. The figures are reported in chronological order, from left to right, and top to bottom.

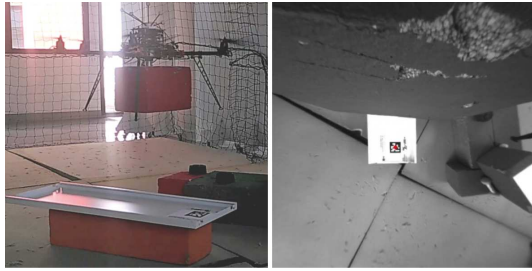
At first, in Figure 3.3a, the robot is scanning the area to search for a collectable brick while following a pre-defined trajectory. Then, in Figure 3.3b, the robot detects the first object in the environment and the visual servoing is now enabled. This module generates velocity commands to drive the robot on top of the brick, which corresponds to the situation in Figure 3.3c. Once the robot is above it, the aerial vehicle initiates the picking phase involving a descending maneuver to bring the robot gripper in contact with the object to collect, as shown in Figure 3.3d. As the MRAV gets closer to the brick, the camera Field of View (FoV) may be obstructed by the surrounding environment or the robot gripper, thus the sight of the detected



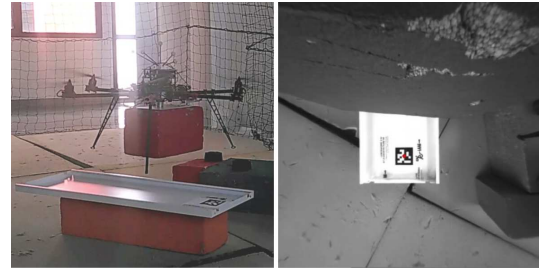
(a) The robot has detected the placing location, which is marked in green. The visual servoing will now drive the robot towards that spot.



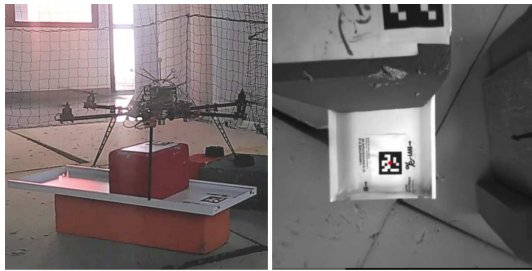
(b) The robot is in the correct position to initiate the placing phase.



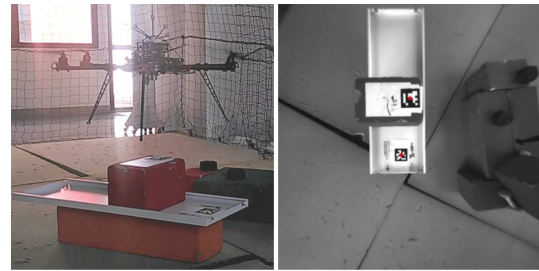
(c) Once the robot is properly positioned, it starts descending to place the brick.



(d) The robot is almost in contact with the placing board and the Aruco tag is still within the camera FoV.



(e) The robot has placed the object, and it is taking off.



(f) The robot is now scanning again the area for a new brick to repeat the process.

Figure 3.4: Second part of the pick-and-place operation.

object may be lost. For this reason, if the visibility is lost, the descending maneuver is performed *blindly* until a sufficiently large external contact force is detected by the wrench observer of the control architecture.

In Figure 3.3e, the robot is in contact with the brick and attempts to pick it. Here, the brick is no more visible, as announced earlier. During the contact phase, the physical interaction module modifies the original desired trajectory with a new one allowing the system to be compliant with the environment. The robot picks up the detected brick by applying a force at the contact location, so that the gripper can effectively grasp the object. At this point, the robot should start scanning the area again, this time searching the placing location. As the robot takes off with the collected object, the placing location is not still completely visible within the onboard camera FoV, as noticeable in Figure 3.3f. When the robot has completed the ascending maneuver, in Figure 3.4a, the placing location is properly detected by the onboard vision system, thus the visual servoing is now driving the AR towards that position. From Figures 3.4b to 3.4d, it performs a new descending maneuver to place the brick on top of the white board, next to the fiducial marker identifying that



Video 3.1: QR code to the video showcasing the pick-and-place validation campaign. Link: <https://peertube.laas.fr/w/eEY4Q8nWdj1z5vs4JBaMpt>.

spot. Lastly, once the object has been placed, the robot takes off to look for other bricks, then it repeats the pick-and-place steps, as shown in Figures 3.4e and 3.4f. In Video 3.1, a clickable link and a scannable QR code are provided redirecting the reader to the video showing the experiments conducted at LAAS to validate the proposed control architecture.

### **Human-to-robot handover**

In this case, we design a validation scenario involving a Human-Aerial robot handover. In particular, a user operator is positioned on a scaffolding elevated from the ground, whose situation is simulated by means of a scaffold lift, as depicted in Figure 3.5a. In that location, we imagine the human has performed a task with a certain tool and needs to return it to a ground station. However, the user cannot descend easily, so we use an AR to retrieve the object from the operator and bring it to the proper location. In the considered HARH scenario, the item transfer is occurring from the human to the robot (H2R handover). The AR is an aerial manipulator composed of the same fully-actuated hexa-rotor base, as in the previous case, but now equipped with a 3-Degrees of Freedom (DoFs) manipulator arm, as shown in Figure 3.5. We assume that the handover location is known a priori and the robot is driven there by following a pre-defined trajectory generated by an external module. However, the user can reposition the robot to a more convenient location by pulling or pushing its EE. In such a case, the AM is made compliant by the physical interaction layer included in the proposed hierarchical control architecture, which allows the human worker to re-position the robot. The final handover location decided by the user is meant to meet the personal preferences of the operator regarding the object exchange. Here, we show some photos taken during the experimental validation conducted at the University of Twente<sup>1</sup>.

The robot starts from an initial location and it is tasked to approach the operator who is signaling the need of returning the tool to the robot, as shown in Figure 3.5a. Therefore, in Figure 3.5b, the robot starts approaching the scaffolding and heading the EE towards the user. The desired end effector pose is pre-determined and the robot joints are commanded accordingly. In Figure 3.5c, the robot has reached the location for the handover and it has its EE closer to the operator. In that configuration, it waits to receive the object. In Figure 3.5d, the operator signals his intention to start the physical interaction phase. However, the current position

---

<sup>1</sup>This work has been conducted during a 3-month visiting period at the Robotics and Mechatronics laboratory in the University of Twente, in Enschede (the Netherlands).



(a) The robot is at its starting location, while a human worker is on a scaffolding signaling the intention to return an object and initiate the handover.



(b) The robot is approaching the operator by moving closer and heading the EE towards the scaffolding.



(c) The robot has reached the handover location. It has its EE closer to the user waiting for the human to return the tool.

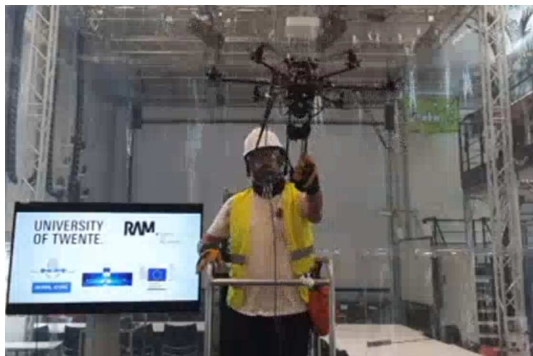


(d) The human signals his intention to start the interaction. At this point, the physical interaction layer of the robot control framework is enabled.

Figure 3.5: Experimental validation of the hierarchical control architecture allowing the robot to physically interact with a human worker. The experiments involve a Human-Aerial robot handover scenario, where an operator is working on a scaffolding and needs to return a tool to the robot. These images depict the approach and reach phases of the H2R handover process. Pictures of the next part of the experiments are reported in Figure 3.6.

of the AM is uncomfortable and unpractical to the user, who decides to move it to a more preferable location. As shown in Figure 3.6a, the worker does that by pushing its end effector and moving it to the new position. The robot behavior is regulated by the physical interaction layer of the control framework, which makes the AR compliant with the human actions and move to the user desired position, as reflected in Figures 3.6b and 3.6c. Lastly, in Figure 3.6d, the operator is attaching the object to be returned to the robot end effector, which consists of a simple hook.

After having handed over the object as depicted in Figure 3.7a, the human worker decides to move back the robot to the initial handover location. Consequently, as shown in Figures 3.7b and 3.7c, the user pushes the robot EE again to make the robot slide back to the point where it originally approached the scaffolding. This demonstrates the validity of the proposed control architecture to account for subsequent applications of external actions and properly stabilize the platform. Lastly, in Figure 3.7d, the experiment concludes with the robot returning to its starting position, where it can finally land. In Video 3.2, a clickable link and a scannable



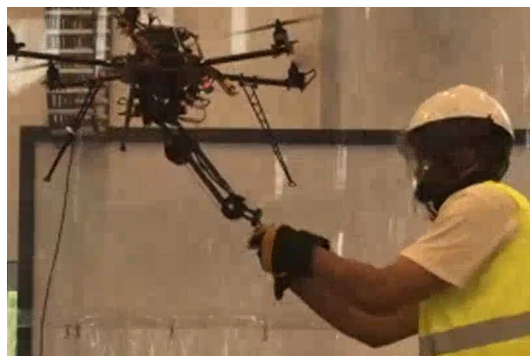
(a) The human worker decides to re-position the robot to a more convenient location matching his preferences by pushing on the robotic arm.



(b) The robot is compliant and moves accordingly to the human pushing force.



(c) The robot has reached the location that the operator prefers.



(d) The user is now handing over the tool to the robot.

Figure 3.6: Photos of the interaction phase of the H2R handover. The last part of this experimental validation is reported in Figure 3.7.

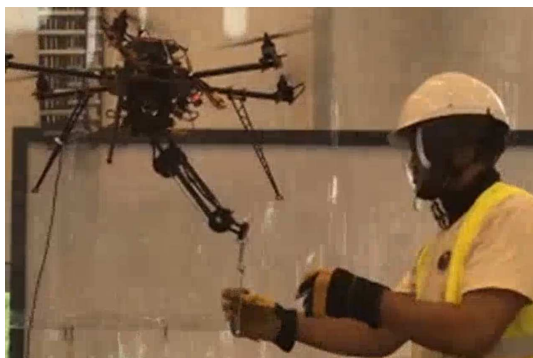
QR code can be used to be redirected to the multimedia material showcasing the experiments conducted in the University of Twente, in the scope of the Aerial-CORE project.

In order to assess the repeatability of the presented results, these experiments have been replicated also at LAAS, since an identical aerial prototype has been assembled within the time duration of the present manuscript.



Video 3.2: QR code to the video presenting the experiments conducted at the University of Twente in the scope of the Aerial-CORE project. Link: <https://youtu.be/LrQxXbQ5IHc>.





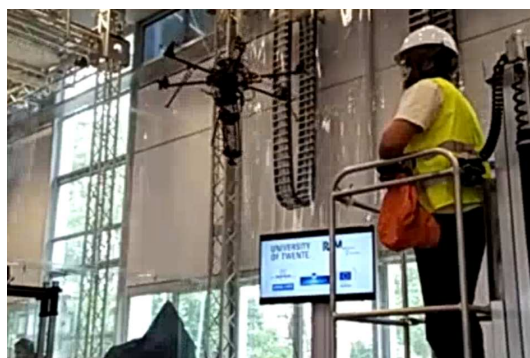
(a) The transferred tool is attached to the robot EE which consists of a simple hook.



(b) The object transfer has terminated. The human worker decides to move back the robot to the initial handover location.



(c) The robot has reached the position from where it originally approached the user.



(d) The robot is now moving back to the starting point where it can land and conclude the handover process.

Figure 3.7: Last part of the H2R handover. Here, the images show the last portion of the interaction between the human and the aerial robot, and the retraction phase where the robot returns to the initial position.

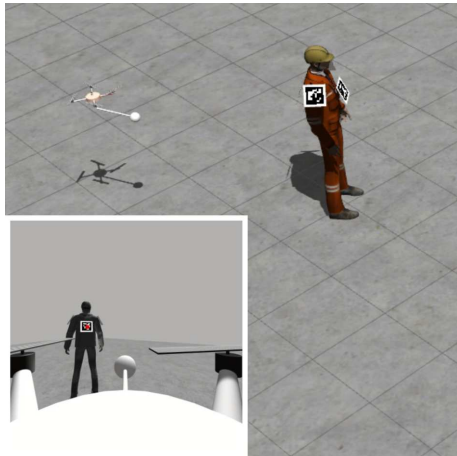
### 3.2.2 Human-aware aerial robot handover

We design another HAR handover scenario to validate the control methodology proposed for the second subproblem. In this case, we imagine a human worker who needs a tool and we employ an AR to deliver it. The operator can move freely in the environment and the robot is an under-actuated quad-rotor equipped with a front-facing monocular camera. The MRAV can detect the pose of the other agent thanks to AruCo tags attached to the human body. The object to be exchanged is represented by a spherical ball attached to the robot by means of a rigid EE.

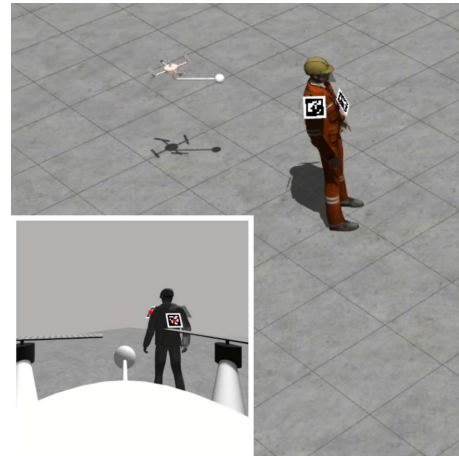
#### Robot-to-human handover

We run a set of simulations in order to validate the control algorithm in a safe environment implemented in the Gazebo simulator. In Figures 3.8 and 3.9, we report some pictures captured during the simulations.

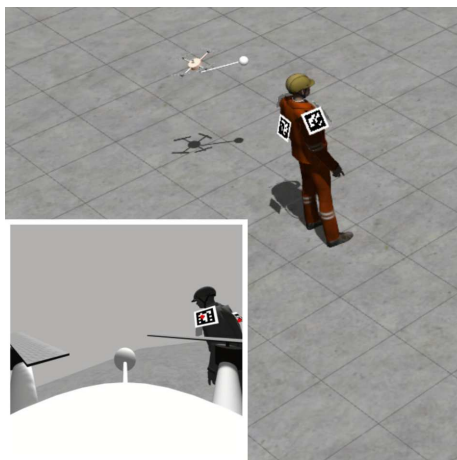
The robot starts from a point located behind the human operator where the human body is visible and contained inside the FoV of the robot onboard sensor, as shown in Figure 3.8a. The MRAV is tasked to reach a waypoint in front of the human and then to hand over the object to the user. While the robot is moving to the



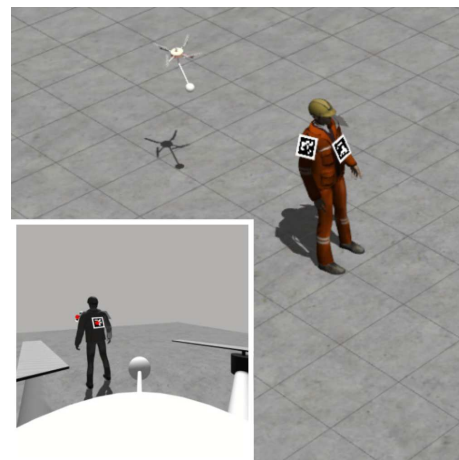
(a) The robot starts behind the human worker with the partner body inside the sensor FoV.



(b) The robot initiates the handover process. It is commanded to approach the partner and to reach an intermediary position in front of the human.

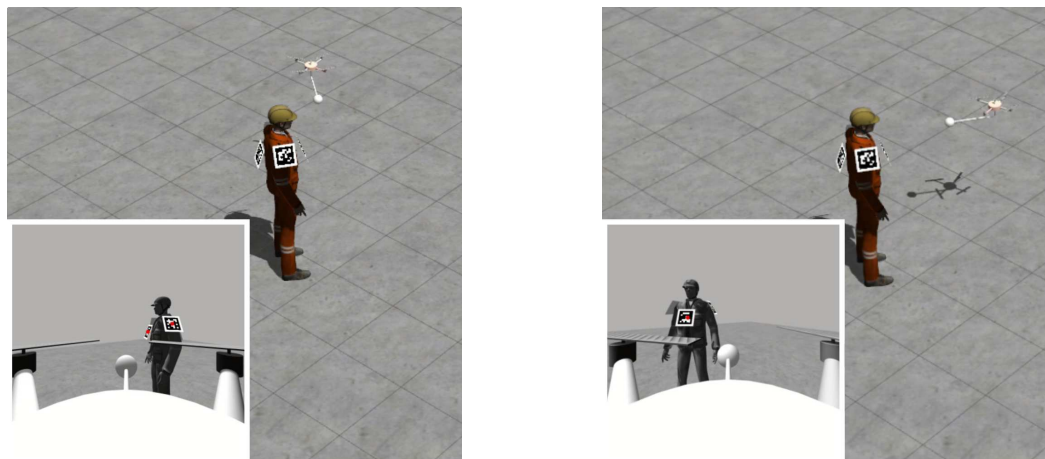


(c) While the robot is moving to that location, the human worker suddenly moves and turns.



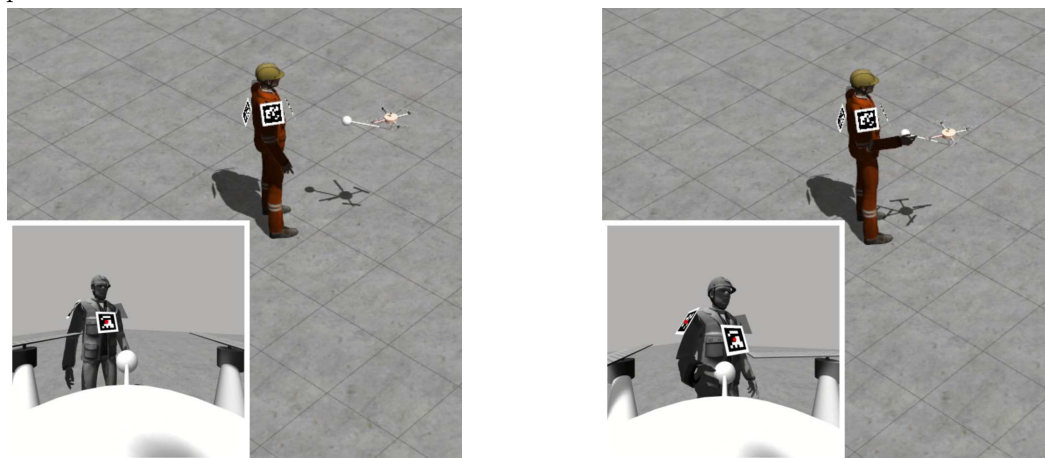
(d) The aerial robot is adapting its motion according to the one of the partner.

Figure 3.8: First part of the simulations of the R2H handover we carried out to validate the human-aware NMPC framework. For each subfigure, the current frame acquired by the onboard camera is reported on the bottom left corner. The tiny red dots over the fiducial markers (AruCo tags) denote their correct detection. The second half of the simulation validation is reported in Figure 3.9.



(a) The human stops moving, thus the robot can continue its motion without performing any adaptation.

(b) The AR has almost reached the intermediary location in front of the human worker.



(c) The robot is in front of the human, thus terminating the approach phase. From here, it can now reach the human.

(d) The robot is handing over the object in an ergonomic location computed by the NMPC-based controller.

Figure 3.9: Snapshots related to the second part of the simulation validation campaign of the “human-aware” controller.

commanded location, as depicted in Figure 3.8b, the human suddenly moves and turns. Thus, the robot has to re-adapt the planned motion accordingly, while maintaining visibility of the human trunk, as reflected in Figures 3.8c and 3.8d. Once the operator stops moving, the robot manages to reach the intermediary position in front of the partner without performing any motion adaptation, as shown in Figures 3.9a and 3.9b. Once the approach is terminated, from that location, the aerial robot can finally reach the worker to perform the tool handover, as depicted in Figure 3.9c. The final handover position, reached by the robot in Figure 3.9d, is computed online by the developed controller according to the proposed OCP formulation, which accounts for the operator ergonomics. The mathematical details are provided in Chapter 6. Additionally, in Video 3.3, a clickable link and a scannable QR code redirect to the media content to appreciate the realized simulations presented earlier.

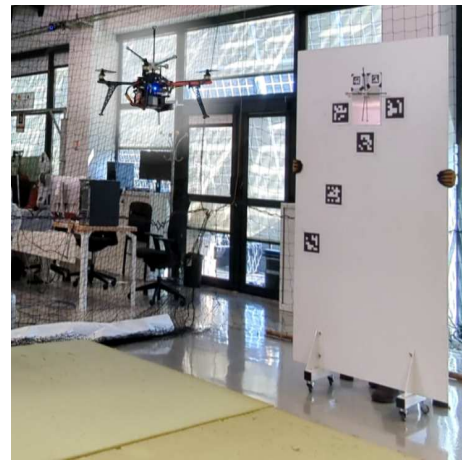
Once validated the effectiveness of the controller in the simulator, we conducted a set of experiments in the indoor arena at LAAS. We decided to employ a moving



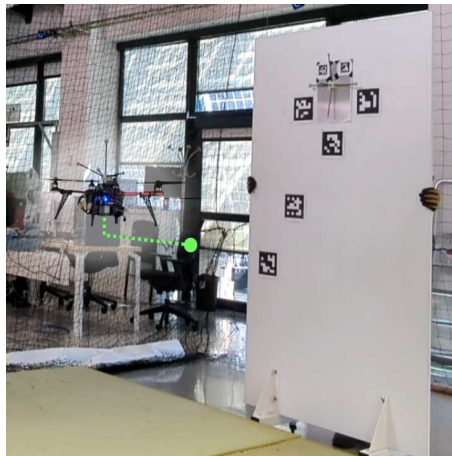
Video 3.3: QR code to the video showing the simulations realized in the Gazebo simulator for validating the “human-aware” controller. Link: <https://peertube.laas.fr/w/fvDnY4ZCaMhd5ztT2aUhS5>.



(a) The robot is at the starting location before approaching the avatar board being manipulated by a user who is behind it.



(b) The user moves the panel while the robot is reaching to perform the handover.



(c) The robot is at the final handover location which is computed as a trade-off between the human ergonomics, visibility and safety costs.

Figure 3.10: Experimental validation of the human-aware NMPC-based controller which achieves a safe and ergonomic robot-to-human handover. The controller is informed that a virtual object is frontally attached to the robot body. This virtual item is visualized in 3.10c as a green dot attached to a virtual rigid arm (the dashed green line) at the robot frame.



Video 3.4: QR code to the video presenting the preliminary experiments realized at LAAS for testing the “human-aware” controller in real-world settings. Link: <https://peertube.laas.fr/w/5aCzSTsi5BMhgXHSUB9Kg4>.

panel as a mock-up of the human operator, in order to test the control framework in a protected and controlled environment. Practically, a human user is standing behind the board to operate and move it. We attach a set of Aruco tags to this panel to exploit the same detection pipeline employed in the simulations. In Figure 3.10, we report some photos taken while performing the real experiments.

In Figure 3.10a, the robot is moving in front of the human operator to hand over the object. After the approaching phase, in Figure 3.10b, the user, who is standing behind the board and manipulating it, is moving the panel. Consequently, the robotic partner adapts its motion accordingly while trying to reach the board. Lastly, in Figure 3.10c, the AR is in the final handover location which is derived from solving the proposed OCP, thus being a trade-off between the safety, visibility, ergonomic costs and satisfying the system constraints. In Video 3.4, a clickable link and a scannable QR code are given and they redirect the reader to the multimedia material to appreciate the preliminary experiments conducted at LAAS to test this controller in real-world settings.

### 3.3 Synthesis

Hereby, we draw some preliminary conclusions about the work briefly presented so far, while more exhaustive considerations will be derived in Chapter 7.

In this chapter, we unveiled the main ideas behind the proposed methodologies addressing each subproblem. Specifically, we presented (i) a modular hierarchical control architecture, and (ii) a predictive optimization-based approach. The former allows to both visually control the EE towards the desired location, and to enhance the AR with the capability to exchange forces and torques with the surrounding environment and a human worker. The latter (NMPC-based controller) introduces human-centric metrics within the control law, such as ergonomics and safety, and it accounts for constraints related to perception and actuation tasks. With the inclusion of the human state within the system dynamics of the OCP, the derived control algorithm is “aware” of the human partner.

To validate the presented methods, we performed a set of simulations and experiments. In particular, for the first subproblem, we designed two scenarios. First, we analyzed the physical interaction capabilities of our hierarchical architecture in a pick-and-place operation with a fully-actuated hexa-rotor. Then, we tested its extended version in a H2R handover comprising the interaction between an aerial manipulator

---

and a human worker on a scaffolding. Instead, for the second subproblem, we first realized realistic simulations in a safe and controlled environment aimed at validating the main features of the NMPC-based controller. This time, the considered scenario involved a R2H handover. Then, we performed an experimental validation to assure its applicability to real-world settings.



**Part II**  
**Scientific work**





# Chapter 4

## Modeling

The aim of this chapter is to provide the reader with the mathematical concepts and tools that will be employed in the two subsequent chapters to detail our control methodologies. In particular, the contribution of this chapter is to present the mathematical models used to describe the state evolution of the main agents involved in the Human-Aerial robot handover (HARH) problem presented in Chapter 2.

In this thesis, we rely on the assumption that the agents can be modeled as the composition of one or more rigid bodies, i.e. physical entities that do not undergo deformation or, more realistically, a deformation whose effect is so small which can be neglected [Goldstein, 1950]. Conversely, the constituting particles of deformable objects change relative spatial arrangement according to the intensity of the applied deformation [Ruina, 2019].

Henceforth, this chapter is organized as follows. In Section 4.1, we offer an overview of the notation that will be used in the remainder of this manuscript. Additionally, in Sections 4.2 and 4.3, we recap the main definitions and concepts in rigid-body kinematics and dynamics. After this introductory part, we unveil the models of the agents involved in the HARH problem. Specifically, in Sections 4.4 and 4.5, we provide the models of the *robotic agents*, respectively a Generically-Tilted Multi-Rotor (GTMR) and an aerial manipulator (AM), the latter being equipped with a poly-articulated robotic arm. Next, in Section 4.6, we discuss the actuation mechanism of the class of aerial robots (ARs) considered in our investigation, namely that of multi-rotor aerial vehicles (MRAVs). Subsequently, in Section 4.7, we introduce a state-space representation for the aerial robotic agents. Later on, in Section 4.8, we continue by describing the perceptive sensors that are typically mounted on-board MRAVs. Finally, in Section 4.9, we present the model associated to the motion of the *human agent* and its upper limb. Similarly, we then show the state-space formulation for the human models. Lastly, in Section 4.10, we conclude this chapter with a brief reminder on the concepts covered in this chapter.

### 4.1 Notation

In this section, the main notation adopted throughout this manuscript is reported.

### 4.1.1 Sets

Sets are written using standard letters, for example  $\mathbb{N}$  and  $\mathbb{R}$  denote the set of natural and real numbers, respectively. Intervals are denoted using a set of parenthesis. Specifically, continuous intervals are indicated with square brackets, as  $[a, b]$  is the interval containing all real numbers within  $a$  and  $b$  given  $a, b \in \mathbb{R}$  and  $a < b$ . Similarly, discrete sets are denoted with curly braces, e.g.  $\{n, m\}$  and  $\{1, \dots, 4\}$  denote, respectively, a set containing two integer numbers  $n$  and  $m$  and a set of integer numbers within 1 and 4.

Variables are written using Greek or Latin letters, with the following font style:

- *Normal* font for scalars, e.g.,  $a \in \mathbb{R}_+$  denotes a real positive scalar.
- *Bold* font for (column) vectors, e.g.,  $\mathbf{v} \in \mathbb{R}^2$  and  $\mathbf{w} \in \mathbb{R}^3$  are two generic vectors of real numbers, the former 2-dimensional (2D) and the latter 3D.
- *Capital bold* font for matrices, for instance,  $\mathbf{A} \in \mathbb{R}^{3 \times 3}$  is a generic 3-by-3 matrix of real numbers.

A vector of dimension  $p$ -by-1 full of zeros is denoted with  $\mathbf{o}_p$ , while null matrices of size  $n \times m$  and  $n \times n$  are respectively denoted as  $\mathbf{O}_{n \times m}$  and  $\mathbf{O}_n$ . An  $n$ -by- $n$  identity matrix is indicated with the symbol  $\mathbf{I}_n$ .

Elements of a generic vector  $\mathbf{v} \in \mathbb{R}^n$  and matrix  $\mathbf{A} \in \mathbb{R}^{p \times q}$  are denoted, respectively, with  $v_k, \forall k \in \{1, \dots, n\}$  and  $a_{i,j}, i \in \{1, \dots, p\}, j \in \{1, \dots, q\}$ .

Lastly, functions are denoted with Latin letters - either in upper or lowercase - and we adopt the same font formatting as variables. Therefore, a generic scalar function will be denoted as  $b = g(a) : \mathbb{R} \rightarrow \mathbb{R}$ , where  $a \in \mathbb{R}$  and  $b \in \mathbb{R}$  are, respectively, the input and output scalars. Contrary, a vector function having as input argument a vector  $\mathbf{v} \in \mathbb{R}^n$  and output a vector  $\mathbf{w} \in \mathbb{R}^m$  ( $n, m \in \mathbb{R}_+$ ), is defined as  $\mathbf{w} = \mathbf{f}(\mathbf{v}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Instead, using capital letters, we denote a function having in input a matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$  and output a matrix  $\mathbf{B} \in \mathbb{R}^{p \times q}$ , as

$$\mathbf{B} = \mathbf{F}(\mathbf{A}) : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{p \times q}. \quad (4.1)$$

### 4.1.2 Operators

Given a generic vector  $\mathbf{v} \in \mathbb{R}^n$ , we denote with  $\mathbf{v}^\top$  the transpose of  $\mathbf{v}$ . The Euclidean (or 2-) norm of a vector  $\mathbf{v} \in \mathbb{R}^n$  is written as  $\|\mathbf{v}\| \in \mathbb{R}_+$ . The skew-symmetric matrix<sup>1</sup> associated to a generic vector  $\mathbf{w} \in \mathbb{R}^3$  is denoted with  $\mathbf{S}(\mathbf{w}) \in \mathbb{R}^{3 \times 3}$ , and it is defined as follows:

$$\forall \mathbf{w} = [w_1 \ w_2 \ w_3]^\top \in \mathbb{R}^3, \quad \mathbf{S}(\mathbf{w}) = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \quad (4.2)$$

Next, the inverse operator is denoted with  $\bullet^\vee$ , thus  $\mathbf{S}(\mathbf{w})^\vee = \mathbf{w}$ .

Time derivation of a vector  $\mathbf{p} \in \mathbb{R}^n$  is denoted with the compact notation  $\frac{d}{dt}\mathbf{p} = \dot{\mathbf{p}} \in \mathbb{R}^n$ . Similar notation is used for higher-order derivatives w.r.t. time: for

<sup>1</sup>A skew-symmetric (or antisymmetric or antimetric) matrix is a square matrix whose transpose equals its negative, i.e.  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is skew-symmetric  $\iff \mathbf{A}^\top = -\mathbf{A}$  [Strang, 2006].

instance,  $\frac{d^2}{dt^2}\mathbf{p} = \ddot{\mathbf{p}}$  and  $\frac{d^3}{dt^3}\mathbf{p} = \dddot{\mathbf{p}}$  are the second- and third-order derivatives of the vector  $\mathbf{p} \in \mathbb{R}^n$ , respectively.

Lastly, when we write  $\underline{\mathbf{u}} < \mathbf{u} < \bar{\mathbf{u}}$ ,  $\mathbf{u} \in \mathbb{R}^n$ , we mean that  $u_i \in [u_i, \bar{u}_i]$ ,  $i \in \{1, \dots, n\}$ . Therefore, the vectors  $\underline{\mathbf{u}}$  and  $\bar{\mathbf{u}}$  collect, respectively, the lower and upper bounds of all the elements of  $\mathbf{u}$ . Moreover,  $u_i \neq u_j$  with  $i, j \in \{1, \dots, n\}$ ,  $i \neq j$ , and similarly for the upper bounds gathered in  $\bar{\mathbf{u}}$ .

### 4.1.3 Reference frames

A generic reference frame is denoted as  $\mathcal{F}_A$ , having origin in  $O_A$  and orthogonal unit axes  $\mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A$ . Alternatively, we write  $\mathcal{F}_A = O_A, \{\mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A\}$ .

Being  ${}^A\mathbf{v} \in \mathbb{R}^3$  a 3D vector expressed w.r.t.  $\mathcal{F}_A$ , it is sometimes more intuitive and convenient to denote its components as  ${}^A\mathbf{v} = [{}^Av_x \ {}^Av_y \ {}^Av_z]$ , rather than relying on numeric subscripts for indexing each element. For simplifying the notation, the reference frame is omitted when implicitly obvious from the context. For instance  ${}^A\mathbf{v} = [v_x \ v_y \ v_z]$ .

The position of  $O_A$  expressed w.r.t. to another reference frame  $\mathcal{F}_B$  is denoted with  ${}^B\mathbf{p}_A$ . For what concerns the orientation of a frame w.r.t. another one, different representations exist, which are discussed in the following section.

### 4.1.4 Subscripts

In this manuscript, when referring to a certain variable, the subscript can be either an upper or lowercase letter. In the former case, the variable is related to a geometrical point, e.g.  $\mathbf{p}_A$  can denote either the position of a point  $A$  or the one of the origin  $O_A$  of  $\mathcal{F}_A$ <sup>2</sup>. If the subscript is a lowercase letter, then the variable is associated to a certain entity. For instance, in the handover problem,  $\mathbf{p}_h$  denotes the position of the human. Similarly, the robot body mass and the configuration of the robotic arm attached to an aerial manipulator are denoted, respectively, with  $m_b$  and  $\mathbf{q}_a$ .

## 4.2 Rigid-body attitude representations

Before presenting the derivation of the dynamic models describing the motion of rigid bodies, it is worth reminding some concepts regarding the various approaches to represent their orientation in 3D space. In general, we denote with  ${}^B\mathbf{r}_A$  the orientation of  $\mathcal{F}_A$  w.r.t.  $\mathcal{F}_B$ , where  $\mathbf{r}$  denotes a generic rotation representation. Different parametrizations are possible which are introduced hereafter.

### 4.2.1 Rotation matrices

Given two reference frames  $\mathcal{F}_A$  and  $\mathcal{F}_B$ , the first method to represent the orientation of  $\mathcal{F}_A$  w.r.t.  $\mathcal{F}_B$  is by means of the rotation matrix  ${}^B\mathbf{R}_A \in \mathbb{R}^{3 \times 3}$ . Each column

<sup>2</sup>The discrimination between those two cases will be clear from the context.

of  ${}^B\mathbf{R}_A$  is composed of the coordinates of the unit vectors of  $\mathcal{F}_A$  expressed w.r.t.  $\mathcal{F}_B$ , namely  ${}^B\mathbf{x}_A$ ,  ${}^B\mathbf{y}_A$ ,  ${}^B\mathbf{z}_A$ . The column vectors composing a rotation matrix are mutually orthogonal, as they represent the unit vectors of an orthonormal frame, and they have unit norm [Siciliano, 2008]. Therefore, it is said that rotation matrices form a group named *Special Orthogonal* group of dimension  $m$  and denoted with  $SO(m)$ , where  $m = 3$  for spatial rotations<sup>3</sup> and  $m = 2$  for planar ones [Siciliano, 2009]. As a result, given  $\mathbf{v} \in \mathbb{R}^3$ , we can write  $\mathbf{S}(\mathbf{v}) \in SO(3)$ , where  $\mathbf{S}(\bullet)$  is the skew operator of Equation (4.2). Additionally, for a given element  $\mathbf{R}$  of the group  $SO(3)$  it can be proven that

$$\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}_3 \quad \text{and} \quad \det(\mathbf{R}) = 1. \quad (4.3)$$

Given a vector  ${}^A\mathbf{p} \in \mathbb{R}^3$  expressed in a frame  $\mathcal{F}_A$ , it is possible to express it in a second frame  $\mathcal{F}_B$  thanks to the rotation matrix  ${}^B\mathbf{R}_A$  as follows

$${}^B\mathbf{p} = {}^B\mathbf{R}_A {}^A\mathbf{p}, \quad (4.4)$$

where  ${}^B\mathbf{p}$  denotes the same vector but expressed w.r.t.  $\mathcal{F}_B$ . Eventually, successive rotations can be obtained by composing a set of finite individual rotation matrices, each one consisting of an intermediary rotation from the previous frame to the next one. For instance, given three reference frames  $\mathcal{F}_A$ ,  $\mathcal{F}_B$  and  $\mathcal{F}_C$ , it is possible to write

$${}^C\mathbf{p} = {}^C\mathbf{R}_A {}^A\mathbf{p}, \quad (4.5a)$$

$${}^C\mathbf{R}_A = {}^C\mathbf{R}_B {}^B\mathbf{R}_A, \quad (4.5b)$$

where the intermediary rotation matrices - namely  ${}^C\mathbf{R}_B$  and  ${}^B\mathbf{R}_A$  - are being concatenated by premultiplication. However, consecutive rotations can be also specified by constantly referring them to a common fixed frame. In the latter case, it is necessary to postmultiply each individual rotation matrix. For more details and proofs the reader is addressed to [Siciliano, 2008; Siciliano, 2009].

Rotation matrices are very easy to manipulate and practical to use when, for instance, it is necessary to rotate a vector and compose rotations, as shown Equations (4.4) and (4.5). However, they contain redundant information as 9 non-independent parameters<sup>4</sup> are required to represent an orientation. Nevertheless, thanks to their practicality and the absence of singularities intrinsic to their redundant nature, they find great application in robotics and especially large adoption in control problems. In the following, other representations are discussed which requires a smaller number of parameters.

## 4.2.2 Euler angles

As mentioned earlier, rotation matrices are redundant as 9 parameters are employed to describe a transformation involving only 3 Degrees of Freedom (DoFs). Differently, Euler angles consist in 3 angles of rotations around 3 chosen axes. As a consequence, they constitute a *minimal representation*, as only 3 coordinates are used to describe

<sup>3</sup>While  $SO(3)$  denotes the special orthogonal group of 3D rotations,  $SE(3)$  is the special Euclidean group comprising 3D rotations and translations.

<sup>4</sup>The 9 parameters are constrained by the orthogonality conditions in Equation (4.3).

the orientation state of a rigid body. As a matter of fact, a minimal representation of a special orthonormal group  $SO(m)$  requires  $m(m-1)/2$  parameters [Siciliano, 2009]. Therefore, for planar rotations in  $SO(2)$  only 2 parameters are needed, while 3 shall be used for spatial rotations in  $SO(3)$ .

There exist different combinations of such angles and axes - up to 12 - which shall satisfy the requirement that two consecutive rotations are not made about parallel axes [Siciliano, 2009]. The set of angles - also known as *convention* - is denoted with three letters to denote the rotating axes. Examples are  $xyz$ ,  $XYZ$ ,  $zxx$ , or again  $ZXZ$ . These letters are reported either in uppercase to denote rotations around moving axes (*intrinsic convention*) or in lowercase for rotations around fixed axes (*extrinsic*)<sup>5</sup>.

The most common convention of Euler angles in robotics, and also in the aeronautic sector, is the (*extrinsic*) *Roll-Pitch-Yaw (RPY) angles* or  $xyz$ . It consists in the successive rotations around the  $\mathbf{x}$  about roll angle,  $\mathbf{y}$  about pitch, and lastly around  $\mathbf{z}$  about yaw w.r.t. a common fixed frame. They are usually denoted with the Greek letters  $\phi, \theta, \psi$  and defined as follows:

$$\boldsymbol{\eta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in ]-\pi, \pi] \times \left] -\frac{\pi}{2}, \frac{\pi}{2} \right] \times ]-\pi, \pi]. \quad (4.6)$$

As elementary rotations about each axis can be identified, a rotation matrix can be expressed as function of the Euler angles. For instance, given the RPY angles  $\boldsymbol{\eta}$ , the rotation matrix representing the same transformation can be obtained as follows:

$$\mathbf{R}(\boldsymbol{\eta}) = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi). \quad (4.7)$$

Other conventions are obtained similarly, and analytical expressions can be found in [Siciliano, 2009].

Euler angles are very convenient and intuitive due their connection to the physical domain. However, singularities can arise during successive vector rotations. Being defined over a discontinuous set, as clearly noticeable in Equation (4.6), the transitions of roll and yaw from  $-\pi$  to  $\pi$ , or vice versa, are not continuous. In addition, a pitch angle of  $\frac{\pi}{2}$  introduces numerical issues in the computation of Equation (4.7), as one obtains

$$\mathbf{R}(\boldsymbol{\eta}) = \begin{bmatrix} 0 & \sin(\phi - \psi) & \cos(\phi - \psi) \\ 0 & \cos(\phi - \psi) & -\sin(\phi - \psi) \\ -1 & 0 & 0 \end{bmatrix}. \quad (4.8)$$

This phenomenon is known with the name of *Gimbal lock*. When this situation occurs, the system is said to be locked as one DoF is lost and the system is constrained to rotate laying on a plane. Indeed, in Equation (4.8), infinite combinations of  $\phi$  and  $\psi$  rotations produce the same rotation. This takes place whenever the first and last rotations both occur about the same axis [Siciliano, 2008].

<sup>5</sup>Extrinsic and intrinsic can be obtained from one to another by inverting the order of rotations and axes, e.g. the extrinsic  $ZYX$  is equivalent to the intrinsic  $xyz$ .

### 4.2.3 Angle and axis

A third common orientation representation consists in defining an axis of rotation and an angle about which rotate the rigid body. This representation is therefore named *angle-axis* and it comprises 4 parameters: 3 for the 3D unit vector  $\mathbf{u}_\theta$  of the rotation axis, and an additional one for the angle  $\theta \in [0, \pi[$  of the rotation. By applying the Rodrigues formula [Dai, 2015], the associated rotation matrix can be computed as

$$\mathbf{R} = \mathbf{I}_3 + \sin \theta \mathbf{S}(\mathbf{u}_\theta) + (1 - \cos \theta) \mathbf{S}(\mathbf{u}_\theta)^2. \quad (4.9)$$

Interestingly, despite the use of 4 parameters, this representation constitutes a minimal representation as one out four of them can be computed from the other three. However, it is not exempt from singularities, which occur at rotations of  $\theta = \pi$  radians, which is consequently removed from the function domain. Indeed, the rotation about  $\mathbf{u}_\theta$  of  $\theta$  and the opposite rotation about  $-\mathbf{u}_\theta$  of  $-\theta$  are equivalent and cannot be distinguished leading to a non-uniqueness in attitude representation [Siciliano, 2008; Siciliano, 2009]<sup>6</sup>. Moreover, a rotation of a null angle makes the axis undetermined as infinite solutions exist, thus introducing a singularity.

### 4.2.4 Unit quaternions

Quaternions are another representation comprising 4 parameters. In fact, they are composed of 4 real numbers belonging to a space denoted with  $\mathbb{Q}$ , which is homeomorphic to  $\mathbb{R}^4$ . A quaternion  $\mathbf{q}$  is defined as follows

$$\mathbf{q} = q_w + \mathbf{i}q_x + \mathbf{j}q_y + \mathbf{k}q_z \in \mathbb{Q}, \quad (4.10)$$

and then conveniently written as

$$\mathbf{q} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \in \mathbb{R}^4, \quad (4.11)$$

where  $\mathbf{i}q_x + \mathbf{j}q_y + \mathbf{k}q_z$  is the vector part of  $\mathbf{q}$  and  $q_w$  the scalar one [Bork, 1966]. It is possible to find other conventions, which would make the whole formulation different [Solà, 2017]. As an example, it is possible to place the scalar  $q_w$  at the end of the definition of  $\mathbf{q}$ . Nevertheless, the convention used through this manuscript is the one shown in Equation (4.11).

Given two quaternions  $\mathbf{q}_1 = [q_{w1} \ q_{x1} \ q_{y1} \ q_{z1}]^\top$  and  $\mathbf{q}_2 = [q_{w2} \ q_{x2} \ q_{y2} \ q_{z2}]^\top$ , an important operator is the Hamilton product [Girard, 2007], which is defined as

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} q_{w1}q_{w2} - q_{x1}q_{x2} - q_{y1}q_{y2} - q_{z1}q_{z2} \\ q_{w1}q_{x2} + q_{x1}q_{w2} + q_{y1}q_{z2} - q_{z1}q_{y2} \\ q_{w1}q_{y2} + q_{y1}q_{w2} + q_{z1}q_{x2} - q_{x1}q_{z2} \\ q_{w1}q_{z2} + q_{z1}q_{w2} + q_{x1}q_{y2} - q_{y1}q_{x2} \end{bmatrix}, \quad (4.12)$$

<sup>6</sup>The non-uniqueness issue does not represent really a problem is one restricts the domain of definition of theta to positive angles only, i.e.  $\theta > 0$ .

from which it can be derived that

$$\|\mathbf{q}_1 \otimes \mathbf{q}_2\| = \|\mathbf{q}_1\| \|\mathbf{q}_2\|. \quad (4.13)$$

Then, for a quaternion  $\mathbf{q} \in \mathbb{Q}$ , we denote its conjugate as  $\mathbf{q}^*$ . It is obtained from  $\mathbf{q}$  by negating the vector part, thus it is defined as follows:

$$\mathbf{q}^* = \begin{bmatrix} q_w \\ -q_x \\ -q_y \\ -q_z \end{bmatrix} \in \mathbb{R}^4. \quad (4.14)$$

Unit quaternions, i.e. those quaternions for which  $\|\mathbf{q}\| = 1$ , can be interpreted as orientation specification or a rotation operator. Indeed, unit quaternions can be related to the Axis-angle representation introduced in Section 4.2.3, as it can be proven that

$$\mathbf{q} = \begin{bmatrix} \cos \frac{\theta}{2} \\ \mathbf{u} \sin \frac{\theta}{2} \end{bmatrix}, \quad (4.15)$$

which in turn satisfies  $\|\mathbf{q}\| = \sin^2 \frac{\theta}{2} + \cos^2 \frac{\theta}{2} = 1$ . In case of unit quaternions the conjugate of a non-zero quaternion can be proven to be equal to the inverse of the same quaternion [Rucker, 2018], since the inverse of a quaternion is defined as

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} \in \mathbb{R}^4. \quad (4.16)$$

As quaternions represent a rotation operator, the same transformation parametrized by means of a rotation matrix  $\mathbf{R}$  can be expressed with quaternions. This is proven to be obtained by means of the Hamiltonian product operator, which provides the compact and elegant form

$$\begin{bmatrix} 0 \\ \mathbf{R}\mathbf{p} \end{bmatrix} = \mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{p} \end{bmatrix} \otimes \mathbf{q}^*, \quad (4.17)$$

which for unit quaternions and their property in Equation (4.16) can be rewritten as  $\mathbf{q} \otimes \begin{bmatrix} 0 \\ \mathbf{p} \end{bmatrix} \otimes \mathbf{q}^{-1}$ .

In a similar fashion to the composition of rotation matrices through multiplication, rotations parametrized as quaternions are composed together again by means of the Hamilton product which leads to

$$\begin{bmatrix} 0 \\ \mathbf{R}_1 \mathbf{R}_2 \mathbf{p} \end{bmatrix} = \mathbf{q}_1 \otimes \mathbf{q}_2 \otimes \begin{bmatrix} 0 \\ \mathbf{p} \end{bmatrix} \otimes \mathbf{q}_1^* \otimes \mathbf{q}_2^*. \quad (4.18)$$

Quaternions, and in particular unit quaternions, are widely adopted in robotics, for instance to represent the attitude of a robot end effector orientation [Xian, 2004]. They are employed to describe the vehicle attitude in the aerospace sector [Costic, 2001] and recently in underwater robotics [Grande, 2018], since a spacecraft or a glider might assume body attitudes that are singular if parametrized, for instance, with Euler angles. Their use is also popular in computer vision and



3D graphics for their computational efficiency, as they do not involve the use of trigonometric functions and their differential equations are linear [Rucker, 2018].

The interested reader can find more details and derivations about this representation in [Kuipers, 1999; Silva, 2002; Diebel, 2006; Girard, 2007; Solà, 2017].

### 4.3 Overview on rigid body dynamics

In this section, the *dynamics* of rigid bodies is discussed. With dynamics we refer to the relationship relating the motion of rigid bodies to the torques and forces (i.e. *wrenches*) originating such a motion [Goldstein, 1950]. In Classical Mechanics<sup>7</sup>, two are the main formalisms used to describe the dynamics of a rigid-body system, precisely the *Newtonian* and the *Lagrangian* formalisms [Kurfess, 2005]. It is important to underline that, while these approaches lead to the same outcome, the backbone ideas are quite dissimilar. Moreover, as discussed hereafter, they can provide different perspectives and insights on the model of the system and about its properties.

The *Lagrangian* approach offers an elegant and systematic way of expressing the dynamic model of rigid-body systems which is independent of the reference coordinate frame [Siciliano, 2009]. It comprises firstly the choice of a set of *generalized coordinates*  $\xi \in \mathbb{R}^n$ , i.e. the free variables which the system dynamics will be expressed with respect to. These coordinates, if properly chosen, allow to fully capture the configuration of the robot and its DoFs, and account at the same time for all the system constraints. First, the kinetic and potential energies are computed as function of the chosen coordinates and, next, the dynamic model of the system can be derived by applying the *Lagrange equations*. These equations establish the relations existing between the generic forces applied to the system and the free coordinates and derivatives [Siciliano, 2009].

This formalism is well-suited to identify the conserved quantities and the dissipative effects in the model, to derive insights on interesting properties of the system, and to deal in a systematic way with constraints [Kurfess, 2005]. Despite its elegant formulation, the method is revealed being unpractical and less computational efficient for complex systems with a large number of DoFs.

A preliminary version of *Newtonian* formalism was firstly published by Isaac Newton in 1687 for point-particle systems and later extended to rigid bodies by Leonhard Euler in 1736. This justifies why in the literature this formalism is commonly known as *Newton-Euler* [Beatty, 1986]. Differently from the Lagrangian formalism, it is particularly well-suited to compute the dynamic model of a multi-body system thanks to the existence of a *recursive* and efficient way to express this methodology. In the literature, this version of the Newton-Euler method is named *Recursive Newton-Euler Algorithm (RNEA)* [Siciliano, 2008].

In general, the Newton-Euler formalism considers the system dynamics in terms of the applied forces and torques, which are expressed w.r.t. a precise reference frame.

---

<sup>7</sup>Classical Mechanics is that branch of Physics studying the motion of macroscopic objects whose dimensions are not (sub-)atomic (field of interest of Quantum Mechanics) and speed is not approaching the speed of light (field of interest of Relativity) [Kibble, 2004].

In the case of multi-body systems, this formalism treat each rigid body independently and model the couplings, which are introduced by the different joint mechanisms, through the forces required to enforce them [Kurfess, 2005].

Given the existence of a recursive form, it is a very scalable method and particularly efficient when implemented as a software algorithm. Moreover, the Newton-Euler approach is systematic and of straightforward understanding, and it is revealed to be effective when considering complex effects, like deformations of flexible links. Nevertheless, it requires special attention and treatment in the case reaction forces arise in constrained systems, or for robots containing a closed-chain kinematics.

Both formalism lead to the same dynamic model for the considered system, as it will be outlined in Section 4.3.1 and Section 4.3.2. However, in this chapter, we will provide the derivation of the dynamic model of a Generically-Tilted Multi-Rotor (GTMR) (Section 4.4) an aerial manipulator (AM) (Section 4.5) solely by means of the Newton-Euler formalism. The motivation lies in aforementioned advantages in ease of application, computational efficiency and scalability of this approach when deriving the dynamic model for complex systems with a large number of DoFs, being the case for MRAVs equipped with robotic arms.

### 4.3.1 Lagrange formalism

As mentioned earlier, when adopting the Lagrange formalism, the first step consists in the definition of the generalized coordinates  $\boldsymbol{\xi} \in \mathbb{R}^{n_\xi}$  and their derivatives  $\dot{\boldsymbol{\xi}} \in \mathbb{R}^{n_\xi}$ , where  $n_\xi \in \mathbb{N}_+$ . Then, as second step, the *Lagrangian* functions can be defined as

$$\mathcal{L}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \mathcal{K}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) - \mathcal{U}(\boldsymbol{\xi}), \quad (4.19)$$

where  $\mathcal{K}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) : \mathbb{R}^{n_\xi \times n_\xi} \rightarrow \mathbb{R}^{n_\xi}$  and  $\mathcal{U}(\boldsymbol{\xi}) : \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}^{n_\xi}$  are the *kinetic* and the *potential* energy of the system [Siciliano, 2008]. Subsequently, the *Lagrange* equations can be computed, which in compact form are given by

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{\xi}}} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = \boldsymbol{\lambda}, \quad (4.20)$$

where  $\boldsymbol{\lambda} \in \mathbb{R}^{n_\xi}$  collects the generalized forces associated with the generalized coordinates  $\boldsymbol{\xi}$  [Siciliano, 2009]. Particularly, the contributions to these generalized forces comprise all the non-conservative effects. Thus, they include the forces and torques of the actuators, the friction forces originating from non-ideal behaviors of the mechanical parts, aerodynamic effects, and the external forces acting on the robot eventually arising from any interaction with the environment or other (robotic and human) agents. As an example, lets consider a set of  $n_f$  forces  $\mathbf{f} = [\mathbf{f}_1^\top \dots \mathbf{f}_{n_f}^\top]^\top \in \mathbb{R}^{3n_f}$ , where each individual force  $\mathbf{f}_i \in \mathbb{R}^3$  is applied to a generic point  $\mathbf{p}_i \in \mathbb{R}^3$  of the system, with  $i \in \{1, \dots, n_f\}$ . The generalized force component  $\lambda_j$  can be computed as follows

$$\lambda_j(\mathbf{f}, \xi_j) = \sum_{i=1}^{n_f} \mathbf{f}_i^\top \frac{\partial \mathbf{p}_i}{\partial \xi_j}, \quad j \in \{1, \dots, n_\xi\}. \quad (4.21)$$

Similarly, if a set of torques is additionally applied on the system, the generalized force coordinates would account also for those torques producing an effect on the considered generalized coordinate.

Substituting the terms in Equation (4.21) into Equation (4.20), and evaluating the system kinetic and potential energies as function of the generalized coordinates, the dynamic model of the rigid-body system can be obtained. For the mechanical systems considered in this manuscript, the kinetic energy can be proven to have a quadratic form of the type  $\mathcal{K}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \frac{1}{2} \dot{\boldsymbol{\xi}}^\top \mathbf{M}(\boldsymbol{\xi}) \dot{\boldsymbol{\xi}}$ , where  $\mathbf{M} \in \mathbb{R}_+^{n_\xi \times n_\xi}$  is the symmetric positive-definite configuration-dependent *generalized inertia matrix* which accounts for the inertial effects. Hence, it is possible to write the system dynamical model in the following compact form

$$\mathbf{M}(\boldsymbol{\xi}) \ddot{\boldsymbol{\xi}} + \mathbf{n}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \boldsymbol{\lambda}, \quad (4.22)$$

where  $\ddot{\boldsymbol{\xi}} \in \mathbb{R}^{n_\xi}$  are the second-order derivatives of the generalized coordinates (i.e. the accelerations), and the vector  $\mathbf{n} \in \mathbb{R}^{n_\xi}$  is the so-called *generalized bias force* comprising all the effects that require a torque for being compensated at zero accelerations. Thus, it accounts for the Coriolis and centrifugal forces, gravity and dissipative effects such as friction [Featherstone, 2008]. As the potential energy corresponds only to the gravitational potential [Siciliano, 2009], it is possible to have the gravity term appear explicitly in Equation (4.22) resulting in

$$\mathbf{M}(\boldsymbol{\xi}) \ddot{\boldsymbol{\xi}} + \mathbf{c}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) + \mathbf{g}(\boldsymbol{\xi}) = \boldsymbol{\lambda}, \quad (4.23)$$

where  $\mathbf{g}(\boldsymbol{\xi}), \mathbf{c}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \in \mathbb{R}^{n_\xi}$  denote, respectively, the gravitational and the Coriolis-centrifugal vectors. This last term can be partitioned into a matrix multiplying the system velocities as  $\mathbf{c}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \mathbf{C}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) \dot{\boldsymbol{\xi}}$ . However, the choice of the partitioned matrix  $\mathbf{C} \in \mathbb{R}^{n_\xi \times n_\xi}$  is not unique and several choices exist. One common choice can be obtained by means of the so-called *Christoffel symbols of the first type*. For more details, the interested reader is addressed to [Siciliano, 2009].

Equations (4.23) are named *Equation of Motion (EoM)*, or alternatively *Canonical equations* [Siciliano, 2008], and they establish the relations existing between the generalized coordinates (positional variables) and their derivatives (i.e. velocities and accelerations), and the generalized force contributions. However, sometimes it is more convenient to consider the system velocities expressed in Cartesian coordinates rather than the derivatives of the generalized coordinates. In such a case, the *twist* vector  $\boldsymbol{\zeta} \in \mathbb{R}^{n_\xi}$  is introduced to collect the velocities of the system and, in general,  $\dot{\boldsymbol{\xi}} \neq \boldsymbol{\zeta}$ , since a mapping  $\mathbf{T}(\boldsymbol{\xi})$  between the two exists which is function of  $\boldsymbol{\xi}$ . Hence, Equation (4.23) can be rewritten by using the twist vector as

$$\mathbf{M}(\boldsymbol{\xi}) \dot{\boldsymbol{\zeta}} + \mathbf{c}(\boldsymbol{\xi}, \dot{\boldsymbol{\zeta}}) + \mathbf{g}(\boldsymbol{\xi}) = \boldsymbol{\lambda}, \quad (4.24)$$

In Section 4.4.1 and Section 4.5.3, this mapping is shown, respectively, for the case under consideration of GTMRs and AMs, as their dynamic models are derived in the same form as Equation (4.24).

*Remark. Direct dynamics.* The *direct dynamics problem* consists in evaluating the robot motion, i.e. the time evolution of the generalized coordinates  $\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}, \ddot{\boldsymbol{\xi}}$  as a result of the given generalized forces  $\boldsymbol{\lambda}$  and the system initial conditions. This corresponds to simulate the robot given its model expressed as in Equation (4.22), since the output of any simulator are the system positions, velocities and accelerations, given as input the forces and torques applied.

*Remark. Inverse dynamics.* The *inverse dynamics problem*, instead, consists in the opposite case of the direct dynamics one. Therefore, given the time evolution of the system coordinates, the generalized forces required to generate the motion are computed. Consequently, this consists in the control problem, where the actuator commands are generated to produce a desired system behavior.

### 4.3.2 Newton-Euler formalism

The Newton-Euler formalism is based on two fundamental physical quantities, namely the linear and angular momenta. For a single rigid body, in the absence of an external wrench (forces and torques), the total momenta - i.e. the sum of the linear and angular ones - will not vary. Conversely, if a wrench is applied on the system, then the time evolution of that quantity, expressed in an inertial frame, is related to the net contribution of all forces and torques acting on the center of mass (CoM) of the system. In mathematical terms, we can write:

$$\frac{d}{dt} \left( m_{\text{CoM}} {}^W \mathbf{v}_{\text{CoM}} \right) = {}^W \mathbf{f}_{\text{CoM}}, \quad (4.25a)$$

$$\frac{d}{dt} \left( {}^W \mathbf{J}_{\text{CoM}} {}^W \boldsymbol{\omega}_{\text{CoM}} \right) = {}^W \boldsymbol{\tau}_{\text{CoM}}, \quad (4.25b)$$

where  ${}^W \mathbf{v}_{\text{CoM}}, {}^W \boldsymbol{\omega}_{\text{CoM}} \in \mathbb{R}^3$  are, respectively, the linear and angular velocities of the CoM expressed w.r.t. an inertial frame denoted with  $\mathcal{F}_W$ . The scalar  $m_{\text{CoM}} \in \mathbb{R}_+$  and the matrix  ${}^W \mathbf{J}_{\text{CoM}} \in \mathbb{R}_+^{3 \times 3}$  are the mass and the inertia computed w.r.t. a frame having its origin in the CoM of the rigid body, but expressed in  $\mathcal{F}_W$ . Lastly, the vectors  ${}^W \mathbf{f}_{\text{CoM}}, {}^W \boldsymbol{\tau}_{\text{CoM}}$  are the net forces and torques applied on the CoM of the body.

In the case of a system composed of multiple rigid bodies, the Newton formalism foresees to treat each individual body independently. Therefore, for each one, a set of equations for the linear and angular motions shall be written down, while accounting for the coupling interactions induced by the joints within the neighboring bodies. This process can be performed in a recursive way by employing the RNEA, which is composed of two main phases: a *forward recursion* and a *backward recursion*. In the former, the velocities and accelerations are propagated from the first rigid body to the last one composing the system. In doing so, the kinematic quantities of each body are computed from those of the previous one in the chain. In the backward recursion, instead, the forces and torques are back-propagated starting from the last rigid body, usually the end effector of the robot, to the first one, i.e. the fixed or moving base of the structure [Siciliano, 2008].

The equations obtained from the application of the RNEA are not in closed form, since the motion of each rigid body is coupled to the ones of the other bodies through the kinematic relationships of the velocities and accelerations [Siciliano, 2008]. However, by applying such an algorithm, it is possible to derive the dynamic model of the system as a whole, resulting in an expression identical to the one in Equation (4.22), which was obtained by means of the Lagrange formalism.

## 4.4 Generically-tilted multi-rotor systems

The robotic agent involved in the handover problem is an aerial robot whose possible designs have been presented in Chapter 1. Despite their mechanical complexity and actuation properties may largely differ, it is possible to model them by adopting a common formulation presented hereafter. We first derive the model for a simple flying aerial vehicle equipped with an arbitrary number of motor-propeller actuators arbitrarily distributed w.r.t. to the body. This category of vehicles falls under the umbrella of the *Generically-Tilted Multi-Rotor* formulation presented in [Bicego, 2020; Jacquet, 2021] and adopted in the following. Once the main results are shown for a simpler AR design, in Section 4.5, we provide the model for a more complex flying robot, namely an aerial manipulator, being a GTMR equipped with a robotic arm.

### 4.4.1 Equations of motion

A GTMR is modeled as a rigid body of mass  $m_b \in \mathbb{R}_+$  and inertia  $\mathbf{J}_b \in \mathbb{R}_+^{3 \times 3}$ . Besides, it is actuated by  $n_p \in \mathbb{N}_+$  motor-propeller actuators, arbitrarily placed and oriented w.r.t. its main body.

As shown in Figure 4.1, we define a world inertial frame as  $\mathcal{F}_W$ , with its origin  $O_W$  and its axes  $\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W$  arbitrarily oriented. We introduce then a body frame  $\mathcal{F}_B$  whose origin  $O_B$  attached to the AR, in a point that usually and conveniently coincides with its geometric center. Differently from the common assumption to have the CoM of the system being coincident with  $O_B$  [Michieletto, 2018; Bicego, 2020], we assume it is located at a generic position  ${}^B\mathbf{p}_{\text{CoM}} \in \mathbb{R}^3$  w.r.t.  $O_B \in SO(3)$ .

Then, we define an actuator frame  $\mathcal{F}_{A_i}$  for each motor-propeller pair  $i \in \{1, \dots, n_p\}$ , having  $\mathbf{z}_{A_i}$  coincident with the motor axis. Therefore, their spatial arrangement w.r.t. the main body is given by  ${}^B\mathbf{p}_{A_i} \in \mathbb{R}^3$  and  ${}^B\mathbf{R}_{A_i}$ . Precisely, the several existing configurations of motor-axis orientations, which generate the wide spectrum of MRAV designs, can be represented with a pair of two angles:  $\alpha_i \in \mathbb{R}$  and  $\beta_i \in \mathbb{R}$ , being respectively the radial and tangential directions of each actuator. As a result, we can write  ${}^B\mathbf{R}_{A_i} = {}^B\mathbf{R}_{A_i}(\alpha_i, \beta_i)$ .

We then denote with  ${}^W\mathbf{p}_B \in \mathbb{R}^3$  the position of  $O_B$  w.r.t.  $\mathcal{F}_W$ , and with  ${}^W\mathbf{R}_B \in SO(3)$  the orientation of the rigid body expressed w.r.t.  $\mathcal{F}_W$ . Therefore, the orientation parametrizations by means of unit quaternions is denoted with  ${}^W\mathbf{q}_B \in \mathbb{R}^4$ , while the one through Euler-Angles with  ${}^W\boldsymbol{\eta}_B \in \mathbb{R}^3$ . We then indicate with  ${}^W\mathbf{v}_B \in \mathbb{R}^3$  the linear speed of  $O_B$  expressed in  $\mathcal{F}_W$ , and with  ${}^B\boldsymbol{\omega}_B \in \mathbb{R}^3$  the angular velocity of  $\mathcal{F}_B$  w.r.t.  $\mathcal{F}_W$  specified in  $\mathcal{F}_B$ .

*Remark. Simplicity of notation.* Hereafter, the reference frame is omitted for any vector expressed w.r.t. the inertial frame  $\mathcal{F}_W$ , and for any rotation expressing the orientation of a frame w.r.t. to  $\mathcal{F}_W$ .

The EoMs of a GTMR can be derived by exploiting the Newton-Euler formalism

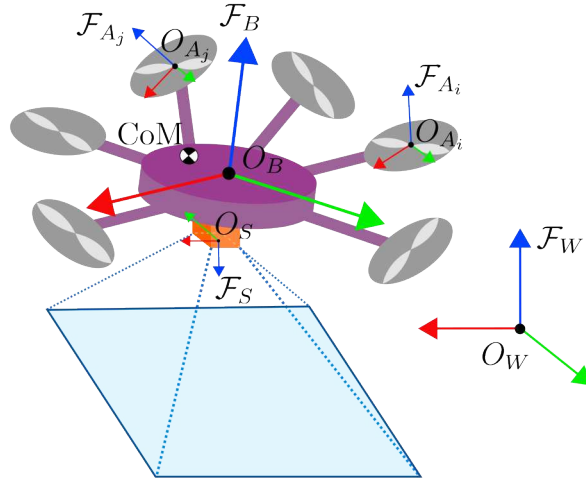


Figure 4.1: Here, a schematic representation of a Generically-Tilted Multi-Rotor (GTMR) is provided as well as the depiction of the main frames used to describe its motion. The  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{z}$  of each frame are respectively drawn using the RGB convention.

introduced in Section 4.3.2, which provide the following set of equations:

$$\dot{\mathbf{p}}_B = \mathbf{v}_B, \quad (4.26a)$$

$$\dot{\mathbf{q}}_B = \frac{1}{2} \mathbf{q}_B \otimes \begin{bmatrix} 0 \\ {}^B \boldsymbol{\omega}_B \end{bmatrix}, \quad (4.26b)$$

$$m_b \ddot{\mathbf{p}}_B = -m_b g \mathbf{z}_W + \mathbf{R}_B ({}^B \mathbf{f}_{\text{act}} + {}^B \mathbf{f}_{\text{ext}}) \quad (4.26c)$$

$$\mathbf{J}_b {}^B \dot{\boldsymbol{\omega}}_B = -\mathbf{S}({}^B \boldsymbol{\omega}_B) \mathbf{J}_b {}^B \boldsymbol{\omega}_B - \mathbf{S}({}^B \mathbf{p}_{\text{CoM}}) \mathbf{R}_B^\top m_b g \mathbf{z}_W + {}^B \boldsymbol{\tau}_{\text{act}} + {}^B \boldsymbol{\tau}_{\text{ext}}, \quad (4.26d)$$

where the scalar  $g = 9.81 \text{ms}^{-1}$  is the intensity of the gravity acceleration. The pair  ${}^B \mathbf{f}_{\text{act}}, {}^B \boldsymbol{\tau}_{\text{act}} \in \mathbb{R}^3$  denote the wrench applied by the actuators on the AR and expressed in  $\mathcal{F}_B$ . It will be further detailed in Section 4.6.1. The pair  ${}^B \mathbf{f}_{\text{ext}}, {}^B \boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^3$ , instead, is the net contribution of the forces and torques exchanged between the AR and the external environment, e.g. an obstacle, and with another (robotic or human) agent. Therefore, given  $n_e \in \mathbb{N}$  contact locations belonging to the robot airframe, the net contribution of the external wrenches can be written as

$${}^B \mathbf{w}_{\text{ext}} = \begin{bmatrix} {}^B \mathbf{f}_{\text{ext}} \\ {}^B \boldsymbol{\tau}_{\text{ext}} \end{bmatrix} = \sum_{e=1}^{n_e} \begin{bmatrix} {}^B \mathbf{R}_{E,e} & \mathbf{O}_3 \\ \mathbf{S}({}^B \mathbf{p}_{E,e}) & {}^B \mathbf{R}_{E,e} \end{bmatrix} {}^E \mathbf{w}_{E,e}, \quad (4.27)$$

where  ${}^E \mathbf{w}_{E,e} \in \mathbb{R}^6$  denotes an individual contribution applied to the contact location  $e \in \{1, \dots, n_e\}$  [Siciliano, 2009]. Each term is expressed in a local frame  $\mathcal{F}_E$  having its origin  $O_E$  in each contact point and roto-translated by  ${}^B \mathbf{p}_{E,e} \in \mathbb{R}^3$ ,  ${}^B \mathbf{R}_{E,e} \in SO(3)$  w.r.t.  $\mathcal{F}_B$ . As discussed in Chapter 1, multi-rotor ARs are usually equipped with an end effector rigidly mounted on the main body for performing tasks requiring physical interaction with the external environment, thus resulting in  $n_e = 1$  and  $\mathcal{F}_E$  being the EE frame.

Equation (4.26b) is the time derivation of a unit quaternion, which is related to the angular velocity of the rigid body  ${}^B \boldsymbol{\omega}_B \in \mathbb{R}^3$ . A proof of such a formula is provided in [Solà, 2017]. Moreover, the same equation can be written in the

alternative form

$$\dot{\mathbf{q}}_B = \frac{1}{2} \begin{bmatrix} 0 \\ {}^W\boldsymbol{\omega}_B \end{bmatrix} \otimes \mathbf{q}_B, \quad (4.28)$$

with the body angular rates expressed w.r.t.  $\mathcal{F}_W$ .

*Remark. Reference frame.* The translational part of the system dynamics in Equation (4.26) is expressed in the inertial frame  $\mathcal{F}_W$ , while the rotational part in  $\mathcal{F}_B$ . The motivation behind this choice lies in the fact that the inertia matrix would be configuration-dependent otherwise, if it was expressed w.r.t.  $\mathcal{F}_W$ . Therefore, as quaternion dynamics, we adopt Equation (4.26b). Consequently, the state vector, which is defined later in Section 4.7, does not contain the body angular rates expressed in world frame but rather in  $\mathcal{F}_B$ , i.e.  ${}^B\boldsymbol{\omega}_B$ .

By rearranging and collecting the terms in Equations (4.26c) and (4.26d), it is possible to write the GTMR dynamics in the same form as of Equation (4.24). As previously carried out in Section 4.3.1, we first define the generalized coordinates and the twist for the GTMR. The former is denoted with  $\boldsymbol{\xi}_{\text{GTMR}}$  and it collects the position and orientation of the robot expressed in  $\mathcal{F}_W$ , thus  $\boldsymbol{\xi}_{\text{GTMR}} = [\mathbf{p}_B^\top \mathbf{q}_B^\top]^\top \in \mathbb{R}^7$ . The latter, instead, gathers the velocities of the aerial robot in Cartesian space [Doty, 1993] and it is defined as  $\boldsymbol{\zeta}_{\text{GTMR}} = [\mathbf{v}_B^\top {}^B\boldsymbol{\omega}_B^\top]^\top \in \mathbb{R}^6$ , where  ${}^B\boldsymbol{\omega}_B$  is related to  $\dot{\mathbf{q}}_B$  by means of Equation (4.26b). At this point, we can conclude that

$$\underbrace{\begin{bmatrix} m_b \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{J}_b \end{bmatrix}}_{\mathbf{M}_{\text{GTMR}}(\boldsymbol{\xi}_{\text{GTMR}}) \in \mathbb{R}^{6 \times 6}} \underbrace{\begin{bmatrix} \ddot{\mathbf{p}}_B \\ {}^B\dot{\boldsymbol{\omega}}_B \end{bmatrix}}_{\dot{\boldsymbol{\zeta}}_{\text{GTMR}} \in \mathbb{R}^6} + \underbrace{\begin{bmatrix} \mathbf{o}_3 \\ \mathbf{S}({}^B\boldsymbol{\omega}_B) \mathbf{J}_b {}^B\boldsymbol{\omega}_B \end{bmatrix}}_{\mathbf{c}_{\text{GTMR}}(\boldsymbol{\xi}_{\text{GTMR}}, \boldsymbol{\zeta}_{\text{GTMR}}) \in \mathbb{R}^6} \\ + \underbrace{\begin{bmatrix} m_b g \mathbf{z}_W \\ \mathbf{S}({}^B\mathbf{p}_{\text{CoM}}) \mathbf{R}_B^\top m_b g \mathbf{z}_W \end{bmatrix}}_{\mathbf{g}_{\text{GTMR}}(\boldsymbol{\xi}_{\text{GTMR}}) \in \mathbb{R}^6} = \underbrace{\begin{bmatrix} \mathbf{R}_B & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{f}_{\text{act}} + {}^B\mathbf{f}_{\text{ext}} \\ {}^B\boldsymbol{\tau}_{\text{act}} + {}^B\boldsymbol{\tau}_{\text{ext}} \end{bmatrix}}_{\boldsymbol{\lambda}_{\text{GTMR}} \in \mathbb{R}^6}, \quad (4.29)$$

where  $\dot{\boldsymbol{\zeta}}_{\text{GTMR}} = [\dot{\mathbf{v}}_B^\top {}^B\dot{\boldsymbol{\omega}}_B^\top]^\top \in \mathbb{R}^6$  is the derivative of the twist vector  $\boldsymbol{\zeta}_{\text{GTMR}}$ , hence it contains the Cartesian accelerations of the aerial robot. The canonical form of the GTMR dynamic model is as follows

$$\mathbf{M}(\boldsymbol{\xi}) \dot{\boldsymbol{\zeta}} + \mathbf{c}(\boldsymbol{\xi}, \boldsymbol{\zeta}) + \mathbf{g}(\boldsymbol{\xi}) = \boldsymbol{\lambda}, \quad (4.30)$$

where we have removed the subscript ‘‘GTMR’’ to lighten the notation.

#### 4.4.2 Center of mass

A CoM location not coincident with the geometric center (i.e.  ${}^B\mathbf{p}_{\text{CoM}} \neq 0$ ) induces a non-zero torque in the rotational dynamics of the system, as noticeable in Equation (4.26b). Moreover, it affects also the computation of the inertia matrix  $\mathbf{J}_b$  of the platform. Specifically, in Equation (4.26d) the notation  ${}^B\mathbf{J}_b$  should be used in order to explicitly indicate that the inertia is expressed w.r.t.  $\mathcal{F}_B$ <sup>8</sup> Therefore any inertia matrix relative to a pole  $O$  expressed w.r.t. a generic frame  $\mathcal{F}_A$  is denoted

<sup>8</sup>Through this manuscript we will use the same notation as in [Siciliano, 2009].

as  ${}^A\mathbf{J}_O$ . Therefore, it is possible to account for a non-zero  ${}^B\mathbf{p}_{\text{CoM}}$  by means of the *Steiner's theorem* or *parallel axis theorem* [Siciliano, 2009] as follows

$${}^B\mathbf{J}_b = {}^B\mathbf{R}_{\text{CoM}} {}^{\text{CoM}}\mathbf{J}_b {}^B\mathbf{R}_{\text{CoM}}^\top + \mathbf{S}({}^B\mathbf{p}_{\text{CoM}})^\top \mathbf{S}({}^B\mathbf{p}_{\text{CoM}}), \quad (4.31)$$

where conveniently it is assumed  ${}^B\mathbf{R}_{\text{CoM}} = \mathbf{I}_3$ , being the choice of those coordinate axes arbitrary. In Equation (4.31),  ${}^{\text{CoM}}\mathbf{J}_b$  represents the inertia of the AR computed w.r.t. the CoM. The value of this parameter is computed by integrating the mass properties on the volume of each object composing the mechanical structure of the robot. This process nowadays is automatized by Computer Aided Design (CAD) software. Nevertheless, precise values of  ${}^B\mathbf{p}_{\text{CoM}}$  and  ${}^{\text{CoM}}\mathbf{J}_b$  are hard to obtain, as a minimal change in the robot configuration, such as the addition of a sensor or the displacement of an hardware component, will affect these parameters.

Knowing the CoM location has been shown to improve the tracking performances of a given controller, thanks to the exploitation of a more precise knowledge of the underlying robot physics [Mellinger, 2011; Jacquet, 2022b]. As a first guess, such a parameter can be set to zero for preliminary experimental tests. However, simple procedures can be designed to compute a rough estimate and consequently obtain better results, as the one proposed in [Jacquet, 2022b].

### 4.4.3 Model assumptions

The dynamic model in Equations (4.26) relies on some assumptions, as it neglects some high-order effects which are instead accounted by adopting more sophisticated and complex models [Hanover, 2023]. To name a few, the model does not take into consideration (i) the gyroscopic effect induced by the conservation of the angular momentum of the propellers, which have a small but non-zero mass and inertia; (ii) the blade flapping phenomenon induced by the non-rigidity of the propellers; (iii) body drag and lift which, for the former, may be observed at very high-speed maneuvers, while for the latter in windy environments for particular robot designs offering specific surfaces that can produce lift if the relative wind speed is large; (iv) ground and ceiling effects which produce additional disturbances on the robot airframe. These effects are usually negligible compared to the main body dynamics, as they can appear in particular conditions like during high-speed translational maneuvers or in specific environmental settings, e.g. windy environments. In the case of small-scale aerial robots and the tasks considered through this thesis, it is particularly safe neglecting these effects. The reduced dimensions of the propeller blades introduce minor flapping phenomena and small ground and ceiling effects appear when interacting with the surrounding environment. On the other side, the inertial parameters of some components, like those of the rotors and propellers, play a bigger role if the scale of the vehicle increases [Pounds, 2010]. Analogously aerodynamic effects cannot be neglected when performing maneuvers involving very large speeds and accelerations [Salzmann, 2022]. Similarly, wind gusts and disturbances with the nearby objects in the environment shall be considered if the action of such a disturbance plays a noticeable effect on the robot dynamics. Usually, and especially in the design phase of a control approach, the model should be capable of taking into account the most important part of the phenomena under investigation



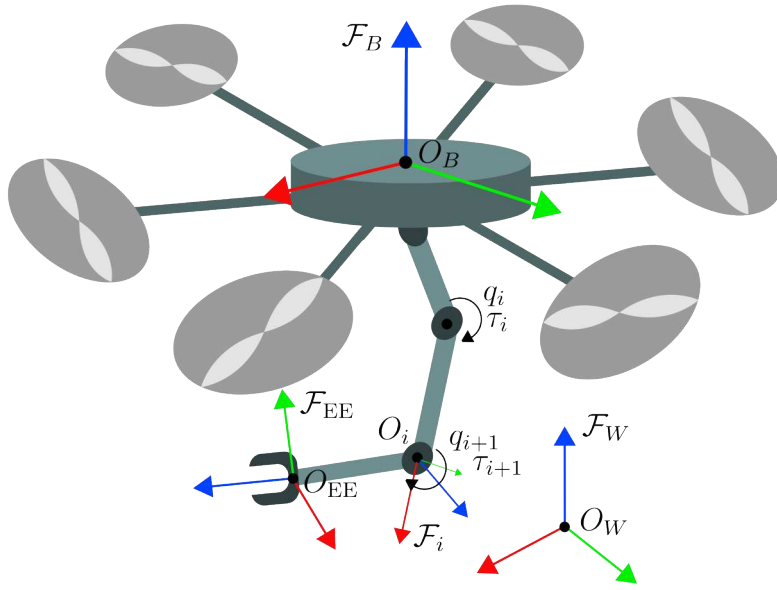


Figure 4.2: This figure presents a schematic drawing of an aerial manipulator, being a composition of a GTMR and a robotic arm.

without requiring an overly complex mathematical description for the class of robots and applications considered.

## 4.5 Aerial manipulator systems

After presenting the model used to describe the dynamics and the motion of a GTMR aerial robot, in this section we extend the formulation to account also for an additional robotic arm mounted below. This extra device enhances the AR with additional DoFs introducing more dexterity, which simplifies the execution of tasks requiring physical interaction, and increasing the level of redundancy in the system. As mentioned in Chapter 1, in this work we consider only aerial robots equipped with serial manipulators, while parallel and dual-arm systems are not considered.

### 4.5.1 A flying base and a robotic arm

An aerial manipulator can be decomposed into two subsystems: a *flying base* and a *robotic arm*. A schematic representation is provided in Figure 4.2. The flying base is the GTMR modeled in Section 4.4. An open-chain serial manipulator is mounted which introduces  $n_q \in \mathbb{N}_+$  DoFs to system. This manipulator is composed by  $n_q$  links interconnected to the flying base and within each other by  $n_q$  joints, either revolute or prismatic. We collect the joint variables in the vector  $\mathbf{q}_a \in \mathbb{R}^{n_q}$ , and we denote its derivatives as  $\dot{\mathbf{q}}_a, \ddot{\mathbf{q}}_a$ . They gather the joint angles and displacements of the revolute and prismatic joints, respectively.

We define a frame attached to each link and we denote it as  $\mathcal{F}_i = O_i, \{\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}, i \in \{1, \dots, n_q\}$ , with  $\mathbf{z}_i$  parallel to the joint axis. The frames are defined according to the *Denavit-Hartenberg (DH) convention*, explained in detail in [Siciliano, 2009]. We

assume that the mass of each link is distributed uniformly and we denote the CoM position of each link with  ${}^i\mathbf{p}_{\text{CoM},i} \in \mathbb{R}^3$ , which is expressed w.r.t. to frame attached to the  $i$ -th link. The inertial properties of the  $i$ -th link are denoted as  $m_i \in \mathbb{R}_+$  and  ${}^i\mathbf{J}_i \in \mathbb{R}^{3 \times 3}$ . Lastly, we denote the position of  $O_i$  w.r.t.  $\mathcal{F}_{i-1}$  as  ${}^{i-1}\mathbf{p}_i \in \mathbb{R}^3$ , and its linear and angular velocities as  ${}^{i-1}\dot{\mathbf{p}}_i, {}^{i-1}\boldsymbol{\omega}_i \in \mathbb{R}^3$ , respectively. Similar notation will be adopted for the linear and angular accelerations of each link.

To derive the EoMs for an AM, we rely on the Newton-Euler formulation and particularly on its recursive approach, i.e. the RNEA mentioned in Section 4.3.2. Therefore, we will detail the mathematical steps of its forward and backward passes, and then how to use them to derive the model for an aerial manipulator.

## 4.5.2 Recursive Newton-Euler algorithm

In the following, we will draw the main results for an AR equipped with a manipulator owing only revolute DoFs. In case some prismatic joints are present, similar results can be obtained. The interested reader is addressed to [Siciliano, 2009] for a full derivation and proofs.

In the forward recursion of the RNEA, the velocities and accelerations of each link  $i \in \{1, \dots, n_q\}$  are computed starting from those of the GTMR base ( $i = 0$ ). Therefore, we write the following initial conditions

$${}^0\ddot{\mathbf{p}}_0 - {}^0\mathbf{g}_0 = {}^W\mathbf{R}_B{}^\top ({}^W\ddot{\mathbf{p}}_B - g\mathbf{z}_W), \quad {}^0\boldsymbol{\omega}_0 = {}^B\boldsymbol{\omega}_B, \quad {}^0\dot{\boldsymbol{\omega}}_0 = {}^B\dot{\boldsymbol{\omega}}_B, \quad (4.32)$$

where the linear accelerations of the base account also for gravity. Consequently, the kinematic quantities of the manipulator links can be obtained using the following equations

$${}^i\boldsymbol{\omega}_i = {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\boldsymbol{\omega}_{i-1} + \dot{q}_i\mathbf{z}_i), \quad (4.33a)$$

$${}^i\dot{\boldsymbol{\omega}}_i = {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\dot{\boldsymbol{\omega}}_{i-1} + \ddot{q}_i\mathbf{z}_i + \dot{q}_i\mathbf{S}({}^{i-1}\boldsymbol{\omega}_{i-1})\mathbf{z}_i), \quad (4.33b)$$

$${}^i\ddot{\mathbf{p}}_i = {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\ddot{\mathbf{p}}_{i-1} + \mathbf{S}({}^i\dot{\boldsymbol{\omega}}_i){}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\mathbf{p}_i + \mathbf{S}({}^i\boldsymbol{\omega}_i)({}^i\mathbf{p}_i)) + \mathbf{S}({}^i\boldsymbol{\omega}_i)({}^i\mathbf{p}_i)), \quad (4.33c)$$

$${}^i\ddot{\mathbf{p}}_{\text{CoM},i} = {}^i\ddot{\mathbf{p}}_i + \mathbf{S}({}^i\dot{\boldsymbol{\omega}}_i){}^i\mathbf{p}_{\text{CoM},i} + \mathbf{S}({}^i\boldsymbol{\omega}_i)({}^i\mathbf{p}_{\text{CoM},i} + {}^i\mathbf{p}_{\text{CoM},i}), \quad (4.33d)$$

given that

$$\begin{aligned} {}^i\ddot{\mathbf{p}}_i &= {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\ddot{\mathbf{p}}_i), & {}^i\dot{\mathbf{p}}_i &= {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\dot{\mathbf{p}}_i), & {}^i\mathbf{p}_i &= {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\mathbf{p}_i), \\ {}^i\dot{\boldsymbol{\omega}}_i &= {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\dot{\boldsymbol{\omega}}_i), & {}^i\boldsymbol{\omega}_i &= {}^{i-1}\mathbf{R}_i{}^\top ({}^{i-1}\boldsymbol{\omega}_i), \end{aligned} \quad (4.34)$$

It is worth it to point out that each variable is expressed w.r.t. the current  $i$ -th link and as function of the velocities and accelerations of the previous link.

Having computed the velocities and accelerations from the base link to the end effector with the forward recursion, a backward step is carried out for the forces and moments. Given the wrench applied to the end effector (i.e. the last link of the manipulator chain), the force and torques applied to each link, respectively  $\mathbf{f}_i \in \mathbb{R}^3$

and  $\boldsymbol{\tau}_i \in \mathbb{R}^3$ , can be computed as

$${}^i\mathbf{f}_i = {}^i\mathbf{R}_{i+1} {}^{i+1}\mathbf{f}_{i+1} + m_i {}^i\ddot{\mathbf{p}}_{\text{CoM},i}, \quad (4.35a)$$

$$\begin{aligned} {}^i\boldsymbol{\tau}_i = & {}^i\mathbf{R}_{i+1} {}^{i+1}\boldsymbol{\tau}_{i+1} - \mathbf{S}({}^i\mathbf{f}_i) ({}^i\mathbf{p}_i + {}^i\mathbf{p}_{\text{CoM},i}) \\ & + {}^i\mathbf{R}_{i+1} \mathbf{S}({}^{i+1}\mathbf{f}_{i+1}) {}^i\mathbf{p}_{\text{CoM},i} + {}^i\mathbf{J}_i {}^i\dot{\boldsymbol{\omega}}_i + \mathbf{S}({}^i\boldsymbol{\omega}_i) ({}^i\mathbf{J}_i {}^i\boldsymbol{\omega}_i). \end{aligned} \quad (4.35b)$$

Then, the generalized force associated to the  $i$ -th joint is obtained as follows

$$\lambda_i = {}^i\boldsymbol{\tau}_i^\top {}^{i-1}\mathbf{R}_i^\top \mathbf{z}_{i-1}, \quad (4.36)$$

where for the EE link, i.e.  $i = n_q + 1$ , the forces and torques are given by the wrench  $\mathbf{w}_E \in \mathbb{R}^6$  as

$$\mathbf{w}_E = \begin{bmatrix} \mathbf{f}_E^\top & \boldsymbol{\tau}_E^\top \end{bmatrix}^\top = \begin{bmatrix} \mathbf{f}_{n_q+1}^\top & \boldsymbol{\tau}_{n_q+1}^\top \end{bmatrix}^\top. \quad (4.37)$$

By means of the equations above, the generalized forces  $\boldsymbol{\lambda}_a = [\lambda_1 \dots \lambda_{n_q}]^\top$  applied to the links can be computed. Those instead related to the base, i.e.  $\boldsymbol{\lambda}_b$ , correspond to the right-handed side of Equation (4.29). Therefore, we rewrite here for convenience that

$$\boldsymbol{\lambda}_b = \begin{bmatrix} {}^W\mathbf{f}_{\text{act}}^\top & {}^B\boldsymbol{\tau}_{\text{act}}^\top \end{bmatrix}^\top \in \mathbb{R}^6, \quad (4.38)$$

$$\boldsymbol{\lambda}_a = [\lambda_1 \quad \dots \quad \lambda_{n_q}]^\top \in \mathbb{R}^{n_q}, \quad (4.39)$$

where in the first equation we neglect the presence of external forces and torques applied to the robot flying base. If instead an external wrench is applied to the main body, we need to include in  $\boldsymbol{\lambda}_b$  also the terms  ${}^W\mathbf{f}_{\text{ext}} \in \mathbb{R}^3$  and  ${}^B\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^3$ .

Next, we can define the generalized coordinates of the aerial manipulator which would intuitively be a combination of those of the aerial base, denoted as  $\boldsymbol{\xi}_b$  and corresponding to  $\boldsymbol{\xi}_{\text{GTMR}} \in \mathbb{R}^7$  of Section 4.4.1, and those for the robotic arm denoted as  $\boldsymbol{\xi}_a$ . Therefore, recalling the definitions in Equation (4.29), we can conclude

$$\boldsymbol{\xi}_{\text{AM}} = \begin{bmatrix} \boldsymbol{\xi}_b^\top & \boldsymbol{\xi}_a^\top \end{bmatrix}^\top = \begin{bmatrix} \mathbf{p}_B^\top & \mathbf{q}_B^\top & \mathbf{q}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{7+n_q}. \quad (4.40)$$

We draw similar definitions for the twist and its time derivative, namely

$$\boldsymbol{\zeta}_{\text{AM}} = \begin{bmatrix} \boldsymbol{\zeta}_b^\top & \boldsymbol{\zeta}_a^\top \end{bmatrix}^\top = \begin{bmatrix} \mathbf{v}_B^\top & {}^B\boldsymbol{\omega}_B^\top & \dot{\mathbf{q}}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{6+n_q}, \quad (4.41)$$

$$\dot{\boldsymbol{\zeta}}_{\text{AM}} = \begin{bmatrix} \dot{\boldsymbol{\zeta}}_b^\top & \dot{\boldsymbol{\zeta}}_a^\top \end{bmatrix}^\top = \begin{bmatrix} \dot{\mathbf{v}}_B^\top & {}^B\dot{\boldsymbol{\omega}}_B^\top & \ddot{\mathbf{q}}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{6+n_q}, \quad (4.42)$$

where  $\boldsymbol{\zeta}_b \in \mathbb{R}^6$  and  $\boldsymbol{\zeta}_a = \dot{\boldsymbol{\xi}}_a \in \mathbb{R}^{n_q}$  are the twist vectors for the aerial base and the manipulator, while  $\dot{\boldsymbol{\zeta}}_b \in \mathbb{R}^6$  and  $\dot{\boldsymbol{\zeta}}_a = \ddot{\boldsymbol{\xi}}_a \in \mathbb{R}^{n_q}$  their time derivatives, respectively.

It is possible to gather together Equations (4.33a)-(4.36) into a generic routine function. It takes as input the initial and terminal conditions given by Equations (4.32) and (4.37), the generalized coordinates  $\boldsymbol{\xi}_{\text{AM}}$  of the AM, the twist  $\boldsymbol{\zeta}_{\text{AM}}$ , and its derivative  $\dot{\boldsymbol{\zeta}}_{\text{AM}}$ . As output, this function provides the generalized forces applied to the whole system, i.e.  $\boldsymbol{\lambda}_{\text{AM}} \in \mathbb{R}^{6+n_q}$ , and it can be denoted as

$$\boldsymbol{\lambda}_{\text{AM}} = \begin{bmatrix} \boldsymbol{\lambda}_b^\top & \boldsymbol{\lambda}_a^\top \end{bmatrix}^\top = \text{RNEA} \left( \boldsymbol{\xi}_{\text{AM}}, \boldsymbol{\zeta}_{\text{AM}}, \dot{\boldsymbol{\zeta}}_{\text{AM}} \right). \quad (4.43)$$

It is important to point out that the function in Equation (4.43) is tailored to the structure of the particular robot considered, as the DH convention is function of the geometrical properties of the system. Additionally, the algorithm requires also the inertial parameters of the AM, like link masses and inertias. By querying the routine in Equation (4.43) for different input combinations, it is possible to obtain the system matrices allowing to write the dynamic model in the same form as the one in Equation (4.24). Precisely, it can be proven that

$$\mathbf{g}_{\text{AM}}(\boldsymbol{\xi}) = \text{RNEA}_g(\boldsymbol{\xi}, \mathbf{o}_{6+n_q}, \mathbf{o}_{6+n_q}) \in \mathbb{R}^{(6+n_q)}, \quad (4.44a)$$

$$\mathbf{c}_{\text{AM}}(\boldsymbol{\xi}, \boldsymbol{\zeta}) = \text{RNEA}_0(\boldsymbol{\xi}, \boldsymbol{\zeta}, \mathbf{o}_{6+n_q}) \in \mathbb{R}^{(6+n_q)}, \quad (4.44b)$$

$$\mathbf{M}_{\text{AM},i}(\boldsymbol{\xi}) = \text{RNEA}_0(\boldsymbol{\xi}, \mathbf{o}_{6+n_q}, \mathbf{e}_i) \in \mathbb{R}^{(6+n_q)}, \quad (4.44c)$$

where  $\mathbf{M}_{\text{AM},i}$  is the  $i$ -th column of the generalized inertia matrix of the aerial manipulator, and  $\mathbf{e}_i \in \mathbb{R}^{6+n_q}$  is the  $i$ -th column of an identity matrix  $\mathbf{I}_{6+n_q}$ . It is important to mention that the subscript 0 and  $g$  denote the application of the algorithm in Equation (4.43), respectively, with and without the gravity in the initial condition given by Equation (4.32).

### 4.5.3 Equations of motion

Similarly to Equation (4.30), we arrive to the canonical form of the AM dynamic model by collecting the previous terms. If the EE of the aerial manipulator is in contact with the environment, a portion of the actuation effort is used to balance the wrench arising at the contact location. According to [Siciliano, 2009], such a contribution is given by  $\mathbf{J}_{\text{AM}}^\top {}^E \mathbf{w}_E$ , where  $\mathbf{J}_{\text{AM}} \in \mathbb{R}^{6 \times (6+n_q)}$  is the *geometric Jacobian* of the aerial manipulator mapping a wrench at the contact location to the robot generalized coordinates. That quantity is function of the robot configuration  $\boldsymbol{\xi}_{\text{AM}}$ . The wrench  ${}^E \mathbf{w}_E = [{}^E \mathbf{f}_E^\top \ {}^E \boldsymbol{\tau}_E^\top]^\top \in \mathbb{R}^6$  is the external wrench arising at the contact location during the interaction. It is expressed in the frame  $\mathcal{F}_E$ , whose origin is located at the end effector tip. Finally, assembling all terms together, we obtain the following dynamic model for the AM:

$$\mathbf{M}(\boldsymbol{\xi}) \dot{\boldsymbol{\zeta}} + \mathbf{c}(\boldsymbol{\xi}, \boldsymbol{\zeta}) + \mathbf{g}(\boldsymbol{\xi}) = \boldsymbol{\lambda} + \mathbf{J}(\boldsymbol{\xi})^\top {}^E \mathbf{w}_E, \quad (4.45)$$

where we have dropped again the subscript “AM” to simplify the notation.

## 4.6 Actuators of aerial robots

As remarked in [Bicego, 2019], the dynamic equations (4.30) and (4.45), can be applied to a broader class of robots than the ones considered so far. As a matter of fact, those equations describe the dynamics of any *floating-base* rigid body [Featherstone, 2008], whose base is not attached to the ground, as in the case of fixed-based manipulators, and optionally in contact with the external environment in a finite set of contact locations, e.g. AMs during physical interaction [Tognon, 2019] or bipedal robots during locomotion [Nakanishi, 2007]. Hereafter, we are going to refine that set of equations to the case of multi-rotor ARs equipped with rotary wings and a robotic

arm. This will be performed by making explicit and describing how the generalized forces  $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_b^\top \boldsymbol{\lambda}_a^\top]^\top \in \mathbb{R}^{6+n_q}$  are physically generated by their actuators, which are the true and active responsible of the robot dynamics.

### 4.6.1 Motor-propeller actuation units

First of all, any GTMR and the flying base of any AM is equipped with  $n_p \in \mathbb{R}_+$  actuation units. Each actuator is composed of one motor and one propeller spinning coaxially at the same speed and same direction [Hamandi, 2021]. Therefore, if we consider the two most widely adopted platforms, an Uni-directional Thrust (UDT) quad-rotor and a Multi-Directional Thrust (MDT) hexa-rotor own  $n_p = 4$  collinear and  $n_p = 6$  tilted motor-propeller pairs, respectively. Each motor-propeller unit produces a *lift force* with intensity  $\gamma_i \in \mathbb{R}$  along the axis of the paired motor and coincident with  $\mathbf{z}_{A_i}, i \in \{1, \dots, n_p\}$  of  $\mathcal{F}_{A_i}$ . Additionally, when spinning, each propeller generates also a torque  ${}^B\boldsymbol{\tau}_{\gamma_i} \in \mathbb{R}^3$  on the body as by-product of the produced force and the arm leverage  ${}^B\mathbf{p}_{A_i} \in \mathbb{R}^3$  (i.e. the position of  $O_{A_i}$  w.r.t.  $\mathcal{F}_B$ ). Moreover, a *drag torque*  ${}^B\boldsymbol{\tau}_{d_i} \in \mathbb{R}^3$  on the main body is induced by the interaction between the rotating propeller and the air. The sum of all these forces and torques, properly expressed w.r.t.  $\mathcal{F}_B$ , constitute the body wrench applied on the robot airframe. Therefore, we can write that

$$\begin{aligned} {}^B\mathbf{w}_{\text{act}} &= \begin{bmatrix} {}^B\mathbf{f}_{\text{act}} \\ {}^B\boldsymbol{\tau}_{\text{act}} \end{bmatrix} = \sum_{i=1}^{n_p} \begin{bmatrix} {}^B\mathbf{R}_{A_i} {}^{A_i}\mathbf{f}_i \\ {}^B\boldsymbol{\tau}_{\gamma_i} + {}^B\boldsymbol{\tau}_{d_i} \end{bmatrix} \\ &= \sum_{i=1}^{n_p} \begin{bmatrix} {}^B\mathbf{R}_{A_i} \mathbf{z}_{A_i} \gamma_i \\ \mathbf{S}({}^B\mathbf{p}_{A_i}) {}^B\mathbf{R}_{A_i} \mathbf{z}_{A_i} \gamma_i + {}^B\mathbf{R}_{A_i} {}^{A_i}\boldsymbol{\tau}_{d_i} \end{bmatrix}. \end{aligned} \quad (4.46)$$

The drag torque  ${}^{A_i}\boldsymbol{\tau}_{d_i}$  are produced by the tangential forces applied along the profile of the spinning rotor by the air resistance, which results in a torque collinear with  $\mathbf{z}_{A_i}$ . This moment can be synthetically modeled as

$${}^{A_i}\boldsymbol{\tau}_{d_i} = \mathbf{z}_{A_i} \bar{c}_{s,i} c_{\tau,i} \gamma_i, \quad (4.47)$$

where  $\bar{c}_{s,i} \in \{-1, +1\}$  denotes the rotating speed direction of each propeller, which is equal to  $+1$  for a propeller shape meant to rotate counter-clockwise (CCW) and  $-1$  when clockwise (CW). The rotation direction can be obtained by directly inspecting the angle of attack  $\mu_i \in \mathbb{R}$  of each propeller, as show in Figure 4.3. The scalar  $c_{\tau,i} \in \mathbb{R}_+$  is a coefficient that relates the amount of drag torque to the intensity of the thrust force  $\gamma_i$  generated by the  $i$ -th propeller. The lift force  $\gamma_i$  is proven to be related to the spinning velocity of the  $i$ -th propeller as follows

$$\gamma_i = c_{s,i} c_{\gamma,i} |\Omega_i| \Omega_i, \quad (4.48a)$$

$$c_{s,i} = -\bar{c}_{s,i}, \quad (4.48b)$$

where  $c_{\gamma,i} \in \mathbb{R}_+$  is a coefficient that provides the mapping from the propeller spinning velocity  $\Omega_i \in \mathbb{R}$  to the amount of lift thrust produced along  $\mathbf{z}_{A_i}$ . Equations (4.46)-(4.48) provide a generic model for the actuation of a MRAV as it depends only on the thrust force  $\gamma_i$  leaving free choice for the particular mechanism of thrust generation. In particular, it allows both for positive (CCW) and negative (CW)

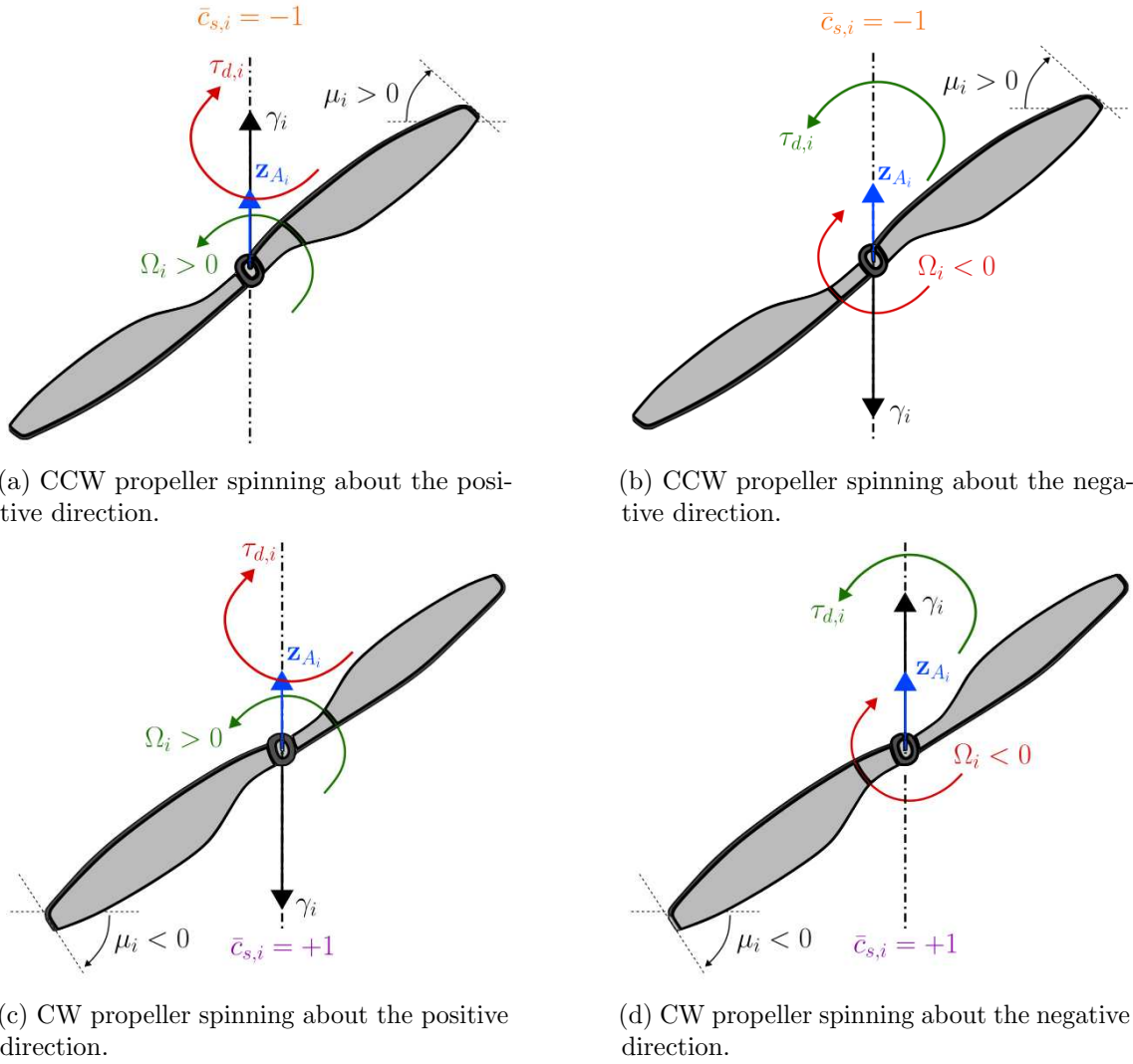


Figure 4.3: This pictures show the angle of attack  $\mu_i$  and the value of  $\bar{c}_{s,i}$  of the  $i$ -th CCW or CW propeller for positive (CCW) and negative (CW) spinning velocities  $\Omega_i$ . Accordingly, the thrust force  $\gamma_i$  along the propeller axis  $\mathbf{z}_{A_i}$  and the drag torque  $\tau_{d,i}$  are drawn.

propeller spinning velocities w.r.t.  $\mathbf{z}_{A_i}$ . The model assumes that the propeller generates symmetrical values of thrust either for positive and negative spinning velocities. However, the propeller blade is designed and optimized for a specific spinning rotation sense, therefore that assumption may result to be too simplistic. To account for more realistic conditions, the propeller coefficients  $c_{\gamma,i}$ ,  $c_{\tau,i}$  could be considered configuration-dependent, i.e. they assume different values according to the spinning directions of the rotors. Nevertheless, this mathematical characterization is extensively adopted in the literature [Hamel, 2002; Bangura, 2014; Zhao, 2022] and is experimentally proven to hold for the application under consideration [Ryll, 2012].

Nowadays, the thrust-generation model has been extended with a more sophisticated mathematical description accounting also for the blade dynamics. For instance, in [Bauersfeld, 2021], the dynamics of a spinning propeller is modeled exploiting a combination of momentum and blade-element theory, which capture the effects of varying relative air speed on the rotor thrust. On the one hand, the simple mathematical description in Equation (4.48a) is valid for most of the user cases which

generally involve maneuvers at low speeds typical during contact-based applications, like surface inspection and sensor placement. On the other hand, more sophisticated models accounting for the rotor aerodynamic effects shall be exploited when fast maneuvering is achieved, e.g. during racing competitions [Bauersfeld, 2021].

*Remark. Drag resistance.* Interestingly, the drag torque is always opposed to the rotor spinning sense and produces a resistant torque. Indeed, by substituting Equation (4.48a) into Equation (4.47), and considering Equation (4.48b), it is possible to note that  $\bar{c}_{s,i}c_{s,i} = -1$ .

Substituting Equation (4.47) into Equation (4.46), and gathering all the lift forces into the vector  $\boldsymbol{\gamma} \in \mathbb{R}^{n_p}$ , we can write

$${}^B\mathbf{w}_{\text{act}} = \begin{bmatrix} {}^B\mathbf{f}_{\text{act}} \\ {}^B\boldsymbol{\tau}_{\text{act}} \end{bmatrix} = \sum_{i=1}^{n_p} \begin{bmatrix} {}^B\mathbf{R}_{A_i}\mathbf{z}_{A_i}\gamma_i \\ \mathbf{S}({}^B\mathbf{p}_{A_i}) {}^B\mathbf{R}_{A_i}\mathbf{z}_{A_i}\gamma_i + {}^B\mathbf{R}_{A_i}\mathbf{z}_{A_i}\bar{c}_{s,i}c_{\tau,i}\gamma_i \end{bmatrix} \quad (4.49a)$$

$$= \begin{bmatrix} \mathbf{G}_{\mathbf{f}} \\ \mathbf{G}_{\boldsymbol{\tau}} \end{bmatrix} [\gamma_1, \dots, \gamma_{n_p}]^\top = \mathbf{G}\boldsymbol{\gamma}, \quad (4.49b)$$

where the matrix  $\mathbf{G}$  is the so-called *allocation matrix* and defined as

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{f_1}, \dots, \mathbf{G}_{f_i}, \dots, \mathbf{G}_{f_{n_p}} \\ \mathbf{G}_{\tau_1}, \dots, \mathbf{G}_{\tau_i}, \dots, \mathbf{G}_{\tau_{n_p}} \end{bmatrix} \in \mathbb{R}^{6 \times n_p}, \quad (4.50a)$$

$$\mathbf{G}_i = \begin{bmatrix} \mathbf{G}_{f_i} \\ \mathbf{G}_{\tau_i} \end{bmatrix} = \begin{bmatrix} {}^B\mathbf{R}_{A_i}\mathbf{z}_{A_i} \\ \left( \mathbf{S}({}^B\mathbf{p}_{A_i}) + c_{s,i}c_{\tau,i}\mathbf{I}_3 \right) {}^B\mathbf{R}_{A_i}\mathbf{z}_{A_i} \end{bmatrix} \in \mathbb{R}^6. \quad (4.50b)$$

By considering Equations (4.38), (4.29) and (4.49), we conclude that

$$\boldsymbol{\lambda}_b = \begin{bmatrix} {}^W\mathbf{f}_B \\ {}^B\boldsymbol{\tau}_B \end{bmatrix} = \begin{bmatrix} \mathbf{R}_B & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{G}_{\mathbf{f}} \\ \mathbf{G}_{\boldsymbol{\tau}} \end{bmatrix} \boldsymbol{\gamma}. \quad (4.51)$$

### Allocation matrix properties

The allocation matrix represents the mapping from the actuator forces to the body wrench. In particular, it can be decomposed into the *force* and *moment* allocation matrices, respectively  $\mathbf{G}_{\mathbf{f}} \in \mathbb{R}^{3 \times n_p}$  and  $\mathbf{G}_{\boldsymbol{\tau}} \in \mathbb{R}^{3 \times n_p}$ , which relate the forces produced by propellers to the forces and moments applied to the body. The allocation matrix depends on the design of the considered MRV, as it is a function of the spatial arrangement of the actuators w.r.t. the main body. Interestingly, it can be noticed from Equation (4.50b), that any displacement of an actuator along its  $\mathbf{z}_{A_i}$  will not alter the wrench applied on the body, as in mathematical terms it can be proven that

$$\mathbf{S}({}^B\mathbf{p}_{A_i} + {}^B\mathbf{R}_{A_i}\mathbf{z}_{A_i}\delta_{\mathbf{z}_{A_i}}) {}^B\mathbf{f}_{A_i} = \mathbf{S}({}^B\mathbf{p}_{A_i}) {}^B\mathbf{f}_{A_i}, \quad (4.52)$$

where  $\delta_{\mathbf{z}_{A_i}} \in \mathbb{R}$  represents a finite displacement along the propeller axis. This property may come in handy during the design phase of the mechanical structure of a MRV, as different rotor arrangements may produce the same wrench on the body, but differ in terms of mass distribution, i.e. different values for the resulting  ${}^B\mathbf{p}_{\text{CoM}}$  and  ${}^B\mathbf{J}_b$ .

Moreover, the rank of the allocation matrix is directly linked to the actuation capabilities of the AR [Hamandi, 2021]. Recalling the taxonomy of Section 1.2.1, under-actuated MRAVs cannot exert independently a 6D wrench  ${}^B\mathbf{w}_{\text{act}}$ . Consequently, the rank of their allocation matrix is smaller than six, even if  $n_p \geq 6$ . An example is the case of collinear hexa-rotors which have six propellers but they fall anyway into the class of under-actuated vehicles. The larger number of actuators than those mounted on UDT quad-rotors ( $n_p = 4$ ) makes them just redundant w.r.t. the task of hovering in a certain waypoint<sup>9</sup>, but not fully-actuated. Differently, fully-actuated ARs have  $\text{rank}(\mathbf{G}) = 6$  as they can decouple their linear and rotational dynamics or, equivalently, they can exert a wrench in any direction while keeping unaltered their body attitude. A MDT hexa-rotor with  $n_p = 6$  tilted propellers is an example of such a category. Lastly, over-actuated multi-rotor aerial vehicles are those flying robots equipped with  $n_p > 6$  and have  $\text{rank}(\mathbf{G}) = 6$ .

### Allocation problem

To have the AR performing a desired maneuver (motion), the necessary body wrench  ${}^B\mathbf{w}_B$  can be computed by using the system dynamics (4.26). Afterwards, it is necessary to compute the necessary actuation commands, i.e. the propeller forces (or equivalent speeds) to effectively generate such a body wrench. In the literature, this problem goes under the name of *Allocation problem* [Santos, 2022]. A simple method to solve it is considering Equation (4.49b) and by computing the (pseudo-)inverse of the allocation matrix [Brescianini, 2016], depending on the number of propellers  $n_p$ . However, the obtained solution may not be feasible, as it is not always possible to guarantee that the computed propeller speeds  $\Omega_i, i \in \{1, \dots, n_p\}$  satisfies the actuator limitations. The authors in [Sun, 2022] show that it is possible to solve such an issue by formulating it as a Quadratic Programming (QP) and find through numerical optimization the propeller inputs that produce a wrench resembling as close as possible the desired one while satisfying the actuator bounds.

### Propeller coefficients

The propellers are characterized by several parameters of the blade, like pitch angle, diameter, angle of attack, rotor disk area and radius [Pounds, 2010; Mahony, 2012]. All these properties influence the amount of generated lift force and drag torque. Moreover, propellers usually have specific requirements for the driving of electrical motors in order to provide the best compromise in terms of produced force and induced drag torques as function of the spinning frequency [Mahony, 2012]. All these characteristics combined together contribute to the values assumed by the so-called *propeller coefficients*, which are represented by the pairs  $c_{\gamma,i}, c_{\tau,i}$ . Their values must be experimentally identified given the particular hardware setup in use. It is worth it to mention that in this manuscript it is assumed that an AR may not be equipped with identical propellers and motors. Consequently, each motor-propeller actuation unit  $i$  has its own pair of coefficients as explicitly pointed out by the subscript  $i$  in  $c_{\gamma,i}, c_{\tau,i}$ . If the actuators used are of the same type, then it is safe assuming

---

<sup>9</sup>For a more detailed definition of the hovering capabilities of ARs, the interested reader is referred to [Baskaya, 2021].



that the variability within the coefficient pairs is negligible. Hence  $c_{\gamma,i} = \tilde{c}_f$  and  $c_{\tau,i} = \tilde{c}_\tau, \forall i \in \{1, \dots, n_p\}$ , where  $\tilde{c}_f \in \mathbb{R}_+$  and  $\tilde{c}_\tau \in \mathbb{R}_+$  are two constant propeller coefficients.

### 4.6.2 Joint actuators

The actuators of an aerial manipulator comprehend those of the flying base - i.e. the  $n_p$  motor-propeller pairs - and  $n_q$  actuator motors, one for each joint. We denote with  $\tau_{a,i} \in \mathbb{R}$  the torque applied by the  $i$ -th joint motor at the joint coordinate  $q_i, i \in \{1, \dots, n_q\}$ . We collect all these torques in the vector  $\boldsymbol{\tau}_a \in \mathbb{R}^{n_q}$ . Therefore, the generalized forces for the robotic arm of the AM is

$$\boldsymbol{\lambda}_a = [\tau_1 \quad \dots \quad \tau_{n_q}]^\top = \boldsymbol{\tau}_a \in \mathbb{R}^{n_q}. \quad (4.53)$$

Therefore, for the full aerial manipulator we conclude that the generalized forces are as follows

$$\boldsymbol{\lambda}_{\text{AM}} = \begin{bmatrix} \boldsymbol{\lambda}_b \\ \boldsymbol{\lambda}_a \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \mathbf{R}_B & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{G}_f \\ \mathbf{G}_\tau \end{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\tau}_a \end{bmatrix} \in \mathbb{R}^{6+n_q}. \quad (4.54)$$

### 4.6.3 Actuation limitations

So far, we have assumed that the desired forces and torques generated by the robot actuators can be varied instantaneously. This is a too simplistic assumption that does not comply with the real behavior of the robot actuators. A real actuator is a mechanical system with non-zero inertial parameters, friction, and non-unlimited actions applied to it, which undergoes its own dynamics. This means that the motor command exhibits a transient before reaching the desired set-point starting from a given initial condition, which justifies why the variation of the actuator command is limited and it shall be constrained.

From the hardware standpoint, the motors and their driving circuits, namely the Electronic Speed Controllers (ESCs), have precise electrical requirements and working conditions which, if not met, jeopardize their performances and may result in a shorter lifespan or ultimate failures. For instance, causes of potential electronic failures are the exceedance of the maximum current that the motor windings can sustain, and the negligence of thermal operating conditions within which the system can operate. Additionally, the ESCs driving a motor-propeller can effectively work only within certain velocity ranges.

#### Constraints for the motor-propeller actuation units

In the case of the motor-propeller actuation units, these operating terms lead to (i) a maximum torque that the motor can effectively apply on the rotor and the propeller, thus a maximum acceleration and change of the produced thrust; (ii) a maximum attainable rotor spinning rate beyond which the ESC cannot drive the

motor and the propeller<sup>10</sup>; (iii) a minimum propeller speed which is required by the ESC to properly operate<sup>11</sup>. This translates into a maximum propeller acceleration and deceleration (alternatively, a maximum positive and negative variation of the thrust force), bounds in the propeller velocities (forces) for the given actuation unit mounted on the AR, and a non-negligible actuator dynamics.

In mathematical terms, this results into the following differential equation

$$\dot{\gamma} = \mathbf{f}_{\gamma}(\gamma), \quad (4.55)$$

describing the actuator dynamics, and a set of inequalities for the actuation constraints, namely

$$\underline{\gamma} \leq \gamma \leq \bar{\gamma}, \quad (4.56a)$$

$$\underline{\dot{\gamma}} \leq \dot{\gamma} \leq \bar{\dot{\gamma}}. \quad (4.56b)$$

In Equation (4.55),  $\mathbf{f}_{\gamma}$  is the actuator dynamics relating the time evolution of the thrust forces  $\dot{\gamma} \in \mathbb{R}^{n_p}$  to the actuator forces  $\gamma$ . Examples of such a function will be introduced later in Section 4.7. In Equation (4.56), the terms  $\underline{\gamma}, \bar{\gamma} \in \mathbb{R}^{n_p}$  are the minimum and maximum forces that the actuators can produce, while  $\underline{\dot{\gamma}}, \bar{\dot{\gamma}} \in \mathbb{R}^{n_p}$  the lower and upper bounds of their derivatives. The values of these constraints shall be experimentally validated according to the hardware available, as they depend on the electronic components adopted, the individual software implementation and the user operating conditions [Shi, 2017; Bicego, 2019].

Moreover, contrary to the bidirectionality of the thrust generation allowed by the model in Equation (4.48a), ESCs normally cannot spin the propeller in both directions. In fact, once being set, the rotation direction is typically kept constant during their operation, causing the  $i$ -th propeller to rotate only CW, thus  $\Omega_i < 0$  and  $c_s < 0$ , or CCW, hence  $\Omega_i > 0$  and  $c_s > 0$ . Therefore, an optimal direction is chosen, and Equation (4.48a) is rewritten as follows

$$\gamma_i = c_{\gamma} \Omega_i^2, \quad i \in \{1, \dots, n_p\}. \quad (4.57)$$

As a result, the thrusts produced along  $\mathbf{z}_{A_i}$  are always positive altogether with their bounds, hence  $\bar{\gamma} > \underline{\gamma} > 0$ . Interestingly, the motor spinning rates and the generated thrust forces are now related by a simple change of coordinates by means of Equation (4.57).

### Constraints for the joint actuators

In the same way as propellers, the joints are typically powered by servo or brushless motors which cannot produce arbitrary amounts of torques due to thermal and electro-mechanical limitations [Chettibi, 2007]. Hence it is possible to define constraints also for the joint torques of the manipulator attached to the flying base as

$$\underline{\tau}_a \leq \tau_a \leq \bar{\tau}_a, \quad (4.58)$$

<sup>10</sup>The maximum allowable torque and the air resistance would reach a certain equilibrium point where they compensate each other. Such an operating point is associated to the maximum speed at which the propeller can be spun.

<sup>11</sup>Normally, ESCs need to detect the rotor position to effectively drive the propeller by producing the required electrical motor commands. If the rotor speed is below a certain threshold, its position cannot be estimated properly and the control becomes unpractical.

where  $\underline{\boldsymbol{\tau}}_a \in \mathbb{R}^{n_a}$  and  $\bar{\boldsymbol{\tau}}_a \in \mathbb{R}^{n_a}$  are the lower and upper bounds for the joint torques, respectively. Lastly, if the dynamics of the joint actuators cannot be neglected, it is possible to define similar expressions to those introduced earlier for the motor-propellers pairs. Therefore, we have that

$$\dot{\boldsymbol{\tau}}_a = \mathbf{f}_{\boldsymbol{\tau}}(\boldsymbol{\tau}_a), \quad (4.59)$$

and we add the inequality  $\dot{\underline{\boldsymbol{\tau}}}_a \leq \dot{\boldsymbol{\tau}}_a \leq \dot{\bar{\boldsymbol{\tau}}}_a$ , where  $\dot{\underline{\boldsymbol{\tau}}}_a \in \mathbb{R}^{n_a}$  and  $\dot{\bar{\boldsymbol{\tau}}}_a \in \mathbb{R}^{n_a}$  are the lower and upper bounds for the derivatives of the joint torques, respectively.

## 4.7 State-space representation of aerial robotic agents

At this point, we can derive the state-space representation of the dynamic models of aerial robots. First of all, we need to define the *state* variables, i.e. those quantities that represent the entire state of the system at any given time [Kelly, 1994]. Usually, for mechanical systems, the state vector gathers positional and velocity variables related to the system. This choice is motivated by the fact that those quantities are related in turn to the total energy, which allows then to derive the dynamic model of the system by means of the Lagrangian formalism described in Section 4.3.1.

### 4.7.1 The GTMR case

The positional and velocity variables for a GTMR are  $\mathbf{p}_B$ ,  $\mathbf{q}_B$ ,  $\mathbf{v}_B$  and  ${}^B\boldsymbol{\omega}_B$ . Moreover, in Section 4.6.3, we discussed that the motor-propeller actuation units undergo a precise transient, which implies that it is not possible to vary arbitrarily the actuator commands. Thus, to effectively take into account the actuator dynamics, the thrust forces  $\boldsymbol{\gamma}$  should be included within the state variables. We therefore define the GTMR state vector  $\mathbf{x}_{\text{GTMR}}$  as

$$\begin{aligned} \mathbf{x}_{\text{GTMR}} &= \left[ \boldsymbol{\xi}_{\text{GTMR}}^\top \quad \boldsymbol{\zeta}_{\text{GTMR}}^\top \quad \boldsymbol{\gamma}^\top \right]^\top \\ &= \left[ \mathbf{p}_B^\top \quad \mathbf{q}_B^\top \quad \mathbf{v}_B^\top \quad {}^B\boldsymbol{\omega}_B^\top \quad \boldsymbol{\gamma}^\top \right]^\top \in \mathbb{R}^{13+n_p}. \end{aligned} \quad (4.60)$$

The time evolution of the state vector is given by the full dynamics of the GTMR aerial robot, which can be obtained by concatenating the flying base dynamics, i.e. Equation (4.26), and the one for the actuators, i.e. Equation (4.55). However, we still have not defined the model for the actuators. A simple way to describe their dynamics is by means of a first-order system, as proposed in [Faessler, 2017], providing

$$\dot{\gamma}_i = f_{\gamma_i}(\gamma_i) = \frac{1}{T_{\gamma,i}}(u_{\gamma,i} - \gamma_i) \quad \forall i \in \{1, \dots, n_p\}, \quad (4.61)$$

where  $T_{\gamma,i} \in \mathbb{R}^+$  is the time constant of the *i*-th actuator (motor-propeller pair), and  $u_{\gamma,i} \in \mathbb{R}$  the real motor input. Since the time constant shall be estimated, another first-order model, which does not require any parameter to be estimated and it is proven to work well, is obtained by adopting a first-order integrator, as shown in [Bicego, 2020]. Therefore, the actuator dynamics is given by

$$\dot{\boldsymbol{\gamma}} = \mathbf{f}_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}) = \mathbf{u}, \quad (4.62)$$

where  $\mathbf{u} \in \mathbb{R}^{n_p}$  is the *input* vector collecting all the real motor commands, in this case the derivatives of the propeller forces. Hence, we can write  $\mathbf{u}_{\text{GTMR}} = \dot{\boldsymbol{\gamma}}$ .

To conclude, we can write the state space representation of a GTMR as

$$\dot{\mathbf{x}}_{\text{GTMR}} = \mathbf{f}_{\text{GTMR}}(\mathbf{x}_{\text{GTMR}}, \mathbf{u}_{\text{GTMR}}), \quad (4.63)$$

where  $\mathbf{f}_{\text{GTMR}}(\mathbf{x}_{\text{GTMR}}, \mathbf{u}_{\text{GTMR}}) : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_x}$  is a nonlinear function of the input and state vectors, with  $n_x = 13 + n_p$  and  $n_u = n_p$  being respectively the number of state and input variables of a GTMR. In this work, the function  $\mathbf{f}_{\text{GTMR}}$  is given by stacking together the Equations (4.26) and (4.62).

### 4.7.2 The AM case

Here, similarly to the procedure described for a GTMR in Section 4.7, we derive the state-space representation of the dynamic model for an aerial manipulator. We first define the state and input variables as

$$\begin{aligned} \mathbf{x}_{\text{AM}} &= [\boldsymbol{\xi}_{\text{AM}}^\top \quad \boldsymbol{\zeta}_{\text{AM}}^\top \quad \boldsymbol{\gamma}^\top]^\top \\ &= [\mathbf{p}_B^\top \quad \mathbf{q}_B^\top \quad \mathbf{q}_a^\top \quad \mathbf{v}_B^\top \quad {}^B\boldsymbol{\omega}_B^\top \quad \dot{\mathbf{q}}_a^\top \quad \boldsymbol{\gamma}^\top]^\top \in \mathbb{R}^{13+2n_q+n_p} \end{aligned} \quad (4.64)$$

$$\mathbf{u}_{\text{AM}} = [\dot{\boldsymbol{\gamma}}^\top \quad \boldsymbol{\tau}_a^\top]^\top \in \mathbb{R}^{n_p+n_q}. \quad (4.65)$$

Then, the state-space AM dynamics is given by

$$\dot{\mathbf{x}}_{\text{AM}} = \mathbf{f}_{\text{AM}}(\mathbf{x}_{\text{AM}}, \mathbf{u}_{\text{AM}}) : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_x}, \quad (4.66)$$

where  $n_x = 13 + n_p + 2n_q$  and  $n_u = n_p + n_q$ . The function  $\mathbf{f}_{\text{AM}}$  is obtained by stacking together the inverted dynamic model of the aerial manipulator in Equation (4.45) and the actuator dynamics in Equation (4.62).

Lastly, if an actuator dynamics is considered also for the joint motors, this shall be included within Equation (4.66). In turn, the input and state vectors for the aerial manipulator become

$$\mathbf{x}_{\text{AM}} = [\mathbf{p}_B^\top \quad \mathbf{q}_B^\top \quad \mathbf{q}_a^\top \quad \mathbf{v}_B^\top \quad {}^B\boldsymbol{\omega}_B^\top \quad \dot{\mathbf{q}}_a^\top \quad \boldsymbol{\gamma}^\top \quad \boldsymbol{\tau}_a^\top]^\top \in \mathbb{R}^{13+3n_q+n_p}, \quad (4.67)$$

$$\mathbf{u}_{\text{AM}} = [\dot{\boldsymbol{\gamma}}^\top \quad \dot{\boldsymbol{\tau}}_a^\top]^\top \in \mathbb{R}^{n_p+n_q}, \quad (4.68)$$

where  $\dot{\boldsymbol{\tau}}_a \in \mathbb{R}^{n_q}$  collects the derivatives of the joint torques  $\boldsymbol{\tau}_a$ . Similarly to what discussed earlier for the propellers, the dynamics of the joint motors can be described, for instance, with a 1st order system, thus

$$\dot{\tau}_i = f_{\tau_i}(\tau_i) = \frac{1}{T_{\tau,i}} (u_{\tau,i} - \tau_i) \quad \forall i \in \{1, \dots, n_q\}, \quad (4.69)$$

where  $T_{\tau,i} \in \mathbb{R}^+$  is the time constant for the  $i$ -th joint actuator, and  $u_{\tau,i} \in \mathbb{R}$  its real command. Alternatively, if available, a more complex model can be used to consider the presence of elasticity or other non-linearities in the joint mechanisms.

In this thesis, we consider the joint motors to be ideal and do not exhibit particular transient dynamics. Therefore, we rely on the state-space representation given by Equations (4.64)-(4.66).

## 4.8 Perceptive sensors on-board aerial robots

GTMRs are commonly equipped with exteroceptive sensors allowing them to retrieve information about the external world and estimate their own state. Different sensor choices are possible, among which monocular or stereo cameras represent the most popular solution thanks to their compact and lightweight dimensions, affordable prices and the availability of open-source software utilities to process the acquired frames. Similarly, depth-infrared cameras, also known as RGB+Depth (RGBD), are gaining a lot of attention by the robotic community as they flooded the market in a large variety of models, many of which fitting the strict requirements in terms of payload and computational power related to ARs. Additionally, their popularity is increasing as they directly provide the depth measurement, which in standard monocular camera is not available and therefore must be estimated with software packages. In the latter case, it is possible to indirectly measure the distance to an observed feature by either assuming some prior geometric knowledge [Thomas, 2017] or relying on some deep learning-based algorithms [Wofk, 2019].

In recent years, thanks to advances in sensor miniaturization, cheaper and lighter lidars are also available on the market. Several years ago, these sensors were solely employed on ground robots due to not only their large weight and dimensions, but also the large amount of computational power required to process the huge number of collected samples. Nowadays, thanks to more affordable and smaller devices, and the availability of powerful and compact computers, they are started to be employed also on ARs, e.g. for environmental mapping [Mohta, 2018].

The sensors described so far are called *range-and-bearing* sensors, since they provide both 2D angular information and a 1D depth measurement [Jacquet, 2022b]. This class naturally encompasses stereo and RGBD cameras. Instead, monocular cameras are *bearing-only* sensors as they do not provide depth information by themselves. However, they can fall in the previous category when accompanied by a software package that allows a posteriori to retrieve such a measure from the acquired frames.

Therefore, through this thesis, we consider only GTMR systems equipped with a range-and-bearing sensor  $S$ , either native (e.g. a RGBD camera) or composed by a sensor and a software package (as in the case of a monocular camera). As in [Jacquet, 2021], and as shown in Figure 4.1, we model this sensing device as a punctual device rigidly mounted on the robot chassis and centered in  $O_S$ , having principal axis  $\mathbf{z}_S$  (i.e. the axis describing its bearing). The other axes,  $\mathbf{x}_S$  and  $\mathbf{y}_S$ , are chosen such that the frame is right-handed and they define the horizontal and vertical directions of the sensor. In addition, the pose transformation between  $\mathcal{F}_S$  and the one of the flying base  $\mathcal{F}_B$  is constant and known. The sensor is capable of retrieving the 3D-pose of an observed entity, generically called feature, in the sensor frame  $\mathcal{F}_S$  and denoted with  ${}^S\mathbf{p}_F \in \mathbb{R}^3$ . The feature can be a geometrical property of an object in the environment (e.g. the corner of a wall), or a particular point on the body of the human interacting with the robot.

## 4.9 Human agent

In this section, we present the model of the human agent involved in the human-aerial robot handover presented in Chapter 2. The human operator is modeled in a simple but effective way, namely as the composition of two subsystems: a trunk and an arm. The model of the trunk is used to describe the motion of the person during the handover, while the one of the arm will serve to evaluate the ergonomics level, which will be further discussed in Chapter 6 and more specifically in Section 6.2.2. Throughout this thesis, we focus only on one human arm being employed during the human-robot collaboration, but a similar model can be applied to both arm, if required. Moreover, we assume that each subsystem can be represented by the interconnection of one or more rigid bodies, as shown in the following.

In the literature, it is possible to find models to describe the motion of the human and of their limbs. For instance, Arachavaleta et al. [Arechavaleta, 2008] show how the human locomotion is effectively described by a simple non-holonomic system, i.e. a system where the forward heading direction is coupled to the body orientation. The authors show that humans, while naturally moving to a desired goal to perform a task, perform trajectories which are well approximated by those of a unicycle vehicle. Moreover, they conclude that the best results are obtained when considering the trajectories of the shoulders, while those measured at the head, the torso and the pelvis are affected by the step alternation and do not provide good repeatability among different users. In [Laumond, 2017], the authors provide a novel interpretation of the human locomotion by means of the so-called *Yoyo model*. This mathematical formulation approximates the motion of the human during a natural walking gait as the trajectory performed by the geometric center of a rimless wheel. The resulting motion is composed of a sequence of circle arcs whose radii correspond to the stand beam of the rimless wheel, and it is showed to match quite accurately the trajectory of the CoM of the walking body. Similarly, in [Carpentier, 2017], the authors show how the trajectory of the CoM of a walking person is a key descriptor in the understanding and the analysis of bipedal locomotion. Specifically, the CoM of the body, in the sagittal plane, follows a curtate cycloid, which is the curve described by a point rigidly attached to a virtual wheel rolling on a flat surface.

While in [Arechavaleta, 2008] the authors provide a model suitable to describe the human motion over the transversal ( $x, y$  or horizontal) plane, the model of Laumond et al. describes the human locomotion along the sagittal ( $x, z$  or vertical) plane. In this thesis, we focus on describing the position over the horizontal plane, and we assume that the height of the human is not changing noticeably during the handover. Therefore, we take inspiration from the work of [Arechavaleta, 2008] to describe the human motion by means of a constant-velocity model, as illustrated shortly. Contrary to the unicycle model of Arachavaleta et al., our modeling choice allows accounting also for lateral movements of the human, which is motivated by the intuitive idea that during a handover the person may also perform lateral movements.

### 4.9.1 Trunk

The human motion is represented by the trajectory performed by the trunk of the person. Hence, accordingly to Figure 4.4, we define a *trunk* reference frame  $\mathcal{F}_{H_t}$

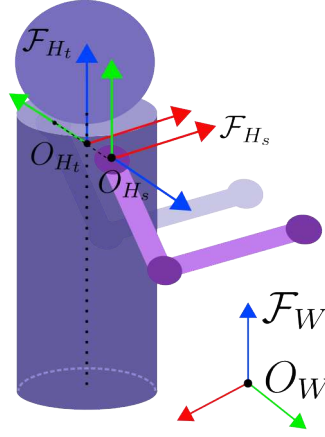


Figure 4.4: This figure illustrates the model used to describe the human worker as a composition of two parts: a trunk and an arm. The second arm is drawn just for visual purposes.

having  $\mathbf{y}_{H_t}$  coincident to the line connecting the human shoulders,  $\mathbf{x}_{H_t}$  pointing in the forward walking direction, and  $\mathbf{z}_{H_t}$  parallel to  $\mathbf{z}_W$ . For the sake of simplicity, we assume that the human maintains a standing position while walking; so the human trunk roll and pitch angles are set constant and null. Therefore, the human motion is given by the position and heading of the trunk expressed in the inertial frame. Consequently, we define the *human state*  $\mathbf{x}_h$  as

$$\mathbf{x}_h = [\mathbf{p}_{H_t}^\top \ \psi_{H_t}]^\top \in \mathbb{R}^4, \quad (4.70)$$

where  $\mathbf{p}_{H_t} \in \mathbb{R}^3$  is the position of  $O_{H_t}$  in  $\mathcal{F}_W$ , and  $\psi_{H_t} \in \mathbb{R}$  is the angle between  $\mathbf{x}_W$  and  $\mathbf{x}_{H_t}$  along  $\mathbf{z}_W$ . As mentioned at the beginning of this section, we adopt a *constant-velocity* model to describe the time evolution of the human state, thus

$$\dot{\mathbf{x}}_h = [\mathbf{v}_{H_t}^\top \ \omega_{H_t\psi}]^\top, \quad (4.71)$$

where  $\mathbf{v}_{H_t} \in \mathbb{R}^3$  are the linear velocities of the human trunk in the inertial frame, and  $\omega_{H_t\psi} \in \mathbb{R}$  is the angular speed about  $\mathbf{z}_{H_t}$  expressed in  $\mathcal{F}_W$ . We select these velocities as the *human inputs* and we denote them with  $\mathbf{u}_h$ . Thus, we can write

$$\dot{\mathbf{x}}_h = \mathbf{u}_h. \quad (4.72)$$

## 4.9.2 Arm

Hereafter, we want to derive a model to describe the ergonomic stress at the human-arm level in the handover scenario. To achieve this goal, we model the upper human limb as a manipulator whose base is connected at the shoulder attaching point, as depicted in Figure 4.1. Therefore, we define a *shoulder* reference frame denoted by  $\mathcal{F}_{H_s}$  and centered at the manipulator base. The relative pose transformation between  $\mathcal{F}_{H_s}$  and  $\mathcal{F}_{H_t}$  is assumed to be known. In particular, we take  $\mathbf{x}_{H_s}$  and  $\mathbf{y}_{H_s}$  to be parallel to  $\mathbf{x}_{H_t}$  and  $\mathbf{z}_{H_t}$ , respectively. This manipulator is composed of a serial chain of rigid links in pairs connected by a 1-revolute joint. Thus, the human arm is

modeled as an  $n_h$ -link manipulator [Siciliano, 2009], where  $n_h \in \mathbb{N}_+$  is the number of joints in the human arm model.

As a result, the dynamics of the upper limb is given by

$$\mathbf{M}_h(\mathbf{q}_h)\ddot{\mathbf{q}}_h + \mathbf{C}_h(\mathbf{q}_h, \dot{\mathbf{q}}_h)\dot{\mathbf{q}}_h + \mathbf{g}_h(\mathbf{q}_h) = \boldsymbol{\tau}_h + \mathbf{J}_h(\mathbf{q}_h)^\top \mathbf{f}_{\text{ext}}, \quad (4.73)$$

where  $\mathbf{M}_h \in \mathbb{R}_+^{n_h \times n_h}$  is the inertia matrix,  $\mathbf{C}_h \in \mathbb{R}^{n_h \times n_h}$  accounts for the centrifugal and Coriolis terms, and  $\mathbf{g}_h \in \mathbb{R}^{n_h}$  collects the gravitational effects of the human arm. The vectors  $\mathbf{q}_h, \dot{\mathbf{q}}_h, \ddot{\mathbf{q}}_h \in \mathbb{R}^{n_h}$  are respectively the joint positions, velocities, and accelerations, while  $\boldsymbol{\tau}_h \in \mathbb{R}^{n_h}$  gathers the human-arm joint torques. The matrix  $\mathbf{J}_h(\mathbf{q}_h) \in \mathbb{R}^{3 \times n_h}$  is the *geometric Jacobian* of the human arm mapping the effect of an external force  $\mathbf{f}_{\text{ext}} \in \mathbb{R}^3$  applied to the hand to the arm dynamics. The model shown in Equation (4.73) is in the canonical form as in Equation (4.23), given  $\boldsymbol{\xi}_h = \mathbf{q}_h$ ,  $\dot{\boldsymbol{\xi}}_h = \dot{\mathbf{q}}_h$  and  $\ddot{\boldsymbol{\xi}}_h = \ddot{\mathbf{q}}_h$ . This is not surprising as we model it as a manipulator composed of rigid bodies, in the same fashion as the robotic arm mounted on an aerial manipulator.

*Remark. Human arm parameters.* The terms  $\mathbf{M}_h$ ,  $\mathbf{C}_h$ , and  $\mathbf{g}_h$  in Equation (4.73) are function of the geometric and inertial parameters of the  $n_h$  links composing the human upper limb. To identify their values, biomechanical datasets [Winter, 2009] or estimation algorithms [Nagano, 2005; Ayusawa, 2011; Jovic, 2016] can be used. Hereafter, we assume that these parameters are known since the proposition of a suitable estimation method is outside the scope of this thesis.

## 4.10 Conclusion

In this chapter, we provided the reader with the theoretical background and the mathematical models of the two agents (aerial robot and human) involved in the Human-Aerial Robot (HAR) handover problem presented in Chapter 2. Specifically, we presented the formulation for a Generically-Tilted Multi-Rotor and an aerial manipulator. Then, we introduced the model describing the human motion and the dynamics of the user upper limb. This allows understanding the control formulation that will be later discussed in the next two chapters. Specifically, in Chapter 5, these models will be employed to derive a control law allowing ARs to achieve physical Human-Aerial Robot Interaction (pHARI), performing a HAR tool handover. Later, in Chapter 6, the human (trunk and arm) models will be embedded into a predictive controller to (i) evaluate the ergonomic stress undertaken by the human worker in the handover process, and (ii) let the AR controller be aware of the partner motion. In this way, the robot control algorithm can compute a human-aware trajectory and actuator commands to successfully and ergonomically hand over a tool to the human operator.





# Chapter 5

## Control methodology for visual and physical Human-Aerial Robot Interaction

The goal of this chapter is to present an in-depth formulation and derivation of the hierarchical control architecture presented in Section 3.1.1. This proposed methodology is designed to address the first subproblem described earlier, which consists in achieving physical Human-Aerial Robot Interaction (pHARI).

As mentioned in Chapter 2, we first empower a fully-actuated aerial robot (AR) with the capability to perform physical interaction tasks in the surrounding environment. In Section 5.1, we detail the internal structure of each module composing such a hierarchical control architecture. The resulting methodology allows the robot to be driven to the goal by a vision-based system, and to apply an arbitrary 6D wrench exploiting the fully actuation property of the robotic platform. We test the method in a pick-and-place operation as previously mentioned. In Section 5.2, we bring more technical and quantitative results related to the performance of the proposed controller architecture in this validation application. Afterwards, in Section 5.3, we extend the formulation of the control framework to the case of an aerial manipulator physically interacting with a human worker. In Section 5.4, we describe promising results obtained during the second validation scenario, namely the human-to-robot (H2R) handover involving an operator returning an object to the robotic partner. Finally, in Section 5.5, we conclude this chapter with some final thoughts on the proposed methodology, we discuss its limitations and sketch potential future work.

### 5.1 Hierarchical control architecture

We present here the internal structure of the proposed hierarchical control architecture allowing fully-actuated aerial robots to autonomously accomplish tasks that require both perception and physical interaction with the external environment.

As shown in Figure 5.1, the vision-based control module is composed of a *Hybrid Visual Servoing* (HVS) scheme. This method generates velocity commands driving an autonomous system towards the desired goal based on the error between the current

value of the selected visual clues, known with the term features, and their desired ones. The formulation of this algorithm is very straightforward, as one has to start from the definition of one or more visual features of interest. The features are local patterns in the image that can be easily identified, such as corners, step edges, lines or curves [Rosenfeld, 1988].

The physical interaction layer combines a *Admittance filter* (AF) and a *Wrench observer* (WO). The AF modifies the input desired trajectory producing a new one based on the external wrench experienced by the robot. The filtering process is such that the robot dynamics acquires a new shape, emulating a virtual mass-damper-spring system whose model parameters are selected according to the user and task requirements. The WO computes the estimation of the external forces and torques applied on the robot body by relying solely on the system model and minimal onboard measurements. These measures can be obtained by a standard IMU, which constitutes the bare minimal sensor equipment on-board any AR.

The output reference trajectory provided by the AF is then fed to a low-level *motion controller* that produces the commands driving the robot actuators by relying on the robot dynamic model. This module pivots on the novel *Flying End-Effector* paradigm earlier mentioned in Chapter 3. It accounts for the fully actuation of the robot to generate a full 6D wrench from which the motor-propellers commands can be computed. Moreover, it considers virtual bounds on the lateral forces that fully-actuated aerial vehicles can in general produce<sup>1</sup>.

In the following subsections, we detail the formulation of each block presented in Figure 5.1. As done in Chapter 4, hereafter, we omit the reference frame for any variable expressed in  $\mathcal{F}_W$ .

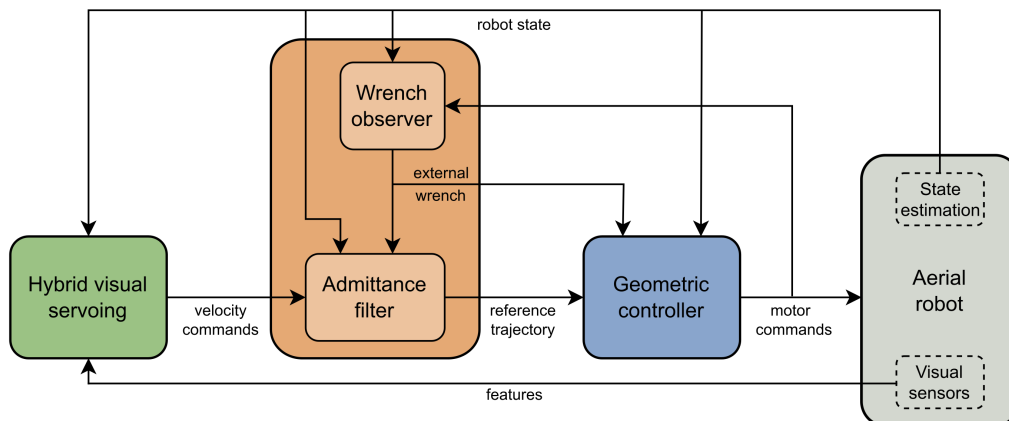


Figure 5.1: Generic visual and physical control architecture enhancing fully-actuated MRAVs with physical interaction capabilities. In green, the vision-based control constituted of a Hybrid Visual Servoing. In orange, the physical interaction layer comprising an external wrench observer and admittance filter. In blue, the motion controller and, lastly, the aerial robotic platform and the robot state estimation module in gray.

<sup>1</sup>These robots fall into the category of laterally-bounded fully-actuated multi-rotor aerial vehicles (MRAVs) which we already presented in Chapter 1.

### 5.1.1 Vision-based control

In order to autonomously generate the motion towards the object based on visual clues we propose the use of a visual servoing scheme. Specifically, we adopt a Hybrid Visual Servoing (HVS) [Chaumette, 2007]. We will show hereafter how this module computes the desired velocity commands of the platform starting from the images captured from the onboard camera.

Before starting, we recall the body and sensor (or camera)<sup>2</sup> frames denoted as  $\mathcal{F}_B = O_B, \{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$  and  $\mathcal{F}_S = O_S, \{\mathbf{x}_S, \mathbf{y}_S, \mathbf{z}_S\}$ , respectively, from Chapter 4. The formulation of the HVS method begins from the definition of the visual feature vector  $\mathbf{s}$  and its reference  $\mathbf{s}^r$ . Next, a tracking error can be formulated as

$$\mathbf{e}_s = \mathbf{s} - \mathbf{s}^r. \quad (5.1)$$

In a classical visual servoing scheme, the feature vector is defined as

$$\mathbf{s} = [\tilde{x} \quad \tilde{y} \quad \log \tilde{z} \quad \theta \mathbf{u}] \in \mathbb{R}^6, \quad (5.2)$$

where  $\tilde{x}, \tilde{y} \in \mathbb{R}$  are the coordinates of the point of interest defined optionally in (i) sensor frame, (ii) normalized camera coordinates, (iii) or as the positions in image plane (pixel coordinates). The quantity  $\tilde{z} \in \mathbb{R}_+$  is the position of the feature along the principal axis of the camera  $\mathbf{z}_S$ , and  $\theta \mathbf{u} \in \mathbb{R}^3$  is the angle-axis representation of the orientation error. In the following, we select  $\tilde{x}$  and  $\tilde{y}$  to be the normalized coordinates of a detected feature F, thus

$$\tilde{x} = \frac{{}^S x_F}{{}^S z_F}, \quad \tilde{y} = \frac{{}^S y_F}{{}^S z_F}, \quad \tilde{z} = {}^S z_F, \quad (5.3)$$

where the position of F expressed in  $\mathcal{F}_S$  is given by  ${}^S \mathbf{p}_F = [{}^S x_F \quad {}^S y_F \quad {}^S z_F]^\top$ . The reference vector  $\mathbf{s}^r$  has to be chosen in order to align the end effector with its target, hence is specific to the considered task. We will show our choice later in Section 5.2.1.

The velocity control in  $\mathcal{F}_S$  is designed to nullify  $\mathbf{e}_s$ . Typically an exponentially decreasing rate is sought, thus we write

$$\dot{\mathbf{e}}_s = -\epsilon_s \mathbf{e}_s, \quad (5.4)$$

where  $\epsilon_s \in \mathbb{R}_+$  represents a constant parameter dictating the convergence rate of the error dynamics. At this point, the interaction matrices  $\mathbf{L}_v$ ,  $\mathbf{L}_\omega$  and  $\mathbf{L}_{\theta \mathbf{u}} \in \mathbb{R}^{3 \times 3}$  can be introduced. These matrices relate the error transient to the sensor twist, denoted as  ${}^S \boldsymbol{\zeta}_S$  and expressed in  $\mathcal{F}_S$ , as follows

$$\dot{\mathbf{e}}_s = \begin{bmatrix} \mathbf{L}_v & \mathbf{L}_\omega \\ \mathbf{O}_3 & \mathbf{L}_{\theta \mathbf{u}} \end{bmatrix} \begin{bmatrix} {}^S \mathbf{v}_S \\ {}^S \boldsymbol{\omega}_S \end{bmatrix}. \quad (5.5)$$

In the equation above, the twist vector  ${}^S \boldsymbol{\zeta}_S = [{}^S \mathbf{v}_S^\top \quad {}^S \boldsymbol{\omega}_S^\top]^\top \in \mathbb{R}^6$  collects the linear and angular velocities of the camera expressed in  $\mathcal{F}_S$ . Clearly, the vision-based system comprises two parts, one exploiting the sensor linear velocities to nullify the error (top part of Equation (5.5)), and another one using the angular speeds (bottom

<sup>2</sup>Hereafter, we will use the term camera and sensor frame to denote the same reference system.

part). Similarly, we can decompose the feature error vector into two portions as follows

$$\mathbf{e}_s = \begin{bmatrix} \mathbf{e}_{st} \\ \mathbf{e}_{sr} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_t - \mathbf{s}_t^r \\ \mathbf{s}_r - \mathbf{s}_r^r \end{bmatrix}, \quad (5.6)$$

where  $\mathbf{e}_{st} \in \mathbb{R}^3$  and  $\mathbf{e}_{sr} \in \mathbb{R}^3$  are the translational and rotational parts of the feature error, respectively. According to the feature selection above, we have that  $\mathbf{e}_{sr} = \theta \mathbf{u}$ , as  $\mathbf{s}_r^r = \theta^r \mathbf{u}^r = \mathbf{o}_3$ .

The angular velocity control is defined as in [Chaumette, 2006]. As a result, we define the orientation interaction matrix  $\mathbf{L}_{\theta \mathbf{u}}$  as

$$\mathbf{L}_{\theta \mathbf{u}} = \mathbf{I}_3 - \frac{\theta}{2} \mathbf{S}(\mathbf{u}) + \left( 1 - \frac{\sin \theta}{\text{sinc}^2 \frac{\theta}{2}} \right) \mathbf{S}(\mathbf{u})^2, \quad (5.7)$$

where  $\text{sinc}(\bullet)$  is the sinus cardinal function and it is defined as  $\text{sinc}(\alpha) = \frac{\sin(\alpha)}{\alpha}$  for a given angle  $\alpha \in \mathbb{R}$ . The determinant of the above matrix has the following expression

$$\det(\mathbf{L}_{\theta \mathbf{u}}) = \frac{1}{\text{sinc}^2 \frac{\theta}{2}}, \quad (5.8)$$

which brings singularities only for  $\theta = 2k\pi$ ,  $k \neq 0$ . However, these cases are out of the potential workspace since  $\theta \in [0, \pi]$ , as mentioned in the definition of the angle-axis representation in Chapter 4. As we can always invert the rotational (bottom) part of Equation (5.5), we can combine Equations (5.2), (5.4), (5.5) and (5.6) to obtain

$${}^S \boldsymbol{\omega}_S = -\lambda \mathbf{L}_{\theta \mathbf{u}}^{-1} \theta \mathbf{u}, \quad (5.9)$$

where the interaction matrix  $\mathbf{L}_{\theta \mathbf{u}}$  is computed from Equation (5.7).

We can now define the linear velocity control scheme, following again [Chaumette, 2007]. We first define  $\mathbf{L}_v$  and  $\mathbf{L}_\omega$  as

$$\mathbf{L}_v = \frac{1}{\rho_z \tilde{z}^r} \begin{bmatrix} -1 & 0 & \tilde{x} \\ 0 & -1 & \tilde{y} \\ 0 & 0 & -1 \end{bmatrix}, \quad (5.10)$$

$$\mathbf{L}_\omega = \begin{bmatrix} \tilde{x}\tilde{y} & -(1 + \tilde{x}^2) & \tilde{y} \\ 1 + \tilde{y}^2 & -\tilde{x}\tilde{y} & -\tilde{x} \\ -\tilde{y} & \tilde{x} & 0 \end{bmatrix}, \quad (5.11)$$

where  $\rho_z = \tilde{z}/\tilde{z}^r$ ,  $\tilde{z}^r \in \mathbb{R}_+$  being the reference for  $\tilde{z}$ , and  $\tilde{x} \in \mathbb{R}$ ,  $\tilde{y} \in \mathbb{R}$  are the normalized coordinates of the detected feature defined as in Equation (5.3). As noted in [Chaumette, 2007],  $\rho_z$  can be obtained from a partial pose estimation scheme. The matrix  $\mathbf{L}_v$  is singular only when  $\tilde{z} \rightarrow \infty$ , making the inversion always feasible also for the linear (top) part of Equation (5.5). Recalling again the definition of the feature vector in Equation (5.2), we can invert Equation (5.5) to obtain

$${}^S \mathbf{v}_S = -\mathbf{L}_v^{-1} (\lambda \mathbf{e}_{st} + \mathbf{L}_\omega {}^S \boldsymbol{\omega}_S), \quad (5.12)$$

where the matrices  $\mathbf{L}_v$  and  $\mathbf{L}_\omega$  are obtained from Equations (5.10) and (5.11).

We can now define the desired linear and angular velocities of the AR in the inertial frame  $\mathcal{F}_W$  by exploiting the kinematic linkage between the robot and sensor bodies. Therefore, we have that

$$\mathbf{p}_S = \mathbf{p}_B + \mathbf{R}_B {}^B \mathbf{p}_S \quad \text{and} \quad \mathbf{R}_S = \mathbf{R}_B {}^B \mathbf{R}_S, \quad (5.13)$$

where  ${}^B \mathbf{p}_S \in \mathbb{R}^3$  and  ${}^B \mathbf{R}_S \in SO(3)$  represent the extrinsic camera parameters. Their values depend on how the camera is mounted on the robot and in turn they can be estimated by means of any CAD software or by inspecting the real hardware. Consequently, by differentiating Equation (5.13) and re-arranging the result, we obtain the desired robot velocities as

$$\mathbf{v}_B = \mathbf{R}_B \left( {}^B \mathbf{R}_S {}^S \mathbf{v}_S - \mathbf{S}(\boldsymbol{\omega}_B) {}^B \mathbf{p}_C \right), \quad (5.14)$$

$$\boldsymbol{\omega}_B = \boldsymbol{\omega}_S = \mathbf{R}_B {}^B \mathbf{R}_S {}^S \boldsymbol{\omega}_S. \quad (5.15)$$

As we assume that the camera is rigidly mounted on the robot body, in the equations above, we have considered null any relative velocity between the camera and the AR.

### 5.1.2 Physical interaction control

Here, we discuss how our control architecture takes into consideration the physical interaction. As visible in Figure 5.1, the physical interaction control features two modules, a Wrench observer (WO) and an Admittance filter (AF). The former estimates the external wrench arising during the interaction applied on the robot body. The latter lets the platform be compliant to the external forces and torques. The AR is modeled as the Generically-Tilted Multi-Rotor (GTMR) vehicle in Section 4.4, and we assume that  ${}^B \mathbf{p}_{\text{CoM}} = \mathbf{o}_3$ , thus  $O_B$  coincides with the robot center of mass (CoM). Moreover, the aerial robot interacts with the environment only through its end effector (EE), hence we consider only one contact point ( $n_e = 1$ ) located at  $O_E$ . In such a position, we define the end effector frame  $\mathcal{F}_E = O_E, \{\mathbf{x}_E, \mathbf{y}_E, \mathbf{z}_E\}$  which is rigidly attached to the robot EE.

#### Wrench observer

To estimate the external wrench that is applied at the robot EE, we adopt the observer proposed in [Tomić, 2017]. Taking inspiration from ground manipulators, the authors propose a *hybrid* Wrench observer tailored to flying robots. It is composed of an acceleration-based estimator for the external forces and a momentum-based one for the external torques. This method is well suited for MRAs since it exploits only robot proprioceptive sensors, such as an IMU. These sensors are nowadays very affordable and typically available on-board any aerial platform [Santamaria-Navarro, 2018]. The structure of the estimator is detailed in Figure 5.2. The WO computes the external forces  ${}^B \mathbf{f}_{\text{ext}} \in \mathbb{R}^3$  and torques  ${}^B \boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^3$  applied on the robot body and expressed in  $\mathcal{F}_B$ , by exploiting the knowledge of the robot dynamical model.

By recalling Equations (4.26), it is possible to estimate the external wrench as

$$\begin{cases} {}^B \hat{\mathbf{f}}_{\text{ext}} = \int_0^t \mathbf{Q}_{I,f} (m_b {}^B \mathbf{a}_B - {}^B \mathbf{f}_{\text{act}} - {}^B \hat{\mathbf{f}}_{\text{ext}}) d\nu \\ {}^B \hat{\boldsymbol{\tau}}_{\text{ext}} = \mathbf{Q}_{I,\tau} \left( \mathbf{J}_b {}^B \dot{\boldsymbol{\omega}}_B + \int_0^t \left( \mathbf{J}_b \mathbf{S}({}^B \boldsymbol{\omega}_B) {}^B \boldsymbol{\omega}_B - {}^B \boldsymbol{\tau}_{\text{act}} - {}^B \hat{\boldsymbol{\tau}}_{\text{ext}} \right) d\nu \right), \end{cases} \quad (5.16)$$

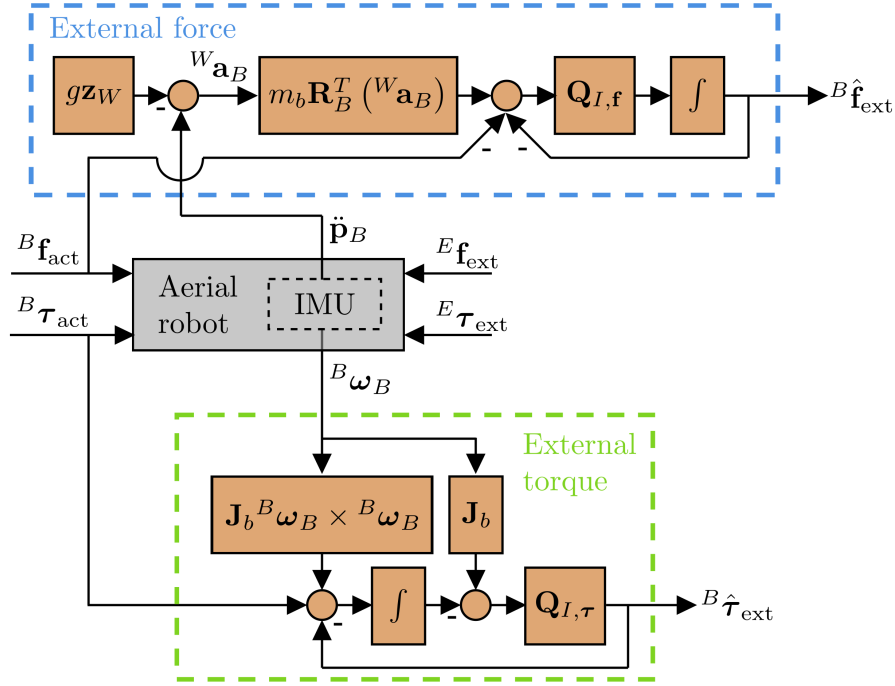


Figure 5.2: Internal structure of the hybrid Wrench observer.

where  $\mathbf{Q}_{I,f}, \mathbf{Q}_{I,\tau} \in \mathbb{R}^{3 \times 3}$  are respectively the estimator gains for the external forces and moments. The quantity  ${}^B\mathbf{a}_B = \mathbf{R}_B^\top (\ddot{\mathbf{p}}_B + g\mathbf{z}_W) \in \mathbb{R}^3$  is the robot acceleration in  $\mathcal{F}_B$  including the gravity contribution, and  ${}^B\boldsymbol{\omega}_B$  the angular velocity of the platform expressed in  $\mathcal{F}_B$ . The robot linear and angular accelerations are, respectively, provided as output by the onboard accelerometer and gyroscope, which typically constitute the sensor set of any standard IMU. Note that the notation  $\hat{\bullet}$ , in the equation above, indicates the estimated value of a given quantity. Lastly, the estimator needs to know the wrench produced by the actuators, i.e. the terms  ${}^B\mathbf{f}_{\text{act}} \in \mathbb{R}^3$  and  ${}^B\boldsymbol{\tau}_{\text{act}} \in \mathbb{R}^3$ . These two quantities are generated by the motion controller presented later in Section 5.1.3.

### Admittance filter

For having the robot compliant in the physical interaction, we resort to an Admittance filter, which is a well-known technique in the literature [Siciliano, 2009]. The AF takes in input the desired robot trajectory  $(\bullet^d)$  expressed in  $\mathcal{F}_W$ . As output, it produces a new reference trajectory  $(\bullet^r)$ , which shapes the robot dynamics as a virtual mass-damper-spring system. Ultimately, the robot is compliant w.r.t. the applied external wrench when tracking closely the new reference trajectory. By exploiting the rigid kinematic linkage between the robot body and its EE, we can relate the motion of  $O_E$  to the one of  $O_B$  and vice versa. Likewise, we can easily transpose the effect of the external wrench in one of the two points, as stated by Equation (4.27). As a result, the admittance filter can be applied to either the robot end effector or its body, leading in both cases to a compliant system. Since the HVS scheme presented earlier generates the desired velocities of  $O_B$  expressed in  $\mathcal{F}_W$ , it is more convenient

to write the admittance filter at the robot body. This results in

$$\mathbf{\Gamma}_v \begin{bmatrix} \ddot{\mathbf{e}}_{\mathbf{p}_B} \\ \dot{\mathbf{e}}_{\boldsymbol{\omega}_B} \end{bmatrix} + \mathbf{\Lambda}_v \begin{bmatrix} \dot{\mathbf{e}}_{\mathbf{p}_B} \\ \mathbf{e}_{\boldsymbol{\omega}_B} \end{bmatrix} + \mathbf{\Delta}_v \begin{bmatrix} \mathbf{e}_{\mathbf{p}_B} \\ \mathbf{e}_{\mathbf{R}_B} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}_{\text{ext}} \\ \hat{\boldsymbol{\tau}}_{\text{ext}} \end{bmatrix}. \quad (5.17)$$

The vector  $\mathbf{e}_{(\bullet)} = (\bullet)^r - (\bullet)^d$  is the error between the reference  $(\bullet)^r$  and desired  $(\bullet)^d$  values of a given quantity. Specifically, in our case, we have

$$\mathbf{e}_{\mathbf{p}_B} = \mathbf{p}_B^r - \mathbf{p}_B^d, \quad (5.18)$$

$$\mathbf{e}_{\mathbf{R}_B} = \frac{1}{2} \left( \mathbf{R}_B^r \mathbf{R}_B^{d\top} - \mathbf{R}_B^d \mathbf{R}_B^{r\top} \right)^\vee, \quad (5.19)$$

$$\mathbf{e}_{\boldsymbol{\omega}_B} = \boldsymbol{\omega}_B^r - \boldsymbol{\omega}_B^d, \quad (5.20)$$

which represent the translational, rotational and angular velocity errors. The notation  $(\bullet)^\vee$  denotes the inverse map of the skew operator  $\mathbf{S}(\bullet)$  [Lee, 2010]. The derivatives of Equations (5.18) and (5.20) are simply obtained exploiting the linearity of the differentiation operator, which results in

$$\dot{\mathbf{e}}_{\mathbf{p},B} = \dot{\mathbf{p}}_B^r - \dot{\mathbf{p}}_B^d, \quad (5.21)$$

$$\ddot{\mathbf{e}}_{\mathbf{p},B} = \ddot{\mathbf{p}}_B^r - \ddot{\mathbf{p}}_B^d, \quad (5.22)$$

$$\dot{\mathbf{e}}_{\boldsymbol{\omega}_B} = \dot{\boldsymbol{\omega}}_B^r - \dot{\boldsymbol{\omega}}_B^d. \quad (5.23)$$

In Equation (5.17), the matrices  $\mathbf{\Gamma}_v \in \mathbb{R}^{6 \times 6}$ ,  $\mathbf{\Lambda}_B \in \mathbb{R}^{6 \times 6}$ ,  $\mathbf{K}_\Delta \in \mathbb{R}^{6 \times 6}$  are the virtual mechanical inertia, damping and stiffness, respectively. According to the desired robot behavior, the user can choose these matrices to tune the virtual mass-damper-spring dynamics that the AR will abide when physically interacting with the environment. Finally, in Equation (5.17), the terms  $\hat{\mathbf{f}}_{\text{ext}}$  and  $\hat{\boldsymbol{\tau}}_{\text{ext}}$  are the estimated external forces and torques computed by the wrench observer through Equations (5.16), but expressed in  $\mathcal{F}_W$ . From Equation (5.17), it is possible to compute the new reference robot accelerations, i.e.  $\ddot{\mathbf{p}}_B^r \in \mathbb{R}^3$  and  $\dot{\boldsymbol{\omega}}_B^r \in \mathbb{R}^3$ , which make the robot behave as the virtual dynamics when an external wrench  ${}^E \mathbf{w}_E = [{}^E \mathbf{f}_E^\top \quad {}^E \boldsymbol{\tau}_E^\top]^\top$  is applied at its EE. By numerically integrating those accelerations, it is possible to compute the rest of the reference trajectory, up to the position  $\mathbf{p}_B^r$  and the attitude  $\mathbf{R}_B^r$ .

### 5.1.3 Motion control

The controller stabilizing the dynamics of the fully-actuated robot is derived from the previous work [Franchi, 2018]. This controller ensures, in nominal conditions, the tracking of a full 6D (position plus orientation) trajectory. Referring again to Figure 5.1, this reference trajectory is provided by the AF after filtering the desired motion commands generated by the HVS. Therefore, the controller receives the following quantities from the Admittance filter: the linear terms  $\mathbf{p}_B^r$ ,  $\dot{\mathbf{p}}_B^r$ ,  $\ddot{\mathbf{p}}_B^r$ , and the angular quantities  $\mathbf{R}_B^r$ ,  $\boldsymbol{\omega}_B^r$ ,  $\dot{\boldsymbol{\omega}}_B^r$ . As shown in Figure 5.3, this controller exploits a cascade structure and it comprises two parts, a translational and a rotational one.

The translational loop, shown on the top of Figure 5.3, computes the necessary thrust force  $\mathbf{f}_{\text{act}}^r \in \mathbb{R}^3$  to track the reference trajectory. In order to do so, it exploits the



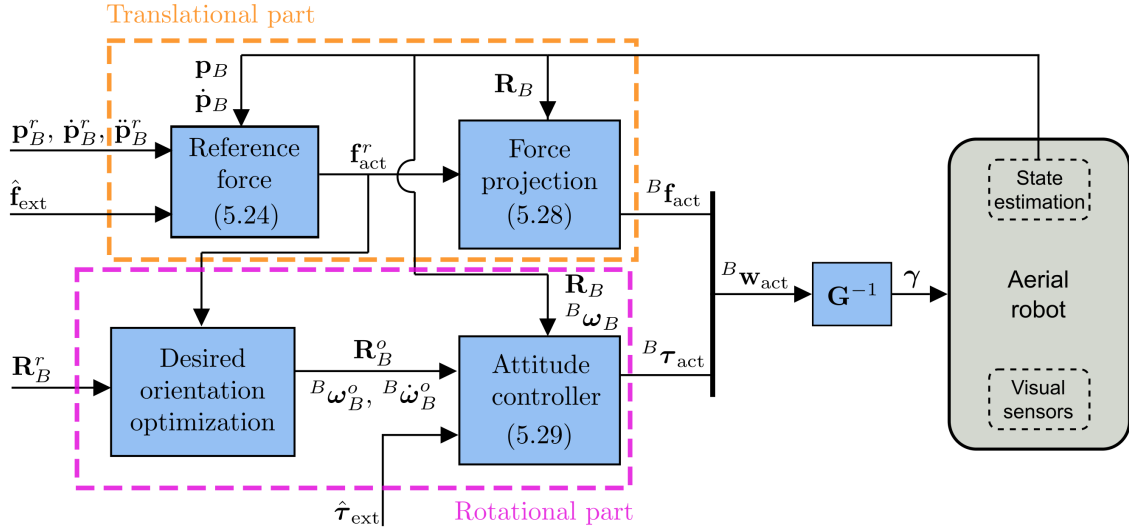


Figure 5.3: Schematic representation of the geometric controller adopted to stabilize the translational and rotational dynamics of the fully-actuated MRAV.

inversion of the translational part of the robot dynamic model, i.e. Equation (4.26c), as

$$\mathbf{f}_{\text{act}}^r = m_b \ddot{\mathbf{p}}_B^r + m_b g \mathbf{z}_W - \hat{\mathbf{f}}_{\text{ext}} - \mathbf{K}_{P,\mathbf{p}} \mathbf{e}_{\mathbf{p}} - \mathbf{K}_{D,\mathbf{v}} \mathbf{e}_{\mathbf{v}} - \mathbf{K}_{I,\mathbf{p}} \int_0^t \mathbf{e}_{\mathbf{p}}(\nu) d\nu, \quad (5.24)$$

where the matrices  $\mathbf{K}_{P,\mathbf{p}} \in \mathbb{R}_+^{3 \times 3}$  and  $\mathbf{K}_{D,\mathbf{v}} \in \mathbb{R}_+^{3 \times 3}$  are diagonal positive-definite gain matrices introducing a feedback term based on the tracking error. In Equation (5.24), the reference translational kinematic quantities  $\mathbf{p}_B^r$ ,  $\dot{\mathbf{p}}_B^r$  are used with the feedback variables  $\mathbf{p}_B$ ,  $\dot{\mathbf{p}}_B$  to compute the kinematic error quantities similarly to Equations (5.18)-(5.23). Therefore, we can write

$$\mathbf{e}_{\mathbf{p}} = \mathbf{p}_B - \mathbf{p}_B^r, \quad (5.25)$$

$$\mathbf{e}_{\mathbf{v}} = \dot{\mathbf{e}}_{\mathbf{p}} = \dot{\mathbf{p}}_B - \dot{\mathbf{p}}_B^r \quad (5.26)$$

Besides, compared to [Franchi, 2018], we add an integral action on the translational error with a diagonal positive-definite gain matrix denoted as  $\mathbf{K}_{I,\mathbf{p}} \in \mathbb{R}_+^{3 \times 3}$ . We will add a similar term also for the orientation control loop, as discussed later. Finally, in Equation (5.24), the estimated external force  $\hat{\mathbf{f}}_{\text{ext}} = \mathbf{R}_B^B \hat{\mathbf{f}}_{\text{ext}}$  appears, so that the requested thrust compensates that force whenever it is present.

In Equation (5.24), the quantity  $\mathbf{f}_{\text{act}}^r$  represents the forces, expressed in  $\mathcal{F}_W$ , that ideally one would like to apply to the aerial vehicle body to track the reference translational trajectory. Such a quantity is related to the body force  ${}^B \mathbf{f}_{\text{act}}$  through the platform orientation  $\mathbf{R}_B$ , precisely  $\mathbf{f}_{\text{act}}^r = \mathbf{R}_B {}^B \mathbf{f}_{\text{act}}$ . As discussed in Chapter 1, this class of ARs can exert only a limited amount of thrust along non-principal (lateral) body directions due to the spatial arrangement of their rotors and the motor limitations. This means that the  $x$  and  $y$  components of  ${}^B \mathbf{f}_{\text{act}}$ , i.e.  ${}^B \mathbf{f}_{\text{act}x} \in \mathbb{R}$  and  ${}^B \mathbf{f}_{\text{act}y} \in \mathbb{R}$ , are bounded, or alternatively

$$\begin{bmatrix} {}^B \mathbf{f}_{\text{act}x} \\ {}^B \mathbf{f}_{\text{act}y} \end{bmatrix} \in \mathcal{U}_{x,y} \subset \mathbb{R}^2, \quad (5.27)$$

where  $\mathcal{U}_{x,y}$  represents the bounds on the lateral forces. Such a subspace has different shapes according to the considered type of aerial vehicle and the size of the bounds. The interested reader is referred to [Franchi, 2018] for different graphical visualizations of the relationship in (5.27) according to different robot designs. Given a certain robot attitude  $\mathbf{R}_B \in SO(3)$ , the requested accelerations derived from  $\mathbf{f}_{\text{act}}^r$  may require an amount of lateral forces that violate the constraint above. As a result, despite being able to follow a 6D trajectory, this class of aerial robots cannot achieve *any* full-pose motion. If the lateral forces are exceeded, the controller could decide either to modify the platform orientation to match the requested linear accelerations in  $\mathcal{F}_W$ , or to jeopardize the positional tracking in favor of maintaining the same attitude.

The controller proposed in [Franchi, 2018] gives higher priority to the positional tracking by sacrificing the attitude in the case the requested lateral forces exceed the bounds. This choice is motivated by the fact that in a typical application a wrong positional tracking is more likely to lead to an obstacle collision than an imperfect orientation accuracy.

It can be proven that it exists a finite non-empty set of orientations that allow tracking any demanded force  $\mathbf{f}_{\text{act}}^r$  without violating the lateral bounds of the actuators. However, this set may or may not contain the reference orientation  $\mathbf{R}_B^r$  that one would like the aerial robot to attain. The rotational part of the controller, shown on the bottom of Figure 5.1, is in charge of finding the closest orientation to meet the desired linear acceleration requirement, when the latter is violating the constraint in Equation (5.27). At each time  $t$ , it selects an orientation  $\mathbf{R}_B^o \in SO(3)$  that (i) belongs to the set of orientations allowing to track the computed reference force  $\mathbf{f}_B^r$  (which in turn allows following the reference position trajectory), and (ii) minimizes a certain cost function w.r.t. the given reference orientation. In the previous work [Franchi, 2018], it is proven that the selected orientation  $\mathbf{R}_B^o$  will exponentially converge to the reference orientation  $\mathbf{R}_B^r$  if the reference position  $\mathbf{p}_B^r$  leads to a feasible  $\mathbf{f}_{\text{act}}^r$ . If the platform constraints are not met, then the closest orientation to the reference one will be selected, which allows tracking the reference position trajectory and satisfying the lateral bounds. In the optimization step, where the new orientation is computed, a new angular velocity  ${}^B\boldsymbol{\omega}_B^o$  and acceleration  ${}^B\dot{\boldsymbol{\omega}}_B^o$  is computed by adding a regularization term in the cost function.

Once the optimal orientation is found, the wrench to be applied on the robot body can be found by employing the following control laws

$${}^B\mathbf{f}_{\text{act}} = \text{sat}_{\mathcal{U}_{x,y}}\left(\left(\mathbf{f}_{\text{act}}^r \top \mathbf{R}_B \mathbf{x}_B\right)\mathbf{x}_B + \left(\mathbf{f}_{\text{act}}^r \top \mathbf{R}_B \mathbf{y}_B\right)\mathbf{y}_B\right) + \left(\mathbf{f}_{\text{act}}^r \top \mathbf{R}_B \mathbf{z}_B\right)\mathbf{z}_B, \quad (5.28)$$

$$\begin{aligned} {}^B\boldsymbol{\tau}_{\text{act}} = \mathbf{S}\left({}^B\boldsymbol{\omega}_B\right) \mathbf{J}_b {}^B\boldsymbol{\omega}_B - {}^B\hat{\boldsymbol{\tau}}_{\text{ext}} - \mathbf{K}_{P,\mathbf{R}}\mathbf{e}_{\mathbf{R}} - \mathbf{K}_{D,\boldsymbol{\omega}}\mathbf{e}_{\boldsymbol{\omega}} - \mathbf{K}_{I,\mathbf{R}}\mathbf{e}_I \\ - \mathbf{J}_b\left(\mathbf{S}\left({}^B\boldsymbol{\omega}_B\right) \mathbf{R}_B^\top \mathbf{R}_B^o {}^B\boldsymbol{\omega}_B^o - \mathbf{R}_B^\top \mathbf{R}_B^o {}^B\dot{\boldsymbol{\omega}}_B^o\right), \end{aligned} \quad (5.29)$$

where  $\text{sat}_{\mathcal{U}_{x,y}}(\bullet)$  is a saturation operator which guarantees that the output vector of the lateral forces belongs to  $\mathcal{U}_{x,y}$ . In Equation (5.29), similarly to Equation (5.24), the term  ${}^B\hat{\boldsymbol{\tau}}_{\text{ext}}$  appears so that the controller computes a torque that can also compensate any external moment applied on the EE, if present. The vectors  $\mathbf{e}_{\mathbf{R}} \in \mathbb{R}^3$  and  $\mathbf{e}_{\boldsymbol{\omega}} \in \mathbb{R}^3$  are the rotational and angular velocity errors and they are computed

similarly to Equations (5.19) and (5.20). Thus

$$\mathbf{e}_{\mathbf{R}} = \frac{1}{2} \left( \mathbf{R}_B^o \top \mathbf{R}_B - \mathbf{R}_B \top \mathbf{R}_B^o \right)^\vee, \quad (5.30)$$

$$\mathbf{e}_{\boldsymbol{\omega}} = {}^B \boldsymbol{\omega}_B - \mathbf{R}_B \top \mathbf{R}_B^o {}^B \boldsymbol{\omega}_B^o. \quad (5.31)$$

The matrices  $\mathbf{K}_{P,\mathbf{R}} \in \mathbb{R}_+^{3 \times 3}$ ,  $\mathbf{K}_{D,\boldsymbol{\omega}} \in \mathbb{R}_+^{3 \times 3}$  and  $\mathbf{K}_{I,\mathbf{R}} \in \mathbb{R}_+^{3 \times 3}$  are diagonal positive-definite gain matrices. For the integral error  $\mathbf{e}_I$ , we take inspiration from [Goodarzi, 2013], where it is defined as

$$\mathbf{e}_I = \int_0^t \mathbf{e}_{\boldsymbol{\omega}}(\nu) + c_2 \mathbf{e}_{\mathbf{R}}(\nu) d\nu, \quad c_2 \in \mathbb{R}^+. \quad (5.32)$$

After computing the actuator wrench  ${}^B \mathbf{w}_{\text{act}} = \left[ {}^B \mathbf{f}_{\text{act}} \top \quad {}^B \boldsymbol{\tau}_{\text{act}} \top \right]^\top \in \mathbb{R}^6$ , the motor-propeller commands can be computed by inverting Equation (4.49). Hence,

$$\boldsymbol{\gamma} = \mathbf{G}^{-1} {}^B \mathbf{w}_{\text{act}}. \quad (5.33)$$

where  $\boldsymbol{\gamma} \in \mathbb{R}^6$  ( $n_p = 6$ ) collects the forces generated by the robot actuators. Such an inversion is always possible as fully-actuated aerial vehicles have a full-rank and squared<sup>3</sup> allocation matrix [Hamandi, 2021]. Then, the propeller speeds can be obtained by employing Equation (4.57), i.e.  $\gamma_i = c_{\gamma,i} \Omega_{p,i}^2$ ,  $i \in \{1, \dots, 6\}$ .

## 5.2 Interaction with the environment

As mentioned in Chapter 3, the control scheme is validated in a pick-and-place application, where multiple brick-shaped objects are autonomously picked and placed at different locations by the AR. Hereby, we first provide more details about the experimental setup used in the validation campaign. Then, we present quantitative and performance results collected in the experimental campaign.

### 5.2.1 Validation setup

We employ a fixedly-tilted-propeller star-shaped hexa-rotor to conduct the experiments. A picture is provided on the left of Figure 5.4. This fully-actuated AR is a custom prototype called *FiberTHex* designed and realized in-house at LAAS, where the experiments are performed. It is sized to have a comfortable payload while providing enough lateral thrust to ensure rapid motion. The platform diameter is about 80cm and is actuated by 6 evenly-spaced 13in propellers. It is equipped with a monocular down-facing camera and an end effector capable of tightly gripping an object, e.g., a simple brick, in order to move it in the workspace. The wrench applied by the brick at the end effector is compensated by the controller using the aforementioned strategy, so no prior knowledge of the brick mass or inertia is required. The robot features an onboard Intel NUC, comprising an Intel Core i7-8565U and 8GB of DDR3 RAM, capable of running the image processing algorithms. It

<sup>3</sup>If the allocation matrix is not squared, one could employ the pseudo-inverse, as mentioned in Section 4.6.1.

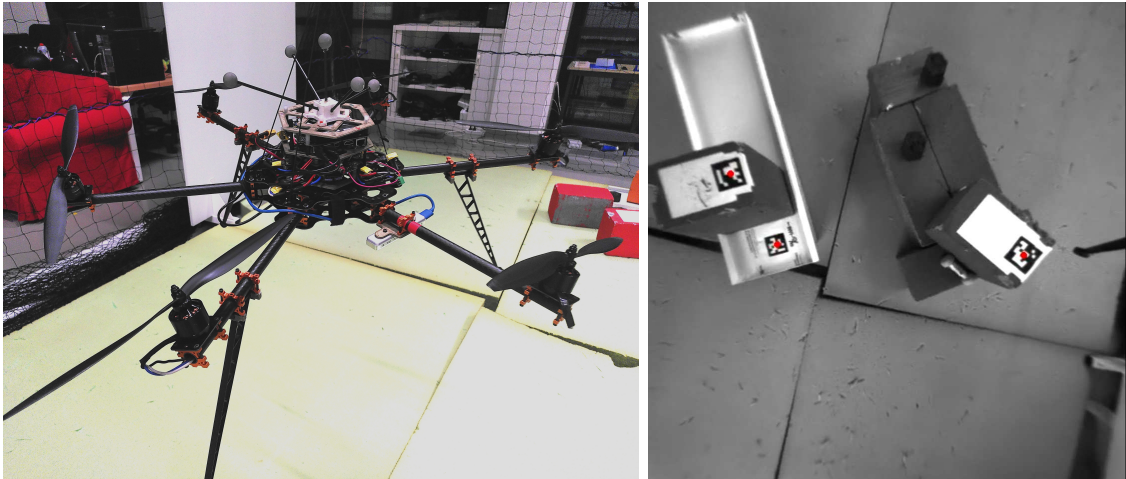


Figure 5.4: On the left, a picture of the fully-actuated hexa-rotor used in the experimental validation, while on the right a picture taken from the onboard camera. In the latter image, the bricks and the placing location are shown. Their successful detection is marked with a red dot displayed over their fiducial markers.

runs Ubuntu 18.04, and the software architecture is implemented in C++ using GenOM [Mallet, 2010], i.e. a middleware agnostic component generator. This tool allows creating software components that can be compiled for a given middleware, e.g., for the Robot Operating System (ROS). We make use of the TeleKyb3 software architecture for the state estimation as well as the low-level motor control, available within the OpenRobots project<sup>4</sup>. The robot state feedback is provided by an onboard IMU at 1kHz and an external Motion capture (Mocap) system at 100Hz. Sensor fusion of the available measurements is realized by means of an Unscented Kalman Filter (UKF), which provides the full state feedback at 1kHz. The onboard camera is an Intel Realsense T265, chosen for its lightweight and commodity, but its odometry feedback is not utilized. The integration of such a sensor in our architecture and the exploitation of its visual-inertial odometry measures for the robot localization and navigation are left for future work. The object detection is performed using Aruco fiducial markers [Garrido-Jurado, 2014], as shown on the right of Figure 5.4, for the sake of simplifying and abstracting the detection process. However, many MRAV-oriented detection algorithms exist in the literature, for instance, deep-learning-based ones, as presented in [Zhu, 2018; Akbari, 2021]. The detection algorithm identifies the Aruco tags in the camera images and then it provides an estimate of the 6D state of their central point knowing the geometrical dimensions of the fiducial markers.

To ensure a safe motion in the workspace, we impose without any loss of generality that the motion shall occur at a constant altitude  $z_B^r$ . We also impose to the motion controller to attain a null roll and pitch, i.e.  $\phi_B^r = \theta_B^r = 0$ , during the whole operation in order to exploit the full actuation property of the platform. Additionally, we enforce the associated  $\phi$  and  $\theta$  components of the angular velocity  $\boldsymbol{\omega}_B = [\omega_{B\phi} \ \omega_{B\theta} \ \omega_{B\psi}]^T$  to be zero in the HVS output, namely in Equation (5.15). Once the end effector is aligned with the object, the AR descends until sufficient force feedback is measured by the wrench observer along the vertical axis (i.e.  $\mathbf{z}_B$ ), ensuring contact for picking. Similarly, the placing operation is autonomously performed by visually servoing to

<sup>4</sup><https://git.openrobots.org/projects/telekyb3>.

the related location and employing contact feedback. Finally, in order to ensure the feasibility of the task when the features are not within the camera Field of View (FoV) or moving away from it, a position-based searching policy is implemented to scan the pre-defined area until the brick is found or its detection recovered. Then, the HVS controller is re-enabled. Similarly, the placing location is not known a priori and, if not already visible, it is searched and detected by adopting the same area-scanning routine.

### Reference values of the feature vector

As mentioned earlier, depending on the nature of the particular application considered, the reference values of the feature vector  $\mathbf{s}^r \in \mathbb{R}^6$  shall be chosen such that to align the end effector with the desired goal location. In our pick-and-place scenario, the gripper of our hexa-rotor shall meet the center of the object upper surface to perform a successful and reliable pick. Considering the situation schematically depicted in Figure 5.5, the previous requirement translates into having  ${}^S\mathbf{p}_F^r = {}^S\mathbf{p}_E$ , where  $\mathbf{p}_F^r$  is the reference value of the feature position and  $\mathbf{p}_E$  the robot EE position, both expressed in camera frame. Hence, by taking into account the kinematic linkage between the robot parts, we can write that

$${}^S\mathbf{p}_F^r = {}^S\mathbf{p}_E = {}^S\mathbf{p}_B + {}^S\mathbf{R}_B {}^B\mathbf{p}_E \quad (5.34)$$

$$= {}^B\mathbf{R}_S^\top (-{}^B\mathbf{p}_S) + {}^B\mathbf{R}_S {}^B\mathbf{p}_E^\top \quad (5.35)$$

$$= {}^B\mathbf{R}_S^\top ({}^B\mathbf{p}_E - {}^B\mathbf{p}_S). \quad (5.36)$$

In the equation above, the extrinsic camera parameters appear again, and the quantity  ${}^B\mathbf{p}_E \in \mathbb{R}^3$  represents the mounting location of the robot EE w.r.t. the robot body. In a similar fashion, to orient the end effector with the object, we can impose

$${}^S\mathbf{R}_F^r = {}^S\mathbf{R}_E = {}^B\mathbf{R}_S^\top {}^B\mathbf{R}_E, \quad (5.37)$$

where  ${}^B\mathbf{R}_E \in SO(3)$  represents the mounting orientation of the end effector expressed in  $\mathcal{F}_B$ . Recalling our choice for the feature selection, namely the normalized camera coordinates in Equation (5.3), we have that their reference values are obtained as

$$\tilde{x}^r = \frac{{}^Sx_F^r}{{}^Sz_F^r}, \quad \tilde{y}^r = \frac{{}^Sy_F^r}{{}^Sz_F^r}, \quad \tilde{z}^r = {}^Sz_F^r, \quad (5.38)$$

where the coordinates  ${}^Sx_F^r$ ,  ${}^Sy_F^r$  and  ${}^Sz_F^r \in \mathbb{R}$  are the components of the vector  ${}^S\mathbf{p}_F^r$ . Therefore, through Equations (5.36) and (5.38), we can obtain the reference values for the translation part of the feature vector, i.e. the vector  $\mathbf{s}_t^r$ . As mentioned earlier in Section 5.1.1, the orientation error  $\mathbf{e}_{s_r}$  is instead given by  $\theta\mathbf{u}$ , as  $\mathbf{s}_r^r = \mathbf{o}_3$ . To compute  $\mathbf{e}_{s_r}$ , we convert the error rotation matrix  $\tilde{\mathbf{R}} = {}^S\mathbf{R}_F^r {}^S\mathbf{R}_F^5$ <sup>5</sup> to its axis-angle representation, where  ${}^S\mathbf{R}_F$  is the current feature orientation in camera frame and  ${}^S\mathbf{R}_F^r$  the feature reference attitude obtained through Equation (5.37).

<sup>5</sup>The interested reader is referred to [Campa, 2009] for details regarding this orientation error for rotation matrices.

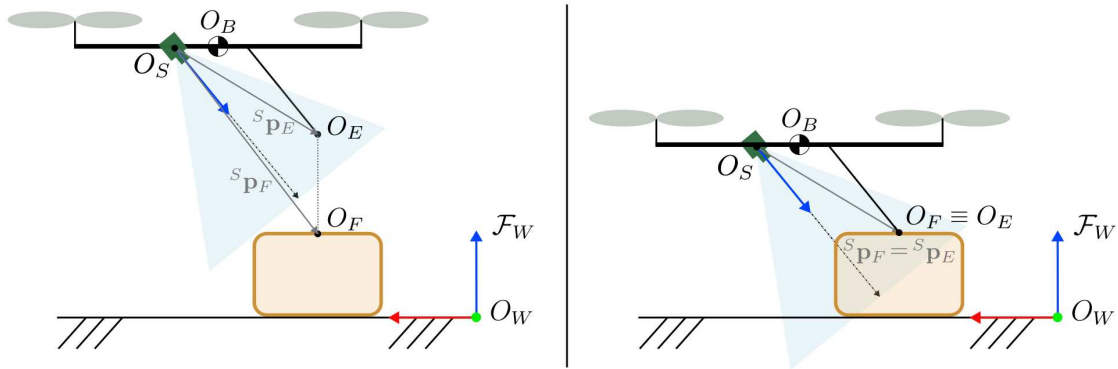


Figure 5.5: Schematic drawing representing the AR closing the distance with the object to pick, on the left, and during the contact phase, on the right. The origins of the main frames are drawn. For instance,  $O_F$  is the origin of the feature frame  $\mathcal{F}_F$ . The vectors  ${}^S\mathbf{p}_F$  and  ${}^S\mathbf{p}_E$  are the positions of the feature point and the robot EE tip expressed in  $\mathcal{F}_S$ , respectively.

### 5.2.2 Experiments

In this section, we present the results of two conducted experiments, each one requiring the robot to find a brick, pick it, and then place it at another location. We remind the reader that a video of the experiments discussed hereafter is available in Video 3.1 or at this link: <https://peertube.laas.fr/w/eEY4Q8nWdj1z5vs4JBaMpt>.

In the first experiment, we employ the full control architecture presented earlier, while in the second one, we disable the modules that handle the physical interaction, precisely the wrench observer and the admittance filter. These two cases aim to demonstrate the validity and effectiveness of our proposed methodology compared to classical control approaches, which do not explicitly take into consideration the physical interaction and rely solely on the disturbance-rejection capabilities of the controller [Gioioso, 2014]. In both experiments, we deactivate the integral action in the positional part of the geometric controller, i.e. we set  $\mathbf{K}_{I,p} = \mathbf{O}_3$  in Equation (5.24). Contrary, we keep it in the attitude part, namely in Equation (5.29), to highlight how the proposed control method can achieve accurate tracking of the reference motion.

In Table 5.1, we report the metrics computed on the results obtained from two experiments. They comprise the means and the standard deviations of the distance and rotation-tracking errors. For each sample of the robot motion, the error terms are calculated as the difference between the reference trajectory and the actual robot

	Full	No phy.
$\mu_d$ m	0.0512	0.1059
$\sigma_d$ m	0.0281	0.0498
$\boldsymbol{\mu}_\eta$ Deg	[0.2634 0.2875 2.8201]	[0.3059 0.2578 2.7623]
$\boldsymbol{\sigma}_\eta$ Deg	[0.2175 0.2235 2.1948]	[0.2063 0.1951 2.0716]

Table 5.1: Means and standard deviations of the distance error  $e_d$ , and rotation-tracking ones  $\mathbf{e}_\eta$ . The two columns refer to the two conducted experimental cases: using our entire proposed control architecture (“Full”) and the same one, but with the physical interaction module disabled (“No phy.”).

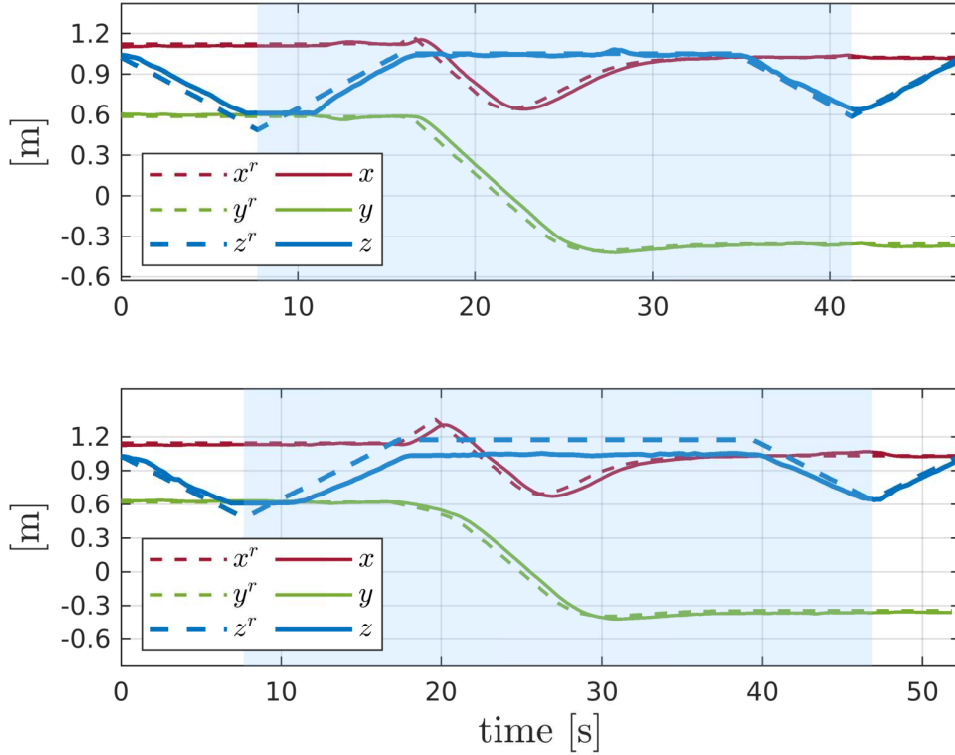


Figure 5.6: Position tracking during the two experiments. In dashed lines, the position references of the robot trajectory, while in continuous lines, the current robot coordinates. The blue area shows a phase of interest, as detailed in Section 5.2.2.

state (position  $\mathbf{p}_B$  or attitude in Euler angles  $\boldsymbol{\eta}_B \in \mathbb{R}^3$ ). Particularly, the distance error is then obtained as the (Euclidean) 2-norm of the position-tracking error. We denote with  $\mathbf{e}_{\mathbf{p},i} \in \mathbb{R}^3$  and  $\mathbf{e}_{\boldsymbol{\eta},i} \in \mathbb{R}^3$  the position and rotation-tracking errors at the  $i$ -th sample of the robot motion, respectively. Therefore, we have that

$$e_{d,i} = \|\mathbf{e}_{\mathbf{p},i}\|_2 = \|\mathbf{p}_{B,i}^r - \mathbf{p}_{B,i}\|_2, \quad \mu_d = \frac{1}{N_s} \sum_{i=1}^{N_s} e_{d,i}, \quad \sigma_d = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} e_{d,i}^2 - \mu_d^2}, \quad (5.39a)$$

$$\mathbf{e}_{\boldsymbol{\eta},i} = \boldsymbol{\eta}_{B,i}^r - \boldsymbol{\eta}_{B,i}, \quad \boldsymbol{\mu}_\eta = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{e}_{\boldsymbol{\eta},i}, \quad \boldsymbol{\sigma}_\eta = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{e}_{\boldsymbol{\eta},i} \mathbf{e}_{\boldsymbol{\eta},i}^T - \boldsymbol{\mu}_\eta \boldsymbol{\mu}_\eta^T}, \quad (5.39b)$$

where  $N_s \in \mathbb{N}_+$  is the number of samples, and  $e_{d,i} \in \mathbb{R}$  is the distance error for the  $i$ -th sample of the robot motion. The pairs  $\mu_d, \sigma \in \mathbb{R}$  and  $\boldsymbol{\mu}_\eta, \boldsymbol{\sigma}_\eta \in \mathbb{R}^3$  are, respectively, the mean and standard deviation of the distance error sequences, and those of the rotation-tracking errors.

The results in Table 5.1 prove that the adoption of the our control architecture (column “Full”) achieves better reference-tracking accuracy compared to a HVS scheme solely combined with a standard controller which is unaware of the physical interaction (column “No phy.”). During the second test, a larger mean and standard deviation of the distance error are obtained on average. On the contrary, almost identical attitude-tracking performance is achieved. To exploit the full actuation property of the platform, we impose it to maintain a flat orientation w.r.t. the ground.

The resulting lateral force required for the motion is not violating the virtual bounds of the actuation forces, thus an accurate orientation tracking is achieved in both cases. The larger position-tracking errors obtained during the second experimental test, can be better appreciated by observing Figure 5.6.

The plot on the top of Figure 5.6 shows the reference trajectory over the robot actual position during the first experiment, while the bottom one displays the same information but for the second experiment. In the region highlighted in light blue, it can be noticed how beneficial is the adoption of a control method which accounts for the physical interaction. During the second experiment, a classical controller (bottom plot) provides a noticeably larger deviation from the vertical motion reference  $z_B^r$ . Whereas, the tracking performances along the  $y$  and  $x$  components are less affected. Since our validation scenario consists of a pick-and-place operation, it is clear that the most stressed motion component is the vertical one, perpendicular to the ground. Along that direction, larger interaction forces are arising due to the picked payload, while smaller ones along the lateral directions. As a result, we expect that the classical controller would deviate more also along the  $x$  and  $y$  coordinates, if a task involving the exchange of noticeably larger forces along the lateral directions is envisioned.

In Figure 5.7, we provide the robot height from the ground (top), and the magnitude of the force exchanged between the robot and the brick along  $\mathbf{z}_B$  (bottom), during the first experiment only. In the regions highlighted in green, the picking phase occurs. In that stage, the robot  $z$  coordinate decreases down to the brick level (denoted as  $z_{B,\text{pick}}$ ), while the vertical component of the estimated contact force  $f_{\text{ext}_z}$  arises. The contact phase is detected by monitoring the vertical external force applied on the robot: when it exceeds the threshold of 2N, the physical interaction is taking place. After that phase,  $f_{\text{ext}_z}$  decreases as the robot and the brick are no longer in contact. The variable  $z_B$  restarts to increase as the robot is taking off with the picked object. In the regions highlighted in red, the placing phase is taking place. Contrary to the pick, the robot height firstly decreases to reach the placing location (denoted by  $z_{B,\text{place}}$ ), with a subsequent increase of the interaction force. Then, after the contact force has reached the pre-defined threshold and the brick has been placed, the robot takes off again. However, as the placing location is very close to the ground, the wind airflow generated by the rotating propellers is causing some *ground effect*, interacting with the robot by applying a vertical force on the platform. This can be clearly seen in Figure 5.7. The estimated external vertical force  $F_{\text{ext}_z}$  increases before the contact is established. As a consequence, the placing phase is much shorter than the picking one. In-between these two parts, while the robot moves with the collected payload towards the placing area,  $F_{\text{ext}_z}$  oscillates around a constant value of about 4N, which in turns corresponds to the object weight (mass of  $\approx 400\text{g}$ ).

In Figure 5.8, we illustrate the displacement of the detected brick in the image plane of the onboard camera with a continuous black line. Every 1s, the current brick position is drawn as a red dot, and its current heading is synthetically displayed through a purple segment. The adopted HVS scheme leads the robot to accurately move the robot EE on top of the object in order to permit its proper pick. As the brick approaches its reference position, denoted by the yellow squared marker, the camera orientation converges to the desired one, displayed as a green segment.



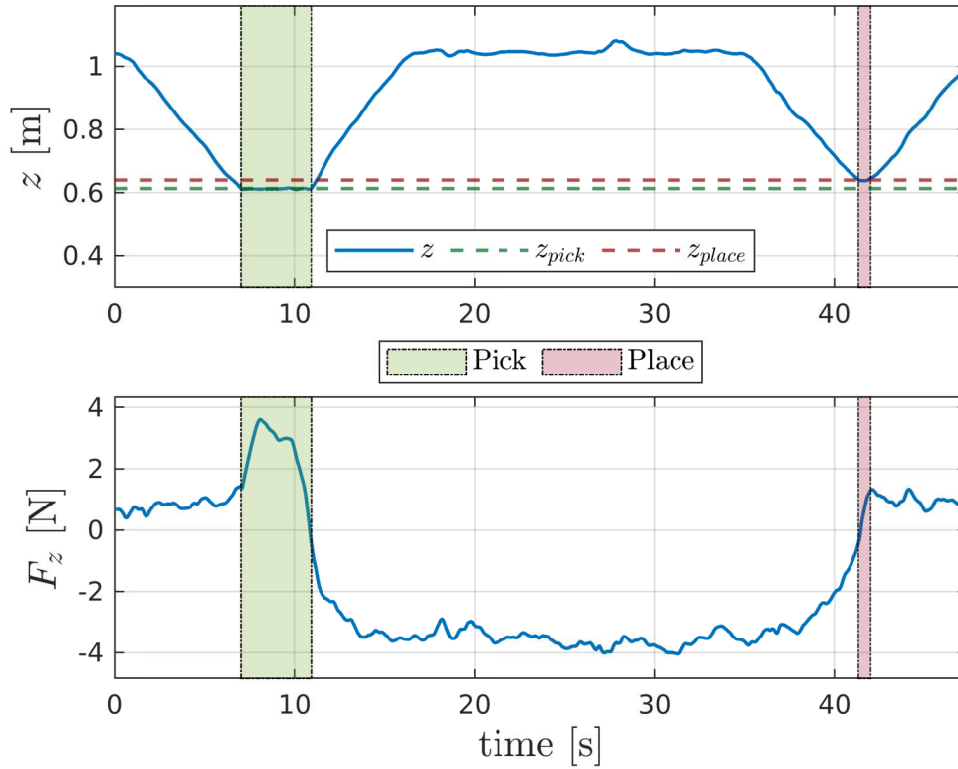


Figure 5.7: Plots related to the first experiment (“Full”) only. In the top, the robot height from the ground in  $\mathcal{F}_W$ , while in the bottom, the estimated contact force  ${}^B\hat{\mathbf{f}}_{\text{ext}z}$  along  $\mathbf{z}_B$ .

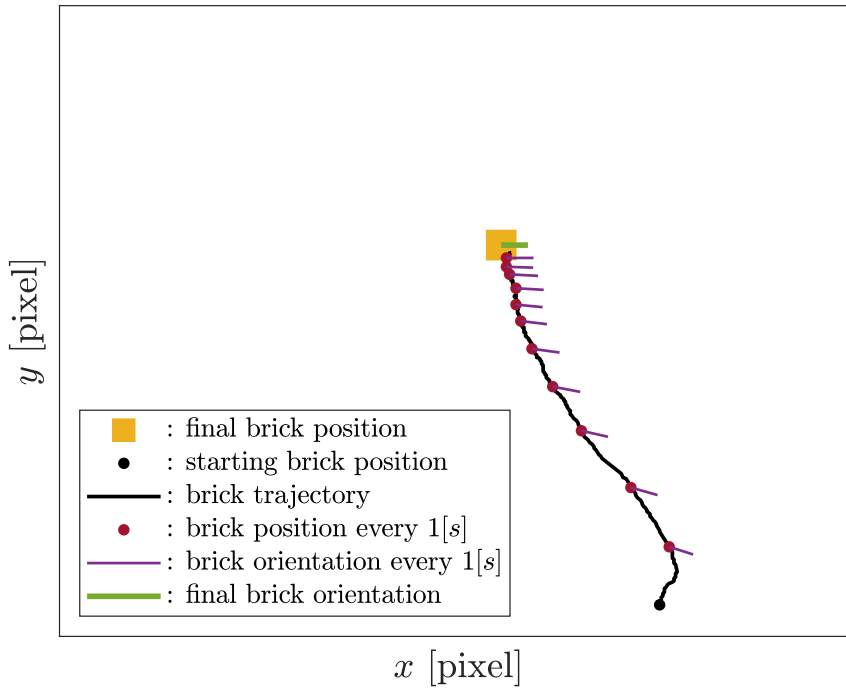


Figure 5.8: Figure related to the first experiment (“Full”) only. The displacement and heading of the detected brick in the image plane of the onboard camera.

The code related to the presented experiments is released open-source<sup>6</sup>.

## 5.3 Extension to aerial manipulators

In this section, we extend the control architecture presented earlier to achieve the second milestone of the same first subproblem. This time the goal is to achieve physical interaction between an aerial manipulator and a human worker.

### 5.3.1 Model of the aerial manipulator

Here, we present the dynamic model of the aerial manipulator (AM) employed in the design phase of the motion controller. We start by recalling the generalized coordinates, the twist and its derivative for an aerial manipulator, which are defined as follows:

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_b^\top & \boldsymbol{\xi}_a^\top \end{bmatrix}^\top = \begin{bmatrix} \mathbf{p}_B^\top & \mathbf{q}_B^\top & \mathbf{q}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{7+n_q}, \quad (5.40)$$

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta}_b^\top & \boldsymbol{\zeta}_a^\top \end{bmatrix}^\top = \begin{bmatrix} \mathbf{v}_B^\top & {}^B\boldsymbol{\omega}_B^\top & \dot{\mathbf{q}}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{6+n_q}, \quad (5.41)$$

$$\dot{\boldsymbol{\zeta}} = \begin{bmatrix} \dot{\boldsymbol{\zeta}}_b^\top & \dot{\boldsymbol{\zeta}}_a^\top \end{bmatrix}^\top = \begin{bmatrix} \dot{\mathbf{v}}_B^\top & {}^B\dot{\boldsymbol{\omega}}_B^\top & \ddot{\mathbf{q}}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{6+n_q}. \quad (5.42)$$

As the robot is composed of two subsystems, when considering the dynamic model in the canonical form as in Equation (4.23), we can decompose into two parts. Specifically, we get

$$\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{ba} \\ \mathbf{M}_{ab} & \mathbf{M}_{aa} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\zeta}}_b \\ \dot{\boldsymbol{\zeta}}_a \end{bmatrix} + \begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_a \end{bmatrix} + \begin{bmatrix} \mathbf{g}_b \\ \mathbf{g}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}_b \\ \boldsymbol{\lambda}_a \end{bmatrix}, \quad (5.43)$$

where  $\bullet_b$  and  $\bullet_a$  denote the parts related to the flying base and the robotic arm, respectively. In particular, the generalized inertia matrix comprises two terms related solely to the flying base, namely  $\mathbf{M}_{bb}$  and  $\mathbf{M}_{ba}$ . The former accounts exclusively for the inertial effects due to the dynamics of the MRAV, while the latter for the dynamic coupling with the robotic arm. Similar considerations apply to the other two terms, i.e.  $\mathbf{M}_{aa}$  and  $\mathbf{M}_{ab}$ , where the first one is related to the robotic arm only, while the second one to the coupling. Exploiting the factorization of the Coriolis vector, one can decompose such a term similarly in four terms, obtaining

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_a \end{bmatrix} = \mathbf{C}\boldsymbol{\zeta} = \begin{bmatrix} \mathbf{C}_{bb} & \mathbf{C}_{ba} \\ \mathbf{C}_{ab} & \mathbf{C}_{aa} \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_b \\ \boldsymbol{\zeta}_a \end{bmatrix}. \quad (5.44)$$

Likewise, we partition the gravitational vector as

$$\mathbf{g} = \begin{bmatrix} \mathbf{g}_b \\ \mathbf{g}_a \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{bb} + \mathbf{g}_{ba} \\ \mathbf{g}_{ab} + \mathbf{g}_{aa} \end{bmatrix}. \quad (5.45)$$

While considering the previous results, we select only the upper line of Equation (5.43) and, by rearranging the terms, we get

$$\mathbf{M}_{bb}\dot{\boldsymbol{\zeta}}_b + \mathbf{C}_{bb}\boldsymbol{\zeta}_b + \mathbf{g}_{bb} = \boldsymbol{\lambda}_b + \left(-\mathbf{M}_{ba}\dot{\boldsymbol{\zeta}}_a - \mathbf{C}_{ba}\boldsymbol{\zeta}_a - \mathbf{g}_{ba}\right) = \boldsymbol{\lambda}_b + \mathbf{w}_a, \quad (5.46)$$

<sup>6</sup><https://redmine.laas.fr/projects/visual-physical-control-architecture>.

where  $\mathbf{w}_a \in \mathbb{R}^6$  collects all the coupling effects induced by the robotic arm on the flying base. If an external wrench  $\mathbf{w}_{\text{ext}} \in \mathbb{R}^6$  is applied on the robot, then an additional term appears on the right-handed side of the equation above.

The terms on the left-handed side are related only to the dynamics of the flying base, thus we have that  $\mathbf{M}_{bb} = \mathbf{M}_{\text{GTMR}}$ ,  $\mathbf{C}_{bb}\boldsymbol{\zeta}_b = \mathbf{c}_{\text{GTMR}}$ ,  $\mathbf{g}_{bb} = \mathbf{g}_{\text{GTMR}}$ . Therefore, by recalling Equation (4.29), it results that

$$\begin{bmatrix} m_b \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{J}_b \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}}_B \\ {}^B \dot{\boldsymbol{\omega}}_B \end{bmatrix} + \begin{bmatrix} \mathbf{O}_3 \\ \mathbf{S}({}^B \boldsymbol{\omega}_B) \mathbf{J}_b {}^B \boldsymbol{\omega}_B \end{bmatrix} + \begin{bmatrix} m_b g \mathbf{z}_W \\ \mathbf{O}_3 \end{bmatrix} = \mathbf{w}_{\text{act}} + \mathbf{w}_a + \mathbf{w}_{\text{ext}}, \quad (5.47)$$

where we substituted  ${}^B \mathbf{p}_{\text{CoM}} = \mathbf{O}_3$ .

*Remark. Reference system.* It is important to point out that the wrench vectors on the right-handed side of the equation above are not expressed fully w.r.t.  $\mathcal{F}_W$ . While the composing forces are expressed in that frame, the torque components are instead defined w.r.t. the body coordinate system, i.e.  $\mathcal{F}_B$ . However, in the following, we will not use any particular superscript to maintain a clean notation.

*Remark. GTMR model.* If one imposes  $\mathbf{w}_a = \mathbf{O}_6$  in Equation (5.47), the dynamic model of just the GTMR is obtained, where

$$\boldsymbol{\lambda}_b = \begin{bmatrix} \mathbf{R}_B & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} {}^B \mathbf{w}_{\text{act}} = \mathbf{w}_{\text{act}}. \quad (5.48)$$

*Remark. Dynamic coupling.* The coupling wrench  $\mathbf{w}_a$  can be computed by means of the Recursive Newton-Euler Algorithm (RNEA) introduced in Chapter 4 by querying it with suitable inputs. The reader is referred to [Orsag, 2018] for more details.

### 5.3.2 Vision-based control

The HVS scheme presented previously allowed an AR to be visually controlled towards the object, either to pick or to place it in the surrounding environment. In a human-robot handover process, a similar vision-based control strategy can be adopted to control the object pose exploiting visual clues. During a H2R handover, the visual servoing could guide the AR to retrieve the object from the partner hand, in a similar way to the previous picking phase. Contrary, in a robot-to-human (R2H) handover, the pose of the object carried by the robot can be visually controlled and moved towards the human hand, performing an action similar to the placing step.

Since hereafter we focus on deriving a controller for achieving physical interaction between a dynamically more complex AR and a human partner, we neglect the presence of the HVS scheme. In the validation scenario, being a H2R handover, we assume that the final object position is known and the controller tracks an ad hoc motion reference. The AF will modify this reference trajectory according to the external wrench applied on the robot. We imagine that the final location may not be comfortable for the human operator, who may want to move the robot EE to a more preferable location. To do so, the user can deliberately apply a force on the robot along a certain axis to convey this intention to the other agent and make the robot move toward that direction.

### 5.3.3 Physical interaction control

For what concerns the interaction layer, the module structure remains unaltered, thus comprising again the same Wrench observer and Admittance filter introduced previously. The main exception is the dynamic model used in the wrench observer: now the estimator will employ the robot model presented earlier. In this way, it can discriminate the internal wrench caused by the dynamic coupling between the two subsystems and the one arising from the human-robot interaction.

#### Wrench observer

In this second milestone, we need to estimate the human wrench applied on the MRAV during the physical interaction. We extended the precedent formulation to account also for the wrench generated by the robotic arm. Therefore, we add to Equations (5.16) the coupling term  $\mathbf{w}_a = [\mathbf{f}_a^\top \ ^B\boldsymbol{\tau}_a^\top]^\top$  as follows

$$\begin{cases} \hat{\mathbf{f}}_{\text{ext}} = \int_0^t \mathbf{Q}_{I,\mathbf{f}} (m_b \ddot{\mathbf{p}}_B + m_b g \mathbf{z}_W - \mathbf{f}_{\text{act}} - \mathbf{f}_a - \hat{\mathbf{f}}_{\text{ext}}) d\nu \\ \left\{ \begin{array}{l} {}^B \hat{\boldsymbol{\tau}}_{\text{ext}} = \mathbf{Q}_{I,\boldsymbol{\tau}} \left( \mathbf{J}_b \ ^B \dot{\boldsymbol{\omega}}_B + \int_0^t \left( \mathbf{J}_b \mathbf{S}({}^B \boldsymbol{\omega}_B) \ ^B \boldsymbol{\omega}_B - {}^B \boldsymbol{\tau}_{\text{act}} - {}^B \boldsymbol{\tau}_a - {}^B \hat{\boldsymbol{\tau}}_{\text{ext}} \right) d\nu \right), \end{array} \right. \end{cases} \quad (5.49)$$

where we remove again the reference frame  $\mathcal{F}_W$  to simplify the notation. Compared to the estimator in Section 5.1.2, here, the force estimation is performed in the inertial reference frame.

#### Admittance filter

In the control scheme for the AM, the admittance filter, which is responsible for the robot interaction behavior, acts only on the MRAV trajectory. This choice is in contrast with the usual design of having the interaction control at the EE level for poly-articulated robots [Bascetta, 2013]. It is necessary to control individually the joint accelerations or the torques of the manipulator arm to successfully implement, respectively, an admittance or an impedance control at the robot end effector. As many aerial manipulator designs presented in the literature are still endowed with position or velocity-controlled robotic arms, they make the application of a pure admittance or impedance control scheme unfeasible. One would need a full torque-controlled AM, as the one recently proposed in [Martí-Saumell, 2023]. This motivate us to implement the AF at the robot base level, and we assume that the MRAV and the joint trajectories are available a priori and they correspond to a desired EE pose. Consequently, given the desired motion, the admittance filter computes a new reference trajectory for the MRAV based on the desired interaction dynamics and the knowledge of the external wrench applied by the human on the robot. Thus, the admittance filter has the same expression as before, i.e. the one given by Equation (5.17).

### 5.3.4 Motion control

As the robotic platform comprises a fully-actuated flying base endowed with an additional robotic arm, we need to replace the previous motion controller. Here, we rely on a *feedback linearization* approach which generates the commands for the robot actuators with the only prerequisite of knowing its dynamical model.

We start by defining a virtual input that is composed of a Proportional-Derivative (PD) and feed-forward term, and it has the objective of zeroing the tracking errors. This is achieved by computing the nominal robot linear and angular accelerations as

$$\dot{\zeta}_b^n = \begin{bmatrix} \ddot{\mathbf{p}}_B^n \\ {}^B\dot{\boldsymbol{\omega}}_B^n \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{P,\mathbf{p}}\mathbf{e}_\mathbf{p} + \mathbf{K}_{D,\mathbf{v}}\mathbf{e}_\mathbf{v} + \ddot{\mathbf{p}}_B^r \\ \mathbf{K}_{P,\mathbf{R}}\mathbf{e}_\mathbf{R} + \mathbf{K}_{D,\boldsymbol{\omega}}\mathbf{e}_\boldsymbol{\omega} + {}^B\boldsymbol{\omega}_B^r \end{bmatrix}, \quad (5.50)$$

where the error terms are computed as in Equations (5.25), (5.30), (5.26) and Equation (5.31). In Equation (5.50), the matrices  $\mathbf{K}_{P,\mathbf{p}}$ ,  $\mathbf{K}_{D,\mathbf{v}}$ ,  $\mathbf{K}_{P,\mathbf{R}}$ , and  $\mathbf{K}_{D,\boldsymbol{\omega}} \in \mathbb{R}_+^{3 \times 3}$  represent the gains of the PD controller.

Then, these virtual inputs are used within a feedback linearization scheme. Therefore, we invert the robot dynamic model obtained in Equation (5.47) as follows

$$\mathbf{w}_{\text{act}} = \begin{bmatrix} \mathbf{f}_{\text{act}} \\ {}^B\boldsymbol{\tau}_{\text{act}} \end{bmatrix} = \mathbf{M}_{bb}\dot{\zeta}_b^n + \mathbf{C}_{bb}\zeta_b + \mathbf{g}_b - \mathbf{w}_a - \hat{\mathbf{w}}_{\text{ext}}, \quad (5.51)$$

where  $\hat{\mathbf{w}}_{\text{ext}} = \begin{bmatrix} \hat{\mathbf{f}}_{\text{ext}}^\top & {}^B\hat{\boldsymbol{\tau}}_{\text{ext}}^\top \end{bmatrix}^\top \in \mathbb{R}^6$  is the external wrench computed by the WO. Lastly, by recalling Equation (4.49), we can compute the motor-propeller actuator commands as

$$\boldsymbol{\gamma} = \mathbf{G}^{-1} \begin{bmatrix} \mathbf{R}_B & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{w}_{\text{act}}, \quad (5.52)$$

where the inversion is always feasible thanks to the full actuation property of the aerial base, as  $\mathbf{G}$  is a square full-rank matrix.

#### Joint velocity controller

If the robotic arm is controlled by servo motors featuring a low-level position or velocity controller, it can be safely assumed that they can track the desired joint commands precisely, while rejecting external disturbances [Dietrich, 2016]. In mathematical terms this translates into having

$$\dot{\mathbf{q}}_a \approx \dot{\mathbf{q}}_a^d, \quad (5.53)$$

if the joints are velocity-controlled. Else, in the case of a low-level position controller, we have that

$$\mathbf{q}_a \approx \mathbf{q}_a^d. \quad (5.54)$$

Finally, since it is usually required to move the arm at low speeds in tasks requiring physical interaction, we can consider the joint accelerations  $\ddot{\mathbf{q}}_a \in \mathbb{R}^{n_a}$  to be null. Consequently, the coupling effects induced by the arm motion are treated as external disturbances at the control level of the aerial robot, while the robotic arm features its own low-level joint-level controller.

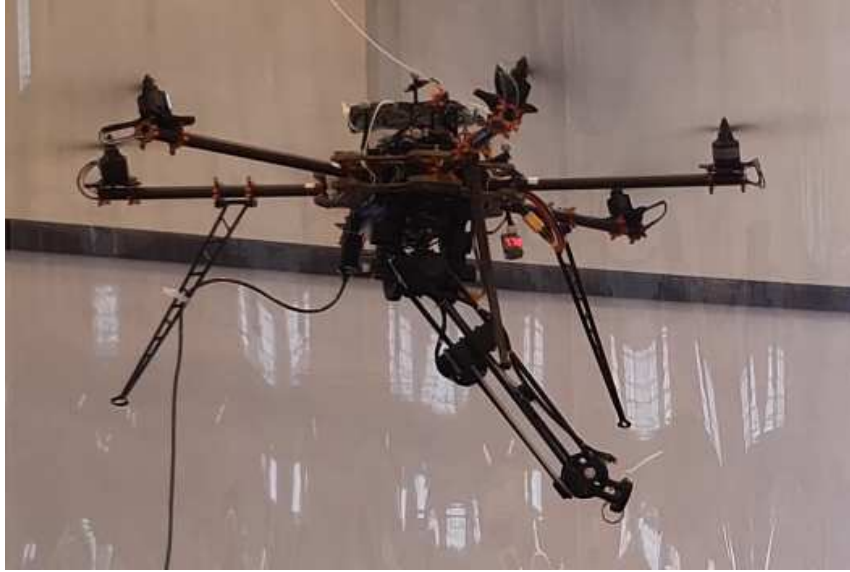


Figure 5.9: A picture of the aerial manipulator adopted in the H2R handover experiment.

## 5.4 Interaction with humans

In this section, we present the results obtained from the pHARI experiment presented in Chapter 3. To briefly recap, it consists of a H2R handover where a human worker placed on a scaffolding needs to return a tool to a robotic partner being an aerial manipulator. Hereby, we first describe the aerial robot employed in the validation campaign, and later we offer the technical outcomes deriving from the experimentation.

### 5.4.1 Validation setup

The MRAV comprises the same fully-actuated hexa-rotor detailed in Section 5.2.1, this time equipped with a 3-Degrees of Freedom (DoFs) manipulator driven by servo motors, specifically by Dynamixel smart actuators<sup>7</sup>. A picture of the prototype employed in the experimental campaign conducted in the University of Twente is available in Figure 5.9. The Dynamixel units feature both an internal velocity and a position-based controller which provides as feedback the measures of the joint angles and their angular rates thanks to the integrated encoders. During the approach and reach phases, we use precomputed joint trajectories to obtain the desired motor velocity commands driving the robot EE to the desired position. Throughout the physical interaction stage, we assume that the robotic arm is able to track precisely the set-points, canceling out the effects induced by the contact wrench. The custom-designed manipulator arm is composed of a differential gear providing two degrees of freedom at the shoulder level, as visible in the design visualization reported in Figure 5.10, and a pulley mechanism introducing an additional one at the elbow. The end effector consists in a hook-like device that allows the human operator to secure the tool to be handed over.

For conducting the experiments in a safe manner, the human worker wears

<sup>7</sup><https://www.robotis.us/dynamixel/>.

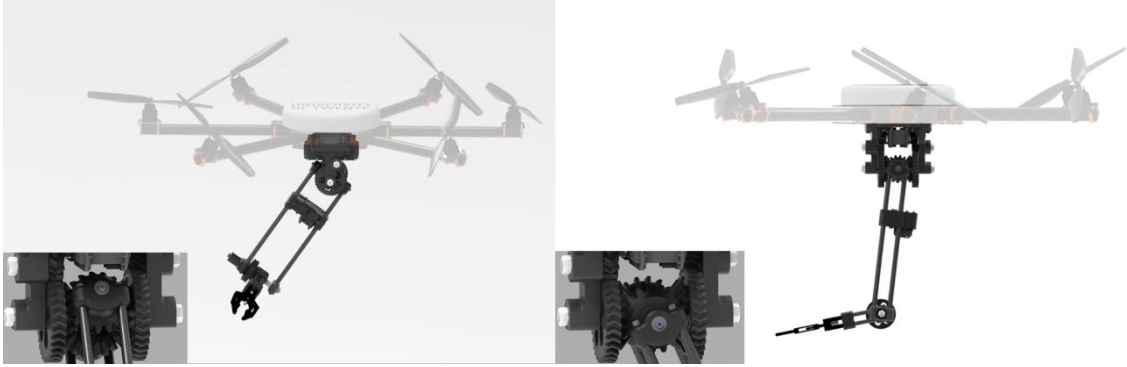


Figure 5.10: A picture of the aerial manipulator adopted in the H2R handover experiment. For visualization purposes, a hand gripper is mounted as end effector.

protective gear, such as an helmet with visor, gloves, and clothing that prevents leaving any body part from being exposed. Moreover, the robot is secured by means of a cable to the ceiling of the experimental room. In this way, if any technical issue occurs, the experiment can be terminated immediately and the robot turned off, without producing any potential danger for the human partner or damages to the platform. Furthermore, an operator placed at a ground station supervises the activities, operates the state machine of the software architecture through a joystick, and decides whether proceeding with the different states or terminating the experiments in case of any unexpected event.

The control method detailed in Section 5.3 is implemented in MATLAB and Simulink, and it interfaces to the real hardware by means of a local network. We rely again on the TeleKyb3 architecture for the robot state estimation and low-level interfaces to the hardware components. A Mocap, installed in the ceiling of the experimental room, is used to obtain the robot position and the attitude measurements at 100Hz. Other sensor measurements are available which comprise linear accelerations and angular velocities provided by an onboard IMU with a sampling frequency of 1kHz. As before, the sensor fusion process is performed by means of an UKF running at 1kHz.

### Deadzone strategy for the external wrench

As the WO module computes an estimation of the external wrench applied on the robot from the available sensor measurements, noise is naturally affecting its output. If the estimated forces and wrenches are directly fed to the admittance filter, it may influence the robot dynamics even though no real external action is applied on the vehicle body by the human. This results in unexpected and unwanted robot motions. To prevent this behavior, we apply a simple deadzoning strategy. We perform a test flight during which we do not apply deliberately any external wrench on the robot body. We record the output of the WO and, from the log files, we manually estimate the maximum and minimum values of each estimated force and torque component. We use these values to build deadzones for the output of the wrench observer. Therefore, the real output of the WO module, i.e. the pair  $\hat{\mathbf{f}}_{\text{ext}}^{\text{out}}$  and  $\hat{\boldsymbol{\tau}}_{\text{ext}}^{\text{out}}$ ,

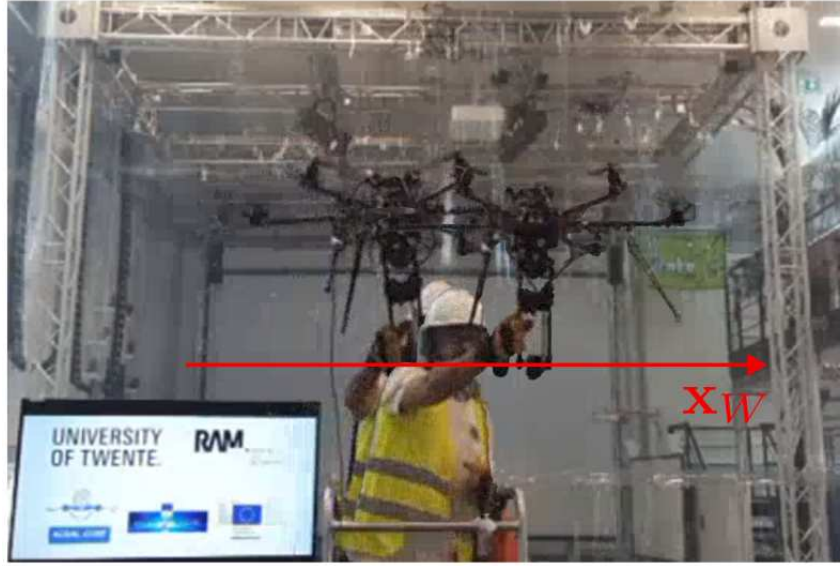


Figure 5.11: A picture of the H2R handover taken during the interaction phase. The direction along which the robot is compliant and the human can guide its EE is shown in red. It is parallel to  $\mathbf{x}_W$ . Along the other axes, the robot exhibits a stiff behavior.

is obtained as

$$\hat{\mathbf{f}}_{\text{ext}}^{\text{out}} = \begin{cases} \hat{\mathbf{f}}_{\text{ext}} - \bar{\mathbf{f}}_{\text{ext}} & \hat{\mathbf{f}}_{\text{ext}} \geq \bar{\mathbf{f}}_{\text{ext}}, \\ 0 & -\underline{\mathbf{f}}_{\text{ext}} < \hat{\mathbf{f}}_{\text{ext}} < \bar{\mathbf{f}}_{\text{ext}}, \\ \hat{\mathbf{f}}_{\text{ext}} + \underline{\mathbf{f}}_{\text{ext}} & \hat{\mathbf{f}}_{\text{ext}} \leq -\underline{\mathbf{f}}_{\text{ext}}, \end{cases} \quad (5.55)$$

$$\hat{\boldsymbol{\tau}}_{\text{ext}}^{\text{out}} = \begin{cases} \hat{\boldsymbol{\tau}}_{\text{ext}} - \bar{\boldsymbol{\tau}}_{\text{ext}} & \hat{\boldsymbol{\tau}}_{\text{ext}} \geq \bar{\boldsymbol{\tau}}_{\text{ext}}, \\ 0 & -\underline{\boldsymbol{\tau}}_{\text{ext}} < \hat{\boldsymbol{\tau}}_{\text{ext}} < \bar{\boldsymbol{\tau}}_{\text{ext}}, \\ \hat{\boldsymbol{\tau}}_{\text{ext}} + \underline{\boldsymbol{\tau}}_{\text{ext}} & \hat{\boldsymbol{\tau}}_{\text{ext}} \leq -\underline{\boldsymbol{\tau}}_{\text{ext}}, \end{cases} \quad (5.56)$$

where we denote with  $\bar{\mathbf{f}}_{\text{ext}}, \underline{\mathbf{f}}_{\text{ext}} \in \mathbb{R}_+^3$  and  $\bar{\boldsymbol{\tau}}_{\text{ext}}, \underline{\boldsymbol{\tau}}_{\text{ext}} \in \mathbb{R}_+^3$  respectively the lower and upper bounds of the external forces and torques obtained from the test flight mentioned above. The quantities  $\hat{\mathbf{f}}_{\text{ext}}^{\text{out}}$  and  $\hat{\boldsymbol{\tau}}_{\text{ext}}^{\text{out}}$  constitute the actual external wrench fed to the admittance filter, and the motion controller.

## 5.4.2 Experiments

In this section, we shortly recap the main stages of the pHARI experiment and then we analyze the collected results. Before continuing, we encourage the reader to watch again the video of the validation campaign, available in Video 3.2, or at this link: <https://youtu.be/LrQxXbQ5IHc>.

The experiment is divided into three main parts, namely the *approach-and-reach*, the *interaction* and the *retraction*, which corresponds to the handover phases described in Chapter 2. These stages also constitute the three cases of the state machine employed during the test, each one enabling or disabling certain functionality. In the experimental validation, we employ only the estimation of the external forces, while we neglect the use of the external torques, which is left as future work.



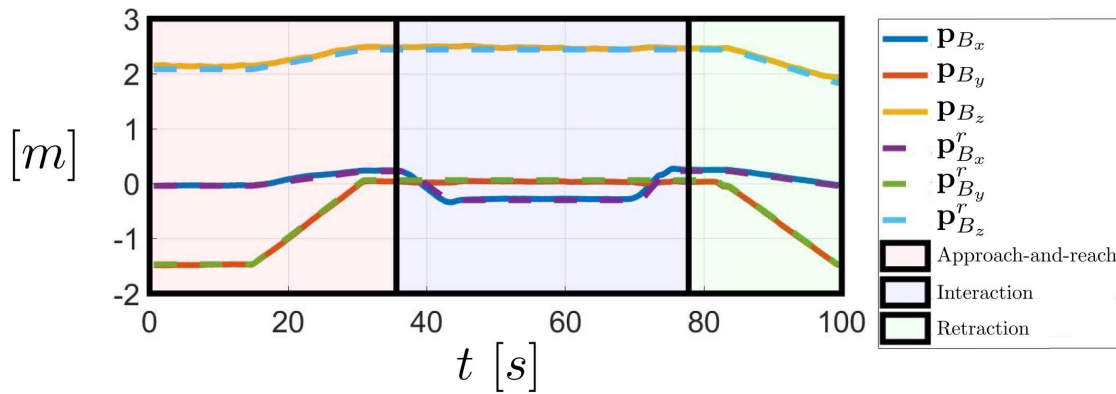


Figure 5.12: The plot is divided into three different highlighted regions representing the three phases of the experiment. Here, the actual position of the AM and its reference trajectory are displayed.

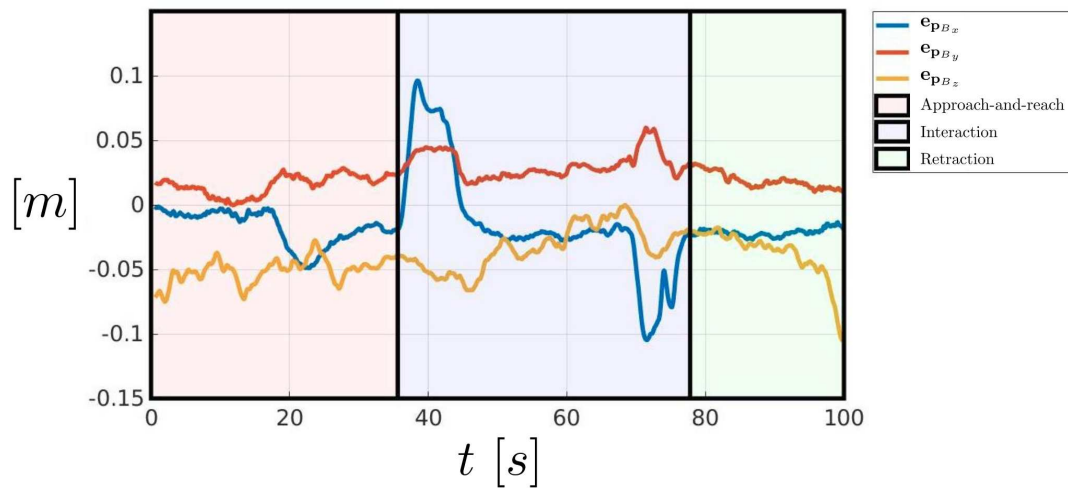


Figure 5.13: Here, we report the position errors of the aerial robot tracking the reference trajectory throughout the experiment. In general, the motion controller provides acceptable and satisfying tracking performance in all phases.

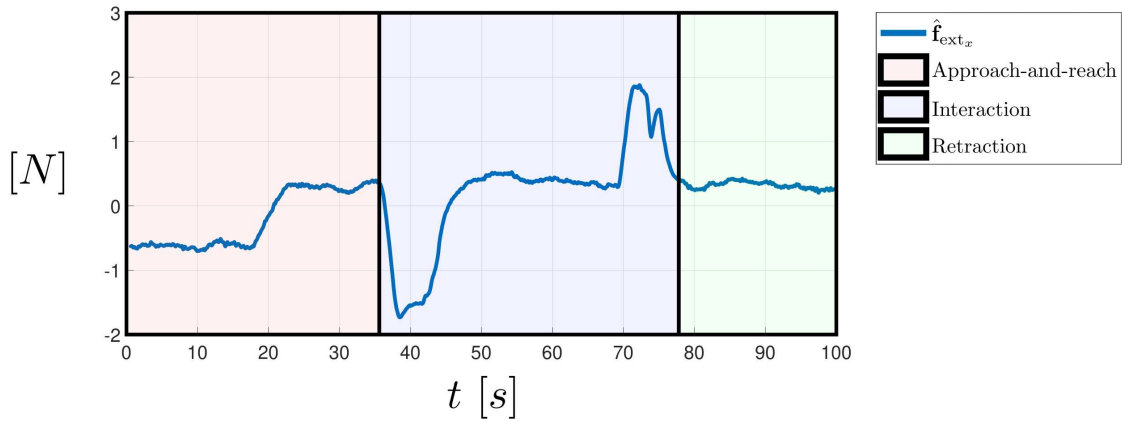
Accordingly, we make use of the deadzone technique only for the force output of the WO. We decide to use only the external forces as we believe that this would constitute the most natural and human-like type of interaction, where the robot can only be pushed or pulled by the partner along certain directions.

Initially, the AM, which is in the first phase (approach-and-reach), moves towards the worker and stops in front of the scaffolding where the user is located. As no contact is expected during the robot motion, the output of the wrench observer is not fed to the admittance filter, as we would like the robot to reject any possible unexpected external disturbance. The actual and reference position of the AR collected during this part are reported in Figure 5.12, respectively as continuous and dashed lines. In this phase, the robot motion controller is able to achieve good tracking performance of the reference set-points. The accuracy of the feedback linearization approach can be clearly appreciated in Figure 5.13, where we provide the trajectory tracking errors. There, it is possible to notice that the position error stays within  $\pm 5\text{cm}$  throughout this first phase. We consider this a satisfying result considering the mechanical complexity and the degree of technological realization of the robotic system in use.

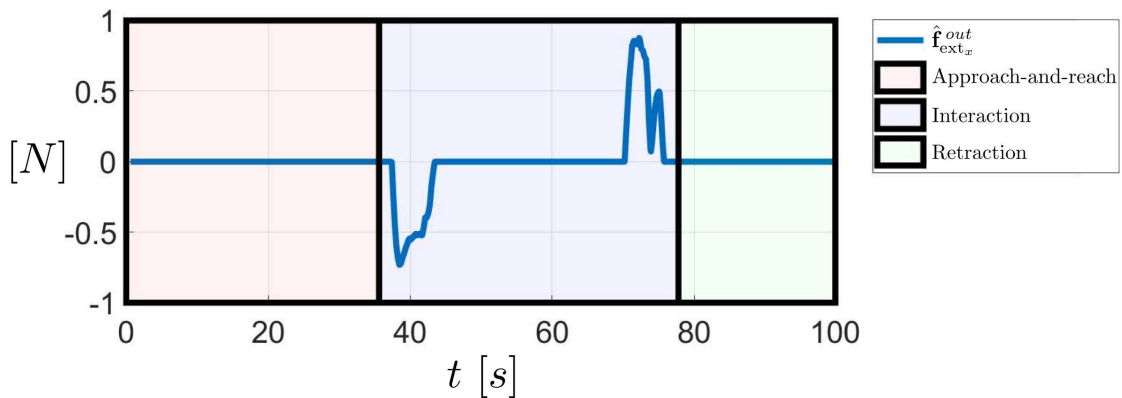
After the approach-and-reach phase, the human on the elevated structure signals the intention to physically interact with the robot, which marks the start of the contact stage. The state machine is changed into the interaction state. This means that the estimated external wrench is now fed to the admittance filter allowing the aerial robot to implement the desired admittance behavior given by the selected admittance parameters in Equation (5.17). In this application, the robot desired interaction behavior is to permit the human to manually guide and position its EE in the space. This is achieved by tuning properly the virtual-dynamics parameters and updating at each control iteration the desired aerial vehicle position with the current estimated position within the AF scheme. As mentioned earlier, the platform is anchored to the ceiling by means of a cable. Therefore, the experimental setup induces an external disturbance on the platform. We evaluated experimentally that the induced disturbances affect the robot mainly along the  $y$  and  $z$  axes of the inertial frame, and additionally the platform attitude. Therefore, in the experiment, the desired admittance behavior is only enabled along  $\mathbf{x}_W$ . Whereas, the robot exhibits a stiff behavior in the other DoFs. The  $x$ -axis of  $\mathcal{F}_W$  is parallel to the direction corresponding to the aerial vehicle moving sideways once in front of the human during the interaction phase, as shown in Figure 5.11. This choice is motivated by the fact that the robot is neither supposed to move away from the scaffolding, otherwise the human would not be able to hand over the object to the partner, nor too close for safety reasons and avoid potential collisions with the structure. For similar concerns, the robotic agent should not move excessively upwards and downwards to facilitate the human-robot interaction.

When the robot has reached the scaffolding, the human worker decides to reposition the robot EE by manually guiding it to a preferred position. By examining the interaction parts of the plots in Figures 5.12 and 5.14b, we see that the robot implements the desired admittance behavior in response to the estimated interaction wrench from approximately  $t > 35\text{s}$  to  $t \leq \approx 45\text{s}$ . We can see a change in the position reference in response to the estimated force provided to the AF, which computes a new reference trajectory for the flying base. Once the robot is positioned in a

comfortable location, the human worker proceeds to attach the tool to be handed over to the robotic agent end effector. Attaching the payload to the robot does not have any effect on the estimated external force, as shown in Figure 5.14b. As a result, the admittance filter does not change the reference set-point along  $\mathbf{x}_W$ . After the tool exchange, the human decides to move back the robot EE to the location where it was when the robot initially reached the elevated structure. Consequently, the operator applies another force on the robotic arm, which is hence followed by a subsequent robot re-positioning, as displayed in Figure 5.12 approximately from  $t \geq \approx 70$ s to  $t < 80$ s.



(a) Estimated force along  $\mathbf{x}_W$  originally provided by the wrench observer.



(b) Estimated force along  $\mathbf{x}_W$  after applying the deadzone strategy.

Figure 5.14: In these plots, we provide the estimated force along  $\mathbf{x}_W$ , which is the direction excited by the human during the interaction phase. Due to real system uncertainties, a non-zero force is estimated even when there is no contact, specifically during the approach-and-reach phases and the retraction part of the handover. This constitutes one downside of using a model-based wrench estimator. This problem is generally solved by using thresholds within a deadzone strategy. Here, we show the output of the wrench observer without (top) and with (bottom) using such a technique. The chosen deadzone thresholds for the force output are  $\bar{\mathbf{f}}_{\text{ext}} = \mathbf{f}_{\text{ext}} = [1 \ 0 \ 0]^T$ . As we consider only the forces along  $\mathbf{x}_W$ , we set only the first element different from zero in the vectors containing the threshold parameters.

The robot behavior during the interaction is properly achieved thanks to the use of the deadzoning strategy described in the previous section. As mentioned there, it allows removing the noise affecting the estimation of the external wrench which would have provoked unwanted and unexpected robot movements during the

interaction phase. In Figure 5.14a, we report the external force estimated without applying the deadzoning strategy, while in Figure 5.14b, the signal obtained after applying it. From these two pictures, it is evident how beneficial is the application of such a technique. However, using a deadzone brings some downsides which will be discussed in the final part of the chapter, precisely in Section 5.5.2.

Finally, once the interaction phase is concluded, the human signals the end of the handover to the supervisor who can initiate the retraction phase, as shown in Figure 5.12 at  $t > \approx 80$ s. During this last phase, the interaction control layer is disabled again, as the robot is supposed to move backwards and then land at the starting position.

## 5.5 Conclusion

In this section, we draw some final considerations regarding the work presented in this chapter. After a brief summary of the main key elements and results, we highlight potential limitations of the methodology we described earlier. Next, we conclude the chapter by discussing possible extensions of the control methods presented so far.

### 5.5.1 Synthesis

We opened the chapter by deriving the formulation of the hierarchical control architecture introduced in Chapter 3. Specifically, we showed how the HVS scheme generates the velocity commands driving the robot towards the desired location based on the selected feature information of the observed entities. We described the twofold functionality of the interaction control layer: (i) estimate the external wrench applied on the robot body, and (ii) shape the robot dynamics during the interaction. Finally, we unveiled the motion control law allowing the generation of motor commands letting a fully-actuated MRAV apply a 6D wrench on the environment. Then, we provided quantitative results showcasing the trajectory-tracking accuracy of the proposed control architecture compared to standard controllers not explicitly accounting for the physical interaction. The obtained results demonstrate the capability of this method to perform task requiring physical interaction. Additionally, we show that the vision-based control system can effectively drive the robot EE to the desired location by exploiting the visual clues provided by an onboard camera.

The features provided by this control framework are essential when considering scenarios where ARs are requested to interact and cooperate with human operators sharing their same workspace. Therefore, we extended the previous methodology to aerial manipulators in physical interaction with humans. As AMs feature an additional robotic arm, we replaced the geometric controller with a feedback linearization approach capable of generating motor commands both for the motor-propeller and the joint actuators. We validated this new architecture in a H2R handover experiment. The conducted test proved the viability of a new use case in aerial robotics, namely physical Human-Aerial Robot Interaction.

### 5.5.2 Limitations

The experimental results shed light on the potential limitations of the derived control framework, which we list hereafter.

The proposed modular architecture comprises several gains and parameters which require proper selection and adjustment. This might end up in a tedious and time-consuming activity. However, automatic strategies, as the one described in [Roveda, 2020], can be potentially envisioned to facilitate this process for the motion controller. However, the choice of the admittance parameters still requires manual effort and several trials to correctly shape the interaction behavior of the robot to match the desired one. Likewise, preliminary tests with the particular robotic platform in use are necessary for estimating the threshold values of the deadzone removing the noise affecting the output of the wrench observer. Additionally, the selection of the threshold parameters is important, as they affect the system sensitivity to the external wrench applied on the robot body. Too large values will require the user to apply very large forces on the robot, thus resulting in an increased physical effort from the human standpoint. At the same time, if the threshold values are selected too small, the output of the wrench observer may be still affected by some noise, which could make the robot move unintentionally.

Besides, the task has to be designed to be feasible both from the actuation and the perception standpoints. The motion controller considers only the virtual bounds on the lateral forces that the robotic platform can apply, while it neglects the real actuation mechanism responsible of the vehicle motion. As discussed in Chapter 4, the motor-propeller and joint actuation units can apply limited forces and torques. Their bounds are neither encapsulated by the virtual constraints considered by the geometric controller, nor accounted by the feedback linearization approach which can command an arbitrary motor wrench. From the perception point of view, the control architecture does not take into consideration any vision-based constraint, such as the loss of visibility of the observed entities. In the conducted experiments, the camera and the targets have to be placed and oriented in the workspace such that the accomplishment of the task is ensured.

Lastly, the human is just a source of external actions applied on the robot body that the controller has to treat either as a mean of conveying an intention (e.g. move to a certain position) or to reject as representing an unwanted interaction. As a matter of fact, the controller accounts neither for the human state nor for the user current activity. Hence, the current control framework can potentially lead to an ergonomic collaboration or an unacceptable collision between the two agents during the object handover.

The derivation of a control method including human-centric metrics, such as safety and ergonomics, and satisfying actuation and perception-based constraints will be analyzed in the next chapter.

### 5.5.3 Perspectives

We relied on an indirect force control method to estimate the external wrench applied on the robot by the surrounding environment. This has the benefit of not requiring

the integration of any auxiliary measurement device. However, the quality of the estimation provided by model-based methods is affected by the measurement noise and the model uncertainties. The addition of a force-torque sensor would allow to directly measure the external forces and torques and fuse them with the other data sources to mitigate these issue and improve the estimation process. Along this line, the availability of a more precise estimation of the external wrench may cause the deadzone strategy to be less necessary.

Nevertheless, in the case this thresholding technique is still required, the use of fixed values for the thresholds, if not well tuned, might lead the system to underrespond to the human actions or overrespond to the external disturbances. In future work, it can be interesting to investigate the use of a dynamic thresholding strategy where the bounds are mathematically derived w.r.t. the system state or the handover phase (e.g., approach, interaction, and retraction). This would make the system more adaptive to the task and it would preclude from performing an experimental campaign to empirically derive the thresholding values for each part of the operation.

Furthermore, in the H2R handover process, the robot approaches and reaches the human without exploiting the visual sensors which are usually available on-board MRAVs. Therefore, in future work, it could be investigated the use of a similar visual servoing scheme to visually guide the robot towards the operator and possibly control the EE position during the object exchange. A similar strategy is employed in [Costanzo, 2023] in order to achieve seamless and natural H2R and R2H handovers.

Lastly, the availability of a fully-torque-controllable aerial manipulator would allow the investigation of a pure impedance or admittance control framework, as the one proposed in [Dietrich, 2016]. This could allow obtaining a robot compliant behavior by taking advantage of the extra DoFs encompassed by the attached robotic arm, which are not actually exploited by our formulation.



# Chapter 6

## Predictive human-aware control

The goal of this chapter is to present an in-depth formulation and derivation of the human-aware predictive controller introduced in Section 3.1.2. This proposed methodology is designed to address the second subproblem described in Chapter 2. To briefly recap, we are interested in deriving a control approach that can include human-centric metrics, such as ergonomics and safety, while respecting the robot actuation limits and satisfying a set of task requirements (e.g., visibility).

Section 6.1 opens this chapter by offering an overview about the main concepts related to Model Predictive Control (MPC). Additionally, it introduces the general formulation of an Optimal Control Problem (OCP), and the technique we utilize to derive a Nonlinear Programming (NLP) problem. The MPC-based controller solves the NLP at a certain sampling frequency in real-time, and the solution comprises the optimal commands to be sent to the robot. Therefore, in Section 6.2, we detail each term composing the OCP presented in Chapter 3. We validate this method in a robot-to-human (R2H) handover operation. In Section 6.3, we present the results collected in a safe and simulated, but yet realistic, environment. Next, in Section 6.4, we analyze the comments received by the reviewers who evaluated our work for a prestigious journal. We provide their valuable feedback, and we analyze it in order to sketch possible improvements for our proposed controller. Afterwards, in Section 6.5, we detail the results obtained by testing our control algorithm in real-world experiments. Lastly, in Section 6.6, we conclude this chapter with some final thoughts on the proposed methodology and a discussion on the potential limitations.

### 6.1 Model Predictive Control

Model Predictive Control (MPC) is a control technique developed in the second half of the 20<sup>th</sup> century and it found great popularity and success in industrial applications [Schwenzer, 2021], such as the control of chemical reactors [Diehl, 2002] and food processing [Qin, 2003]. Later on, it started to be adopted in highly-demanding applications, for instance in the automotive [Swief, 2019] and aerospace sectors [Eren, 2017], and subsequently in the field of mobile and autonomous robotics, e.g. for aerial and legged robots [HNgyen, 2021; Katayama, 2023].

As mentioned in Chapter 3, this control method consists in the formulation of



an *Optimal Control Problem* (OCP), where a cost function is minimized and a set of constraints shall be satisfied. The solution to this problem is obtained by means of optimization process, where the future system behavior is predicted over a finite time window by exploiting the model knowledge. This allows obtaining a sequence of control actions over the finite-time prediction that lets the controlled dynamic system achieve the desired goal. In the practice, as the optimization problem is normally solved online at a given *sampling* frequency, only the part of the control plan related to the nearest future is actually used to control the system, while the rest discarded. Then, at the next sampling time instant, the computation is repeated again by initializing the optimization problem possibly with a new state feedback provided by the available sensors. This re-initialization procedure serves to mitigate possible deviations of the real system from the expected one predicted by means of its nominal model [Bicego, 2020]. The characteristic of repeatedly solving the optimization problem grants this control technique the alternative name of *Receding Horizon Control* (RHC) [Alessio, 2009]. Additionally, for its ability to naturally include constraints, MPC is also known in the literature as *constrained optimal control*.

As mentioned earlier, the solution is typically computed by finding *online* a numerical solution to the finite-horizon open-loop OCP. As the optimization procedure is usually computationally expensive, this results in mediocre running frequencies which are not suitable for controlling fast-varying dynamic systems. Moreover, the code implementing the solver might encounter numerical issues which, if not properly treated, jeopardize the system stability and they can additionally create software certification issues for safety-critical applications [Alessio, 2009]. The research community, to decrease the online computational burden, has developed a method to solve the optimization problem off-line so that the operations on-line reduce to a simple function evaluation. By exploiting multi-parametric programming techniques, it is possible to solve the optimization problem off-line for a given range of operating conditions of interest [Alessio, 2009]. The result is usually a lookup table of linear gains to be used on-line, which offers incredible fast software implementations. Nevertheless, as it requires the evaluation of the entire workspace of a given system and related application, it generates complex problems that admit feasible solutions only for small-scale systems and low-dimensional state spaces. Despite removing numerical issues of classic solvers and moving the optimization to a non-time-critical off-line calculation, it additionally requires larger memory sizes which are not always available on-board integrated processing units [Schwenzer, 2021]. Therefore, this type of MPC, denominated *explicit*, still finds marginal use compared to the standard (*implicit*) one where the solution is computed numerically and on-line. Moreover, the recent advances in embedded processors and the availability of methods to rapidly find a solution to the given OCP [Bock, 2007; Houska, 2011; Nurkanović, 2019; Gros, 2020] have enabled the application of MPC to the real-time control of fast-varying dynamic systems with high-sampling rates. Hence, in this thesis, we will employ an implicit solver for the OCP discussed later. The use of an explicit MPC technique might be considered in future work.

### 6.1.1 Optimal Control Problem formulation

Given a *time-invariant continuous-time system*, where  $t$  is the time, its dynamic behavior is typically described by a system of Ordinary Differential Equations (ODEs) with a certain initial condition, which can be written as

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{x}(t_0) = \mathbf{x}_0, \end{cases} \quad (6.1)$$

where  $\mathbf{x} \in \mathbb{R}^{n_x}$  and  $\mathbf{u} \in \mathbb{R}^{n_u}$  denote, respectively, the system state and input vectors, and the scalars  $n_x \in \mathbb{N}_+$  and  $n_u \in \mathbb{N}_+$  the number of state and input variables. The function  $\mathbf{f} : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_x}$  represents the system dynamic model and  $\mathbf{x}_0 \in \mathbb{R}^{n_x}$  is its initial condition at time  $t = t_0$ .

The solution computed by the MPC is the input vector that solves the following OCP [Bicego, 2019]:

$$\mathbf{u}^* = \arg \min_{\mathbf{x}, \mathbf{u}} J(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{l}_f(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} \mathbf{l}(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (6.2a)$$

$$s.t. \quad \mathbf{r}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{o}_{n_r}, \quad (6.2b)$$

$$\mathbf{s}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{o}_{n_s}, \quad (6.2c)$$

where the time window  $[t_0, t_f]$  is the *prediction horizon*, whose length is named *time horizon* and denoted as  $T_h \in \mathbb{R}_+$ . Therefore, the MPC solver computes the optimal control input vector  $\mathbf{u}^* \in \mathbb{R}^{n_u}$  as the system inputs that minimize the cost function  $J : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}$ , while satisfying the set of constraints. The objective function is composed of two parts. The first one, namely  $\mathbf{l} : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}$ , is the *running-cost* and it provides the cost along the time horizon. The second part, i.e.  $\mathbf{l}_f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ , is instead the *terminal cost* and it is related to the last time instant  $t_f$  of the time horizon. The set of constraints are composed of  $n_r \in \mathbb{N}$  *equality* and  $n_s \in \mathbb{N}$  *inequality* expressions. The former relationships, i.e. the equality constraints, are embodied by the function  $\mathbf{r} : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_r}$ , which normally include the system dynamics and the initial condition given by Equation (6.1). The inequality ones, instead, are synthetically represented by the function  $\mathbf{s} : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_s}$ , which usually comprise three types of constraints.

- The first one comprises the *input constraints* which act on the system control variables. They typically encompass the saturation of the system actuators and other physical limitations related to the actuation units.
- Then, we have the *state constraints*, which limit the search space of the system state variables. These bounds are usually related to restrictions on the robot operating workspace, e.g. joint limits, safety boundaries, or kinematic constraints related to the robot trajectory, like maximum and minimum velocities and accelerations.
- The third type includes the so-called *general constraints*, i.e. those bounds whose expression is function of the state and input variables, or just one of the two quantities. Examples of general constraints are the limited sensing capabilities of the onboard sensors [Jacquet, 2022b] or the formulation of collision-avoidance requirements [Lindqvist, 2020].

The OCP, which is given by the Equations (6.2), is iteratively solved at each *sampling time*  $T_s \in \mathbb{R}_+$ , and it provides the optimal control inputs  $\mathbf{u}^*$  to be applied to the robotic system.

*Remark. Parameters.* Additionally, one can parametrize the OCP in (6.2) by means of a vector of  $n_{pm} \in \mathbb{N}$  parameters, denoted as  $\mathbf{p}(t) \in \mathbb{R}^{n_{pm}}$ , resulting in

$$\mathbf{u}^* = \arg \min_{\mathbf{x}, \mathbf{u}} \quad J(\mathbf{x}, \mathbf{u}, \mathbf{p}) = l_f(\mathbf{x}, \mathbf{p}) + \int_{t_0}^{t_f} l(\mathbf{x}, \mathbf{u}, \mathbf{p}) dt \quad (6.3a)$$

$$s.t. \quad \mathbf{r}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \mathbf{o}_{n_r}, \quad (6.3b)$$

$$\mathbf{s}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \leq \mathbf{o}_{n_s}. \quad (6.3c)$$

In the OCP above, we have dropped the time dependency for simplicity of notation. The values of  $\mathbf{p}$  are already known when trying to solve the optimization problem, thus they do not constitute part of the solution computed by the numerical solver [Chen, 2019].

When either the cost function or the constraints, or both, have a nonlinear dependency on the state and input variables, the OCP in Equation (6.2) leads to a nonlinear optimization problem. As we will see later, this is the case for many robotic systems, since their dynamic model comprises a set of nonlinear equations. Therefore, from now on, we prefer to use the term Nonlinear Model Predictive Control (NMPC) rather than simply MPC to highlight the nonlinearity of the optimization problem.

## 6.1.2 Nonlinear Programming Problem

The well-known classification of ways for solving optimal control problems is mainly divided into *indirect* and *direct* methods<sup>1</sup> [Drag, 2016]. On the one hand, the former (indirect) optimal control methods, also known as *first-optimize-then-discretize* approaches, are characterized by the formulation of the infinite dimensional first-order necessary optimality conditions in order to solve the OCP [Quirynen, 2017]. However, they are hardly used in today's NMPC applications as they require particular attention when deriving the solution of the OCP [Yutao, 2019]. On the other hand, the direct methods reformulate the original infinite dimensional optimization problem adaptively into a finite-dimensional one through a parametrization of the controls and states [Bock, 2000]. For this characteristic, they are also known as *first-discretize-then-optimize* methods [Quirynen, 2017]. The resulting optimization problem can be efficiently solved by numerical solvers, whose fast development in the recent years contributed to the popularity of this type of approaches [Yutao, 2019]. For this reason, in this thesis, we will consider only direct methods, and particularly the popular *multiple shooting* approach. However, as the scope of this thesis is not the one of contributing to these methods, we will not provide all the details regarding its formulation, for which we refer the interested reader to [Yutao, 2019].

By applying the direct *multiple shooting* method, the time horizon  $T_h$  is discretized over  $N$  samples, known as *shooting points*, and the time-continuous OCP is

<sup>1</sup>However, a third type of solving OCPs exists which is known as *Dynamic Programming* (DP). The interested reader is referred to [Bellman, 1957; Bertsekas, 2010] for more details about this third approach.

transformed into a *Nonlinear Programming* (NLP) having the following form:

$$\min_{\substack{\mathbf{x}_0 \dots \mathbf{x}_N \\ \mathbf{u}_0 \dots \mathbf{u}_{N-1}}} J_N(\mathbf{x}_N, \mathbf{p}_N) + \sum_{k=0}^N J_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k) \quad (6.4a)$$

$$s.t. \quad \mathbf{r}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k) = \mathbf{o}_{n_r}, \quad k \in \{0, \dots, N-1\} \quad (6.4b)$$

$$\mathbf{r}_N(\mathbf{x}_N, \mathbf{p}_N) = \mathbf{o}_{n_{rN}}, \quad (6.4c)$$

$$\mathbf{s}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k) \leq \mathbf{o}_{n_s}, \quad k \in \{0, \dots, N-1\} \quad (6.4d)$$

$$\mathbf{s}_N(\mathbf{x}_N, \mathbf{p}_N) \leq \mathbf{o}_{n_{sN}}. \quad (6.4e)$$

In the optimization problem above, we use the subscript  $k$  to denote the discretized version of the variable evaluated at the  $k$ -th shooting point. Thus, if we consider for instance the state vector, we have that  $\mathbf{x}_k = \mathbf{x}(kT_{st})$ ,  $\forall k \in \{0, \dots, n\}$ , where  $T_{st} \in \mathbb{R}_+$  is the *shooting time*, i.e. the time within each shooting point (or equivalently the discretization time). where the equality and inequality constraints have been split into two, since the input vector is defined up to  $k = N - 1$ . Consequently, we have  $n_r \in \mathbb{N}_+$  equalities and  $n_s \in \mathbb{N}_+$  inequalities for all the shooting points within the time horizon, and  $n_{rN} \in \mathbb{N}_+$  plus  $n_{sN} \in \mathbb{N}_+$  constraints for the last shooting point of the sequence.

Typically, the cost function represents the need to drive certain system output variables to their desired values. If we denote with  $\mathbf{y} \in \mathbb{R}^{n_y}$  the outputs of the considered dynamic system, the objective function is typically given by the weighted squared 2-norm of the difference between the output vector and its reference one, denoted as  $\mathbf{y}^r \in \mathbb{R}^{n_y}$ . Moreover, to prevent the NMPC controller from stressing excessively the actuators, a similar regularization term is usually added to penalize large actuation actions. Therefore, the minimization in (6.4a) can be rewritten as

$$\min_{\substack{\mathbf{x}_0 \dots \mathbf{x}_N \\ \mathbf{u}_0 \dots \mathbf{u}_{N-1}}} \sum_{k=0}^N \|\mathbf{y}_k - \mathbf{y}_k^r\|_{\mathbf{W}_y}^2 + \sum_{k=0}^{N-1} \|\mathbf{u}_k\|_{\mathbf{W}_u}^2 \quad (6.5)$$

where the matrices  $\mathbf{W}_y \in \mathbb{R}^{n_y \times n_y}$  and  $\mathbf{W}_u \in \mathbb{R}^{n_u \times n_u}$  collect the weights for the output and input variables respectively. The typical choice is to select the coefficients such that those matrices are semi-definite positive and diagonal. The output vector is given as

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k) \in \mathbb{R}^{n_y}, \quad \forall k \in \{0, \dots, N-1\}, \quad (6.6)$$

$$\mathbf{y}_N = \mathbf{h}_N(\mathbf{x}_N, \mathbf{p}_N) \in \mathbb{R}^{n_{yN}}. \quad (6.7)$$

The scalars  $n_y \in \mathbb{N}_+$  and  $n_{yN} \in \mathbb{N}_+$  denote, respectively, the number of the system output variables during the horizon and at its end, and they are not necessarily equal. The functions  $\mathbf{h}_k : \mathbb{R}^{n_x \times n_u \times n_{pm}} \rightarrow \mathbb{R}^{n_y}$  and  $\mathbf{h}_N : \mathbb{R}^{n_x \times n_{pm}} \rightarrow \mathbb{R}^{n_{yN}}$  provide the system output variables.

## 6.2 Human-aware NMPC formulation

In this section, we mathematically describe each term of the OCP qualitatively introduced in Chapter 3. For convenience, we report here the same picture in Figure 6.1.

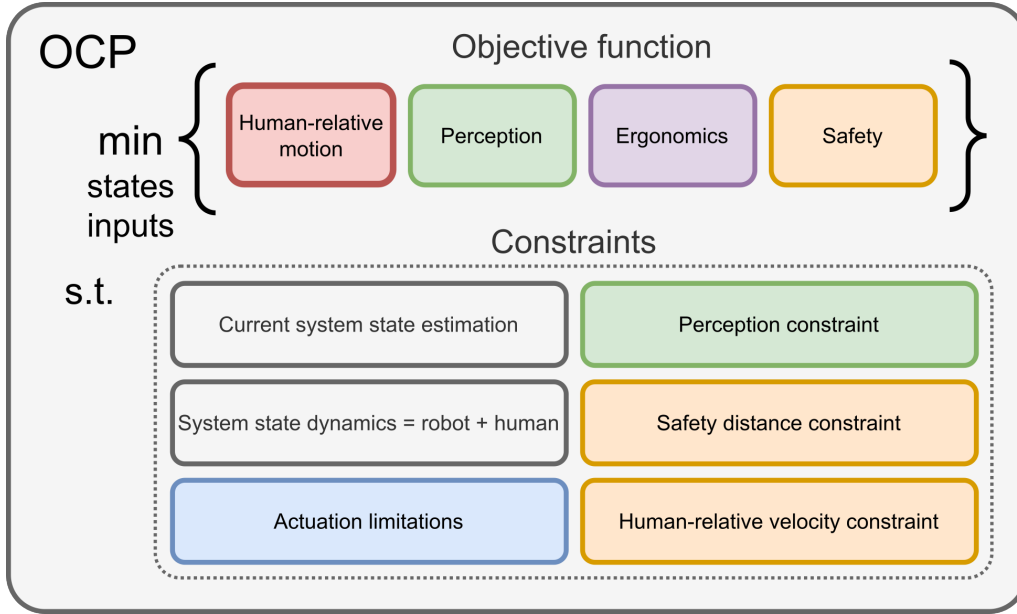


Figure 6.1: The visual description of the OCP already presented in Chapter 6, and specifically in Figure 3.2.

Once, every term is derived, we will arrive to a NLP as the one in Equations (6.4), but with the cost function in the same form as in Equation (6.5).

As discussed in Chapter 3, the OCP in Figure 6.1 aims to successfully hand over an object handover to a human coworker by means of an multi-rotor aerial vehicle (MRAV), acting as the giver, while guaranteeing human safety and accounting for the user ergonomics. To achieve the goal, the controller has to: (i) execute a human-centric motion that allows approaching the receiver; (ii) guarantee the safety of the worker, thus avoiding unwanted collisions while handing over the carried tool; (iii) evaluate the human articular stress to determine the most ergonomic handover location; (iv) constantly observe the human, to maintain visibility of its partner trunk; and (v) stabilize the robot dynamics by generating torque-level commands, which are compatible with its actuation limitations. Hereby, we first define the state and input variables. Then, we detail all the terms composing the objective function, and the set of constraints incorporated in the optimization problem. In this way, in Section 6.2.5, we can collect all the terms previously introduced, and consequently compose the NLP.

### 6.2.1 Definition of system state, input and parameter variables

The system under consideration is composed of the two agents involved in the Human-Aerial robot handover (HARH): the human as the receiver, and the aerial robot (AR) as the giver, the latter being a Generically-Tilted Multi-Rotor (GTMR) MRAV. By recalling the GTMR and the human state-space formulations introduced in Section 4.7.1 and Section 4.9.1, respectively, we can define the *system state vector* as the combination of the robot and human states, namely as

$$\mathbf{x} = \left[ \mathbf{x}_r^\top \quad \mathbf{x}_h^\top \right]^\top \in \mathbb{R}^{n_x}. \quad (6.8)$$

We conveniently report here the robot and human states:

$$\mathbf{x}_{\text{GTMR}} = \mathbf{x}_r = \left[ \mathbf{p}_B^\top \quad \mathbf{q}_B^\top \quad \mathbf{v}_B^\top \quad {}^B\boldsymbol{\omega}_B^\top \quad \boldsymbol{\gamma}^\top \right]^\top \in \mathbb{R}^{13+n_p}, \quad (6.9)$$

$$\mathbf{x}_h = \left[ \mathbf{p}_{H_t}^\top \quad \psi_{H_t} \right]^\top \in \mathbb{R}^4, \quad (6.10)$$

hence  $n_x = 17 + n_p$ . Furthermore, we recall that  $\mathbf{p}_B \in \mathbb{R}^3$ ,  $\mathbf{q}_B \in \mathbb{R}^4$  and  $\mathbf{p}_{H_t} \in \mathbb{R}^3$ ,  $\psi_{H_t} \in \mathbb{R}$  constitute the robot and human poses in  $\mathcal{F}_W$ , and the pair  $\mathbf{v}_B \in \mathbb{R}^3$ ,  ${}^B\boldsymbol{\omega}_B \in \mathbb{R}^3$  groups the robot linear and angular velocities, and  $\boldsymbol{\gamma} \in \mathbb{R}^{n_p}$  the propeller forces generated by the  $n_p \in \mathbb{N}_+$  robot actuators. The inclusion of the human trunk coordinates within the system state allows the NMPC controller to predict the user motion along the prediction horizon. As we will show later, this offers the robot more reactivity to the partner displacements during the handover process.

Next, we define the *system inputs* as the control commands to drive the robot actuators. According to the discussion in Section 4.7.1, as we included the propeller forces as part of the state vector, we select as system inputs the first-order derivatives of the forces generated by the motor-propeller units. Thus, we can write that

$$\mathbf{u} = \mathbf{u}_r = \mathbf{u}_{\text{GTMR}} = \dot{\boldsymbol{\gamma}} \in \mathbb{R}^{n_p}, \quad (6.11)$$

leading to  $n_u = n_p$ .

Lastly, we consider the human trunk velocities  $\mathbf{v}_{H_t} \in \mathbb{R}^3$  and  $\omega_{H_t} \in \mathbb{R}$  as the *system parameters*, hence recalling Equations (4.71) we have that and (4.72)

$$\mathbf{p} = \mathbf{u}_h = \left[ \mathbf{v}_{H_t}^\top \quad \omega_{H_t} \right]^\top \in \mathbb{R}^4, \quad (6.12)$$

which implies that  $n_{pm} = 4$ . Therefore, their values is known when computing the solution of the NLP problem and, in our case, they will be estimated by the robot on-board vision system, as we will describe later in Section 6.3.2.

## 6.2.2 Objective function

As clear from Figure 6.1, the cost function is composed of four terms, namely (i) a *human-relative motion* task, (ii) a *perception* objective, (iii) a *ergonomic* goal, and lastly (iv) the *safety* task.

### Human-relative motion

ARs are typically requested to follow a sequence of waypoints specified w.r.t. an inertial reference frame to accomplish a given task. However, in Human-Robot Interaction (HRI) scenarios, and especially during the exchange phase of a handover, the robot is typically maintaining a certain relative position and orientation, in conjunction with a precise velocity profile, w.r.t. its human partner [Strabala, 2013]. Hence, to ensure that the aerial robot tracks a human-relative trajectory, we introduce a motion term expressed in the human trunk frame  $\mathcal{F}_{H_t}$  as part of the cost function.

We start by deriving the human-relative coordinates of the robot expressed in  $\mathcal{F}_{H_t}$ , which are given by

$${}^{H_t}\mathbf{p}_B = \mathbf{R}_{H_t}^\top (\mathbf{p}_B - \mathbf{p}_{H_t}), \quad (6.13a)$$

$${}^{H_t}\mathbf{R}_B = \mathbf{R}_{H_t}^\top \mathbf{R}_B. \quad (6.13b)$$

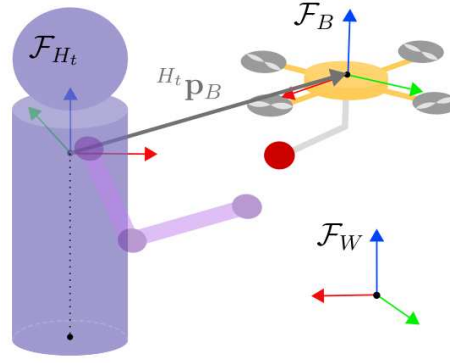


Figure 6.2: This picture shows the vector representing the robot position expressed in the human-trunk frame  $\mathcal{F}_{H_t}$ , i.e., the quantity  ${}^{H_t}\mathbf{p}_B$ .

The rotation matrices  $\mathbf{R}_B \in SO(3)$  and  $\mathbf{R}_{H_t} \in SO(3)$  represent the robot and human orientations expressed in the inertial frame. An graphical representation of the vector  ${}^{H_t}\mathbf{p}_B$  is given in Figure 6.2.

Then, we compute the derivatives of the equations above, which result into the relative robot linear and angular velocities expressed in  $\mathcal{F}_{H_t}$ . Therefore, we can write that

$${}^{H_t}\mathbf{v}_B = \mathbf{R}_{H_t}^\top \left( \mathbf{v}_B - \mathbf{v}_{H_t} - \mathbf{S}(\boldsymbol{\omega}_{H_t}) \mathbf{R}_{H_t} {}^{H_t}\mathbf{p}_B \right), \quad (6.14a)$$

$$\mathbf{S}({}^{H_t}\boldsymbol{\omega}_B) = {}^{H_t}\mathbf{R}_B^\top \left[ \left( \mathbf{S}({}^{H_t}\boldsymbol{\omega}_W) \mathbf{R}_{H_t}^\top \right) \mathbf{R}_B + \mathbf{R}_{H_t}^\top \left( \mathbf{S}(\boldsymbol{\omega}_B) \mathbf{R}_B \right) \right], \quad (6.14b)$$

where  $\mathbf{v}_B, \boldsymbol{\omega}_B \in \mathbb{R}^3$ , and  $\mathbf{v}_{H_t}, \boldsymbol{\omega}_{H_t} \in \mathbb{R}^3$  are the linear and angular velocities of the robot and the human trunk, respectively. The velocity term  ${}^{H_t}\boldsymbol{\omega}_W \in \mathbb{R}^3$  represents the angular velocity of the inertial frame as seen from a theoretical observer fixed to the human trunk, and it is given by

$${}^{H_t}\boldsymbol{\omega}_W = \mathbf{R}_{H_t}^\top \left( -\boldsymbol{\omega}_{H_t} \right). \quad (6.15)$$

The vector  $\boldsymbol{\omega}_{H_t} \in \mathbb{R}^3$  collects the angular velocities of the human trunk frame expressed in  $\mathcal{F}_W$ .

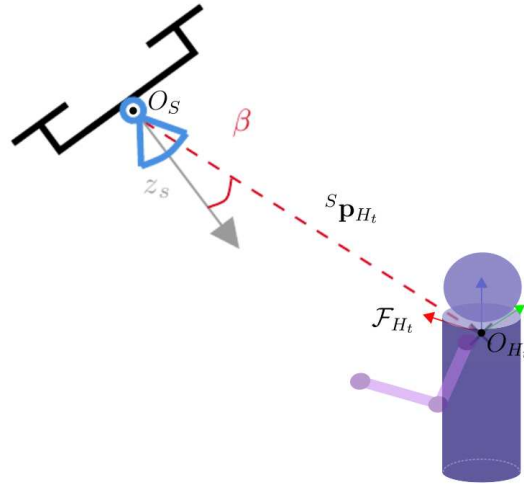
The terms derived within this section in Equations (6.13a)-(6.14b) represent the motion of the AR expressed as seen from the reference frame attached to the human trunk. As one could expect, those quantities depend on the robot and human state variables. If we collect those terms together, we can write that

$$\mathbf{y}_m = \left[ {}^{H_t}\mathbf{p}_B^\top \quad {}^{H_t}\mathbf{q}_B^\top \quad {}^{H_t}\mathbf{v}_B^\top \quad {}^{H_t}\boldsymbol{\omega}_B^\top \right]^\top \in \mathbb{R}^{13}, \quad (6.16)$$

where  $\mathbf{y}_m$  denotes the *human-relative motion task*, i.e. the red block in the objective criterion depicted in Figure 6.1.

## Perception

During the handover, the robot has to maintain the human trunk always visible within the *limited* Field of View (FoV) of the onboard sensor, so that a pose estimation of



Courtesy of [Jacquet, 2022b].

Figure 6.3: A schematic drawing showing the bearing vector going from the camera origin to the feature  $F$ , and the related angular distance  $\beta$ .

the human partner can be computed. This can be achieved by requiring the desired feature point(s) to stay as close as possible to the center of the sensor FoV [Jacquet, 2022b]. Therefore, we can include this goal as part of the cost function, by adding a perception task. As we discussed in Chapter 2, we assume that the human is detected by the robot onboard vision system, as soon as the user trunk is visible from the onboard sensor (a standard camera). Hence, we select as feature of interest the origin  $O_{H_t}$  of the human trunk frame. Thus, recalling Section 4.8, we can write that

$${}^S \mathbf{p}_F = {}^S \mathbf{p}_{H_t} = [x_{H_t} \quad y_{H_t} \quad z_{H_t}]^\top \in \mathbb{R}^3. \quad (6.17)$$

Based on the work of Jacquet [Jacquet, 2022b], to achieve robust visual tracking of the feature, we need to maximize the cosine of the angular distance between the camera axis  $\mathbf{z}_S$  and the bearing vector going from the camera origin  $O_S$  to the feature, thus the vector  ${}^S \mathbf{p}_{H_t} \in \mathbb{R}^3$ . This vector is shown in Figure 6.3 as a red dashed line, while the angular distance is defined as the angle  $\beta \in \mathbb{R}$ . Henceforth, according to [Jacquet, 2022b], we define  $c\beta = \cos(\beta)$  and we can compute it through the projection of the bearing vector on the axis  $\mathbf{z}_S$  as:

$$c\beta = \frac{{}^S \mathbf{p}_{H_t} \cdot \mathbf{z}_S}{\|{}^S \mathbf{p}_{H_t}\|_2} \in \mathbb{R}. \quad (6.18)$$

The bearing vector can be calculated by considering some kinematic relationships as follows

$${}^S \mathbf{p}_{H_t} = {}^S \mathbf{p}_B + {}^S \mathbf{R}_B {}^B \mathbf{p}_{H_t} \quad (6.19a)$$

$$= {}^S \mathbf{p}_B + {}^S \mathbf{R}_B \left( \mathbf{R}_B^\top (\mathbf{p}_{H_t} - \mathbf{p}_B) \right) \quad (6.19b)$$

$$= {}^B \mathbf{R}_S^\top \left( \mathbf{R}_B^\top (\mathbf{p}_{H_t} - \mathbf{p}_B) - {}^B \mathbf{p}_S \right), \quad (6.19c)$$

where  ${}^S \mathbf{p}_B = {}^B \mathbf{R}_S^\top (-{}^B \mathbf{p}_S)$ .

The *perception task* can be defined as

$$\mathbf{y}_p = 1 - c\beta \in \mathbb{R}, \quad (6.20)$$



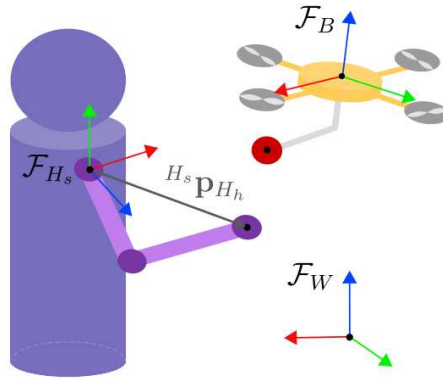


Figure 6.4: This picture shows the vector representing the human hand position expressed in the shoulder frame  $\mathcal{F}_{H_s}$ , i.e., the quantity  $H_s \mathbf{p}_{H_h}$ .

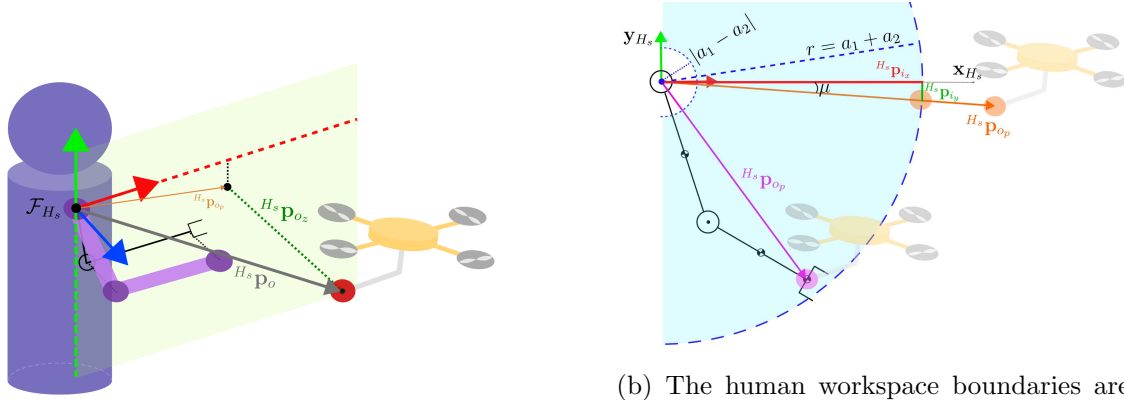
since the natural maximum value for  $c\beta$  is 1. In this way, the perception objective is maximized, i.e. it equals 1, when  $\mathbf{y}_p = 0$ , i.e. is minimum. The term in Equation (6.20) allows the controller to maintain the human trunk as close as possible to the center of the FoV of the sensor, while coping with the other tasks. In Figure 6.1, the perception task is represented by the green block in the cost function. However, the minimization of that term does not guarantee that the human trunk is always visible and not exiting the camera FoV. Therefore, we will introduce also a perception constraint, as we will show later in Section 6.2.4.

## Ergonomics

The human-robot interaction has to occur in the most natural and comfortable way. Hence, we would like the robot to hand over the object in a position that is both reachable and ergonomic from the human point of view. In this work, we relate the degree of human ergonomics to the physical effort the human has to undertake to receive an object in the handover process. Inspired by the work in [Peternel, 2017], we quantify the ergonomics by estimating the human joint torques that the user needs to apply to hold the received object in a given hand location. Additionally, we perform this evaluation in the moment just after the object being received by the human operator, and we consider only the contribution given by the human arm joint torques. This is contrast with the work of Peternel et al, where also the user postural changes and the related Degrees of Freedom (DoFs) are taken into consideration. However, in the HARH, this does not constitute a conservative assumption, since the objects which are typically transportable by means of medium-small scale MRAV are lightweight. As a consequence, the user does not need to significantly perform postural adjustments to receive these items from the robotic partner and being able to sustain their weight. Besides, we evaluate the human ergonomics for a given handover location from a quasi-static point of view, thus we assume the human joint velocities and acceleration to be zero. Hence, we assume that

$$\ddot{\mathbf{q}}_h = \dot{\mathbf{q}}_h = 0. \quad (6.21)$$

Recalling the dynamic model introduced in Section 4.9.2 for the human arm, i.e. Equation (4.73), and by substituting the condition in Equation (6.21), we can



(a) This picture shows the shoulder-frame  $\mathbf{x}_{H_s}$ ,  $\mathbf{y}_{H_s}$  plane, and the related projected quantities. In gray, the object position expressed in  $\mathcal{F}_{H_s}$ . In green and orange, respectively, the  $z$  coordinate of  ${}^{H_s}\mathbf{p}_o$ , and its projection in the vertical shoulder-frame plane.

(b) The human workspace boundaries are marked by the blue dashed circles. In orange, the object position projected in the  $\mathbf{x}_{H_s}$ ,  $\mathbf{y}_{H_s}$  plane, which is not reachable from the planar human arm. In red and green, the object  $x$  and  $y$  coordinates, respectively. In fuchsia, an object position that is contained inside the human arm workspace.

Figure 6.5: 3D (left) and 2D (right) representations of the shoulder-frame  $\mathbf{x}_{H_s}$ ,  $\mathbf{y}_{H_s}$  plane.

compute the human arm joint torques as:

$$\boldsymbol{\tau}_h = \mathbf{g}_h(\mathbf{q}_h) - \mathbf{J}_h(\mathbf{q}_h)^\top \mathbf{f}_o. \quad (6.22)$$

After the object handover, the only external forces  $\mathbf{f}_o$  acting on the human hand are the gravity force given by the object mass  $m_o \in \mathbb{R}_+$ . In the equation above, the external force  $\mathbf{f}_o$  is expressed w.r.t. the human shoulder frame  $\mathcal{F}_{H_s}$ , thus

$$\mathbf{f}_o = \mathbf{R}_S^\top \begin{bmatrix} 0 & 0 & -m_o g \end{bmatrix}^\top, \quad (6.23)$$

where the rotation matrix  $\mathbf{R}_S \in SO(3)$  is the orientation of  $\mathcal{F}_{H_s}$  expressed w.r.t.  $\mathcal{F}_W$ .

At this point, by means of Equation (6.22), we can evaluate the human arm joint torques as function of the arm coordinates  $\mathbf{q}_h \in \mathbb{R}^{n_h}$ ,  $n_h \in \mathbb{N}_+$ . Since we would like to express the torques  $\boldsymbol{\tau}_h \in \mathbb{R}^{n_h}$  as function of the hand location, we need to rely on the *inverse kinematics* of the human arm. This allows expressing the Cartesian hand position in terms of the arm joint variables. If we denote with  ${}^{H_s}\mathbf{p}_{H_h}$  the hand location w.r.t. the shoulder frame  $\mathcal{F}_{H_s}$ , the following equation holds

$$\mathbf{q}_h = \boldsymbol{\Phi}_{\text{IK}}^h \left( {}^{H_s}\mathbf{p}_{H_h} \right), \quad (6.24)$$

where  $\boldsymbol{\Phi}_{\text{IK}}^h \left( {}^{H_s}\mathbf{p}_{H_h} \right) : \mathbb{R}^3 \rightarrow \mathbb{R}^{n_h}$  denotes the *inverse kinematics* function of the *human arm*. In Figure 6.4, we provide a drawing that illustrates the quantity  ${}^{H_s}\mathbf{p}_{H_h}$ .

During the transfer phase of the handover, the human hand and the exchanged object must be in the same position to successfully complete the passing action. Therefore, the following condition shall be met:

$${}^{H_s}\mathbf{p}_{H_h} = {}^{H_s}\mathbf{p}_o, \quad (6.25)$$

where  ${}^{H_s}\mathbf{p}_O \in \mathbb{R}^3$  denotes the position of the object expressed in  $\mathcal{F}_{H_s}$ , as shown in Figure 6.5a. Therefore, substituting Equations (6.24) and (6.25) into Equation (6.22), it results

$$\boldsymbol{\tau}_h = \mathbf{g}_h\left(\Phi_{\text{IK}}^h\left({}^{H_s}\mathbf{p}_o\right)\right) - \mathbf{J}_h\left(\Phi_{\text{IK}}^h\left({}^{H_s}\mathbf{p}_o\right)\right)^\top \mathbf{f}_o. \quad (6.26)$$

By considering the kinematic linkage from the robot body to its end effector (EE), where typically there is a gripper to hold the object to be exchanged, it is possible to relate the quantity  ${}^{H_s}\mathbf{p}_O$  to the robot pose expressed in  $\mathcal{F}_{H_t}$ . Specifically, it is possible to write the following relationship:

$${}^{H_s}\mathbf{p}_o = {}^{H_t}\mathbf{R}_{H_s}^\top \left( {}^{H_t}\mathbf{p}_o - {}^{H_t}\mathbf{p}_{H_s} \right) \quad (6.27a)$$

$$= {}^{H_t}\mathbf{R}_{H_s}^\top \left( \left( {}^{H_t}\mathbf{p}_B + {}^{H_t}\mathbf{R}_B {}^B\mathbf{p}_o \right) - {}^{H_t}\mathbf{p}_{H_s} \right). \quad (6.27b)$$

In the equation above, we can recognize the quantities  ${}^{H_t}\mathbf{p}_B \in \mathbb{R}^3$  and  ${}^{H_t}\mathbf{R}_B \in SO(3)$  being the human-relative robot pose, thus it depends in turn on the robot state  $\mathbf{x}_r$ . The vector  ${}^B\mathbf{p}_o \in \mathbb{R}^3$  being the object position expressed in the robot body frame. In the case of a GTMR MRV with a fixedly attached EE, that quantity is constant and known. Lastly,  ${}^{H_t}\mathbf{p}_{H_s} \in \mathbb{R}^3$  is the distance between the two human frame origins, and the rotation matrix  ${}^{H_t}\mathbf{R}_{H_s} \in SO(3)$  their relative orientation, as mentioned in Section 4.9.2.

Using Equation (6.26), we can compute the human arm joint torques necessary to hold an object as function of the robot position and orientation expressed w.r.t. the shoulder frame  $\mathcal{F}_{H_s}$ . Therefore, by including it within the cost function, the NMPC controller can compute a robot handover pose that reduces the human joint articular stress. To embed Equation (6.26) in the NMPC controller, we need to have an analytical expression for the function  $\Phi_{\text{IK}}^h(\bullet)$ , since the numerical solvers typically use gradient-based methods to compute the solution of a NLP [Yutao, 2019]. In general, the inverse kinematics problem of a manipulator involves the solution of nonlinear equations, and it may have multiple, infinite, or no solution at all [Siciliano, 2009].

In the case of non-redundant manipulators with a small number of DoFs, it is possible to derive geometrical relationships that allow solving the problem analytically. Therefore, we decide to reduce the human arm to a simple 2-DoFs planar manipulator ( $n_h = 2$ ), for which closed-form results are available in textbooks, and we select only the elbow-down configuration to comply with the human elbow articulation. We assume that the links of this human arm manipulator are all laying on the same plane and the revolute joints have parallel axes. This assumption is justified by the intuition that a human would naturally move the arm alongside the body during the handover. Consequently, the human arm workspace lays on the vertical plane  $(\mathbf{x}_{H_s}, \mathbf{y}_{H_s})$ , or equivalently  $(\mathbf{x}_{H_t}, \mathbf{z}_{H_t})$ , and the inverse kinematic function is given according to [Siciliano, 2009] as

$$\mathbf{q}_h = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \Phi_{\text{IK}}^h\left({}^{H_s}\mathbf{p}_o\right) = \begin{bmatrix} \text{atan2}(s_1, c_1) \\ \text{atan2}(s_2, c_2) \end{bmatrix}, \quad (6.28)$$

where  $s_i$  and  $c_i$  are, respectively, the sine and cosine of the  $i$ -th joint coordinate  $q_i$ , where  $i \in \{1, 2\}$ , and atan2 the 2-argument arctangent function. In turn, it can be

proven that [Siciliano, 2009]:

$$c_2 = \frac{{}^{H_s}\mathbf{p}_{o_x}^2 + {}^{H_s}\mathbf{p}_{o_y}^2 - a_1^2 - a_2^2}{2a_1a_2}, \quad (6.29)$$

$$s_1 = +\sqrt{1 - c_2^2}, \quad (6.30)$$

$$s_1 = \frac{(a_1 + a_2c_2){}^{H_s}\mathbf{p}_{o_y} - a_2s_2{}^{H_s}\mathbf{p}_{o_x}}{{}^{H_s}\mathbf{p}_{o_x}^2 + {}^{H_s}\mathbf{p}_{o_y}^2}, \quad (6.31)$$

$$c_1 = \frac{(a_1 + a_2c_2){}^{H_s}\mathbf{p}_{o_x} + a_2s_2{}^{H_s}\mathbf{p}_{o_y}}{{}^{H_s}\mathbf{p}_{o_x}^2 + {}^{H_s}\mathbf{p}_{o_y}^2}, \quad (6.32)$$

where  $a_1 \in \mathbb{R}_+$  and  $a_2 \in \mathbb{R}_+$  are the human arm link lengths, and the quantities  ${}^{H_s}\mathbf{p}_{o_x} \in \mathbb{R}$ ,  ${}^{H_s}\mathbf{p}_{o_y} \in \mathbb{R}$  are the  $x$ ,  $y$  coordinates of the object position vector  ${}^{H_s}\mathbf{p}_o$ .

Moreover, the existence of solutions for the inverse kinematics problem is guaranteed only if the given object position,  ${}^{H_s}\mathbf{p}_o$ , belongs to the human arm workspace [Siciliano, 2009]. For a 2-DoF planar manipulator, the workspace is the space in-between two concentric co-planar circles [Siciliano, 2009], whose outer radius is equal to the sum of the link lengths, and the inner radius to their difference. As a result, the NMPC controller cannot evaluate the human ergonomics until the robot gets close enough to enter the human arm workspace. To overcome this problem, in (6.26), we consider the closest object position that belongs to the human arm workspace. The procedure that follows can be better understood by referring to Figure 6.5.

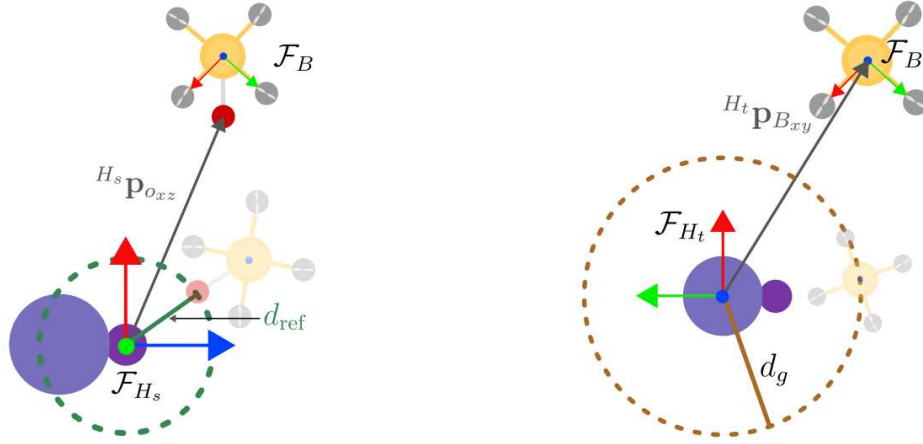
We first consider the projection of the current object position onto the arm plane, being the vertical plane spanned by the axes  $(\mathbf{x}_{H_s}, \mathbf{y}_{H_s})$  of  $\mathcal{F}_{H_s}$ . As shown in Figure 6.5b, we denote the projection of the object position with the 2D vector  ${}^{H_s}\mathbf{p}_{o_p} \in \mathbb{R}^2$ , whose elements are given by the  $x$ ,  $y$  coordinates of  ${}^{H_s}\mathbf{p}_o$ . If that projected point is already part of the human arm workspace, the inverse kinematics admits a feasible solution. If not, we radially project it onto the outer border of the human arm workspace. This is obtained by computing the intersection between the circular workspace centered in  $O_{H_s}$ , and the vector  ${}^{H_s}\mathbf{p}_{o_p}$ . Then, we select the closest point to  $O_{H_s}$  between the planar object position and the computed intersection point. Therefore, the coordinates of the closest point, which we denote as  $\tilde{x}_o$ ,  $\tilde{y}_o$  and  $\tilde{z}_o$ , are obtained through geometrical considerations and by means of the following equations

$$\tilde{x}_o = \min \left( {}^{H_s}\mathbf{p}_o^\top \mathbf{x}_{H_s}, {}^{H_s}\mathbf{p}_{i_x} \right), \quad \tilde{y}_o = \mu \tilde{x}_o, \quad \tilde{z}_o = {}^{H_s}\mathbf{p}_o^\top \mathbf{z}_{H_s}, \quad (6.33a)$$

$$\mu = \frac{\mathbf{n}_y}{\mathbf{n}_x}, \quad \mathbf{n} = \frac{{}^{H_s}\mathbf{p}_{o_p}}{\|{}^{H_s}\mathbf{p}_{o_p}\|_2}, \quad (6.33b)$$

where  $r = a_1 + a_2 \in \mathbb{R}_+$  denotes the total length of the human arm,  $\mathbf{n}_x \in \mathbb{R}$ ,  $\mathbf{n}_y \in \mathbb{R}$  are the  $x$ ,  $y$  coordinates obtained by normalizing the planar vector  ${}^{H_s}\mathbf{p}_{o_p} \in \mathbb{R}^2$ , which is then denoted as  $\mathbf{n} \in \mathbb{R}^2$ . The quantity  $\mu \in \mathbb{R}$  is the angular coefficient (elevation angle) of the vector  $\mathbf{n}$ , as illustrated in Figure 6.5b. Lastly, the quantity  ${}^{H_s}\mathbf{p}_{i_x} \in \mathbb{R}$  denotes the  $x$  coordinate of the vector  ${}^{H_s}\mathbf{p}_i \in \mathbb{R}^2$ , where the latter variable is the vector going from  $O_{H_s}$  to the intersection point between the circle of radius  $r$  and the vector  ${}^{H_s}\mathbf{p}_{o_p}$ . The coordinates of this intersection point are given by computing the geometrical intersection between a circle and a line, thus as follows:

$${}^{H_s}\mathbf{p}_{i_x} = \pm \sqrt{\frac{r^2}{1 + \mu^2}}, \quad {}^{H_s}\mathbf{p}_{i_y} = \mu {}^{H_s}\mathbf{p}_{i_x}. \quad (6.34)$$



(a) Top view showing the  $\mathbf{x}_{H_s}, \mathbf{z}_{H_s}$  plane. In gray, the planar  $(x, z)$  object position expressed in  $\mathcal{F}_{H_s}$ . In green, the desired handover distance.

(b) Top view showing the  $\mathbf{x}_{H_t}, \mathbf{y}_{H_t}$  plane. In gray, the planar  $(x, y)$  robot position expressed in  $\mathcal{F}_{H_t}$ . In brown, the minimum guard distance.

Figure 6.6: Schematic top view representation of the human-aerial robot handover. The human trunk is the larger purple circle, while the smaller one represents the user shoulder. The robot is depicted in yellow, while the carried object as a red circle.

In the equation above, two solutions exist: one for positive  $x$  coordinates, and another one for negatives, which correspond to the handover performed in front and behind the human trunk, respectively. As the handover shall take place in a comfortable and safe configuration, we consider only the front half of such a region as the human workspace. Hence, in Equation (6.33), we consider  ${}^{H_s}\mathbf{p}_{i_x} = +\sqrt{\frac{r^2}{1+\mu^2}}$ . The *projected* object position, given by the coordinates  $\tilde{x}_o, \tilde{y}_o$  and  $\tilde{z}_o$  in Equations (6.33), constitute a handover location that assures a feasible solution to the inverse kinematics problem and allows computing the human torques. Hence, in Equation (6.26), we consider  ${}^{H_s}\mathbf{p}_o = [\tilde{x}_o \tilde{y}_o \tilde{z}_o]^\top$ . At this point, we can finally define the *articular stress* objective as

$$\mathbf{y}_\tau = \boldsymbol{\tau}_h, \quad (6.35)$$

where  $\boldsymbol{\tau}_h$  is given by Equation (6.26). This cost finds its origins in the work of Katayama et al. [Katayama, 2003], where the authors proposed five different optimization models to characterize human comfort, including this joint torque approach.

Besides, to assure having the object in the human arm plane, the controller is tasked to minimize the normal projection of  ${}^{H_s}\mathbf{p}_o$  to the plane  $(\mathbf{x}_{H_s}, \mathbf{y}_{H_s})$ . This is achieved by introducing an additional term in the cost function, which we name *projection* objective, and it is defined as:

$$y_z = {}^{H_s}\mathbf{p}_{O_z} \in \mathbb{R}, \quad (6.36)$$

where  ${}^{H_s}\mathbf{p}_{O_z} \in \mathbb{R}$  represents the  $z$  coordinate of  ${}^{H_s}\mathbf{p}_o$ . This quantity is depicted in Figure 6.5a.

By minimizing only the human joint torques, the robot would prefer to hand over the object at either the hand location when the arm is fully stretched down or when fully stretched up, since these two configurations constitute the two global minima of (6.26). Obviously, these solutions are incompatible with the objective of achieving a safe and natural interaction. Therefore, we introduce a third term representing

the distance at which the handover appears comfortable and natural from the user perspective. Hence, we define a *handover* objective function  $y_o$  as

$$y_o = \frac{b}{(d_{ho} - d_{\text{ref}})}, \quad (6.37)$$

where  $d_{\text{ref}} \in \mathbb{R}_+$  is a desirable distance at which letting the object transfer phase occur. In particular, we have that  $d_g < d_{\text{ref}} < (a_1 + a_2)$  to assure having the object exchange at a reachable position and, at the same time, not violating the operator safety distance. The quantity  $b \in \mathbb{R}_+$  is a scaling factor, while  $d_{ho} \in \mathbb{R}$  is the relative distance between the object and the human on the horizontal plane spanned by  $(\mathbf{x}_{H_s}, \mathbf{z}_{H_s})$ . Thus, the latter term is computed as

$$d_{ho} = \|\mathbf{p}_{o_{xz}}\|_2 = \sqrt{{}^{H_s}\mathbf{p}_{ox}^2 + {}^{H_s}\mathbf{p}_{oz}^2}, \quad (6.38)$$

where  ${}^{H_s}\mathbf{p}_{ox} \in \mathbb{R}$  and  ${}^{H_s}\mathbf{p}_{oz} \in \mathbb{R}$  denote the  $x$ ,  $y$  coordinates of  ${}^{H_s}\mathbf{p}_o$ . Figure 6.6a offers a schematic illustration of the quantities previously introduced.

At this stage, the total *ergonomics task*  $\mathbf{y}_e$  is obtained as the combination of the individual terms (the articular stress, the projection, and handover ones) given by Equations (6.35)-(6.37), resulting in

$$\mathbf{y}_e = [\mathbf{y}_\tau^\top \quad y_z \quad y_o]^\top \in \mathbb{R}^4. \quad (6.39)$$

This term is represented as a purple block in the cost function of Figure 6.1. By inspecting the composition of the individual quantities within the ergonomic task, we can notice that all of them depend on the robot state  $\mathbf{x}_r$ . Therefore, in our proposed approach, the controller can compute the best handover location based on the trade-off between minimizing the human articular effort, and maximizing the spontaneity of the interaction.

## Safety

During the whole operation and while handing over the object, the robot has to ensure the safety of the human coworker. Thus, in the cost function, we introduce a penalization term that precludes the robot from crossing a safety distance, which guarantees to avoid collisions between the two agents. This quantity, which is represented by the purple block in Figure 6.1, has to strongly affect the robot behavior only in the near proximities of the chosen distance and provide a null contribution to the net cost anywhere else to avoid disturbing the optimization process. Therefore, we define the *safety task* as

$$y_s = \frac{\epsilon_m}{d_{hr} - d_g} \in \mathbb{R}, \quad (6.40)$$

where  $d_g \in \mathbb{R}_+$  is a minimum guard distance, and  $\epsilon_m \in \mathbb{R}_+$  is a scaling factor to shape  $y_s$  according to an additional precautionary margin. Lastly,  $d_{hr}$  is the relative distance between the robot and the human on the plane  $(\mathbf{x}_{H_t}, \mathbf{y}_{H_t})$ , which is computed as

$$d_{hr} = \|\mathbf{p}_{B_{xy}}\|_2 = \sqrt{{}^{H_t}\mathbf{p}_{B_x}^2 + {}^{H_t}\mathbf{p}_{B_y}^2}, \quad (6.41)$$

where  ${}^{H_t}\mathbf{p}_{B_x} \in \mathbb{R}$  and  ${}^{H_t}\mathbf{p}_{B_y} \in \mathbb{R}$  denote the  $x$ ,  $y$  coordinates of  ${}^{H_t}\mathbf{p}_B$  given by Equation (6.13a). Figure 6.6b provides a drawing clearly showing the quantities related to the safety task. Additionally, this safety term corresponds to the yellow block in the cost function of Figure 6.1.

Similarly, the perception task, the safety objective term does not guarantee alone the avoidance of potential collisions between the human and the robotic partner, even though it introduces a degree of robustness in the OCP. Hence, we will introduce also a safety constraint, as we will discuss later in Section 6.2.4.

### 6.2.3 Equality constraints

The equality constraints are given by the system dynamics and the system initialization, which are illustrated as gray blocks in Figure 6.1.

#### System dynamics

Henceforth, recalling the definitions introduced in Section 6.2.1, we have that

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \begin{cases} \mathbf{f}_r(\mathbf{x}_r, \mathbf{u}_r) \\ \mathbf{f}_h(\mathbf{x}_h, \mathbf{u}_h) \end{cases} \quad (6.42)$$

where the system dynamics  $\mathbf{f}$  is composed of the robot dynamic model  $\mathbf{f}_r$ , corresponding to  $\mathbf{f}_{\text{GTMR}}$  of Equation (4.63), and the human dynamics  $\mathbf{f}_h$ , given by Equations (4.71).

#### System initial condition

The initial system state is  $\mathbf{x}_0 = [\mathbf{x}_{r_0}^\top \mathbf{x}_{h_0}^\top]^\top$ , where  $\mathbf{x}_{r_0}$  and  $\mathbf{x}_{h_0}$  are the estimated robot and human states provided, respectively, by the robot state estimation and the onboard camera. More details will be provided later when describing the validation setup in Sections 6.3.2 and 6.5.1.

### 6.2.4 Inequality constraints

To ensure the task feasibility, we need to impose other constraints on the OCP, which result into inequalities.

#### Perception

According to [Jacquet, 2022b], the perception cost alone does not guarantee that the human feature does not exist the onboard camera FoV. Thus, to guarantee constant visibility over the human trunk, we introduce a constraint on the trunk position, expressed in  $\mathcal{F}_S$ , similarly to what has been proposed in [Jacquet, 2022b]. Hence, we impose the following inequality relationships:

$$\left| \frac{x_{H_t}}{z_{H_t}} \right| \leq \tan \frac{\alpha_h}{2}, \quad \left| \frac{y_{H_t}}{z_{H_t}} \right| \leq \tan \frac{\alpha_v}{2}, \quad (6.43)$$

where the sensor FoV boundaries are characterized by a vertical and horizontal angle denoted with  $\alpha_v \in \mathbb{R}_+$  and  $\alpha_h \in \mathbb{R}_+$ , respectively. In the equation above, the quantity  ${}^S\mathbf{p}_{H_t} = [x_{H_t} \ y_{H_t} \ z_{H_t}]^\top$  is the human position vector expressed in  $\mathcal{F}_{H_s}$ . Equations (6.43) represent the *perception constraints*, which are depicted as the green rectangle in Figure 6.1.

### Safety

To guarantee the requirement of avoiding collisions, we embed a *safety constraint* within the OCP, which is defined as

$$c_{hrp} = d_{hr} = \|\mathbf{p}_{B_{xy}}\|_2 \geq d_g. \quad (6.44)$$

In a similar fashion, we introduce a *human-relative velocity constraint* as

$$c_{hrv} = -{}^{H_t}\mathbf{v}_B \leq \mathbf{v}_{B_g}, \quad (6.45)$$

where  ${}^{H_t}\mathbf{v}_B \in \mathbb{R}^3$  are the human-relative robot velocities obtained from Equation (6.14a), and  $\mathbf{v}_{B_g} \in \mathbb{R}_+^3$  collects their bounding values. The constraint in Equation (6.45) represents the need to limit the velocities in the direction of the human (this explains the minus sign) in order to limit the amount of energy transferred to the human in case of an unavoidable collision. This term is derived from the work in [Haddadin, 2007], where the authors performed different crash tests with a robotic manipulator to quantify the potential injury risk caused by a front impact of the EE to several human parts. The carried experiments, involving a mannequin typically used for the evaluation of car crashes, derived metrics to evaluate the risks of an unexpected impact. In our case, we derive the values for  $\mathbf{v}_{B_g}$  by inspecting the tables reporting the data of an injury risk and the robot mass. Indeed, in view of the compact dimensions of the ARs considered in our work, we can approximate the robot to a single rigid body object of equivalent inertial properties. The constraints in Equations (6.44) and (6.45) constitute the *safety constraints* and they are illustrated in yellow in Figure 6.1.

### Actuation limitations

Next, to account for the physical limitations of the motor-propeller actuators, which we discussed in Section 4.6.3, we impose bounds on the generated forces  $\boldsymbol{\gamma}$  and their derivatives  $\dot{\boldsymbol{\gamma}}$  as

$$\underline{\boldsymbol{\gamma}} \leq \boldsymbol{\gamma} \leq \bar{\boldsymbol{\gamma}}, \quad (6.46a)$$

$$\underline{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma}) \leq \dot{\boldsymbol{\gamma}} \leq \bar{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma}). \quad (6.46b)$$

In the expressions above, the upper and lower bounds  $\underline{\boldsymbol{\gamma}}$ ,  $\bar{\boldsymbol{\gamma}}$ ,  $\underline{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma})$ ,  $\bar{\dot{\boldsymbol{\gamma}}}(\boldsymbol{\gamma})$  are obtained through an identification campaign on the actual hardware, as detailed in [Bicego, 2019]. Those terms are depicted as the blue block in Figure 6.1.

### Numerical feasibility

Lastly, the inverse kinematic function of the human arm discussed earlier does not only admit a feasible solution beyond the circle of radius equal to the total human



arm length, but also within the inner part of that workspace. Indeed, we mentioned that the human arm workspace is the area in-between two concentric co-planar circles [Siciliano, 2009], whose outer radius is equal to  $r = a_1 + a_2$ , and the inner radius to  $|a_1 - a_2|$ . Henceforth, we need to prevent the object from entering the inner circle and make the solution of the inverse kinematics function numerically feasible. To obtain that, we impose a *feasibility constraint* on the squared 2-norm of the object position in the human arm vertical plane spanned by  $(\mathbf{x}_{H_s}, \mathbf{y}_{H_s})$ , which is not reported in Figure 6.1. We denote this term with  $c_{ho}$  and we compute it as

$$c_{ho} = \|\mathbf{p}_{oxy}^{H_s}\|_2 \geq (a_1 - a_2)^2, \quad (6.47)$$

where  $\mathbf{p}_{oxy}^{H_s}$  are the  $x, y$  coordinates of  $\mathbf{p}_o$ , and  $a_1, a_2 \in \mathbb{R}_+$  are the link lengths of the human arm.

### 6.2.5 Human-aware NLP

In this section, we formulate the discrete-time OCP, sampled in  $N$  shooting points, which is solved by the controller at each sampling instant  $T_s$ , over the receding horizon  $T_h$ .

First, we define the output map  $\mathbf{y} \in \mathbb{R}^{n_y}$  as the combination of all the objective costs introduced previously as

$$\mathbf{y} = [\mathbf{y}_m^\top \quad \mathbf{y}_p^\top \quad \mathbf{y}_e^\top \quad \mathbf{y}_s^\top]^\top, \quad (6.48)$$

where  $\mathbf{y}_m, \mathbf{y}_s, \mathbf{y}_e$ , and  $\mathbf{y}_p$  are the *motion, perception, ergonomics*, and *safety* tasks, respectively. In turn, the individual objectives are given by considering Equations (6.16), (6.20), (6.39) and (6.40), which we conveniently report here:

$$\mathbf{y}_m = [\mathbf{p}_B^{H_t \top} \quad \mathbf{q}_B^{H_t \top} \quad \mathbf{v}_B^{H_t \top} \quad \boldsymbol{\omega}_B^{H_t \top}]^\top \in \mathbb{R}^{13}, \quad (6.49a)$$

$$\mathbf{y}_p = 1 - c\beta \in \mathbb{R}, \quad (6.49b)$$

$$\mathbf{y}_e = [\boldsymbol{\tau}_h^\top \quad y_z \quad y_o]^\top \in \mathbb{R}^4, \quad (6.49c)$$

$$\mathbf{y}_s = y_s \in \mathbb{R}. \quad (6.49d)$$

Therefore, we have that  $n_y = 19$ . The reference values for each task are gathered in the vector  $\mathbf{y}_\bullet^r$ . Specifically, the motion reference  $\mathbf{y}_m^r$  is provided by an external trajectory planner, and  $\mathbf{y}_p^r, \mathbf{y}_e^r$ , and  $\mathbf{y}_s^r$  are set to 0. The complete objective function is obtained as the summation of the weighted squared 2-norms of the difference between each task objective and its reference vector.

As a result, we can formulate the OCP as

$$\min_{\substack{\mathbf{x}_0 \dots \mathbf{x}_N \\ \mathbf{u}_0 \dots \mathbf{u}_{N-1}}} \sum_{k=0}^N \sum_{j \in \{m, \dots, v\}} \|\mathbf{y}_{j,k} - \mathbf{y}_{j,k}^r\|_{\mathbf{W}_j}^2 + \sum_{k=0}^{N-1} \|\mathbf{u}_r\|_{\mathbf{W}_{\mathbf{u}_r}}^2 \quad (6.50a)$$

$$s.t. \quad \mathbf{x}_0 = \mathbf{x}(t), \quad (6.50b)$$

$$\mathbf{x}_{k+1} = \mathbf{\Pi}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k), \quad k \in \{0, \dots, N-1\} \quad (6.50c)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{p}_k), \quad k \in \{0, \dots, N\} \quad (6.50d)$$

$$\underline{\gamma}_k \leq \gamma_k \leq \bar{\gamma}_k, \quad k \in \{0, \dots, N\} \quad (6.50e)$$

$$\dot{\underline{\gamma}}_k \leq \mathbf{u}_k \leq \dot{\bar{\gamma}}_k, \quad k \in \{0, \dots, N-1\} \quad (6.50f)$$

$$c_{hr_p,k} \geq d_g, \quad k \in \{0, \dots, N\} \quad (6.50g)$$

$$c_{hr_v,k} \leq \mathbf{v}_{B_g}, \quad k \in \{0, \dots, N\} \quad (6.50h)$$

$$c_{ho,k} \geq (a_1 - a_2)^2, \quad k \in \{0, \dots, N\} \quad (6.50i)$$

$$\left| x_{H_t} / z_{H_t} \right|_k \leq \tan \frac{\alpha_h}{2}, \quad k \in \{0, \dots, N\} \quad (6.50j)$$

$$\left| y_{H_t} / z_{H_t} \right|_k \leq \tan \frac{\alpha_v}{2}, \quad k \in \{0, \dots, N\} \quad (6.50k)$$

where the function  $\mathbf{\Pi}$  represents the discretized version of  $\mathbf{f}$ . The optimization problem given by Equations (6.50a)-(6.50k) can be rewritten in the same form as in Equations (6.4a)-(6.4e). By solving the NLP above, the proposed method computes the robot commands that achieve the handover, taking into consideration all the tasks, while satisfying the system constraints.

### Numerical issues

The cost function as written in Equation (6.50a) would require to compute the algebraic difference between two quaternions, since the motion task contains such an orientation parametrization. However, the Euclidean distance between two unit quaternions is not suited to express the dissimilarity between two orientations, as  $\mathbf{q}$  and  $-\mathbf{q}$  represent the same orientation. Based on [Huynh, 2009], we rather compute the distance between two unit quaternions, denoted as  $\mathbf{q}_1$  and  $\mathbf{q}_2$ , by means of the following distance function:

$$d(\mathbf{q}_1, \mathbf{q}_2) = 1 - |\mathbf{q}_1 \cdot \mathbf{q}_2| \in [0, 1]. \quad (6.51)$$

Moreover, a known issue when numerically integrating the quaternion dynamics, e.g. by means of a Runge-Kutta integration scheme [Press, 1992; Butcher, 1996], is constituted by the fact that the resulting quaternion may not adhere to the quaternion manifold [Rucker, 2018]. Rucker et al. propose a solution involving the use of regularized non-unit quaternions and a redundant manifold mapping which allows preserving the unitary modulus. Their proposed method leads to an efficient, singularity-free numerical integration of quaternions over long intervals which preserves the structure of  $SO(3)$  during the integration step [Jacquet, 2022b].

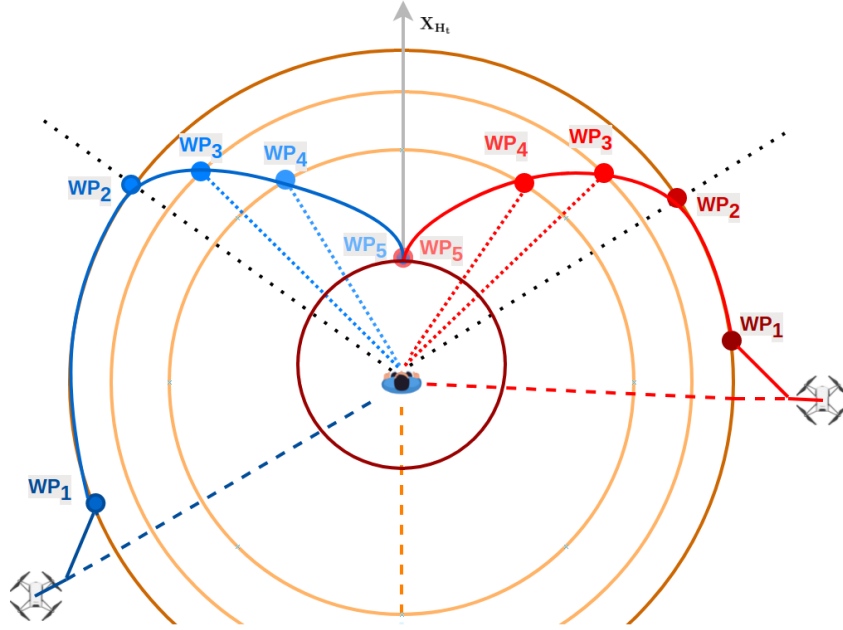


Figure 6.7: Two trajectories generated by the reference motion algorithm are provided for two different starting robot positions, which are marked in blue and red, respectively. Each trajectory is composed of five waypoints: WP<sub>1</sub> to WP<sub>5</sub>.

### 6.3 Simulation validation

In this section, we describe first how the reference for the motion task is generated. Then, we describe the setup for the simulation campaign and, lastly, we provide the results obtained from the conducted simulations.

#### 6.3.1 Motion reference generation

The motion reference task  $\mathbf{y}_m^r$  is designed to drive the AR in front of the human. One could set directly that final position as desired waypoint in the OCP. However, due to limited time horizon and the necessity to maintain real-time performance, the solver is provided with an initial guess of the trajectory that the robot should perform. This guess is generated by a simple motion planning strategy employing a spline interpolation to connect a set of intermediary waypoints. Another possibility, which we leave for future work, is the use of a more sophisticated planner, which could provide a trajectory on-line and possibly account for the human state, e.g., the one presented in [Truc, 2022].

In our approach, the trajectory consists of five intermediary waypoints, as shown in Figure 6.7. The first one is the initial location of the AR, while the final position is in front of the human, from where the robot can reach the partner. Therefore, the approaching phase encompasses the remaining waypoints. Specifically, while approaching, the robot is tasked to maintain a suitable distance to the human, which is not too close to result unpleasant and intimidating from the user perspective. The approaching phase comprises two parts. A first motion is performed towards the human until that suitable distance is reached. Hence, from its starting location, the robot reaches the first WP<sub>1</sub> by performing a straight displacement towards

the human.  $WP_1$  is located at the intersection of this straight line with a circle whose radius is given by the suitable distance mentioned earlier. From this first waypoint, the robot moves along the circumference until it enters the FoV of the human sight. The intersection between the circular path and the line identifying one of the boundaries of the human sight FoV represents  $WP_2$ . From the latter position, the robot approaches the final position in front of the human, which is marked as  $WP_5$ . The intermediary waypoints, i.e.  $WP_i$ ,  $i \in \{1, \dots, 3, 4, 5\}$ , are situated at different distances from  $WP_5$ , and chosen such that the robot performs an arc to reach the last waypoint.

In Figure 6.7, we illustrate two possible trajectories generated by our simple strategy according to two different starting configurations of the aerial robot.

### 6.3.2 Simulation setup

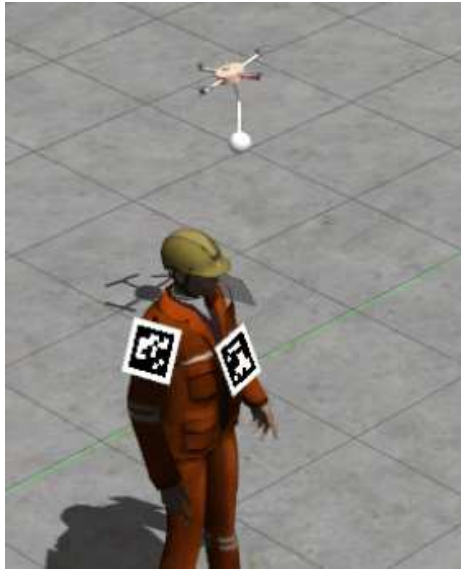
Hereby, we depict the simulation setup utilized to validate the human-aware NMPC controller. We select the Gazebo simulator<sup>2</sup> as simulation environment.

The robot is the collinear quad-rotor shown in Figure 6.8a, which transports a small spherical item of mass  $m_o = 250\text{g}$ , representing a small tool. The object is attached on a fixed bar placed in front of the AR at  $45^\circ$  w.r.t. two of its arms. The simulated AR is equipped with a front-facing 60Hz monocular camera, and in Figure 6.8b we report one frame acquired by this sensor. Then, in the simulator, we have a simulated human featuring a set of Aruco fiducial tags [Garrido-Jurado, 2014], as shown in Figure 6.8. These markers are used to retrieve the position of the human trunk in the world frame, i.e.  $\mathbf{p}_{H_t}$ . This choice is motivated by the practicality of such markers, and by the low computational time and little power required for the detection process. Nevertheless, recent developments in machine learning algorithms allow embedding fast, computationally efficient, and reliable entity detection solutions on-board ARs, e.g., [Zhang, 2019]. The use of such algorithms would relieve the human coworker from wearing markers. However, they are usually trained on specific datasets, and might not provide the desired pose estimate in a handover configuration where the AR is standing very close to the human. The integration of such tools is promising, but left for future work. Lastly, in the simulator, we neglect the weight of the carried tool and the wrench arising from the physical interaction. The latter point will be later discussed in Section 6.4.3.

As for the experimental setup presented in Chapter 5, the framework is implemented in C++ using GenOM [Mallet, 2010], our middleware-independent component generator. Only the reference generation and the necessary scripts to initialize the simulations and to control their evolution are implemented in MATLAB. The NMPC implementation is the one introduced in the Appendix of [Jacquet, 2022b], based on the work in [Chen, 2019]. The simulated hardware interface, the state estimation, and the path planning rely again on the TeleKyb3 software, available on the Open-Robots platform<sup>3</sup>. The software framework is connected to the Gazebo simulated environment that emulates the actual platform interface, whose inputs are the rotor velocities (as it would be for the real robot). Furthermore, in the simulator, we can

<sup>2</sup><https://gazebo.org/home>

<sup>3</sup><https://git.openrobots.org/projects/telekyb3>.



(a) The simulated collinear quad-robot.



(b) An image acquired by the robot onboard monocular camera.

Figure 6.8: On the left, a snapshot of the simulated platform during the approaching phase. On the right, a frame of the robot onboard camera taken during that motion. Some Aruco markers are attached to the human body to allow the robot perception system to detect and estimate the human pose from the camera images.

control the planar position, the yaw, and the arm motion of the simulated human coworker by means of a joystick device. Details on how to use this software and the related source code are made available to the community in our git repository<sup>4</sup>.

In Figure 6.9, we provide a block diagram of the implemented framework. The robot state estimation is achieved using a simulated Motion capture (Mocap) and IMU (accelerometer and gyroscope), whose respective frequencies are set to 50Hz and 500Hz. The rotor velocities are retrieved at 100Hz. Gaussian noise is applied to each type of the collected measurements in the simulator. We apply a standard deviation of 0.003m to position measures, 0.003rad/s and 0.02m/s<sup>2</sup> to velocities and accelerations, and lastly we set a standard deviation of 0.03rad/s to rotor measurements. The human state estimation module uses the measures received at 60Hz from the onboard camera to provide as output the human positions and velocities at higher frequency. The NMPC controller generates the robot commands to be sent to the actuators. As mentioned earlier, the motion task is provided by an external reference generator, while the remaining tasks have reference values set to zero.

### 6.3.3 Results analysis and discussion

The simulation is composed of two phases. First, the robot performs the *approaching* phase where, from its initial position, it moves in front of the human using the algorithm described in Section 6.3.1. Later, in the *reaching* phase, it narrows the distance to the human partner to perform the object handover.

We start by presenting the greater reactivity allowed by considering the motion

<sup>4</sup><https://redmine.laas.fr/projects/nmpc-handover>.

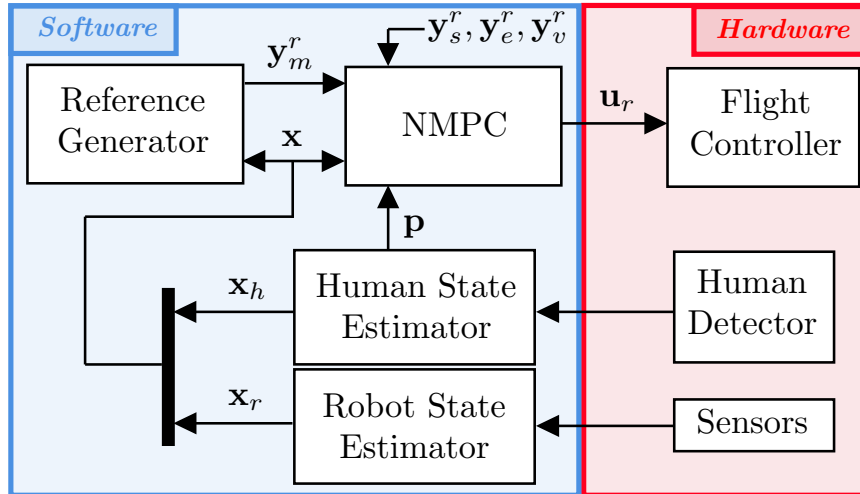


Figure 6.9: Block diagram showing the implementation of the “human-aware” control framework.

in  $\mathcal{F}_{H_t}$ , which is specifically noticeable during the approaching phase. Then, we demonstrate the benefits of the ergonomic cost function during the reaching phase, whose conclusion marks the start of the actual object exchange.

We remind the reader that the video of the simulations, that are hereafter described, can be found in Video 3.3 or at this link: <https://peertube.laas.fr/w/fvDnY4ZCaMhd5ztT2aUhS5>.

### Approaching

During the approaching phase, the human performs a sudden motion while the robot is moving. The  $x, y$  displacements of the human and the robot during this part of the simulations are provided in Figure 6.10. The blue (1) and orange (2) curves correspond to simulations with and without the prediction of the human motion in the controller, i.e. respectively with  $\mathbf{u}_h^{(1)} \neq 0$  and  $\mathbf{u}_h^{(2)} = 0$ . The initial AR position is marked as a blue square while the starting human location as a red circle. The several positions reached by the robot and the human during the simulations are denoted using markers having different shapes and the same color scheme. Consequently, in chronological order, the next robot positions are marked as a blue (or orange) circle, a hexagonal star, and finally as a triangle. Likewise, for the human worker, we use the same sequence, except that the operator starting position is denoted with the red circle, as mentioned earlier.

We first consider the first simulation, i.e. the blue curve, where we enable the human anticipation within the OCP, thus  $\mathbf{u}_h \neq 0$ . When the AR reaches the blue circle, after leaving the blue square, the human suddenly moves from the red circle towards the location denoted by the red star. Then, after waiting a little in that location, the human decides to move to the final location marked as a red triangle. When that motion begins, the robot is in the position denoted by the blue star. Then, while the human moves towards the location of the red triangle, the AR instead moves from the star to the blue triangular shape. As the trajectory is specified w.r.t.  $\mathcal{F}_{H_t}$ , we can notice how the controller modifies the robot motion according to the

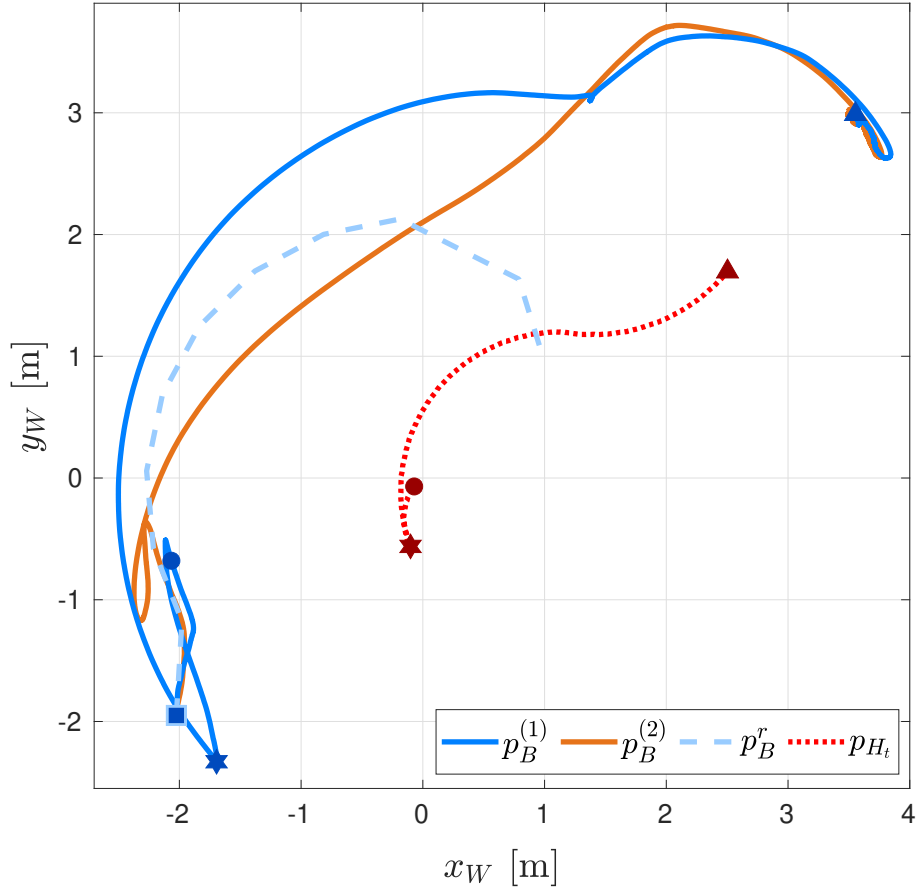


Figure 6.10: Top view of the approaching phase, i.e., in the  $\mathbf{x}_W$ ,  $\mathbf{y}_W$  plane. In light blue, the initial reference motion task generated by our motion planning strategy. In blue and orange, we draw the executed trajectory of the robot in two simulations, (1) and (2), while the trajectory of the human is illustrated in red. The human motion is the same one for both cases.

displacement of the human partner.

The orange curve shows a replica of the previous simulation. However, this time, we disable the human anticipation within the controller by setting  $\mathbf{u}_h = 0$ , while the human partner will move in the same exact way. In this case, we can notice from Figure 6.10 that the motion of the AR reflects with less fidelity the original planned path. Moreover, the distance between the robot and the human is shorter, which could induce safety hazards. This aspect clearly shows the benefit of including the human model within the system dynamics of the NMPC-based controller. A larger robot reactivity is achieved, which in turn makes the interaction safer, as the human-relative motion reference is tracked more accurately. This is possible thanks to the predictive nature of the control technique, which exploits the available model to compute the future positions of the human in a finite-time horizon, and to better position the AR. Furthermore, the motion plan of Section 6.3.1, which is fed to the controller, is computed only once, off-line, and then executed on-line. As showcased by these simulations, our proposed control scheme allows positioning the AR in front of the human regardless of the unknown human motion, and without updating online the given trajectory.

In Figure 6.11, we show the quality of the estimation associated to the human

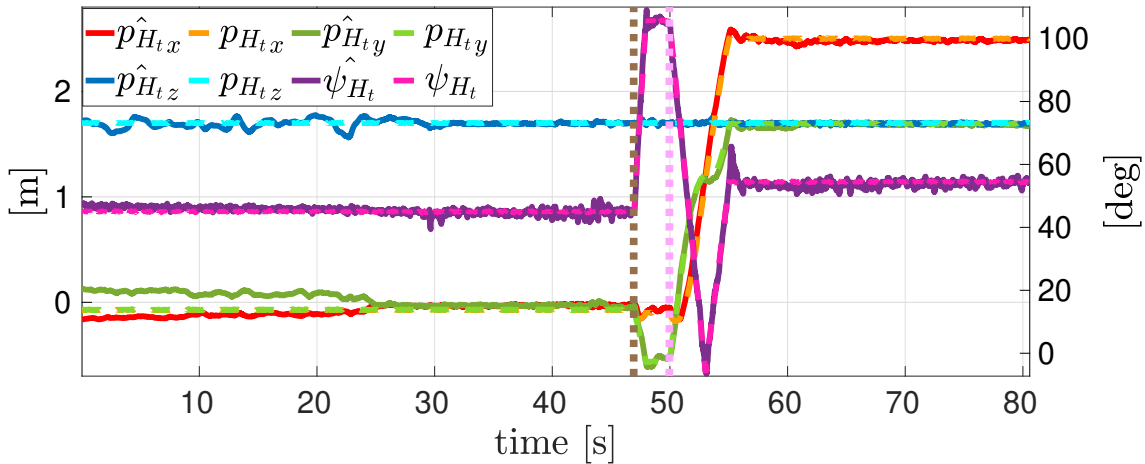


Figure 6.11: Estimated and ground-truth values of the human position and yaw angle during the approaching phase.

position and yaw angle. The dashed lines are the ground-truth values, while the solid ones are the estimated quantities. The brown and pink vertical dotted lines corresponds, respectively, to the first and second displacements of the human worker during the simulation.

In general, the human position is well estimated, except for the first part of the simulation. There, the distance separating the robot to the human is larger than the one during the second part of the simulation, thus increasing the difficulty of detecting precisely the Aruco markers. This is connected to the finite resolution of onboard cameras, which make the detection harder at larger distances from the observed entity [Jacquet, 2022b].

Counter-intuitively, when the human agent moves, we can notice that the estimation is less affected by noise. The reason can be appreciated in Figure 6.8, where an image acquired by the camera on-board the AR is shown. That photo is taken while the robot moves around the human partner to reach the position in front of the operator. At that moment, we can observe that the camera is detecting two fiducial markers, which provide more measurements and in turn improve the estimation process of the human pose.

Lastly, Figure 6.12 shows the visibility task during the first phase (approaching) of the HARH. As the plot suggests, the controller can maintain the human trunk inside the FoV of the camera during the whole simulation, and close to the center, being characterized by  $c_\beta = 1$ . Large deviations from the reference value are obtained when the human is moving. This is related to the under actuation of the AR: as the robot is forced to tilt in order to move, the sensor is no more pointed towards the human worker, in turn jeopardizing the visibility task. Likewise, a similar behavior is observable in the last portion of the plot, where the robot has to stop in the final position. When the robot is moving forward, in order to decelerate, the quad-rotor has to tilt backwards, making the onboard camera looking upwards. Thus the human moves towards the edges of the sensor FoV.

Before concluding this part, it is important to mention that the lower bound of the visibility constraint is given by  $c_\beta = c\alpha_\beta$ , where  $\alpha_\beta \in \mathbb{R}$  is the angle denoting the FoV limit of the onboard camera. For more details about its derivation, we refer



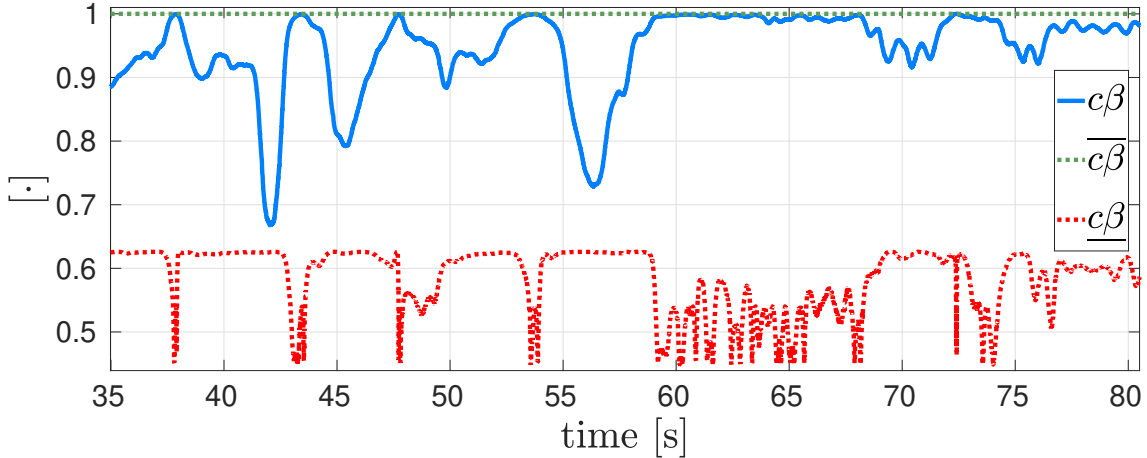


Figure 6.12: Visibility constraint over time during the approaching phase. The quantity  $c_{\beta}$  synthetically summarizes the FoV constraints Equations (6.50j) and (6.50k) in a 1D representation.

the reader to [Jacquet, 2022b].

## Reaching

Once the AR is in front of the operator, the reaching phase starts. At this stage, the ergonomic objective cost is enabled, and the motion task is disabled, as we would like to compute the final handover location based on the terms related to the human comfort only. Indeed, the ergonomic part of the objective function can be computed only when the robot is in the front portion of the human workspace, as mentioned earlier during its derivation.

Figure 6.13 shows the path of the object carried by the AR in the  $\mathbf{x}_{H_s}, \mathbf{y}_{H_s}$  plane. The color gradient is used to represent the level of human comfort, which is computed as the sum of the absolute values of the human joint torques, as function of the object position w.r.t. the human shoulder. The darker the background color is, the smaller is the amount of the articular stress experienced by the operator when positioning the hand in that location for the object exchange. Additionally, the limits of the human arm workspace and the reference handover distance ( $d_{\text{ref}}$ ) are, respectively, illustrated by means of the solid white lines and a brown vertical dashed line.

From that plot, we can appreciate the object displacement in the shoulder plane, which is drawn in fuchsia. Clearly, the ergonomics cost makes the robot move the transported item towards the reference handover distance, while maintaining it in the region that minimizes the human joint torques. The absolute minimum, which corresponds to the resting configuration of the arm, is not reached due to the trade-off between the visibility, the desired handover distance, and the joint torque-related terms. Indeed, moving towards the most ergonomic location would jeopardize the detection of the human trunk and result in an unnatural arm configuration from the user standpoint. Henceforth, the controller computes a robot pose that would result in a comfortable handover position for the object transfer phase.

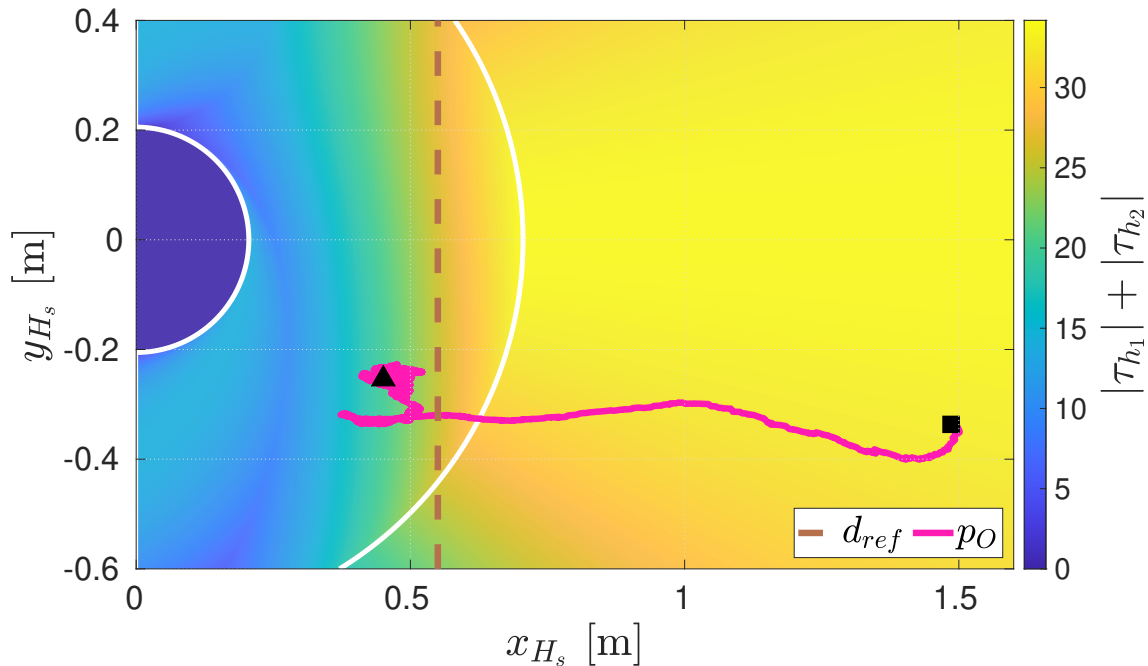


Figure 6.13: Side view of the human arm plane. The color gradient shows the sum of the absolute values of the human arm joint torques.  $p_O$  denotes the projection of the object point  $O$  in the shoulder plane, during the reaching phase. In particular, the carried item starts from the initial position, being the black square, and it moves to the final handover location, which is marked as a black triangle.

## 6.4 Improvements

The work detailed in this chapter has been part of a publication jointly submitted to the prestigious 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) and the journal venue IEEE Robotics and Automation Letters (RA-L). Due to the non yet mature control formulation, it has been accepted only to the former one, and rejected from the journal committee. From the latter venue, we collected the precious comments of the reviewers, whom we deeply thank for their work and critical effort. Our aim is to improve the proposed method, bringing it to a more mature level, and publish it soon to a major venue. Therefore, hereby, we will first discuss the comments of the reviewers and, later, we will present how we can address the raised points.

### 6.4.1 Comments from the reviewers

The comments received by the reviewers pivot on three main points discussed hereafter.

1. Model generalization and its applicability to *any* aerial robot.
2. Inclusion of physical interaction within the control framework.
3. Validation of the method in real experiments.

The first two points are not yet fully addressed and implemented, hence we will just discuss them in the following. Conversely, since a preliminary experimental campaign

has been realized, we will present the related results in the next section.

## 6.4.2 Generalization

The method can be applied to any floating-base robot. Therefore, we show how it can be easily extended to the case of an aerial manipulator (AM) featuring an  $n$ -DoFs robotic arm.

### Aerial manipulators

To broaden the application of the method to AMs, we need to modify the terms composing the OCP that we introduced previously. Therefore, it is necessary to define again the state, the input and the parameter vectors. Then, we perform the same operation for the set of equality and inequality constraints.

Henceforth, we start by recalling the state-space formulation of an AM, which has been introduced in Chapter 4. By using the main results presented in Section 4.7.2, we can write that:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_r^\top & \mathbf{x}_h^\top \end{bmatrix}^\top \in \mathbb{R}^{n_x}, \quad (6.52)$$

$$\mathbf{u} = \mathbf{u}_r \in \mathbb{R}^{n_u}, \quad (6.53)$$

$$\mathbf{p} = \mathbf{u}_h \in \mathbb{R}^{n_{pm}}, \quad (6.54)$$

where  $\mathbf{x}_h$  and  $\mathbf{u}_h$  have the same definitions as in Equations (6.10) and (6.12). Thus, we have that  $\mathbf{x}_h \in \mathbb{R}^4$  and  $n_{pm} = 4$ .

Contrary, the robot states and inputs are now given by those of the AM, namely:

$$\mathbf{x}_r = \mathbf{x}_{AM} = \begin{bmatrix} \mathbf{p}_B^\top & \mathbf{q}_B^\top & \mathbf{q}_a^\top & \mathbf{v}_B^\top & {}^B\boldsymbol{\omega}_B^\top & \dot{\mathbf{q}}_a^\top & \boldsymbol{\gamma}^\top \end{bmatrix}^\top \in \mathbb{R}^{13+2n_q+n_p}, \quad (6.55)$$

$$\mathbf{u}_r = \mathbf{u}_{AM} = \begin{bmatrix} \dot{\boldsymbol{\gamma}}^\top & \boldsymbol{\tau}_a^\top \end{bmatrix}^\top \in \mathbb{R}^{n_p+n_q}, \quad (6.56)$$

where we recall that  $\mathbf{q}_a \in \mathbb{R}^{n_q}$  and  $\dot{\mathbf{q}}_a \in \mathbb{R}^{n_q}$  are the joint positions and velocities, and  $\boldsymbol{\tau}_a \in \mathbb{R}^{n_q}$  the joint torques of the attached robotic arm. Hence, we can write that  $n_x = 17 + 2n_q + n_p$  and  $n_u = n_p + n_q$ .

For what concern the system dynamics, it is similarly given by the composition of the robot and the human models. While the latter is given again by Equation (4.71), the former is composed of the AM dynamic function, which is denoted as  $\mathbf{f}_{AM}$  and obtained from Equation (4.66). Accordingly, the initial system condition is given by  $\mathbf{x}_0 = [\mathbf{x}_{r0}^\top \ \mathbf{x}_{h0}^\top]^\top \in \mathbb{R}^{17+2n_q+n_p}$ .

At this point, we have defined the necessary variables and the equality constraints of our previous OCP for the case of a poly-articulated AM. Now, it just remains to adjust the terms in the cost function and the set of inequality constraints.

Starting with the terms in the objective function, we can notice that the motion, the perception and safety tasks remain unaltered. Therefore,  $\mathbf{y}_m$ ,  $\mathbf{y}_p$ , and  $\mathbf{y}_s$  are given by Equations (6.16), (6.20), and (6.40), respectively. For what concern the last task, namely the ergonomics task  $\mathbf{y}_e$  given by Equation (6.39), we can notice that all its composing terms are function of the same quantity, precisely  ${}^{H_s}\mathbf{p}_o \in \mathbb{R}^3$ .

Therefore, we recall Equation (6.27), which provides the relationship between the vector  ${}^{H_s}\mathbf{p}_o$  and the robot position expressed in  $\mathcal{F}_{H_t}$ , as follows:

$${}^{H_s}\mathbf{p}_o = {}^{H_t}\mathbf{R}_{H_s}^\top \left( ({}^{H_t}\mathbf{p}_B + {}^{H_t}\mathbf{R}_B {}^B\mathbf{p}_o) - {}^{H_t}\mathbf{p}_{H_s} \right). \quad (6.57)$$

In the equation above, the only term which needs to be properly re-defined is  ${}^B\mathbf{p}_o \in \mathbb{R}^3$ . This quantity represents the position of the carried object w.r.t. the robot body. If the robot is a GTMR with a fixedly-attached EE, then the latter is constant. If instead the AR is a poly-articulated AM, then  ${}^B\mathbf{p}_o$  depends on the joint configuration of the robotic arm attached to the flying robot. Therefore, we have that

$${}^B\mathbf{p}_o = \Phi_{\text{FK}}^r(\mathbf{q}_a), \quad (6.58)$$

where  $\Phi_{\text{FK}}^r(\mathbf{q}_a) : \mathbb{R}^{n_a} \rightarrow \mathbb{R}^3$  is the *forward kinematics* of the *robotic arm*. This function maps the joint configuration space of a manipulator to its end-effector position [Siciliano, 2009]. Additionally,  $\Phi_{\text{FK}}^r(\mathbf{q}_a)$  depends on the mounting location of the robotic arm w.r.t. the MRV base. This information represents a constant parameter in the robot forward kinematics function.

*Remark. Forward kinematics.* Differently from the inverse kinematics, the *forward* one can be computed analytically. Hence, contrary to  $\Phi_{\text{IK}}^h(\bullet)$ , the relationship  $\Phi_{\text{FK}}^r(\mathbf{q}_a)$  can be easily embedded within the OCP.

Subsequently, for what concerns the inequality constraints, we can observe that the perception and safety bounds remain unchanged as well. Thus, we can use again Equations (6.43), (6.44), and Equation (6.45). Then, in the expression of  $c_{ho}$  given by Equation (6.47), we need to use the new expression for  ${}^{H_s}\mathbf{p}_o$  obtained previously. Lastly, in the case of a GTMR, we had as actuation constraints Equations (6.46a) and (6.46b), which represent the bounds on the motor-propeller thrusts and their first-order derivatives. In the case of an aerial manipulator, we need to consider also the limited torques generated by the joint motors, as discussed in Section 4.6.3. Therefore, we have that

$$\begin{aligned} \gamma &\leq \gamma \leq \bar{\gamma}, \\ \begin{bmatrix} \dot{\gamma} \\ \tau_a \end{bmatrix} &\leq \underbrace{\begin{bmatrix} \dot{\gamma} \\ \tau_a \end{bmatrix}}_{\mathbf{u}_r} \leq \begin{bmatrix} \dot{\bar{\gamma}} \\ \bar{\tau}_a \end{bmatrix}. \end{aligned} \quad (6.59a)$$

Arrived at this point, we can collect all the terms and apply again the multiple shooting technique to obtain a new discrete-time NLP. The optimal sequence of inputs  $\mathbf{u}_k, k \in \{1, \dots, N-1\}$ , allows controlling both the propellers and the joint motors of the considered AM. Moreover, the solution of that optimization problem will drive the new aerial robot towards a safe and comfortable handover location.

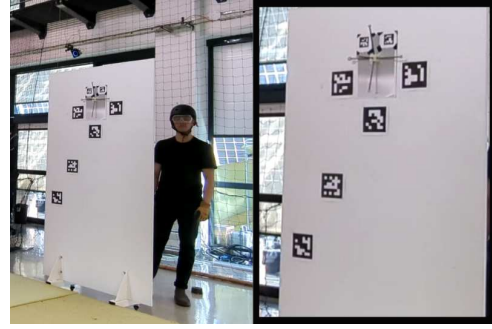
### 6.4.3 Physical interaction

The physical interaction can be incorporated within the previous predictive controller by including the contact wrench arising during the transfer phase into the OCP. This is possible by using Equation (4.45) as part of the robot dynamic function  $\mathbf{f}_{\text{AM}}$ . In



Courtesy of © Cyril FRESILLON / LAAS / CNRS Photothèque.

(a) Uni-directional Thrust (UDT) quad-rotor adopted in the experimental validation of the “human-aware” controller. The position of the virtual object is qualitatively displayed in green.



(b) On the left, the wooden panel board utilized as user avatar, and the human operator moving it. On the right, a zoom-in image on the several fiducial markers attached to the board.

Figure 6.14: In this picture, we show the testbed setup used in the experiments related to the “human-aware” predictive controller.

that equation, the external wrench  $\mathbf{w}_{\text{ext}} \in \mathbb{R}^6$  is applied at the robot end-effector and it affects the robot generalized coordinates  $\boldsymbol{\xi}_{\text{AM}}$ .

The main difficulty is represented by providing the OCP with a mean to predict the future behavior of the contact wrench [Alharbat, 2022]. A simple solution is offered by treating the contact wrench as a constant parameter over the horizon. In such a case, we have that the system parameter vector is given as

$$\mathbf{p} = \left[ \mathbf{u}_h^\top \quad \mathbf{w}_{\text{ext}}^\top \right]^\top, \quad (6.60)$$

where  $n_{pm} = 10$ . This approximation is valid until the external actions can be modeled as slow time-varying disturbances. If that assumption is not holding, then more sophisticated solutions and models can be used [Kocer, 2019; Tzoumanikas, 2020; Peric, 2021].

To obtain the current estimate of  $\mathbf{w}_{\text{ext}}$ , a force-torque sensor or the wrench estimator described in Chapter 5 could be employed.

## 6.5 Experimental validation

In this section, we first describe the testbed setup adopted in the experiments aimed at validating the applicability of our predictive controller in real-world settings. Then, we provide preliminary quantitative results.

### 6.5.1 Experimental setup

In the experimental campaign, we employ a collinear quad-rotor whose picture is provided in Figure 6.14a. This AR is a custom prototype designed and realized in-house at LAAS, where the experiments are performed. The platform diameter is about 50cm and is actuated by 4 collinear 10in propellers. It is equipped with a

monocular front-facing camera, namely an Intel Realsense T265, and its odometry feedback is not utilized. The robot features an onboard Intel NUC, comprising an Intel Core i7-8565U and 8GB of DDR3 RAM, capable of running the image processing algorithms. It runs Ubuntu 18.04, and the software architecture is implemented in C++ using the same software stack described in Section 6.3.2.

The robot state feedback is provided by an onboard IMU at 1kHz and an external Mocup system at 100Hz. Sensor fusion of the available measurements is realized by means of an Unscented Kalman Filter (UKF), which provides the full state feedback at 1kHz. The object detection is performed again using Aruco fiducial markers, which are attached to a wooden panel. This board, whose picture is provided in Figure 6.14b, is used as an avatar to substitute a real participant in the Human-Aerial Robot (HAR) interaction. However, a human operator is standing behind the panel in order to control the board movements. The detection algorithm identifies the Aruco tags in the camera images, and it provides an estimate of the 6D state of the central point thanks to the knowledge of their geometrical dimensions.

Lastly, we consider that a virtual object is rigidly attached to the robot body by means of a fixed EE, which is aligned with the same facing direction of the camera. The qualitative placement w.r.t. the robot body is shown in Figure 6.14a as a green circle. Accordingly, the controller considers this information when computing the item position by means of Equation (6.27).

## 6.5.2 Experimental results

We remind the reader that the videos of the reported simulations can be found in Video 3.4 or at this link: <https://peertube.laas.fr/w/5aCzSTsi5BMhgXHSUB9Kg4>.

In Figure 6.15, we report both the perception objective in the first top plot, the panel board motion expressed in  $\mathcal{F}_W$  in the middle, and its yaw angle in the bottom part. As it can be appreciated from the central plot, the human standing behind the avatar board is moving the panel three times: the first at  $t \approx 10$  along  $\mathbf{x}_W$ , then at  $t \approx 70$  along  $\mathbf{y}_W$ , and lastly at  $t \approx 85$  along  $\mathbf{x}_W$ . While moving the board, the human applies also some rotations around  $\mathbf{z}_W$ , thus modifying the heading (yaw) of the human mockup, as show in the bottom plot. From the top part of Figure 6.15, we can observe that the human remains always visible throughout the whole experiment, despite the performed displacements. As a matter of fact,  $c\beta$  is well contained within its bounds, and very close to its reference value (being 1).

A good visibility of the human board allows the robot to properly estimate the pose of its partner. This can be noticed by inspecting Figure 6.16. There, the top plot shows the estimated position of the human trunk compared against the recorded ground-truth. Except a small error along the vertical axis, the human pose estimation matches well the real values. Similar considerations apply to the yaw angle estimation, which is shown in the bottom of Figure 6.16.

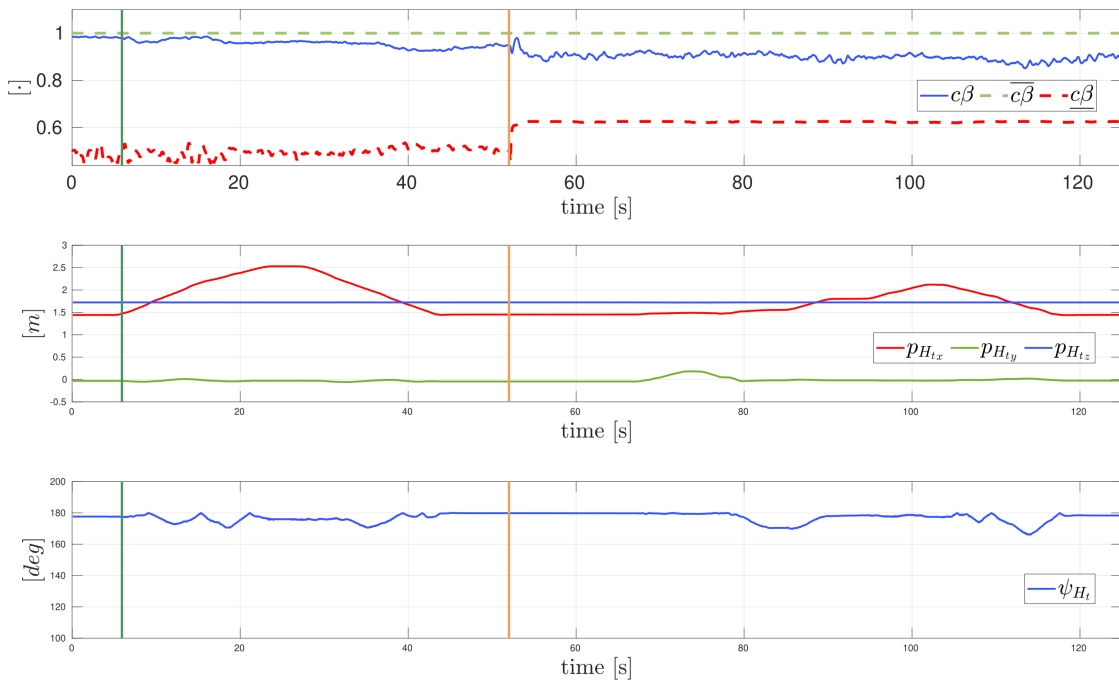


Figure 6.15: In the top, the perception objective is shown. In the middle, the  $x$ ,  $y$  and  $z$  coordinates related to the ground-truth position of the human mockup panel, respectively in red, green, and blue. In the bottom, the yaw angle of the human avatar board.

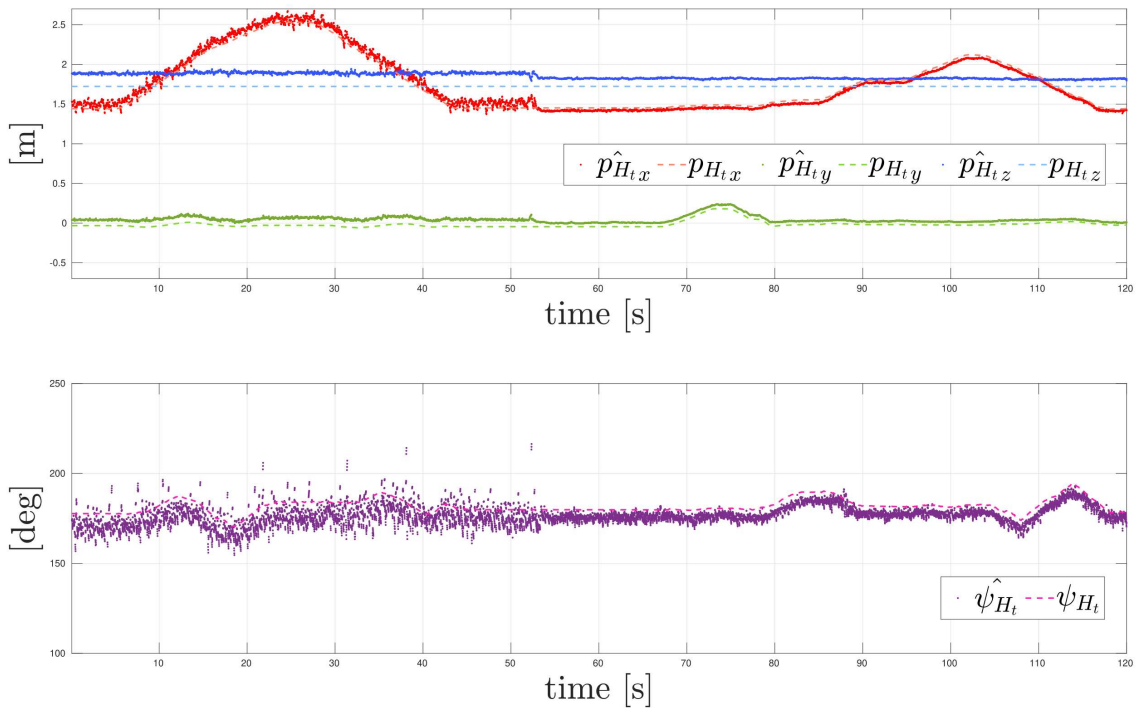


Figure 6.16: In the top, the estimated and the ground-truth position coordinates of the human (trunk) board, respectively as dotted and dashed lines. In the bottom, and in a similar fashion, its estimated and ground-truth yaw angles.

## 6.6 Conclusion

In this work, we propose a human-aware NMPC designed to let a MRAV autonomously perform the handover of a tool to a human worker. We start by introducing the main working principles of the underlying approach, thus we offer an overview on the Model Predictive Control (MPC) technique. Subsequently, we introduce all the terms composing our “human-aware” Optimal Control Problem (OCP). First, the formulation considers the human ergonomics, which is based on the concept of the human overloading joint torques. The minimization of this term, whose expression is composed of the closed-form equations of the human shoulder and elbow torques, allows determining the handover position minimizing the human physical effort required to receive the object. Moreover, the robot motion is computed relative to the partner trunk which increases the reactivity of the framework to any unexpected human motion, without the need for online re-planning. The worker motion model is included within the NMPC system dynamics allowing the controller to predict the future human poses along the horizon. Additionally, the relative formulation allows embedding a safety barrier to avoid collisions with the human, and a velocity-based term to reduce the injury level in case of an unexpected impact. Besides, we build upon previous works to ensure that the robot actuation limitations are strictly observed during the motion, while maintaining the human observable inside the FoV of an onboard camera. The visibility of the worker body is crucial as its detection allows the robot to estimate the human-AR relative pose. Initially, we test our proposed framework in a Gazebo simulation. We show that the controller tracks the desired path and brings the robot to a comfortable position for performing the handover, even in the presence of sudden human displacements. Then, we discuss a series of improvements related to the control formulation based on the comments received by the reviewing committee of a journal venue. Next, we show the applicability of the method in real-world experiments.

### 6.6.1 Limitations and perspectives

Despite its richness, the presented work still leaves open challenges which will be addressed in future work. Firstly, the physical interaction between the two agents has still to be fully investigated. The current formulation neglects the possible wrench that may arise during the handover transfer phase. The controller should compute motor commands to compensate for the human actions and, at the same time, prevent the risk of losing stability and impacting the partner. Therefore, in future work, we aim at addressing this point based on the considerations we drew in Section 6.4. Secondly, the model used for the human arm considers only the shoulder and the elbow articulations, and it neglects other human DoFs, like the neck and the torso. It would be possible to take into consideration also the trunk torques and relate the ergonomics to the variations in the partner body posture. Thirdly, the use of higher DoF models makes finding a closed-form solution to the human inverse kinematics problem challenging or even unfeasible. Consequently, the use of numerical optimization to solve the human inverse kinematics problem could be investigated. This allows to directly embed the mapping between the joint torques and the object pose inside the OCP. Fourthly, we neglected the problem of occlusion.



While this problem might not be critical in case of objects rigidly-attached to the robot body, it could instead occur if the platform is equipped with a poly-articulated robotic arm. The motion of the arm could indeed jeopardize the visibility task, as the manipulator could occupy the camera FoV, in turn deteriorating the human pose detection and estimation. Lastly, so far, we have not consider any human socio-psychological factor related the collaboration, such as legibility, predictability and social acceptability. In future work, we aim at investigating these aspects by testing the controller either in a virtual-reality environment or by means of user studies with several participants.

**Part III**  
**Conclusions**



# Chapter 7

## Conclusion

In recent years, the field of aerial robotics is rapidly growing, driven by the possibility to deploy aerial robots (ARs), and particularly multi-rotor aerial vehicles (MRAVs), to perform physical tasks in dangerous and challenging environments. This brings the possibility to either avoid exposing human operators to hazardous working conditions or to assist them in their activities. Aerial manipulators (AMs) feature the required skills for being valuable aerial co-workers (ACWs), e.g., by delivering the necessary tools and gathering useful data on the environment. Nowadays, they are equipped with powerful sensing and decision-making algorithms and devices, which fuse well with their maneuvering and manipulation capabilities. Therefore, the new domain encompassing physical Human-Aerial Robot Interaction (pHARI) is emerging.

Despite the problem of physical Human-Robot Interaction (pHRI) has been already explored with ground manipulators, the same cannot be said with ARs. Except the few works that started investigating the social aspects of a Human-Aerial Robot (HAR) interaction, still the problem of a physical and safe collaboration has not been fully addressed by the research community. Henceforth, proper control methods for a safe pHARI are necessary.

The contribution of this thesis is the proposition of control algorithms enabling aerial robots to achieve a safe and comfortable physical interaction with their human partners. In particular, this work focuses on the challenge of letting an AR deliver (or receive) an object to (from) a human operator, i.e., on the Human-Aerial robot handover (HARH) problem.

In this conclusive chapter, we first summarize the main contributions brought by our research and the topics covered by this thesis in Section 7.1. Then, in Section 7.2, we conclude the manuscript with final considerations on the challenges that have not yet been addressed in our investigation, and the research perspectives for future work. Regarding the latter point, we sketch an idea allowing combining the two proposed methodologies and address the HARH thoroughly.

### 7.1 Overall contributions

This thesis focused on the Human-Aerial robot handover (HARH) problem involving the object transfer between a robotic and a human agent.

In Chapter 1, we provide an overview of the background related to our work and motivate the following research problem. Therefore, we offer an overview over the fields of Human-Robot Interaction (HRI), physical Aerial Robot Interaction (pARI), and some of the recent articles published within the domain of pHARI. Moreover, this first chapter serves to introduce the taxonomy of the aerial vehicles considered in this thesis: MRAVs and AMs equipped with a multi-Degree of Freedom (DoF) robotic arm.

Then, in Chapter 2, we describe the problem under analysis, highlighting the different phases composing HARH process (namely the approach, reach, and interaction phases) and the underlying assumptions we made. Then, we analyze the research challenges associated with the considered problem. Here, we provide a concise list.

- The study of the physical interaction arising between the two agents when the object is exchanged.
- The problem of maintaining visibility of the human operator during the whole process, as the robot relies only on the onboard visual sensor to determine the human presence and estimate the user body pose.
- The inclusion of human safety at the control level ensuring at whole time to avoid potential human-robot collisions, or to minimize the energy transferred in case of an unexpected impact.
- The formulation of a suitable metric to account for the human ergonomics when controlling the robot during the handover task.

Subsequently, we propose a way to address these challenges, in particular we decompose the original problem into two complementary parts. This simplifies the analysis of the HARH process, as we consider only a subset of the analyzed challenges at once. Moreover, it facilitates the derivation of effective control methodologies. In particular, Chapter 3 proposes two control architectures addressing these two subproblems. Chapter 4 offers the mathematical tools which are used in the derivation of the two control methods.

The first subproblem focuses on the interaction phase between the two agents and the exchanged wrench arising when the object is transferred. In Chapter 3, we introduce a hierarchical control architecture composed of a low-level controller, a physical interaction layer, and a vision system. In Chapter 5, we present the details behind the formulation of this control framework. Specifically, the first module generates the robot motor commands exploiting the system model and its actuation properties (full actuation) in order to apply a 6D wrench. The interaction layer features a wrench observer and an admittance filter. The former estimates the external disturbances applied on the robot which arise during the interaction, while the latter enhances the robot with compliance during the HARH. The last vision-based module comprises a visual servoing algorithm which generates the velocity commands driving the robot towards the goal by observing a set of visual clues.

Conversely, the second subproblem pivots on the inclusion of human awareness and related metrics at the control level in order to successfully perform the approach and reach phases. In Chapter 3, we unveil our Nonlinear Model Predictive Control (NMPC) formulation that can exploit the human and the robot models to predict the future system behavior. Accordingly, in Chapter 6, we offer an overview of the

MPC technique, and we detail the general structure of an Optimal Control Problem (OCP). Henceforth, we derive all the terms that allow composing our “human-aware” NMPC-based controller. The solution of the OCP results in the generation of feasible motor commands and the computation of an ergonomic handover location minimizing the articular stress undertaken by the operator. In addition, the solution satisfies a set of task requirements. Specifically, the robot can maintain visibility of the human partner during the handover. In this way, the onboard estimation architecture can effectively provide the controller with an estimation of the user body. Moreover, the robot motion minimizes both the risk of colliding with the operator and, in case of an unwanted impact, the amount of transferred energy. Lastly, the human-relative formulation allows natural re-plan in case of an unexpected human motion.

Chapters 3, 5, and 6 show the achieved results in several simulated scenarios and real-world experiments justifying the soundness and effectiveness of the proposed methodology.

## 7.2 Limitations and perspectives

The research presented along this manuscript left open many questions, both from the theoretical and practical standpoint, which could lead to fruitful works in the field of pHARI.

On the theoretical side, the first notable aspect not considered in this work is the investigation of the HARH problem as a whole by addressing all the highlighted challenges at the same time. A possibility could be the development of a hierarchical control architecture obtained by combining our two proposed control methodologies. Therefore, a visual servoing modules, as the one presented in Chapter 5, could be used to drive the end effector (EE) pose towards the desired location, i.e., the hand of the operator to hand over the object. Hence, visual clues could be extrapolated from the human body and used within the vision-based control loop to visually guide the robot to approach and reach the human partner. Likewise, a wrench observer and an admittance filter can be again employed to deal with the physical interaction, since they are revealed to be effective force control methods. Then, the low-level motion system could be replaced by our NMPC-based algorithm. In the new architecture, the latter module would receive the desired position and velocity commands from the higher-level components, and generate feasible motor commands allowing the robot to perform successfully the handover. Moreover, as proposed in this thesis, it could consider the task requirements associated to the handover case, such as human visibility, safety and ergonomics. As MPC allows controlling the platform from a centralized perspective, it could make the method applicable not only to simple MRAVs but also to more dexterous AMs equipped with multi-DoF robotic arms.

If the platform is equipped with a robotic arm, a very important aspect, that has not been considered in this work, is the exploitation of the system redundancy to actually perform a set of secondary tasks. For instance, the robot dexterity could be used to avoid obstacles, prevent occlusions of the onboard visual sensors, or to better reach the partner. By stretching the robotic arm towards the user, a larger distance could be imposed between the operator and the dangerous parts of the robot body, e.g., the sharp propellers. Within this context, a term in the cost function could be

introduced to let the optimization-based controller find the best solution among the several possible ones that improve the quality of the interaction.

The integration of a controllable gripper would allow investigating the problem of modulating the gripping force during the object transfer phase. In the research community, it has been observed that, during a human-human handover, there is a correlation between the amount of gripping force of the giver and the evolution of the amount of load shared between the two agents [Strabala, 2013; Ortenzi, 2021]. Additionally, if a force-torque sensor is added to such an EE, the contact wrench could be used to detect the starting of the exchange phase, and hence to control the robot grasping and releasing actions.

Another important aspect that has not been studied in this work is the problem of taking into account the system uncertainty and the external disturbances (e.g., wind). Their consideration is crucial, especially when considering real-world scenarios, as their negligence may endanger the user. On this regard, a *robust* NMPC controller could be investigated. In the literature, there is an emerging interest in designing robust predictive controllers that can cope with the system uncertainties [Mesbah, 2016], e.g., an imprecise robot and human pose estimation. Among the different possibilities, particularly interesting is *chance-constrained* NMPC. This variant of the standard approach can account for the stochasticity associated to a real system, and it treats the external disturbances as probabilistic events. The main idea consists in defining an expression relating a certain event to its probability of occurring. Then, a deterministic constraint can be derived expressing the need that such an event (e.g., the occurrence of a collision) shall not occur given a desired probability threshold. Once this constraint is incorporated inside the OCP, the solution computed by the NMPC solver assures that the considered event will occur within the chosen probability limits.

Lastly, as mentioned in Chapter 6, socio-psychological factors have not been taken into account in our investigation. However, for effectively deploying ACWs in collaborative scenarios, user studies are necessary to assess the level of social acceptability and the robot factors that affect the collaboration. For instance, as ARs have neither limbs nor a face, it is important to find other means by conveying the robot intentions, e.g., by requesting the robot to create legible motions. Therefore, new metrics could be defined and included within the optimization problem to generate more predictable and legible robot movements, thus improving the quality of the interaction.

On the practical side, still many efforts are necessary to realize a stable and secure ACW. As discussed in Chapter 5, to allow the actual implementation of a real force control method (e.g., impedance/admittance control), a fully torque-controlled aerial manipulator is necessary, like the one developed in [Martí-Saumell, 2023]. Furthermore, when considering more realistic experiments, it is essential to rely on the sensors on-board the robot. Thus, GPS systems and visual-inertia odometry algorithms can be employed to achieve a reliable robot pose localization and estimation. Additionally, when moving to real experiments, fiducial markers attached to the user body are unpractical for allowing the robot to detect and estimate the partner pose. Therefore, more sophisticated algorithms, such as learning-based methods, could be explored to provide the robot controller with a reliable estimate of the human position [Fisch, 2022; Zheng, 2023]. In turn, this would relieve the

operator from wearing particular vests featuring the AruCo markers. Lastly, it would be interesting to conceive a suitable gripping mechanism whose design can actually facilitate the object exchange between an aerial robot and a human operator.





# Bibliography

- [Abeywardena, 2015] D. Abeywardena, P. Pounds, D. Hunt, and G. Dissanayake. “Design and development of ReCOPTER: An open source ROS-based multi-rotor platform for research”. In: *2015 Australasian Conference on Robotics and Automation*. 2015 (cit. on p. 8).
- [Aboudorra, 2023] Y. Aboudorra, C. Gabellieri, Q. Sablé, and A. Franchi. “Modelling, Analysis and Control of OmniMorph: an Omnidirectional Morphing Multi-rotor UAV”. 2023 (cit. on p. 11).
- [Acharya, 2017] U. Acharya, A. Bevins, and B. A. Duncan. “Investigation of human-robot comfort with a small Unmanned Aerial Vehicle compared to a ground robot”. In: *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2017, pp. 2758–2765 (cit. on p. 15).
- [Afifi, 2022] A. Afifi, M. van Holland, and A. Franchi. “Toward Physical Human-Robot Interaction Control with Aerial Manipulators: Compliance, Redundancy Resolution, and Input Limits”. In: *2022 IEEE Int. Conf. on Robotics and Automation*. 2022, pp. 4855–4861 (cit. on pp. 12, 14, 16, 27, 33).
- [Afifi, 2023] A. Afifi, G. Corsini, Q. Sable, Y. Aboudorra, D. Sidobre, and A. Franchi. “Physical Human-Aerial Robot Interaction and Collaboration: Exploratory Results and Lessons Learned”. In: *2023 Int. Conf. on Unmanned Aircraft Systems*. Warsaw, Poland, 2023, pp. 956–962 (cit. on p. 21).
- [Akalin, 2022] N. Akalin, A. Kristoffersson, and A. Loutfi. “Do you feel safe with your robot? Factors influencing perceived safety in human-robot interaction based on subjective and objective measures”. In: *International Journal of Human-Computer Studies* 158 (2022), p. 102744 (cit. on p. 39).
- [Akbari, 2021] Y. Akbari, N. Almaadeed, S. Al-maadeed, and O. Elharrouss. “Applications, databases and open computer vision research from drone videos and images: a survey”. In: *Artificial Intelligence Review* 54.5 (2021), pp. 3887–3938 (cit. on p. 107).

- [Aleotti, 2012] J. Aleotti, V. Micelli, and S. Caselli. “Comfortable robot to human object hand-over”. In: *2012 IEEE Int. Symp. on Robots and Human Interactive Communications*. 2012, pp. 771–776 (cit. on pp. 17, 23, 24).
- [Alessio, 2009] A. Alessio and A. Bemporad. “A Survey on Explicit Model Predictive Control”. In: *Nonlinear Model Predictive Control: Towards New Challenging Applications*. Ed. by L. Magni, D. M. Raimondo, and F. Allgöwer. Springer Berlin Heidelberg, 2009, pp. 345–369 (cit. on p. 128).
- [Alharbat, 2022] A. Alharbat, H. Esmaeeli, D. Bicego, A. Mersha, and A. Franchi. “Three Fundamental Paradigms for Aerial Physical Interaction Using Nonlinear Model Predictive Control”. In: *2022 Int. Conf. on Unmanned Aircraft Systems*. 2022, pp. 39–48 (cit. on p. 156).
- [Allenspach, 2022] Mike Allenspach, Yash Vyas, Matthias Rubio, Roland Siegwart, and Marco Tognon. “Human-State-Aware Controller for a Tethered Aerial Robot Guiding a Human by Physical Interaction”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 2827–2834 (cit. on p. 16).
- [Allenspach, 2023] M. Allenspach, S. Laasch, N. Lawrance, M. Tognon, and R. Siegwart. “Mixed Reality Human-Robot Interface to Generate and Visualize 6DoF Trajectories: Application to Omnidirectional Aerial Vehicles”. In: *2023 Int. Conf. on Unmanned Aircraft Systems*. 2023, pp. 395–400 (cit. on p. 10).
- [Angerer, 2013] A. Angerer, A. Hoffmann, A. Schierl, M. Vistein, and W. Reif. “Robotics API: Object-oriented software development for industrial robots”. In: *Journal of Software Engineering for Robotics* 4.1 (2013), pp. 1–22 (cit. on p. 4).
- [Anglade, 2019] A. Anglade, J-M. Kai, T. Hamel, and C. Samson. “Automatic control of convertible fixed-wing drones with vectorized thrust”. In: *2019 IEEE Conf. on Decision and Control*. 2019, pp. 5880–5887 (cit. on p. 7).
- [Arechavaleta, 2008] G. Arechavaleta, J-P. Laumond, H. Hicheur, and A. Berthoz. “On the nonholonomic nature of human locomotion”. In: *Autonomous Robots* 25.1-2 (2008), pp. 25–35 (cit. on p. 93).
- [Asimov, 1950] I. Asimov. *I, Robot*. eng. Greenwich, Conn.: Fawcett Publications Greenwich, Conn., 1950 (cit. on p. 4).
- [Auda, 2021] J. Auda, M. Weigel, J. R. Cauchard, and S. Schneegass. “Understanding Drone Landing on the Human Body”. In: *2021 Int. Conf. on Mobile Human-Computer Interaction*. 2021 (cit. on p. 15).

- [Augugliaro, 2013] F. Augugliaro and R. D’Andrea. “Admittance control for physical human-quadrocopter interaction”. In: *2013 European Control Conference*. 2013, pp. 1805–1810 (cit. on pp. 16, 33).
- [Augugliaro, 2014] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D’Andrea. “The Flight Assembled Architecture installation: Cooperative construction with flying machines”. In: *IEEE Control Systems Magazine* 34.4 (2014), pp. 46–64 (cit. on p. 13).
- [Ayusawa, 2011] K. Ayusawa, G. Venture, and Y. Nakamura. “Real-time implementation of physically consistent identification of human body segments”. In: *2011 IEEE Int. Conf. on Robotics and Automation*. 2011, pp. 6282–6287 (cit. on p. 95).
- [Baizid, 2017] K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero. “Behavioral control of unmanned aerial vehicle manipulator systems”. In: *Autonomous Robots* 41.5 (2017), pp. 1203–1220 (cit. on p. 12).
- [Balaram, 2021] J. Balaram, M. Aung, and M. P. Golombek. “The Ingenuity Helicopter on the Perseverance Rover”. In: *Space Science Reviews* 217.4 (May 2021), p. 56 (cit. on p. 6).
- [Bangura, 2014] M. Bangura and R. Mahony. “Real-time Model Predictive Control for Quadrotors”. In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 11773–11780 (cit. on p. 85).
- [Bascetta, 2013] L. Bascetta, G. Ferretti, G. Magnani, and P. Rocco. “Walk-through programming for robotic manipulators based on admittance control”. In: *Robotica* 31.7 (2013), pp. 1143–1153 (cit. on p. 115).
- [Baskaya, 2021] E. Baskaya, M. Hamandi, M. Bronz, and A. Franchi. “A Novel Robust Hexarotor Capable of Static Hovering in Presence of Propeller Failure”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 4001–4008 (cit. on p. 87).
- [Bauersfeld, 2021] L. Bauersfeld, E. Kaufmann, P. Foehn, S. Sun, and D. Scaramuzza. “NeuroBEM: Hybrid Aerodynamic Quadrotor Model”. In: *2021 Robotics: Science and Systems*. 2021 (cit. on pp. 85, 86).
- [Beatty, 1986] M. F. Beatty. *Principles of Engineering Mechanics*. Mathematical Concepts and Methods in Science and Engineering 32-33. New York: Springer, 1986 (cit. on p. 72).

- [Bekey, 2008] G. Bekey, R. Ambrose, V. Kumar, D. Lavery, A. Sanderson, B. Wilcox, J. Yuh, and Y. Zheng. *Robotics: State of the Art and Future Challenges*. Imperial College Press, 2008 (cit. on p. 4).
- [Bellman, 1957] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957 (cit. on p. 130).
- [Bertsekas, 2010] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. 2010 (cit. on p. 130).
- [Bicego, 2019] D. Bicego. “Design and Control of Multi-Directional Thrust Multi-Rotor Aerial Vehicles with applications to Aerial Physical Interaction Tasks”. Theses. INSA de Toulouse, 2019 (cit. on pp. 9–11, 15, 35, 36, 83, 89, 129, 143).
- [Bicego, 2020] D. Bicego, J. Mazzetto, M. Farina, R. Carli, and A. Franchi. “Nonlinear Model Predictive Control with Enhanced Actuator Model for Multi-Rotor Aerial Vehicles with Generic Designs”. In: *Journal of Intelligent & Robotics Systems* 100 (2020), pp. 1213–1247 (cit. on pp. 7, 36, 37, 76, 90, 128).
- [Bock, 2000] H. G. Bock, M. Diehl, D. B. Leineweber, and J. P. Schlöder. “A Direct Multiple Shooting Method for Real-Time Optimization of Nonlinear DAE Processes”. In: *Nonlinear Model Predictive Control*. Ed. by Frank Allgöwer and Alex Zheng. Birkhäuser Basel, 2000, pp. 245–267 (cit. on p. 130).
- [Bock, 2007] H. G. Bock, M. Diehl, P. Kühn, E. Kostina, J. P. Schiöder, and L. Wirsching. “Numerical Methods for Efficient and Fast Nonlinear Model Predictive Control”. In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. Ed. by R. Findeisen, F. Allgöwer, and L. T. Biegler. Springer Berlin Heidelberg, 2007, pp. 163–179 (cit. on p. 128).
- [Bodie, 2021a] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto. “Active Interaction Force Control for Contact-Based Inspection With a Fully Actuated Aerial Vehicle”. In: *IEEE Transactions on Robotics* 37.3 (2021), pp. 709–722 (cit. on p. 12).
- [Bodie, 2021b] K. Bodie, M. Tognon, and R. Siegwart. “Dynamic End Effector Tracking With an Omnidirectional Parallel Aerial Manipulator”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 8165–8172 (cit. on p. 13).
- [Bork, 1966] A. M. Bork. ““Vectors Versus Quaternions”—The Letters in *Nature*”. In: *American Journal of Physics* 34.3 (1966), pp. 202–211 (cit. on p. 70).

- [Brescianini, 2016] D. Brescianini and R. D’Andrea. “Design, modeling and control of an omni-directional aerial vehicle”. In: *2016 IEEE Int. Conf. on Robotics and Automation*. 2016, pp. 3261–3266 (cit. on p. 87).
- [Busch, 2017] B. Busch, G. Maeda, Y. Mollard, M. Demangeat, and M. Lopes. “Postural optimization for an ergonomic human-robot interaction”. In: *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2017, pp. 2778–2785 (cit. on p. 41).
- [Busch, 2018] B. Busch, M. Toussaint, and M. Lopes. “Planning Ergonomic Sequences of Actions in Human-Robot Interaction”. In: *2018 IEEE Int. Conf. on Robotics and Automation*. 2018, pp. 1916–1923 (cit. on p. 42).
- [Butcher, 1996] J.C. Butcher. “A history of Runge-Kutta methods”. In: *Applied Numerical Mathematics* 20.3 (1996), pp. 247–260 (cit. on p. 145).
- [Cacace, 2016] J. Cacace, A. Finzi, V. Lippiello, M. Furci, N. Mimmo, and L. Marconi. “A control architecture for multiple drones operated via multimodal interaction in search & rescue mission”. In: *2016 IEEE Int. Symp. on Safety, Security and Rescue Robotics*. 2016, pp. 233–239 (cit. on p. 16).
- [Cacace, 2021] J. Cacace, S. M. Orozco-Soto, A. Suarez, A. Caballero, M. Orsag, S. Bogdan, G. Vasiljevic, E. Ebeid, J. A. A. Rodriguez, and A. Ollero. “Safe Local Aerial Manipulation for the Installation of Devices on Power Lines: AERIAL-CORE First Year Results and Designs”. In: *Applied Sciences* 11.13 (2021), p. 6220 (cit. on pp. 13, 14).
- [Campa, 2009] R. Campa and H. de la Torre. “Pose control of robot manipulators using different orientation representations: A comparative review”. In: *2009 American Control Conference*. 2009, pp. 2855–2860 (cit. on p. 108).
- [Carpentier, 2017] J. Carpentier, M. Benallegue, and J-P. Laumond. “On the centre of mass motion in human walking”. In: *International Journal of Automation and Computing* 14.5 (2017), pp. 542–551 (cit. on p. 93).
- [Castillo-Lopez, 2018] M. Castillo-Lopez, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos. “Model Predictive Control for Aerial Collision Avoidance in Dynamic Environments”. In: *2018 Mediterranean Conf. on Control and Automation*. 2018, pp. 1–6 (cit. on p. 40).

- [Cataldi, 2019] E. Cataldi, F. Real, A. Suarez, P.A. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, G. Heredia, and A. Ollero. “Set-based Inverse Kinematics Control of an Anthropomorphic Dual Arm Aerial Manipulator”. In: *2019 IEEE Int. Conf. on Robotics and Automation*. 2019, pp. 2960–2966 (cit. on p. 13).
- [Cauchard, 2015] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay. “Drone & Me: An Exploration into Natural Human-Drone Interaction”. In: *2015 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*. 2015, pp. 361–365 (cit. on p. 15).
- [Cauchard, 2016] J. R. Cauchard, K. Y. Zhai, M. Spadafora, and J. A. Landay. “Emotion encoding in Human-Drone Interaction”. In: *2016 ACM/IEEE Int. Conf. on Human-Robot Interaction*. 2016, pp. 263–270 (cit. on p. 15).
- [Chaumette, 2006] F. Chaumette and S. Hutchinson. “Visual servo control. I. Basic approaches”. In: *IEEE Robotics & Automation Magazine* 13.4 (2006), pp. 82–90 (cit. on pp. 37, 100).
- [Chaumette, 2007] F. Chaumette and S. Hutchinson. “Visual servo control. II. Advanced approaches [Tutorial]”. In: *IEEE Robotics & Automation Magazine* 14.1 (2007), pp. 109–118 (cit. on pp. 99, 100).
- [Chen, 2019] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi. “MATMPC - A MATLAB Based Toolbox for Real-time Nonlinear Model Predictive Control”. In: *2019 European Control Conference*. 2019, pp. 3365–3370 (cit. on pp. 130, 147).
- [Chettibi, 2007] T. Chettibi and P. Lemoine. “Generation of Point to Point Trajectories for Robotic Manipulators Under Electro-Mechanical Constraints”. In: *International Review of Mechanical Engineering, IREME, ISSN 1970-8734* 1.2 (2007), pp. 131–143 (cit. on p. 89).
- [Clark, 2022] A. B. Clark, N. Baron, L. Orr, M. Kovac, and N. Rojas. “On a Balanced Delta Robot for Precise Aerial Manipulation: Implementation, Testing, and Lessons for Future Designs”. In: *2022 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2022, pp. 7359–7366 (cit. on p. 13).
- [Conticelli, 1999] F. Conticelli, B. Allotta, and C. Colombo. “Hybrid visual servoing: A combination of nonlinear control and linear vision”. In: *Robotics and Autonomous Systems* 29.4 (1999), pp. 243–256 (cit. on p. 37).

- [Corsini, 2021] G. Corsini, M. Jacquet, A. E. Jimenez-Cano, A. Afifi, D. Sidobre, and A. Franchi. “A General Control Architecture for Visual Servoing and Physical Interaction Tasks for Fully-actuated Aerial Vehicles”. In: *1st Work. on Aerial Robotic Systems Physically Interacting with the Environment*. Biograd na Moru, Croatia, 2021, pp. 1–8 (cit. on p. 21).
- [Corsini, 2022] G. Corsini, M. Jacquet, H. Das, A Afifi, D. Sidobre, and A. Franchi. “Nonlinear Model Predictive Control for Human-Robot Handover with Application to the Aerial Case”. In: *2022 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Kyoto, Japan, 2022, pp. 7597–7604 (cit. on p. 21).
- [Costanzo, 2023] M. Costanzo, C. Natale, and M. Selvaggio. “Visual and Haptic Cues for Human-Robot Handover”. In: *2023 IEEE Int. Symp. on Robots and Human Interactive Communications. 2023* (cit. on p. 125).
- [Costic, 2001] B. T. Costic, D. M. Dawson, M. S. de Queiroz, and V. Kapila. “Quaternion-based adaptive attitude tracking controller without velocity measurements”. In: *AIAA Journal of Guidance, Control, and Dynamics* 24.6 (2001), pp. 1214–1222 (cit. on p. 71).
- [Croon, 2009] G.C.H.E. de Croon, K.M.E. de Clercq, R. Ruijsink, B. Remes, and C. de Wagter. “Design, Aerodynamics, and Vision-Based Control of the DelFly”. In: *International Journal of Micro Air Vehicles* 1.2 (2009), pp. 71–97 (cit. on p. 7).
- [Cutler, 2015] M. Cutler and J. P. How. “Analysis and control of a variable-pitch quadrotor for agile flight”. In: *ASME Journal on Dynamic Systems, Measurement, and Control* 137 (2015) (cit. on p. 7).
- [Dai, 2015] J. S. Dai. “Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections”. In: *Mechanism and Machine Theory* 92 (2015), pp. 144–152 (cit. on p. 70).
- [Darivianakis, 2014] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart. “Hybrid predictive control for aerial robotic physical interaction towards inspection operations”. In: *2014 IEEE Int. Conf. on Robotics and Automation*. 2014, pp. 53–58 (cit. on p. 36).
- [Darvish, 2023] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci. “Teleoperation of Humanoid Robots: A Survey”. In: *IEEE Trans. on Robotics* 39.3 (2023), pp. 1706–1727 (cit. on p. 5).



- [de Angelis, 2019] E. L. de Angelis, F. Giulietti, and G. Pipeleers. “Two-time-scale control of a multirotor aircraft for suspended load transportation”. In: *Aerospace Science and Technology* 84 (2019), pp. 193–203 (cit. on p. 8).
- [Diebel, 2006] J. Diebel. “Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors”. In: *Matrix* 58.15-16 (2006), pp. 1–35 (cit. on p. 72).
- [Diehl, 2002] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations”. In: *Journal of Process Control* 12.4 (2002), pp. 577–585 (cit. on p. 127).
- [Dietrich, 2016] A. Dietrich, K. Bussmann, F. Petit, P. Kotyczka, C. Ott, B. Lohmann, and A. Albu-Schäffer. “Whole-body impedance control of wheeled mobile manipulators”. In: *Autonomous Robots* 40.3 (2016), pp. 505–517 (cit. on pp. 116, 125).
- [Dimeas, 2016] F. Dimeas and N. Aspragathos. “Online Stability in Human-Robot Cooperation with Admittance Control”. In: *IEEE Trans. on Haptics* 9.2 (2016), pp. 267–278 (cit. on p. 32).
- [Djuric, 2016] A. M. Djuric, R.J. Urbanic, and J.L. Rickli. “A Framework for Collaborative Robot (CoBot) Integration in Advanced Manufacturing Systems”. In: *SAE International Journal of Materials and Manufacturing* 9.2 (2016), pp. 457–464 (cit. on p. 4).
- [Dmytruk, 2022] A. Dmytruk, G. Silano, D. Bicego, D. B. Licea, and M. Saska. “A Perception-Aware NMPC for Vision-Based Target Tracking and Collision Avoidance with a Multi-Rotor UAV”. In: *2022 Int. Conf. on Unmanned Aircraft Systems*. 2022, pp. 1668–1673 (cit. on p. 40).
- [Doty, 1993] K. L. Doty, C. Melchiorri, and C. Bonivento. “A Theory of Generalized Inverses Applied to Robotics”. In: *The International Journal of Robotics Research* 12.1 (1993), pp. 1–19 (cit. on p. 78).
- [Drag, 2016] P. Drag, K. Styczeń, M. Kwiatkowska, and A. Szczurek. “A Review on the Direct and Indirect Methods for Solving Optimal Control Problems with Differential-Algebraic Constraints”. In: *Recent Advances in Computational Optimization: Results of the Workshop on Computational Optimization WCO 2014*. Ed. by S. Fidanova. Springer International Publishing, 2016, pp. 91–105 (cit. on p. 130).

- [Duncan, 2013] B. A. Duncan and R. R. Murphy. “Comfortable approach distance with small Unmanned Aerial Vehicles”. In: *2013 IEEE Int. Symp. on Robots and Human Interactive Communications*. 2013, pp. 786–792 (cit. on p. 15).
- [Eren, 2017] U. Eren, A. Prach, B. B. Koçer, S. V. Rakovic, E. Kayacan, and B. Açikmese. “Model predictive control in aerospace systems: Current state and opportunities”. In: *AIAA Journal of Guidance, Control, and Dynamics* 40.7 (2017), pp. 1541–1566 (cit. on p. 127).
- [Faessler, 2017] M. Faessler, D. Falanga, and D. Scaramuzza. “Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight”. In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 476–482 (cit. on p. 90).
- [Falanga, 2018] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza. “PAMPC: Perception-Aware Model Predictive Control for Quadrotors”. In: *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2018, pp. 1–8 (cit. on p. 38).
- [Fantoni, 2001] Isabelle Fantoni and Rogelio Lozano. *Non-linear Control for Underactuated Mechanical Systems*. Springer, 2001 (cit. on p. 9).
- [Featherstone, 2008] R. Featherstone. *Rigid body dynamics algorithms*. New York: Springer, 2008 (cit. on pp. 74, 83).
- [Fernández, 2016] R. A. Suárez Fernández, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy. “Natural user interfaces for human-drone multi-modal interaction”. In: *2016 Int. Conf. on Unmanned Aircraft Systems*. 2016, pp. 1013–1022 (cit. on p. 16).
- [Feron, 2008] E. Feron and E. N. Johnson. “Aerial Robotics”. In: *Springer Handbook of Robotics*. Ed. by B. Siciliano and O. Khatib. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1009–1029 (cit. on p. 5).
- [Ferraguti, 2020] F. Ferraguti, R. Villa, C. T. Landi, A. M. Zanchettin, P. Rocco, and C. Secchi. “A Unified Architecture for Physical and Ergonomic Human–Robot Collaboration”. In: *Robotica* 38.4 (2020), pp. 669–683 (cit. on p. 41).
- [Fink, 2011] J. Fink, N. Michael, S. Kim, and V. Kumar. “Planning and control for cooperative manipulation and transportation with aerial robots”. In: *The International Journal of Robotics Research* 30.3 (2011), pp. 324–334 (cit. on p. 13).

- [Fisch, 2022] M. Fisch and R. Clark. “Orientation Keypoints for 6D Human Pose Estimation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12 (2022), pp. 10145–10158 (cit. on p. 166).
- [Foehn, 2022] P. Foehn et al. “Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight”. In: *Science Robotics* 7.67 (2022) (cit. on p. 8).
- [Franchi, 2018] A. Franchi, R. Carli, D. Bicego, and M. Ryll. “Full-Pose Tracking Control for Aerial Robotic Systems with Laterally-Bounded Input Force”. In: *IEEE Trans. on Robotics* 34.2 (2018), pp. 534–541 (cit. on pp. 10, 35, 103–105).
- [Gabellieri, 2023] C. Gabellieri, M. Tognon, D. Sanalidro, and A. Franchi. “Equilibria, Stability, and Sensitivity for the Aerial Suspended Beam Robotic System Subject to Parameter Uncertainty”. In: *IEEE Trans. on Robotics* (2023), pp. 1–17 (cit. on pp. 12, 13).
- [Garrido-Jurado, 2014] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292 (cit. on pp. 50, 107, 147).
- [Gio, 2021] Nicolas Gio, Ross Brisco, and Tijana Vuletic. “Control of a drone with body gestures”. In: *Proceedings of the Design Society* 1 (2021), pp. 761–770 (cit. on p. 15).
- [Gioioso, 2014] G. Gioioso, M. Ryll, D. Prattichizzo, H. H. Bühlhoff, and A. Franchi. “Turning a near-hovering controlled quadrotor into a 3D force effector”. In: *2014 IEEE Int. Conf. on Robotics and Automation*. 2014, pp. 6278–6284 (cit. on pp. 12, 109).
- [Girard, 2007] P. R. Girard. *Quaternions, Clifford Algebras and Relativistic Physics*. Birkhäuser Basel, 2007 (cit. on pp. 70, 72).
- [Goldstein, 1950] H. Goldstein. *Classical Mechanics*. A-W series in advanced physics. Addison-Wesley Press, 1950 (cit. on pp. 65, 72).
- [González-Morgado, 2023] A. González-Morgado, C. Álvarez-Cía, G. Heredia, and A. Ollero. “Fully-actuated, corner contact aerial robot for inspection of hard-to-reach bridge areas”. In: *2023 Int. Conf. on Unmanned Aircraft Systems*. 2023, pp. 1191–1198 (cit. on p. 13).
- [Goodarzi, 2013] F. Goodarzi, D. Lee, and T. Lee. “Geometric nonlinear PID control of a quadrotor UAV on SE(3)”. In: *2013 European Control Conference*. 2013, pp. 3845–3850 (cit. on p. 106).

- [Goodrich, 2008] M. A. Goodrich and A. C. Schultz. “Human–Robot Interaction: A Survey”. In: *Foundations and Trends® in Human–Computer Interaction* 1.3 (2008), pp. 203–275 (cit. on pp. 3, 5).
- [Goodrich, 2013] Michael A. Goodrich, Jacob W. Crandall, and Emilia Barakova. “Teleoperation and Beyond for Assistive Humanoid Robots”. In: *Reviews of Human Factors and Ergonomics* 9.1 (2013), pp. 175–226 (cit. on pp. 5, 17).
- [Grande, 2018] D. Grande, L. Bascetta, and A. Martins. “Modeling and simulation of a spherical vehicle for underwater surveillance”. In: *OCEANS 2018 MTS/IEEE Charleston*. 2018, pp. 1–7 (cit. on p. 71).
- [Gros, 2020] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl. “From linear to nonlinear MPC: bridging the gap via the real-time iteration”. In: *International Journal of Control* 93.1 (2020), pp. 62–80 (cit. on p. 128).
- [Haddadin, 2007] S. Haddadin, A. O. Albu-Schäffer, and G. Hirzinger. “Safety Evaluation of Physical Human-Robot Interaction via Crash-Testing”. In: *Robotics: Science and Systems*. 2007 (cit. on pp. 39, 143).
- [Haddadin, 2009] S. Haddadin, A. Albu-Schaffer, M. Frommberger, J. Rossmann, and G. Hirzinger. “The “DLR Crash Report”: Towards a standard crash-testing protocol for robot safety - Part I: Results”. In: *2009 IEEE Int. Conf. on Robotics and Automation*. 2009, pp. 272–279 (cit. on p. 39).
- [Haddadin, 2017] S. Haddadin, A. De Luca, and A. Albu-Schäffer. “Robot Collisions: A Survey on Detection, Isolation, and Identification”. In: *IEEE Trans. on Robotics* 33.6 (2017), pp. 1292–1312 (cit. on pp. 29, 39).
- [Hallworth, 2023] Ben Hallworth, Mike Allenspach, Roland Siegwart, and Marco Tognon. “State-Aware Path-Following with Humans Through Force-based Communication via Tethered Physical Aerial Human-Robot Interaction”. In: *2023 Int. Conf. on Unmanned Aircraft Systems*. 2023, pp. 183–190 (cit. on p. 16).
- [Hamandi, 2020] Mahmoud Hamandi, Kapil Sawant, Marco Tognon, and Antonio Franchi. “Omni-Plus-Seven (O7+): An Omnidirectional Aerial Prototype with a Minimal Number of Unidirectional Thrusters”. In: *2020 Int. Conf. on Unmanned Aircraft Systems*. 2020, pp. 754–761 (cit. on pp. 10, 11).

- [Hamandi, 2021] M. Hamandi, F. Usai, Q. Sable, N. Staub, M. Tognon, and A. Franchi. “Design of Multirotor Aerial Vehicles: a Taxonomy Based on Input Allocation”. In: *The International Journal of Robotics Research* 40.8-9 (2021), pp. 1015–1044 (cit. on pp. 10, 11, 84, 87, 106).
- [Hamel, 2002] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. “Dynamic modelling and configuration stabilization for an X4-flyer.” In: *IFAC Proceedings Volumes* 35.1 (2002), pp. 217–222 (cit. on p. 85).
- [Hanover, 2023] D. Hanover, A. Loquercio, L. Bauersfeld, A. Romero, R. Penicka, Y. Song, G. Cioffi, E. Kaufmann, and D. Scaramuzza. “Autonomous Drone Racing: A Survey”. 2023 (cit. on pp. 7, 79).
- [Hattenberger, 2022] G. Hattenberger, T. Verdu, N. Maury, P. Narvor, F. Couvreur, M. Bronz, S. Lacroix, G. Cayez, and G. C. Roberts. “Field report: deployment of a fleet of drones for cloud exploration”. In: *Int. Journ. of Micro Air Vehicles* 14 (2022) (cit. on p. 6).
- [He, 2023] G. He, Y. Jangir, J. Geng, M. Mousaei, D. Bai, and S. Scherer. “Image-based Visual Servo Control for Aerial Manipulation Using a Fully-Actuated UAV”. 2023 (cit. on p. 38).
- [Hehn, 2014] M. Hehn and R. D’Andrea. “A frequency domain iterative learning algorithm for high-performance, periodic quadcopter maneuvers”. In: *Mechatronics* 24.8 (2014), pp. 954–965 (cit. on p. 7).
- [Hentout, 2019] A. Hentout, M. Aouache, A. Maoudj, and I. Akli. “Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017”. In: *Advanced Robotics* 33.15-16 (2019), pp. 764–799 (cit. on p. 4).
- [Heyer, 2010] C. Heyer. “Human-robot interaction and future industrial robotics applications”. In: *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2010, pp. 4749–4754 (cit. on p. 4).
- [Hignett, 2000] S. Hignett and L. McAtamney. “Rapid Entire Body Assessment (REBA)”. In: *Applied Ergonomics* 31.2 (2000), pp. 201–205 (cit. on p. 41).
- [Hirzinger, 1999] G. Hirzinger, M. Fischer, B. Brunner, R. Koeppel, M. Otter, M. Grebenstein, and I. Schäfer. “Advances in Robotics: The DLR Experience”. In: *The International Journal of Robotics Research* 18.11 (1999), pp. 1064–1087 (cit. on p. 4).

- [HNguyen, 2021] H.Nguyen, M. Kamel, K. Alexis, and R. Siegwart. “Model Predictive Control for Micro Aerial Vehicles: A Survey”. In: *2021 European Control Conference*. 2021, pp. 1556–1563 (cit. on p. 127).
- [Hoeijmakers, 2022] H. W. M. Hoeijmakers, L. H. G. Koerkamp, L. D. de Santana, C. H. Venner, S. Stramigioli, J. L. Mulder, A. Brentjes, F. Gijsman, and S. A. Hartman. “Investigation flapping-flight aerodynamics of a robotic bird”. In: *2022 33rd Congress of the International Council of the Aeronautical Sciences*. 2022, pp. 3326–3351 (cit. on p. 7).
- [Hofer, 2016] M. Hofer, M. Muehlebach, and R. D’Andrea. “Application of an approximate model predictive control scheme on an unmanned aerial vehicle”. In: *2016 IEEE Int. Conf. on Robotics and Automation*. 2016, pp. 2952–2957 (cit. on p. 36).
- [Houska, 2011] B. Houska, H. J. Ferreau, and M. Diehl. “An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range”. In: *Automatica* 47.10 (2011), pp. 2279–2285 (cit. on p. 128).
- [Hutchinson, 1996] S. Hutchinson, G. D. Hager, and P. I. Corke. “A tutorial on visual servo control”. In: *IEEE Trans. on Robotics and Automation* 12.5 (1996), pp. 651–670 (cit. on p. 37).
- [Huynh, 2009] D. Q. Huynh. “Metrics for 3D Rotations: Comparison and Analysis”. In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164 (cit. on p. 145).
- [Isidori, 2013] A. Isidori. *Nonlinear Control Systems*. Springer, 2013 (cit. on p. 34).
- [Jacquet, 2021] M. Jacquet and A. Franchi. “Motor and Perception Constrained NMPC for Torque-controlled Generic Aerial Vehicles”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 518–525 (cit. on pp. 31, 38, 39, 76, 92).
- [Jacquet, 2022a] M. Jacquet and A. Franchi. “Enforcing Vision-Based Localization using Perception Constrained N-MPC for Multi-Rotor Aerial Vehicles”. In: *2022 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2022, pp. 1818–1824 (cit. on p. 29).
- [Jacquet, 2022b] M. Jacquet. “Methods for Online Predictive Control of Multi-rotor Aerial Robots with Perception-driven Tasks subject to Sensing and Actuation Constraints”. Theses. INSA de Toulouse, 2022 (cit. on pp. 37, 39, 79, 92, 129, 135, 142, 145, 147, 151, 152).

- [Jiang, 2018] G. Jiang, R. M. Voyles, and J. J. Choi. “Precision Fully-Actuated UAV for Visual and Physical Inspection of Structures for Nuclear Decommissioning and Search and Rescue”. In: *2018 IEEE Int. Symp. on Safety, Security and Rescue Robotics*. 2018, pp. 1–7 (cit. on p. 13).
- [Jimenez-Cano, 2017] A. E. Jimenez-Cano, G. Heredia, and A. Ollero. “Aerial manipulator with a compliant arm for bridge inspection”. In: *2017 Int. Conf. on Unmanned Aircraft Systems*. 2017, pp. 1217–1222 (cit. on p. 13).
- [Jovic, 2016] J. Jovic, A. Escande, K. Ayusawa, E. Yoshida, A. Kheddar, and G. Venture. “Humanoid and Human Inertia Parameter Identification Using Hierarchical Optimization”. In: *IEEE Trans. on Robotics* 32.3 (2016), pp. 726–735 (cit. on p. 95).
- [Kamel, 2017] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto. “Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control”. In: *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2017, pp. 236–243 (cit. on p. 40).
- [Kamel, 2018] M. Kamel, S. Verling, O. Elkhatib, C. Sprecher, P. Wulkop, Z. Taylor, R. Siegwart, and I. Gilitschenski. “The Voliro Omniorientational Hexacopter: An Agile and Maneuverable Tilttable-Rotor Aerial Vehicle”. In: *IEEE Robotics & Automation Magazine* 25.4 (2018), pp. 34–44 (cit. on p. 11).
- [Katayama, 2003] M. Katayama and H. Hasuura. “Optimization principle determines human arm postures and ”comfort””. In: *2003 SICE Annual Conference*. Vol. 1. 2003, pp. 1000–1005 (cit. on pp. 42, 140).
- [Katayama, 2023] S. Katayama, M. Murooka, and Y. Tazaki. “Model predictive control of legged and humanoid robots: models and algorithms”. In: *Advanced Robotics* 37.5 (2023), pp. 298–315 (cit. on p. 127).
- [Keipour, 2022] A. Keipour, G. A. S. Pereira, R. Bonatti, R. Garg, P. Rastogi, G. Dubey, and S. Scherer. “Visual Servoing Approach to Autonomous UAV Landing on a Moving Vehicle”. In: *Sensors* 22.17 (2022) (cit. on p. 38).
- [Kelly, 1994] A. Kelly. *A 3D state space formulation of a navigation Kalman filter for autonomous vehicles*. Tech. rep. Carnegie-Mellon Univ Pittsburgh Pa Robotics Inst, 1994 (cit. on p. 90).
- [Kibble, 2004] T. W. B. Kibble and F. H. Berkshire. *Classical mechanics*. Imperial College Press, 2004 (cit. on p. 72).

- [Kim, 2013] S. Kim, S. Choi, and H. J. Kim. “Aerial manipulation using a quadrotor with a two DOF robotic arm”. In: *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Tokyo, Japan, 2013, pp. 4990–4995 (cit. on p. 12).
- [Kim, 2018] W. Kim, J. Lee, L. Peternel, N. Tsagarakis, and A. Ajoudani. “Anticipatory Robot Assistance for the Prevention of Human Static Joint Overloading in Human–Robot Collaboration”. In: *IEEE Robotics and Automation Letters* 3.1 (2018), pp. 68–75 (cit. on p. 42).
- [Kim, 2019] W. Kim, M. Lorenzini, P. Balatti, Y. Wu, and A. Ajoudani. “Towards Ergonomic Control of Collaborative Effort in Multi-human Mobile-robot Teams”. In: *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2019, pp. 3005–3011 (cit. on p. 42).
- [Kirsch, 2010] A. Kirsch, T. Kruse, E. A. Sisbot, R. Alami, M. Lawitzky, D. Brščić, S. Hirche, P. Basili, and S. Glasauer. “Plan-Based Control of Joint Human-Robot Activities”. In: *KI - Künstliche Intelligenz* 24.3 (2010), pp. 223–231 (cit. on p. 5).
- [Kocer, 2018] B. B. Kocer, T. Tjahjowidodo, and G. G. L. Seet. “Centralized predictive ceiling interaction control of quadrotor VTOL UAV”. In: *Aerospace Science and Technology* 76 (2018), pp. 455–465 (cit. on p. 36).
- [Kocer, 2019] B. B. Kocer, T. Tjahjowidodo, and G. G. L. Seet. “Model predictive UAV-tool interaction control enhanced by external forces”. In: *Mechatronics* 58 (2019), pp. 47–57 (cit. on p. 156).
- [Korpela, 2014] C. Korpela, M. Orsag, and P. Oh. “Towards valve turning using a dual-arm aerial manipulator”. In: *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2014, pp. 3411–3416 (cit. on p. 13).
- [Kotarski, 2021] Denis Kotarski, Petar Piljek, Marko Pranjić, Carlo Giorgio Grlj, and Josip Kasać. “A Modular Multirotor Unmanned Aerial Vehicle Design Approach for Development of an Engineering Education Platform”. In: *Sensors* 21.8 (2021) (cit. on p. 8).
- [Kragic, 2002] D. Kragic and H. I. Christensen. *Survey on Visual Servoing for Manipulation*. Tech. rep. 2002 (cit. on p. 38).
- [Krishna, 2015] R. V. Krishna, B. S. Sathish, P. Ganesan, P. J. Babu, and R. Abilash. “Design of voice and gesture controlled Quadcopter”. In: *2015 Int. Conf. on Innovations in Information, Embedded and Communication Systems*. 2015, pp. 1–6 (cit. on p. 16).



- [Kuipers, 1999] J. B. Kuipers. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press, 1999 (cit. on p. 72).
- [Kulić, 2005] D. Kulić and E. A. Croft. “Safe planning for human-robot interaction”. In: *Journal of Robotic Systems* 22.7 (2005), pp. 383–396 (cit. on p. 30).
- [Kurfess, 2005] T. R. Kurfess, ed. *Robotics and automation handbook*. CRC Press, 2005 (cit. on pp. 72, 73).
- [Laumond, 2017] J-P. Laumond, M. Benallegue, J. Carpentier, and A. Berthoz. “The Yoyo-Man”. In: *The International Journal of Robotics Research* 36.13-14 (2017), pp. 1508–1520 (cit. on p. 93).
- [Lee, 2010] T. Lee, M. Leoky, and N. H. McClamroch. “Geometric tracking control of a quadrotor UAV on SE(3)”. In: *49th IEEE Conf. on Decision and Control*. Atlanta, GA, 2010, pp. 5420–5425 (cit. on p. 103).
- [Li, 2021] S. Li, K. Han, X. Li, S. Zhang, Y. Xiong, and Z. Xie. “Hybrid Trajectory Replanning-Based Dynamic Obstacle Avoidance for Physical Human-Robot Interaction”. In: *Journal of Intelligent & Robotics Systems* 103.3 (2021), p. 41 (cit. on p. 40).
- [Li, 2023] G. Li, X. Liu, and G. Loianno. “Safety-Aware Human-Robot Collaborative Transportation and Manipulation with Multiple MAVs”. 2023 (cit. on p. 16).
- [Liang, 2018] X. Liang, Y. Fang, N. Sun, and H. Lin. “Dynamics analysis and time-optimal motion planning for unmanned quadrotor transportation systems”. In: *Mechatronics* 50 (2018), pp. 16–29 (cit. on p. 12).
- [Lieser, 2021] M. Lieser, U. Schwanecke, and J. Berdux. “Evaluating Distances in Tactile Human-Drone Interaction”. In: *2021 IEEE Int. Symp. on Robots and Human Interactive Communications*. 2021, pp. 1275–1282 (cit. on p. 15).
- [Lindqvist, 2020] B. Lindqvist, S. S. Mansouri, A. Agha-mohammadi, and G. Nikolakopoulos. “Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6001–6008 (cit. on p. 129).
- [Lindsey, 2012] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. “Construction with quadrotor teams”. In: *Autonomous Robots* 33.3 (2012), pp. 323–336 (cit. on pp. 8, 13).

- [Lippiello, 2018] V. Lippiello, G. A. Fontanelli, and F. Ruggiero. “Image-Based Visual-Impedance Control of a Dual-Arm Aerial Manipulator”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1856–1863 (cit. on p. 38).
- [Liu, 2013] H. Liu, D. Derawi, J. Kim, and Y. Zhong. “Robust optimal attitude control of hexarotor robotic vehicles”. In: *Nonlinear Dynamics* 74.4 (2013), pp. 1155–1168 (cit. on p. 10).
- [Lorenzini, 2018] Marta Lorenzini, Wansoo Kim, Elena De Momi, and Arash Ajoudani. “A Synergistic Approach to the Real-Time Estimation of the Feet Ground Reaction Forces and Centers of Pressure in Humans With Application to Human–Robot Collaboration”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3654–3661 (cit. on p. 42).
- [Lorenzini, 2023] M. Lorenzini, M. Lagomarsino, L. Fortini, S. Gholami, and A. Ajoudani. “Ergonomic human-robot collaboration in industry: A review”. In: *Frontiers in Robotics and AI* 9 (2023) (cit. on pp. 30, 41).
- [Luca, 1998] A. De Luca. “Decoupling and feedback linearization of robots with mixed rigid/elastic joints”. In: *International Journal on Robust and Nonlinear Control* 8.11 (1998), pp. 965–977 (cit. on p. 34).
- [Mahony, 2004] R. Mahony and T. Hamel. “Robust trajectory tracking for a scale model autonomous helicopter”. In: *International Journal on Robust and Nonlinear Control* 14.12 (2004), pp. 1035–1059 (cit. on p. 35).
- [Mahony, 2012] R. Mahony, V. Kumar, and P. Corke. “Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor”. In: *IEEE Robotics & Automation Magazine* 19.3 (2012), pp. 20–32 (cit. on p. 87).
- [Makrini, 2022] I. El Makrini, G. Mathijssen, S. Verhaegen, T. Verstraten, and B. Vanderborght. “A Virtual Element-Based Postural Optimization Method for Improved Ergonomics During Human-Robot Collaboration”. In: *IEEE Trans. on Automation Science and Engineering* 19.3 (2022), pp. 1772–1783 (cit. on p. 41).
- [Malis, 1999] E. Malis, F. Chaumette, and S. Boudet. “2 1/2 D visual servoing”. In: *IEEE Trans. on Robotics and Automation* 15.2 (1999), pp. 238–250 (cit. on p. 37).
- [Mallet, 2010] A. Mallet, C. Pasteur, M. Herrb, S. Lemaignan, and F. Ingrand. “GenoM3: Building middleware-independent robotic components”. In: *2010 IEEE Int. Conf. on Robotics and Automation*. 2010, pp. 4627–4632 (cit. on pp. 107, 147).

- [Maroger, 2022] I. Maroger, O. Stasse, and B. Watier. “From the Study of Table Trajectories during Collaborative Carriages toward Pro-active Human-Robot Table Handling Tasks”. In: *2022 IEEE-RAS 21st Int. Conf. on Humanoid Robots*. 2022, pp. 911–918 (cit. on p. 17).
- [Martí-Saumell, 2021] J. Martí-Saumell, J. Solà, A. Santamaria-Navarro, and J. Andrade-Cetto. “Full-Body Torque-Level Non-linear Model Predictive Control for Aerial Manipulation”. 2021 (cit. on p. 37).
- [Martí-Saumell, 2023] J. Martí-Saumell, H. Duarte, P. Grosch, J. Andrade-Cetto, A. Santamaria-Navarro, and J. Solà. “Borinot: an open thrust-torque-controlled robot for research on agile aerial-contact motion”. 2023 (cit. on pp. 115, 166).
- [McAtamney, 1993] L. McAtamney and E. Nigel Corlett. “RULA: a survey method for the investigation of work-related upper limb disorders”. In: *Applied Ergonomics* 24.2 (1993), pp. 91–99 (cit. on p. 41).
- [Medina, 2016] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard. “A human-inspired controller for fluid human-robot handovers”. In: *2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots*. 2016, pp. 324–331 (cit. on pp. 17, 23).
- [Mellinger, 2011] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. “Design, modeling, estimation and control for aerial grasping and manipulation”. In: *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2011, pp. 2668–2673 (cit. on p. 79).
- [Mellinger, 2014] D. Mellinger, N. Michael, and V. Kumar. “Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors”. In: *Tracts in Advanced Robotics*. Ed. by Oussama Khatib, Vijay Kumar, and Gaurav Sukhatme. Vol. 79. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2014, pp. 361–373 (cit. on p. 7).
- [Mesbah, 2016] A. Mesbah. “Stochastic Model Predictive Control: An Overview and Perspectives for Future Research”. In: *IEEE Control Systems Magazine* 36.6 (2016), pp. 30–44 (cit. on p. 166).
- [Michael, 2010] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. “The GRASP Multiple Micro-UAV Testbed”. In: *IEEE Robotics & Automation Magazine* 17.3 (2010), pp. 56–65 (cit. on p. 7).

- [Michieletto, 2018] G. Michieletto, M. Ryll, and A. Franchi. “Fundamental Actuation Properties of Multirotors: Force-Moment Decoupling and Fail-Safe Robustness”. In: *IEEE Trans. on Robotics* 34.3 (2018), pp. 702–715 (cit. on pp. 10, 76).
- [Mohta, 2018] K. Mohta, K. Sun, S. Liu, M. Watterson, B. Pfrommer, J. Svacha, Y. Mulgaonkar, C. J. Taylor, and V. Kumar. “Experiments in fast, autonomous, gps-denied quadrotor flight”. In: *2018 IEEE Int. Conf. on Robotics and Automation*. 2018, pp. 7832–7839 (cit. on p. 92).
- [Morin, 2015] P. Morin. “Modeling and control of convertible Micro Air Vehicles”. In: *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*. 2015, pp. 188–198 (cit. on p. 7).
- [Moritz, 2001] D. Moritz. “Real-Time Optimization for Large Scale Nonlinear Processes”. Dissertation. Heidelberg University, 2001 (cit. on p. 36).
- [Morlando, 2023] V. Morlando, V. Lippiello, and F. Ruggiero. “Tethering a Human with a Quadruped Robot: A Guide Dog to Help Visually Impaired People”. In: *2023 Mediterranean Conf. on Control and Automation*. 2023, pp. 547–553 (cit. on p. 5).
- [Nagano, 2005] A. Nagano, S. Yoshioka, T. Komura, R. Himeno, and S. Fukushima. “A Three-Dimensional Linked Segment Model of the Whole Human Body”. In: *International Journal of Sport and Health Science* 3 (2005), pp. 311–325 (cit. on p. 95).
- [Nagi, 2014] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella. “Human Control of UAVs using Face Pose Estimates and Hand Gestures”. In: *2014 ACM/IEEE Int. Conf. on Human-Robot Interaction*. 2014, pp. 1–2 (cit. on p. 16).
- [Nakanishi, 2007] J. Nakanishi, M. Mistry, and S. Schaal. “Inverse Dynamics Control with Floating Base and Constraints”. In: *2007 IEEE Int. Conf. on Robotics and Automation*. 2007, pp. 1942–1947 (cit. on p. 83).
- [Naseer, 2013] T. Naseer, J. Sturm, and D. Cremers. “FollowMe: Person following and gesture recognition with a quadrotor”. In: *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2013, pp. 624–630 (cit. on p. 15).
- [Nava, 2020] G. Nava, Q. Sablé, M. Tognon, D. Pucci, and A. Franchi. “Direct Force Feedback Control and Online Multi-Task Optimization for Aerial Manipulators”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 331–338 (cit. on pp. 12, 33, 36).

- [Nejadasl, 2022] A. M. Nejadasl, O. Gheibi, G. Van de Perre, and B. Vanderborght. “NeuroErgo: A Deep Neural Network Method to Improve Postural Optimization for Ergonomic Human-Robot Collaboration”. In: *2022 IEEE Int. Conf. on Robotics and Automation*. 2022, pp. 7372–7378 (cit. on pp. 41, 42).
- [Newcome, 2004] L. R. Newcome. *Unmanned aviation: a brief history of unmanned aerial vehicles*. Aiaa, 2004 (cit. on pp. 6, 7).
- [Nguyen, 2013] H-N. Nguyen and D. Lee. “Hybrid force/motion control and internal dynamics of quadrotors for tool operation”. In: *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2013, pp. 3458–3464 (cit. on pp. 11, 12, 33, 35).
- [Nurkanović, 2019] A. Nurkanović, A. Zanelli, S. Albrecht, and M. Diehl. “The Advanced Step Real Time Iteration for NMPC”. In: *2019 IEEE Conf. on Decision and Control*. 2019, pp. 5298–5305 (cit. on p.128).
- [Odelga, 2016] M. Odelga, P. Stegagno, and H. H. Bühlhoff. “A fully actuated quadrotor UAV with a propeller tilting mechanism: Modeling and control”. In: *2016 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*. 2016, pp. 306–311 (cit. on p. 11).
- [Ollero, 2018] A. Ollero et al. “The AEROARMS Project: Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance”. In: *IEEE Robotics & Automation Magazine* 25.4 (2018), pp. 12–23 (cit. on p. 13).
- [Ollero, 2021] A. Ollero, M. Tognon, A. Suarez, D. J. Lee, and A. Franchi. “Past, Present, and Future of Aerial Robotic Manipulators”. In: *IEEE Trans. on Robotics* 38.1 (2021), pp. 626–645 (cit. on pp. 11, 13, 17, 32–35).
- [Oriolo, 2002] G. Oriolo, A. De Luca, and M. Vendittelli. “WMR control via dynamic feedback linearization: design, implementation, and experimental validation”. In: *IEEE Trans. on Control Systems Technology* 10.6 (2002), pp. 835–852 (cit. on p. 34).
- [Orsag, 2018] M. Orsag, C. Korpela, P. Oh, S. Bogdan, and A. Ollero. *Aerial Manipulation*. Springer, 2018 (cit. on p. 114).
- [Ortenzi, 2021] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kulić. “Object Handovers: A Review for Robotics”. In: *IEEE Trans. on Robotics* 37.6 (2021), pp. 1855–1873 (cit. on pp. 17, 23–26, 166).

- [Paneque, 2022] J. L. Paneque, J. R. Martinez-de Dios, A. Ollero, D. Hanover, S. Sun, A. Romero, and D. Scaramuzza. “Perception-Aware Perching on Powerlines With Multirobotors”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3077–3084 (cit. on pp. 13, 14).
- [Parastegari, 2017] S. Parastegari, B. Abbasi, E. Noohi, and M. Zefran. “Modeling human reaching phase in human-human object handover with application in robot-human handover”. In: *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2017, pp. 3597–3602 (cit. on p. 42).
- [Penin, 2017] B. Penin, R. Spica, P. Robuffo Giordano, and F. Chaumette. “Vision-based minimum-time trajectory generation for a quadrotor UAV”. In: *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2017, pp. 6199–6206 (cit. on p. 38).
- [Penin, 2018] B. Penin, P. Robuffo Giordano, and F. Chaumette. “Vision-Based Reactive Planning for Aggressive Target Tracking While Avoiding Collisions and Occlusions”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3725–3732 (cit. on p. 38).
- [Peric, 2021] L. Peric, M. Brunner, K. Bodie, M. Tognon, and R. Siegwart. “Direct Force and Pose NMPC with Multiple Interaction Modes for Aerial Push-and-Slide Operations”. In: *2021 IEEE Int. Conf. on Robotics and Automation*. 2021, pp. 131–137 (cit. on pp. 34, 156).
- [Peringal, 2022] A. Peringal, M. Chehadeh, R. Azzam, M. Hamandi, I. Boiko, and Y. Zweiri. “Design of Dynamics Invariant LSTM for Touch Based Human-UAV Interaction Detection”. In: *IEEE Access* 10 (2022), pp. 116045–116058 (cit. on p. 34).
- [Peternel, 2017] L. Peternel, W. Kim, J. Babic, and A. Ajoudani. “Towards ergonomic control of human-robot co-manipulation and handover”. In: *2017 IEEE Int. Conf. on Humanoid Robots*. 2017, pp. 55–60 (cit. on pp. 42, 136).
- [Petráček, 2020] P. Petráček, V. Krátký, and M. Saska. “Dronument: System for Reliable Deployment of Micro Aerial Vehicles in Dark Areas of Large Historical Monuments”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2078–2085 (cit. on p. 8).
- [Pierri, 2018] F. Pierri, G. Muscio, and F. Caccavale. “An adaptive hierarchical control for aerial manipulators”. In: *Robotica* 36.10 (2018), pp. 1527–1550 (cit. on p. 46).

- [Pose, 2022] Claudio Pose, Juan Giribet, and Ignacio Mas. “Adaptive Center-of-Mass Relocation for Aerial Manipulator Fault Tolerance”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5583–5590 (cit. on p. 12).
- [Pounds, 2010] P. Pounds, R. Mahony, and P. Corke. “Modelling and control of a large quadrotor robot”. In: *Control Engineering Practice* 18.7 (2010), pp. 691–699 (cit. on pp. 79, 87).
- [Press, 1992] W. H. Press and S. A. Teukolsky. “Adaptive Stepsize Runge-Kutta Integration”. In: *Computers in Physics* 6.2 (1992), p. 188 (cit. on p. 145).
- [Qin, 2003] S. J. Qin and T. A. Badgwell. “A survey of industrial model predictive control technology”. In: *Control Engineering Practice* 11.7 (2003), pp. 733–764 (cit. on p. 127).
- [Quirynen, 2017] R. Quirynen. “Numerical Simulation Methods for Embedded Optimization”. Dissertation. Katholieke Universiteit Leuven, 2017 (cit. on p. 130).
- [Rajappa, 2015] S. Rajappa, M. Ryll, H. H. Bühlhoff, and A. Franchi. “Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers”. In: *2015 IEEE Int. Conf. on Robotics and Automation*. 2015, pp. 4006–4013 (cit. on p. 35).
- [Rajappa, 2017] S. Rajappa, H. Bühlhoff, and P. Stegagno. “Design and implementation of a novel architecture for physical human-UAV interaction”. In: *The International Journal of Robotics Research* 36.5-7 (2017), pp. 800–819 (cit. on pp. 16, 33).
- [Raković, 2018] S. V. Raković and W. S. Levine, eds. *Handbook of Model Predictive Control*. Control Engineering. Birkhäuser Cham, 2018 (cit. on p. 36).
- [Rapetti, 2023] L. Rapetti, C. Sartore, M. Elobaid, Y. Tirupachuri, F. Draicchio, T. Kawakami, T. Yoshiike, and D. Pucci. “A Control Approach for Human-Robot Ergonomic Payload Lifting”. In: *2023 IEEE Int. Conf. on Robotics and Automation*. 2023, pp. 7504–7510 (cit. on p. 17).
- [Rashad, 2020] R. Rashad, J. Goerres, R. Aarts, J. B. C. Engelen, and S. Stramigioli. “Fully Actuated Multirotor UAVs: A Literature Review”. In: *IEEE Robotics & Automation Magazine* 27.3 (2020), pp. 97–107 (cit. on p. 11).
- [Ristorito, 2015] G. Ristorito, F. Mazzetto, G. Guglieri, and F. Quagliotti. “Monitoring performances and cost estimation of multirotor Unmanned Aerial Systems in precision farming”. In: *2015 Int. Conf. Unmanned Aircr. Syst. ICUAS 2015*. IEEE, 2015, pp. 502–509 (cit. on p. 8).

- [Rosenfeld, 1988] A. Rosenfeld. “Computer vision: basic principles”. In: *Proceedings of the IEEE* 76.8 (1988), pp. 863–868 (cit. on p. 98).
- [Roveda, 2020] L. Roveda, M. Forgione, and D. Piga. “Robot control parameters auto-tuning in trajectory tracking applications”. In: *Control Engineering Practice* 101 (2020), p. 104488 (cit. on p. 124).
- [Rucker, 2018] C. Rucker. “Integrating rotations using nonunit quaternions”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 2979–2986 (cit. on pp. 71, 72, 145).
- [Ruggiero, 2014] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello. “Impedance control of VTOL UAVs with a momentum-based external generalized forces estimator”. In: *2014 IEEE Int. Conf. on Robotics and Automation*. 2014, pp. 2093–2099 (cit. on pp. 33, 35).
- [Ruggiero, 2015] F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Pérez, V. Lippiello, A. Ollero, and B. Siciliano. “A multilayer control for multirotor UAVs equipped with a servo robot arm”. In: *2015 IEEE Int. Conf. on Robotics and Automation*. 2015, pp. 4014–4020 (cit. on p. 36).
- [Ruggiero, 2018] F. Ruggiero, V. Lippiello, and A. Ollero. “Aerial Manipulation: A Literature Review”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1957–1964 (cit. on pp. 7, 12–14, 17, 35, 36).
- [Ruina, 2019] A. Ruina and R. Pratap. *Introduction to Statics and Dynamics*. 2019 (cit. on p. 65).
- [Ryll, 2012] M. Ryll, H. H. Bulthoff, and P. R. Giordano. “Modeling and control of a quadrotor UAV with tilting propellers”. In: *2012 IEEE Int. Conf. on Robotics and Automation*. IEEE, May 2012, pp. 4606–4613 (cit. on pp. 11, 85).
- [Ryll, 2017] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi. “6D physical interaction with a fully actuated aerial robot”. In: *2017 IEEE Int. Conf. on Robotics and Automation*. 2017, pp. 5190–5195 (cit. on p. 28).
- [Ryll, 2019] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi. “6D interaction control with aerial robots: The flying end-effector paradigm”. In: *The International Journal of Robotics Research* 38.9 (2019), pp. 1045–1062 (cit. on pp. 12, 13, 33, 35).



- [Ryll, 2022] M. Ryll, D. Bicego, M. Giurato, M. Lovera, and A. Franchi. “FAST-Hex—A Morphing Hexarotor: Design, Mechanical Implementation, Control and Experimental Validation”. In: *IEEE/ASME Trans. on Mechatronics* 27.3 (2022), pp. 1244–1255 (cit. on p. 11).
- [Salzmann, 2022] T. Salzmann, E. Kaufmann, M. Pavone, D. Scaramuzza, and M. Ryll. “Neural-MPC: Deep Learning Model Predictive Control for Quadrotors and Agile Robotic Platforms”. 2022 (cit. on p. 79).
- [Sanalitra, 2022] D. Sanalitra, M. Tognon, A.E. Jimenez A. E. Jimenez Cano, J. Cortés, and A. Franchi. “Indirect Force Control of a Cable-Suspended Aerial Multi-Robot Manipulator”. In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 6726–6733 (cit. on pp. 12, 13).
- [Sanchez-Cuevas, 2017] P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero. “Multirotor UAS for bridge inspection by contact using the ceiling effect”. In: *2017 Int. Conf. on Unmanned Aircraft Systems*. 2017, pp. 767–774 (cit. on p. 13).
- [Santamaria-Navarro, 2018] A. Santamaria-Navarro, G. Loianno, J. Solà, V. Kumar, and J. Andrade-Cetto. “Autonomous navigation of micro aerial vehicles using high-rate and low-cost sensors”. In: *Autonomous Robots* 42.6 (2018), pp. 1263–1280 (cit. on p. 101).
- [Santamaria-Navarro, 2019] A. Santamaria-Navarro, J. Solà, and J. Andrade-Cetto. *Visual Guidance of Unmanned Aerial Manipulators*. Springer, 2019 (cit. on p. 38).
- [Santis, 2008] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi. “An atlas of physical human–robot interaction”. In: *Mechanism and Machine Theory* 43.3 (2008), pp. 253–270 (cit. on pp. 5, 29, 32, 39).
- [Santos, 2022] D. A. Santos and J. A. Bezerra. “On the control allocation of fully actuated multirotor aerial vehicles”. In: *Aerospace Science and Technology* 122 (2022), p. 107424 (cit. on p. 87).
- [Sarafianos, 2016] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris. “3D Human pose estimation: A review of the literature and analysis of covariates”. In: *Computer Vision and Image Understanding* 152 (2016), pp. 1–20 (cit. on p. 29).
- [Scherer, 2015] J. Scherer, B. Rinner, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, and H. Hellwagner. “An Autonomous Multi-UAV System for Search and Rescue”. In: *Proc. First Work. Micro Aer. Veh. Networks, Syst. Appl. Civ. Use - DroNet '15*. New York, USA: ACM Press, 2015, pp. 33–38 (cit. on p. 8).

- [Schwenzer, 2021] M. Schwenzer, M. Ay, T. Bergs, and D. Abel. “Review on model predictive control: an engineering perspective”. In: *The International Journal of Advanced Manufacturing Technology* 117.5 (2021), pp. 1327–1349 (cit. on pp. 127, 128).
- [Shafti, 2019] A. Shafti, A. Ataka, B. U. Lazpita, A. Shiva, H. A. Wurdemann, and K. Althoefer. “Real-time Robot-assisted Ergonomics”. In: *2019 IEEE Int. Conf. on Robotics and Automation*. 2019, pp. 1975–1981 (cit. on p. 41).
- [Shi, 2017] D. Shi, X. Dai, X. Zhang, and Q. Quan. “A Practical Performance Evaluation Method for Electric Multicopters”. In: *IEEE/ASME Trans. on Mechatronics* 22.3 (2017), pp. 1337–1348 (cit. on p. 89).
- [Siciliano, 2008] B. Siciliano, O. Khatib, and T. Kröger. *Springer Handbook of Robotics*. Springer, 2008 (cit. on pp. 68–70, 72–75).
- [Siciliano, 2009] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2009 (cit. on pp. 13, 32, 33, 36, 68–70, 72–74, 77–81, 83, 95, 102, 138, 139, 144, 155).
- [Silva, 2002] C. C. Silva and R. De Andrade Martins. *Polar and axial vectors versus quaternions*. 2002 (cit. on p. 72).
- [Sisbot, 2012] E. A. Sisbot and R. Alami. “A Human-Aware Manipulation Planner”. In: *IEEE Trans. on Robotics* 28.5 (2012), pp. 1045–1057 (cit. on p. 40).
- [Solà, 2017] J. Solà. *Quaternion kinematics for the error-state Kalman filter*. Tech. rep. 2017 (cit. on pp. 70, 72, 77).
- [Soratana, 2018] Teerachart Soratana, Mythra V. S. M. Balakuntala, Praveen Abbaraju, Richard Voyles, Juan Wachs, and Mohammad Mahoor. “Glovebox Handling of High-Consequence Materials with Super Baxter and Gesture-Based Programming - 18598”. In: (2018) (cit. on p. 5).
- [Spica, 2012] R. Spica, A. Franchi, G. Oriolo, H. H. Bühlhoff, and P. Robuffo Giordano. “Aerial grasping of a moving target with a quadrotor UAV”. In: *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2012, pp. 4985–4992 (cit. on p. 35).
- [Stephens, 2022] B. Stephens, L. Orr, B. B. Kocer, H-N. Nguyen, and M. Kovac. “An Aerial Parallel Manipulator With Shared Compliance”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11902–11909 (cit. on p. 13).

- [Strabala, 2013] K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. S. Srinivasa, M. Cakmak, and V. Micelli. “[Toward seamless human-robot handovers](#)”. In: *Journal of Human-Robot Interaction* 2.1 (2013), pp. 112–132 (cit. on pp. 17, 24, 25, 133, 166).
- [Strang, 2006] G. Strang. *Linear algebra and its applications*. Belmont, CA: Thomson, Brooks/Cole, 2006 (cit. on p. 66).
- [Suarez, 2022] A. Suarez, R. Salmoral, A. Garofano-Soldado, G. Heredia, and A. Ollero. “[Aerial Device Delivery for Power Line Inspection and Maintenance](#)”. In: *2022 Int. Conf. on Unmanned Aircraft Systems*. 2022, pp. 30–38 (cit. on pp. 16, 17).
- [Sun, 2022] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza. “[A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight](#)”. In: *IEEE Trans. on Robotics* 38.6 (2022), pp. 3357–3373 (cit. on p. 87).
- [Swief, 2019] A. Swief, A. El-Zawawi, and M. El-Habrouk. “[A Survey of Model Predictive Control Development in Automotive Industries](#)”. In: *2019 Int. Conf. on Applied Automation and Industrial Diagnostics*. Vol. 1. 2019, pp. 1–7 (cit. on p. 127).
- [Thomas, 2017] J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar. “[Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor](#)”. In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1762–1769 (cit. on p. 92).
- [Tognon, 2017] M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi. “[Dynamic Decentralized Control for Protocentric Aerial Manipulators](#)”. In: *2017 IEEE Int. Conf. on Robotics and Automation*. Singapore, May 2017, pp. 6375–6380 (cit. on p. 12).
- [Tognon, 2018] M. Tognon and A. Franchi. “[Omnidirectional Aerial Vehicles With Unidirectional Thrusters: Theory, Optimal Design, and Control](#)”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 2277–2282 (cit. on p. 10).
- [Tognon, 2019] M. Tognon et al. “[A Truly-Redundant Aerial Manipulator System With Application to Push-and-Slide Inspection in Industrial Plants](#)”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1846–1851 (cit. on pp. 13, 14, 83).

- [Tognon, 2021] M. Tognon, R. Alami, and B. Siciliano. “Physical Human-Robot Interaction With a Tethered Aerial Vehicle: Application to a Force-Based Human Guiding Problem”. In: *IEEE Trans. on Robotics* 37.3 (2021), pp. 723–734 (cit. on p. 16).
- [Tomić, 2017] T. Tomić, C. Ott, and S. Haddadin. “External Wrench Estimation, Collision Detection, and Reflex Reaction for Flying Robots”. In: *IEEE Trans. on Robotics* 33.6 (2017), pp. 1467–1482 (cit. on pp. 34, 101).
- [Truc, 2022] J. Truc, P.-T. Singamaneni, D. Sidobre, S. Ivaldi, and R. Alami. “KHAOS: a Kinematic Human Aware Optimization-based System for Reactive Planning of Flying-Coworker”. In: *2022 IEEE Int. Conf. on Robotics and Automation*. 2022, pp. 4764–4770 (cit. on pp. 17, 31, 40, 47, 146).
- [Truc, 2023] J. Truc, D. Sidobre, and R. Alami. “Reactive Planning for Coordinated Handover of an Autonomous Aerial Manipulator”. In: *Companion of the 2023 ACM/IEEE Int. Conf. on Human-Robot Interaction*. 2023, pp. 122–126 (cit. on pp. 17, 40).
- [Trujillo, 2016] M. M. Trujillo, M. Darrah, K. Speransky, B. Deroos, and M. Wathen. “Optimized flight path for 3D mapping of an area with structures using a multirotor”. In: *2016 Int. Conf. on Unmanned Aircraft Systems*. 2016, pp. 905–910 (cit. on p. 8).
- [Tzoumanikas, 2020] D. Tzoumanikas, F. Graule, Q. Yan, D. Shah, M. Popovic, and S. Leutenegger. “Aerial Manipulation Using Hybrid Force and Position NMPC Applied to Aerial Writing”. 2020 (cit. on p. 156).
- [Vianello, 2021] L. Vianello, L. Penco, W. Gomes, Y. You, S. M. Anzalone, P. Maurice, V. Thomas, and S. Ivaldi. “Human-Humanoid Interaction and Cooperation: a Review”. In: *Current Robotics Reports* 2.4 (2021), pp. 441–454 (cit. on p. 5).
- [Villa, 2020] D. K. D. Villa, A. S. Brandão, and M. Sarcinelli-Filho. “A Survey on Load Transportation Using Multirotor UAVs”. In: *Journal of Intelligent & Robotics Systems* 98.2 (May 2020), pp. 267–296 (cit. on p. 13).
- [Villani, 2018] V. Villani, F. Pini, F. Leali, and C. Secchi. “Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications”. In: *Mechatronics* 55 (2018), pp. 248–266 (cit. on p. 49).
- [Welde, 2021] J. Welde, J. Paulos, and V. Kumar. “Dynamically Feasible Task Space Planning for Underactuated Aerial Manipulators”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3232–3239 (cit. on p. 35).

- [Wilson-Small, 2023] N. J. Wilson-Small, D. Goedicke, K. Petersen, and S. Azenkot. “A Drone Teacher: Designing Physical Human-Drone Interactions for Movement Instruction”. In: *2023 ACM/IEEE Int. Conf. on Human-Robot Interaction*. 2023, pp. 311–320 (cit. on p. 15).
- [Winter, 2009] D. A. Winter. “Anthropometry”. In: *Biomechanics and Motor Control of Human Movement*. 2009. Chap. 4, pp. 82–106 (cit. on p. 95).
- [Wofk, 2019] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze. “Fastdepth: Fast monocular depth estimation on embedded systems”. In: *2019 IEEE Int. Conf. on Robotics and Automation*. 2019, pp. 6101–6108 (cit. on p. 92).
- [Xian, 2004] B. Xian, M. S. de Queiroz, D. Dawson, and I. Walker. “Task-space tracking control of robot manipulators via quaternion feedback”. In: *IEEE Trans. on Robotics* 20.1 (2004), pp. 160–167 (cit. on p. 71).
- [Xiao, 2021] A. Xiao, W. Tong, L. Yang, J. Zeng, Z. Li, and K. Sreenath. “Robotic Guide Dog: Leading a Human with Leash-Guided Hybrid Physical Interaction”. In: *2021 IEEE Int. Conf. on Robotics and Automation*. 2021, pp. 11470–11476 (cit. on p. 5).
- [Yang, 2020] W. Yang, C. Paxton, M. Cakmak, and D. Fox. “Human Grasp Classification for Reactive Human-to-Robot Handovers”. In: *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2020, pp. 11123–11130 (cit. on pp. 17, 24).
- [Yang, 2021] W. Yang, C. Paxton, A. Mousavian, Y.-W. Chao, M. Cakmak, and D. Fox. “Reactive Human-to-Robot Handovers of Arbitrary Objects”. In: *2021 IEEE Int. Conf. on Robotics and Automation*. 2021, pp. 3118–3124 (cit. on pp. 17, 23).
- [Yang, 2022] W. Yang, B. Sundaralingam, C. Paxton, I. Akinola, Y.-W. Chao, M. Cakmak, and D. Fox. “Model Predictive Control for Fluid Human-to-Robot Handovers”. In: *2022 IEEE Int. Conf. on Robotics and Automation*. 2022, pp. 6956–6962 (cit. on pp. 17, 23).
- [Yazdani, 2022] A. Yazdani, R. S. Novin, A. Merryweather, and T. Hermans. “DULA and DEBA: Differentiable Ergonomic Risk Models for Postural Assessment and Optimization in Ergonomically Intelligent pHRI”. 2022 (cit. on p. 42).
- [Yeh, 2017] A. Yeh, P. Ratsamee, K. Kiyokawa, Y. Uranishi, T. Mashita, H. Takemura, M. Fjeld, and M. Obaid. “Exploring Proxemics for Human-Drone Interaction”. In: *2017 Int. Conf. on Human Agent Interaction*. 2017, pp. 81–88 (cit. on p. 15).

- [Yüksel, 2016] B. Yüksel, G. Buondonno, and A. Franchi. “Differential flatness and control of protocentric aerial manipulators with any number of arms and mixed rigid-/elastic-joints”. In: *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2016, pp. 561–566 (cit. on p. 35).
- [Yüksel, 2019] Burak Yüksel, Cristian Secchi, Heinrich H. Bühlhoff, and Antonio Franchi. “Aerial physical interaction via IDA-PBC”. In: *The International Journal of Robotics Research* 38.4 (2019), pp. 403–421 (cit. on pp. 8, 13).
- [Yutao, 2019] C. Yutao. “Algorithms and Applications for Nonlinear Model Predictive Control with Long Prediction Horizon”. Dissertation. University of Padua, 2019 (cit. on pp. 48, 130, 138).
- [Zacharaki, 2020] A. Zacharaki, I. Kostavelis, A. Gasteratos, and I. Dokas. “Safety bounds in human robot interaction: A survey”. In: *Safety Science* 127 (2020), p. 104667 (cit. on pp. 29, 39).
- [Zanchettin, 2019] A. M. Zanchettin, E. Lotano, and P. Rocco. “Collaborative Robot Assistant for the Ergonomic Manipulation of Cumbersome Objects”. In: *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2019, pp. 6729–6734 (cit. on p. 41).
- [Zhang, 2019] Pengyi Zhang, Yunxin Zhong, and Xiaoqiong Li. “SlimYOLOv3: Narrower, faster and better for real-time UAV applications”. In: *2020 IEEE/CVF Int. Conf. on Computer Vision*. 2019, pp. 37–45 (cit. on p. 147).
- [Zhang, 2022] K. Zhang et al. “Aerial additive manufacturing with multiple autonomous robots”. In: *Nature* 609.7928 (2022), pp. 709–717 (cit. on p. 13).
- [Zhao, 2022] M. Zhao, K. Okada, and M. Inaba. “Versatile articulated aerial robot DRAGON: Aerial manipulation and grasping by vectorable thrust control”. In: *The International Journal of Robotics Research* 42.4-5 (2022), pp. 214–248 (cit. on p. 85).
- [Zheng, 2018] D. Zheng, H. Wang, W. Chen, and Y. Wang. “Planning and Tracking in Image Space for Image-Based Visual Servoing of a Quadrotor”. In: *IEEE Trans. on Industrial Electronics* 65.4 (2018), pp. 3376–3385 (cit. on p. 38).
- [Zheng, 2020] P. Zheng, X. Tan, B. B. Kocer, E. Yang, and M. Kovac. “TiltDrone: A Fully-Actuated Tilting Quadrotor Platform”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6845–6852 (cit. on p. 11).

- [Zheng, 2023] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah. “Deep Learning-Based Human Pose Estimation: A Survey”. In: *ACM Computing Surveys* 56.1 (2023) (cit. on p. 166).
- [Zhou, 2021] J. Zhou, S. Yang, and Q. Xue. “Lower limb rehabilitation exoskeleton robot: A review”. In: *Advances in Mechanical Engineering* 13.4 (2021), p. 16878140211011862 (cit. on p. 5).
- [Zhu, 2018] P. Zhu et al. “Visdrone-DET2018: The vision meets drone object detection in image challenge results”. In: *2018 IEEE/CVF European Conf. on Computer Vision*. 2018 (cit. on p. 107).
- [Zufferey, 2021] R. Zufferey, J. Tormo-Barbero, M. M. Guzmán, F. J. Maldonado, E. Sanchez-Laulhe, P. Grau, M. Pérez, J. A. Acosta, and A. Ollero. “Design of the High-Payload Flapping Wing Robot E-Flap”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3097–3104 (cit. on p. 6).

# Résumé - Abstract

Les robots aériens et plus particulièrement les véhicules multi-rotors connaissent un large intérêt pour diverses applications. Plus récemment l'intégration d'outils fixes ou de bras articulés à plusieurs degrés de liberté sur les robots aériens a permis à ces robots d'interagir physiquement avec leur environnement, ouvrant ainsi la voie à la réalisation de tâches variées en collaboration entre humains et drones. Toutefois le déploiement de robots aériens pour l'interaction humain-robot présente de nouveaux défis. Du point de vue du contrôle, des algorithmes novateurs sont nécessaires pour que l'interaction physique soit sûre.

Cette thèse aborde ces défis en proposant différentes architectures de contrôle pour les véhicules aériens multi-rotors qui visent à assurer une interaction physique sûre avec les humains. Des résultats analytiques et expérimentaux confirment la validité et la pertinence pratique de la méthodologie proposée.

**Mots clefs** - Robotique aérienne, Contrôle des robots, Manipulation aérienne, Interactions physiques entre humain et robot, Échange d'objet et co-manipulation, Robotique collaborative

---

Aerial robots and more particularly multi-rotor aerial vehicles have experienced a large interest for various applications, encompassing contact-less operations as well as contact-based scenarios. Recently, the integration of fixed tools or multi-degree-of-freedom robotic arms on aerial robots has enabled them to achieve physical interaction with the environment. This has driven the robotic community to envision the realization of collaborative tasks between humans and flying vehicles. However, deploying aerial robots in scenarios involving human-robot interaction poses new challenges. From a control standpoint, novel control algorithms are necessary for a safe physical interaction.

This thesis tackles these challenges by presenting different control architectures for multi-rotor aerial vehicles ensuring both physical and safe interaction with humans. Analytical and experimental results substantiate the validity and practical relevance of the proposed methodology.

**Keywords** - Aerial Robotics, Control of Robots, Aerial Manipulation, Physical Human-Robot Interaction, Handover and co-manipulation, Collaborative Robotics